DIMACS Technical Report 2002-32
August 2002

Error exponents of expander codes under linear-complexity decoding

by

Alexander Barg[1 ,2]
Math. Research
Bell Labs, Lucent Technologies
700 Mountain Ave.,
Murray Hil,. NJ 07974, USA
and IPPI RAN, Moscow, Russia

Gilles Zémor[3]
École Nationale Supérieure des Télécommunications
46 rue Barrault
75 634 Paris 13, France

[1]Permanent Member
[2]Supported in part by the Binational Science Foundation (USA-Israel), grant no. 1999099.
[3]Research done while visiting DIMACS Center, Rutgers University, Piscataway, NJ, in August 2001

DIMACS is a partnership of Rutgers University, Princeton University, AT&T Labs-Research, Bell Labs, NEC Research Institute and Telcordia Technologies (formerly Bellcore).

DIMACS was founded as an NSF Science and Technology Center, and also receives support from the New Jersey Commission on Science and Technology.

ABSTRACT

A class of codes is said to reach capacity $\mathcal{C}$ of the binary symmetric channel if for any rate $R < \mathcal{C}$ and any $\varepsilon > 0$ there is a sufficiently large N such that codes of length $\geq N$ and rate R from this class provide error probability of decoding at most ε, under some decoding algorithm.

The study of the error probability of expander codes was initiated in Barg and Zémor (2002), where it was shown that they attain capacity of the binary symmetric channel under a linear-time iterative decoding with error probability falling exponentially with code length N. In this work we study variations on the expander code construction and focus on the most important region of code rates, close to the channel capacity. For this region we estimate the decrease rate (the error exponent) of error probability of decoding for randomized ensembles codes. The resulting estimate gives a substantial improvement of previous results for expander codes and some other explicit code families.

1 Introduction

We study transmission of information with linear codes over the *binary symmetric channel* (BSC). An $[N, K]$ binary linear code C is a K-dimensional linear subspace of $\{0,1\}^N$. The number N is called the length of the code and the relative dimension K/N is called the rate of the code, denoted by $R = R(C)$. The BSC is a random $\{0,1\} \to \{0,1\}$ map acting independently on the coordinates of the transmitted code vector x. A binary digit sent over the channel is received correctly with probability $1 - p$ and flipped with probability $p < 1/2$. The objective of the decoder is to restore correctly the transmitted codeword. *Maximum likelihood* (or complete) *decoding* of C consists of choosing a codeword x' closest to the received vector y. The event that $x \neq x'$ corresponds to a decoding error. The probability $P_e(C)$ that decoding goes wrong is independent of the transmitted codeword and is a polynomial function of p. This polynomial is notoriously difficult to compute exactly but estimating the value $P_e(C)$ can be somewhat simplified: when p is small enough $P_e(C)$ behaves like its lowest-degree term and the lowest degree equals half the minimum Hamming distance of C. For this reason, combinatorial coding theory is concerned with the construction of large codes with large minimum distance.

However, for both theoretical and practical reasons (like the emergence of mobile communications and their very noisy channels), there has been a renewed interest for studying the situation when p is large, or equivalently when R is close to capacity. Shannon's theorem states that $P_e(C)$ can stay close to zero only as long as we have $R < \mathcal{C}$ where $\mathcal{C}$ is the *channel capacity* and depends only on p. Furthermore, when R is close to $\mathcal{C}$, we know that for large N, fixed rate $R < \mathcal{C}$, and for the best possible codes of rate R, the decoding error probability $P_e(C)$ takes the form $P_e(C) = 2^{-E(R,p)N+o(N)}$ where $E(R, p)$, called the *error exponent*, is a positive quantity that depends only on p and the rate R and can be computed exactly [7]. Unfortunately, the only known way to achieve this best possible error exponent is to use a random code C together with decoding algorithms that find the closest codeword by essentially performing exhaustive search over C.

Turning to manageable decoding algorithms, until fairly recently only one class of codes was known to achieve a non-zero decoding exponent in polynomial time (though less than $E(R, p)$) for rates arbitrarily close to channel capacity: the class of concatenated codes, introduced by Forney [8] and extensively studied through the mid-eighties (see [1], [6] for overviews).

In the nineties the discovery of turbo-codes [3] with their largely unexplained close-to-capacity performance shifted emphasis to iterative decoding techniques. One particular class of codes that can be iteratively decoded is that of expander codes. An expander code is constructed by assigning binary digits to edges of a bipartite graph in a way that was introduced by Tanner [14]. A surge of interest in them occurred after it was shown in [13] that if the underlying graph is an expander graph, then they correct an $\Omega(N)$ number of errors under an $O(N)$ iterative decoding. Not only was this a significant achievement for iterative decoding but it was the first example of this kind in coding theory at large; indeed, the concatenated decoding rival requires an $O(N^2)$ decoding time.

In our work [2] we employ the iterative decoding algorithm of [15] to show that expander codes actually reach the capacity of the BSC with a positive error exponent. Though the error exponent of concatenated codes [8] is better than that of expander codes of [2], their performance again relies on a quadratic-time decoding algorithm as opposed to linear-time decoding of [15],[2].

An expander code C in [13], [15] is constructed from a bipartite Δ-regular graph $G = [A \cup B, E]$ with both parts of size n and a binary code $[\Delta, R\Delta]$ code C_0. Coordinates of a codeword in C are in one-to-one correspondence with edges of G and satisfy the condition that the subvector incident to every vertex $v \in A \cup B$ is a codevector in C_0. The decoding algorithm of [15] consists of iterations that alternate between A and B. In each iteration all the vertices of the respective part are decoded in parallel with the code C_0. Paper [15] also provides a bound on the number of errors correctable by this decoding.

Among the new ideas introduced in [2] is the use of two different codes: C_0 for the part A and C_1 for B. The value of the resulting error exponent is then optimized on the choice of the rates of these codes. Another idea of [2] is replacing each edge by t parallel edges and employing both binary and q-ary structure in the analysis of component codes, $q = 2^t$.

In this paper we obtain a substantial improvement of the estimate of the error exponent for expander codes, focusing on R close to capacity. In particular, we surpass in this region the error exponent of Forney's concatenated codes [8], a benchmark for a long time. The improvement relies on the following ideas which were not employed in earlier analysis. In the first iteration, again relying on the q-ary structure of the code C_0, we employ detailed information on the error events. Introducing an element of randomization in the overall code construction, we can then make a stronger statement about performance of the code C_1 in the second decoding iteration. Finally, we modify the original construction by adjoining a number of vertices of degree one to the expander graph. This decreases the error correcting potential but improves the overall code rate and the resulting trade-off improves the overall error exponent. This idea borrows from turbo codes for which the underlying graph has many degree one vertices contrary to other classes of codes amenable to iterative decoding.

The rest of the paper is organized as follows. In Sections 2 and 3 we introduce the necessary coding background. In Section 4 we summarize previous work on expander codes. In Section 5 we introduce our new variation on decoding and give a refined probabilistic analysis of its behavior: this results in a first error exponent in Theorem 7. In Section 6 this is improved to Theorem 10 through a modified construction that we analyze. Finally we give some concluding comments.

2 Codes and their parameters

Although our ultimate goal will be binary codes, we will also consider codes over larger alphabets of size $q = 2^t$. By $\mathcal{H}_q = \mathcal{H}_q^N$ we denote the q-ary Hamming space, i.e., the N-dimensional coordinate space over the field of q elements. The number of nonzero coordinates of a vector $x \in \mathcal{H}_q$ is called the (Hamming) weight, denoted $|x|$. The Hamming distance is defined by $\mathrm{d}(x, y) = |x - y|$.

For a given linear code C the minimum weight of a nonzero codeword in C is called its *distance*. For a given q we will use notation $C[N,K,D]$ to refer to a linear code of length N, dimension K and distance $D = D(C)$, occasionally omitting the distance.

One of the key problems of combinatorial coding theory is finding the maximum size of a code C of length N and distance D. Consider families of codes $C_i, i = 1,2,\dots$ of growing length N_i, rate R_i and relative distance $\delta_i = D(C_i)/N_i$. According to the well known Gilbert-Varshamov (GV) bound there exist sequences of codes of rate $R(C_i) \to R$ and relative distance $\delta_i \to \delta$ for any

$$R < 1 - H_q(\delta),$$

where

$$H_q(x) = -x \log_q \frac{x}{q-1} - (1-x)\log_q(1-x)$$

is the q-ary entropy function. For a given R we let $\delta_{\mathrm{GV}}^{(q)}(R)$ denote the GV distance: $\delta_{\mathrm{GV}}^{(q)}(R) = H_q^{-1}(1-R)$. Note that $\delta_{\mathrm{GV}}^{(q)}(0) = (q-1)/q$, $\delta_{\mathrm{GV}}^{(q)}(1) = 0$. For $q = 2$ we omit the superscript and write simply $\delta_{\mathrm{GV}}(R)$. Likewise, throughout the paper if the base of the logarithms and exponents is missing, it is equal to 2.

A number of upper bounds are known on the relative distance $\delta(R)$ of a code sequence of rate R. We mention the Bassalygo-Elias bound which asserts that, for binary codes

$$\delta(R) \le \delta_E(R) := 2\delta_{\mathrm{GV}}(R)(1 - \delta_{\mathrm{GV}}(R)).$$

We call the quantity $\delta_E(R) := 2\delta_{\mathrm{GV}}(R)(1-\delta_{\mathrm{GV}}(R))$ the *Elias radius*; its relevance to decoding will become apparent from Prop. 1.

As $q = 2^t$ gets large, the quantity $(q-1)/q$ comes close to one, and the inverse entropy function gets close to the linear function $1 - R$. More precisely, for any $\alpha, \beta > 0$ there exists a value t_0 such that for all $t > t_0$ and all $\alpha < R < 1 - \alpha$

$$0 < (1-R) - \delta_{\mathrm{GV}}^{(q)}(R) < \beta. \tag{1}$$

3 Decoding and error exponents

3.1 Decoding

Given a code $C \subset \mathcal{H}_q$, any mapping $\psi : \mathcal{H}_q \to C$ that takes a given point y to one of the codewords of C closest to it is called *maximum likelihood decoding.*

Given an $[N,K]$ binary code C and a BSC(p), we define the error probability for a given code vector x as follows:

$$P_e(x) = \Pr\big[\psi(y) \ne x \mid x \text{ is sent and } y \text{ is received}\big].$$

Because C is a linear code this probability is independent of the choice of the mapping ψ and of the code vector x and equals the average error probability

$$P_e(C) = 2^{-K} \sum_{x \in C} P_e(x)$$

Therefore, for the purposes of analysis we always assume that the transmitted vector always is the all-zero one: $x = 0$ and write $P_e(C)$ instead of $P_e(0)$.

By the classical results of coding theory [7], [10] there exist sequences of binary linear codes such that the probability $P_e(C)$ under maximum likelihood decoding falls exponentially with the code length N. Therefore, define the error exponent $E(C) = -N^{-1} \log P_e(C)$. We define the best attainable error exponent for the rate R as

$$E(R, p) = \liminf_{N \to \infty} \max_{C \subset \mathcal{H}_2^N : R(C) \geq R} E(C).$$

It was proved in [7] (see also [10]) that $E(R) > 0$ for $0 \leq R < \mathcal{C}$.

3.2 Random coding exponent and typical events

The best known existence bound $E_0(R, p)$ on the error exponent $E(R, p)$ of binary codes (linear or not) is obtained by the random coding method [10]. The function $E_0(R, p)$ is positive for all rates below the channel capacity $\mathcal{C} = 1 - H(p)$. It is easy to prove by random choice [10] that there exist sequences of binary linear $[N, RN]$ codes that attain the GV bound on the minimum distance and reach the error exponent $E_0(R, p)$ under maximum likelihood decoding. We assume that p is fixed and R varies from zero to $\mathcal{C}$. The form of the bound depends on the location of R with respect to $\mathcal{C}$. We are interested in the high-rate region, i.e., R close to $\mathcal{C}$. For high rates we obtain the bound [7], [10]

$$E_0(R, p) = D(\delta_{\mathrm{GV}}(R) \| p),$$

where

$$D(x \| y) := x \log(x/y) + (1 - x) \log((1 - x)/(1 - y)).$$

This bound is actually tight in the region $R_{\mathrm{crit}} \leq R \leq \mathcal{C}$, where the value $R_{\mathrm{crit}} = 1 - H(\rho_0)$,

$$\rho_0 = \frac{\sqrt{p}}{\sqrt{p} + \sqrt{1 - p}}, \tag{2}$$

is called the critical rate of the channel. In other words, for $R_{\mathrm{crit}} \leq R \leq \mathcal{C}$ we have $E(R, p) = E_0(R, p)$ [7]. Note that $E_0(\mathcal{C}, p) = 0$.

For rates $0 \leq R \leq R_{\mathrm{crit}}$ the random coding exponent has the following form:

$$\begin{aligned} E_0(R, p) &= -\delta_{\mathrm{GV}}(R) \log 2\sqrt{p(1 - p)} \quad (0 \leq R \leq R_x) \\ E_0(R, p) &= D(\rho_0 \| p) + R_{\mathrm{crit}} - R \quad (R_x \leq R \leq R_{\mathrm{crit}}), \end{aligned}$$

where $R_x = 1 - H(2\rho_0(1-\rho_0))$.

Both construction and decoding complexity of known code sequences that attain this error exponent grow exponentially with the code length N.

It is possible to say more about the geometry of typical error events under maximum likelihood decoding of random codes. In the high-rate region which interests us, the following proposition is true.

Proposition 1 *Let p be the parameter of a BSC. Assume that the transmitted vector is the all-zero one, and denote by z the decoded vector output by maximum likelihood decoding.*

For any R and for any large enough N, there exists a code C such that, for any p such that $R_{crit} \le R < \mathcal{C}$,

- *C has error exponent $E_0(R,p)$,*
- *for any $\alpha > 0$ we have*

$$\Pr\left[\left|\delta_E(R) - \frac{|z|}{N}\right| \ge \alpha \quad \Big| \quad z \neq 0\right] < 2^{-Nc(\alpha)} \quad (c(\alpha) > 0 \textit{ is independent of } N),$$

where $\Pr[\cdot]$ is the probability that the random vector z fulfills the condition in the brackets.

In other words, for high rates the typical relative distance between the transmitted codeword and the decoded codeword in the case of decoding error equals $\delta_E(R)$. The proof of this result (see Appendix) relies on a combination of facts known to coding theorists, but not spelled out in the literature.

3.3 Error exponents for $O(N^2)$ decoding: Concatenated codes

An important example of codes that attain channel capacity under low-complexity decoding is given by Forney's concatenated codes [8] (see also [6]). Concatenated codes form a generalization of Elias's product codes and provide code families with better parameters in terms of both code distance and error exponents.

Binary $[N = nm, k\ell]$ concatenated codes are constructed by first encoding the k q-ary message symbols, $q = 2^\ell$, with an $[n,k]$ q-ary code C_1, then representing every code symbol back as a string of ℓ bits, and then encoding it with an $[m,\ell]$ binary code C_0. We assume that both rates $R_0 = \ell/m$ of the inner code C_0 and $R_1 = k/n$ of the outer code are fixed.

Let $m = \log_2 N$. As with all linear codes, properties of a typical concatenated code found by random choice are much better than those of the known explicit families. In particular, there exists a sequence of binary $[m, \ell = mR_0]$ codes C_0 for which the error probability of maximum likelihood decoding falls as $2^{-mE_0(R_0,p)}$. Moreover, a brute-force implementation of decoding of the code C_0 has complexity $O(m2^{R_0 m}) = O(N \log N)$. An explicit family is obtained by taking as outer codes a sequence of Reed-Solomon codes C_1 of growing length

n over the alphabet of growing size $q = 2^\ell$. Performing algebraic (generalized minimum distance) decoding of the code C_1 with complexity $O(n^2)$, we obtain the error exponent [8]

$$E_{\mathrm{F}}(R) = \max_{R<R_0<\mathcal{C}} E_0(R_0, p)(1 - \frac{R}{R_0}).$$

The overall decoding complexity is $O(N^2)$.

The error exponent E_{F} was improved by Blokh and Zyablov [4] (see also [6]) who consider *multilevel* concatenated codes: this involves replacing the inner code C_0 by i different binary codes and the outer code C_1 by i Reed-Solomon codes of different rates. They obtain the exponent:

$$E_i(R) = \max_{R<R_0<\mathcal{C}} \left(\frac{i(R_0 - R)}{R_0 \sum_{j=1}^{i} \left(E_0(R_0 \frac{i-j+1}{i}, p)\right)^{-1}} \right) \tag{3}$$

under an $O(N^2)$ multistage decoding algorithm. We have $E_{\mathrm{F}}(R) = E_1(R)$, and

$$E_i(R) < E_{i+1}(R) \quad 0 < R < \mathcal{C}, i = 1, 2, \ldots.$$

We note that for code rates not too close to capacity, the quotient $E_0(R,p)/E_{\mathrm{F}}(R)$ is uniformly bounded from above by a function that depends only on p. Moreover, for $R \to 0$ we even obtain $E_{\mathrm{F}}(R) \sim E_0(R,p)$. The situation changes dramatically for $R \to \mathcal{C}$. Indeed, letting $\mathcal{C} - R = \varepsilon$, we obtain $E_0(R,p) \approx c(p)\varepsilon^2$, where $c(p)$ depends only on the channel. In contrast, $E_{\mathrm{F}}(R) = O(\varepsilon^3)$. Thus, the quotient $E_0(R,p)/E_{\mathrm{F}}(R) \to \infty$ as $\varepsilon \to 0$. This is also the case for the multilevel exponents: we have $\lim_{R\to\mathcal{C}}(E_0(R,p)/E_i(R)) = \infty$ for any finite i and even in the limit $i \to \infty$. Improving error exponents of low-complexity decoding algorithms in the range of code rates R close to $\mathcal{C}$ is therefore of most interest.

4 Expander graphs and expander codes

4.1 Expander graphs

Consider a balanced Δ-regular bipartite graph $G = [A \cup B, E]$ with the vertex set $A \cup B$, where $|A| = |B| = n$ and where every edge has one endpoint in A and one in B. Let λ be the second eigenvalue of G. We have $|E| = N := n\Delta$. Any vertex $v \in A$ ($v \in B$) will be called left (right) vertex.

For a vertex v and a subset of vertices S, denote by $d_S(v)$ the S-degree of v, i.e., the number of edges that connect v to some vertex of S.

We need one result from [15]:

Lemma 2 [15] *Let $S \subset A$ and $T \subset B$. Then the average degree $\overline{d}_{ST}$ of the subgraph induced by $S \cup T$ satisfies :*

$$\overline{d}_{ST} \leq \frac{2|S||T|}{|S| + |T|} \frac{\Delta}{n} + \lambda.$$

This lemma implies that expander graphs approximate well the behavior of random bipartite graphs. Namely, if λ is small compared to Δ, the size of the set T of right vertices whose S-degrees exceed substantially the expected S-degree of a vertex in a random graph, can be made arbitrarily small by choosing a sufficiently large Δ. This is made formal in the following lemma.

Lemma 3 *Let $S \subset A$ with $|S| = s = \sigma n$. Let $\alpha > \alpha_\sigma$ where $\alpha_\sigma = \frac{\lambda}{2\sigma\Delta}$. Let T be the subset of B defined by $T = \{v \in B,\ d_S(v) \geq (1+\alpha)\sigma\Delta\}$. Then we have :*

$$|T| \leq \frac{\alpha_\sigma}{\alpha - \alpha_\sigma}|S|.$$

PROOF. The number of edges of the graph $G_{S\cup T}$ induced by $S \cup T$ is at least $|T|(1+\alpha)\sigma\Delta$ and therefore the average degree $\overline{d}_{ST}$ satisfies

$$\frac{2|T|(1+\alpha)\sigma\Delta}{|S|+|T|}.$$

Applying Lemma 2 we therefore have :

$$\frac{2|T|(1+\alpha)\sigma\Delta}{|S|+|T|} \leq \frac{2|S||T|}{|S|+|T|}\frac{\Delta}{n} + \lambda$$

whence,

$$\begin{aligned} 2|T|(1+\alpha)\sigma\Delta &\leq 2|S||T|\frac{\Delta}{n} + \lambda(|S|+|T|) \\ (\alpha\sigma\Delta - \lambda/2) &\leq \frac{\lambda}{2}|S| \end{aligned}$$

and the result follows after some rearranging. □

4.2 Expander codes

Let $G = [A \cup B, E]$ be as above, with $\lambda \ll \Delta$. Let us fix an arbitrary order of the edges in E. For a vertex v this defines an ordering of edges $v(1), \ldots, v(\delta)$ incident to it. Given a binary vector $x \in \mathcal{H}_2^N$, this ordering induces a subvector $x_v = (x_{v(1)}, \ldots, x_{v(\Delta)})$.

To construct a linear code C associated with the graph G we also need two binary codes $C_0[\Delta, R_0\Delta, \delta_0\Delta]$ and $C_1[\Delta, R_1\Delta, \delta_1\Delta]$. An N-vector x is a codeword of C if and only if for every left vertex v the subvector x_v is a codeword in C_0 and for every right vertex w, the subvector x_w is a codeword in C_1. The code C has length $N = n\Delta$ and rate $R \geq R_0 + R_1 - 1$.

Codes associated with bipartite graphs were considered in [2], [5], [9], [12]–[16]. In particular, Sipser and Spielman [13] introduced an important idea of estimating properties of a simple iterative decoding procedure via spectral properties of the graph G. The main result of [13] is given by the following theorem.

Theorem 4 [13] *For any $\varepsilon > 0$ there exists a polynomial-time constructible family of codes with distance $\delta - \varepsilon$ and rate $1 - 2H(\sqrt{\delta})$ for which any $\alpha < \delta/48$ fraction of errors can be corrected by a circuit of size $O(N \log N)$ and depth $O(\log N)$. The complexity of a sequential implementation of this decoding is $O(N)$.*

This theorem gives codes with positive rate $R > 0$ for $0 < \delta < 0.011$. We note that [13] used the above family of expander codes with $C_0 = C_1$ but with a somewhat different decoding algorithm than the ones used in this paper. The present construction together with the decoding procedure (4) below was used in [15] to improve the fraction of correctable errors in Theorem 4 to $\delta/4$.

4.3 Replicated expander codes

A modification of the above construction was introduced in [2]. Namely, assume that every edge $(v, w) \in E$ is really a bundle of t parallel edges, each with one end in v and the other in w. Further, assume that C_0 is a $[t\Delta, t\Delta R_0]$ code and C_1 a $[t\Delta, t\Delta R_1]$ code. Both codes can be considered as binary linear codes, and also as q-ary additive codes (additive subgroups of $\mathbb{F}_q, q = 2^t$).

The purpose of introducing replication is to make use of the remark (1). In particular, with $C_0 = C_1$ we obtain:

Theorem 5 [2] *For any $R_0, 0 < R_0 < 1$, and $\varepsilon > 0$ there exists an expander code C of rate $R \geq 2R_0 - 1$ and relative distance $\delta = (1 - R_0)H^{-1}(1 - R_0) - \varepsilon$. Iterative decoding applied to this code corrects any $\alpha < \delta/4$ fraction of errors.*

This is an improvement over Theorem 4: in particular, we obtain codes C with positive rates for all $0 \leq \delta < 0.055$. This is the best result known to-date for the fraction of errors correctable in linear time with expander codes.

5 Attaining capacity with linear complexity

5.1 A simple bound

Replication also enables us to obtain good estimates of the error exponent of iterative decoding. Consider the following decoding algorithm of an expander code C. For a vector $y \in \mathcal{H}_2^N$ let $L(y)$ $(R(y))$ be the vector z such that for every $v \in A$ $(v \in B)$ the vector z_v is one of the codewords of C_0 (C_1) closest to y_v. Suppose that the transmitted zero vector is received as $y \neq 0$. Consider the decoding procedure of [15], [2]:

$$y^{(0)} = y, y^{(1)} = L(y^{(0)}), y^{(2)} = R(y^{(1)}), y^{(3)} = L(y^{(2)}) \dots \tag{4}$$

We assume that in computing $L(y^{(0)})$ the algorithm relies on the representation of C_0 as a binary code. In other words, for every left vertex v the associated binary subvector

$(y^{(0)}_{v(1)}, y^{(0)}_{v(2)}, \ldots, y^{(0)}_{v(t\Delta)})$ is decoded with the binary code $C_0[t\Delta, t\Delta R_0]$ into one of the closest codevectors of C_0. All the subsequent stages use the q-ary structure of C_1 and C_0. For instance, in the second stage this amounts to grouping consecutive groups of t bits of the vector $y^{(1)}$ into q-ary symbols. More precisely, for every right vertex w the q-ary subvector $y^{(1)}_w = (y^{(1)}_{w,1}, y^{(1)}_{w,2}, \ldots, y^{(1)}_{w,\Delta})$ associated to it can be written in binary representation as

$$[(y^{(1)}_{w(1)}, \ldots, y^{(1)}_{w(t)}), \ldots, (y^{(1)}_{w((i-1)t+1)}, \ldots, y^{(1)}_{w(it)}), \ldots, (y^{(1)}_{w((\Delta-1)t+1)}, \ldots, y^{(1)}_{w(\Delta t)})].$$

All the n Δ-subvectors y_w are independently decoded with the q-ary code C_1 according to the minimum of the q-ary Hamming distance.

The procedure stops after either having met a fixed point (i.e., when $y^{(i+2)} = y^{(i+1)} = y^{(i)}$ for some i) or after having made $O(\log N)$ steps.

Error exponents of this decoding algorithm for expander codes (replicated or not) were analyzed in [2]. The strongest result obtained in that paper is as follows:

Theorem 6 [2] *For a given rate R, any $\varepsilon > 0$ and $\alpha < 1$ there exists a polynomial-time constructible family of replicated expander codes of length N such that $P_e(C, p) \leq 2^{-\alpha N F_1(R,p)}$, where*

$$F_1(R, p) = \max_{R<R_0<C} E_0(R_0, p)(R_0 - R)/2 - \varepsilon. \tag{5}$$

The decoding complexity of these codes is the same as in Theorem 4.

The idea behind the proof is as follows. We take a code C_0 of rate $R_0 > R$ and a code C_1 of rate R_1 close to one. The first decoding step, $y^{(1)} = L(y^{(0)})$, removes most of the errors from the received word, which is possible because the code C_0 has relatively large distance. By taking a large but fixed Δ we ensure that, for every left vertex, the error exponent after the first step is close the random coding bound $E_0(R_0, p)$ (Sect. 3.2).

At the second step, most vertices correspond to subvectors containing less than $D_1/2 \approx (R_0 - R)/2$ errors and are therefore corrected. The expanding properties of the graph G ensure that the remaining atypical subvectors, call them "badly in error", are corrected at later iterations if their number is small enough. Small enough means smaller than $n\delta_1^q/2$ where $\delta_1^q = D_1/\Delta$ is the q-ary relative minimum distance of C_1. Specifically, Lemma 3 ensures that if at some point of the decoding procedure, the number of wrongly decoded subvectors (or vertices of the graph) is less than $n\delta_1^q/2$, then the number of subvectors in error must decrease geometrically at the next iteration (see [2] for details).

Summarizing, we simply upperbound the probability of a decoding error by the probability that the number of left subvectors remaining in error after the first iteration is more than $n\delta_1^q/2$.

5.2 Refined technique : overview

What precludes one from obtaining a better error exponent in this way is the fact that at the second step the proof relies on a very strong convergence condition, namely that most right vertices receive fewer than $D_1/2$ errors. Since the length of the code C_1 is a fixed constant, we could in principle involve more powerful decoding, but it is unclear how to bound its error rate. Moreover, incorrect codewords obtained in left vertices after the first iteration tend to be of one and the same Hamming weight (by Proposition 1). This information is also unclaimed in the proof of Theorem 6.

Our strategy will therefore be to study the probability that the number of vertices in error after the *second* iteration is sufficiently small for Lemma 3 to apply and again guarantee convergence of the decoding algorithm.

Our analysis is first based on the observation that, given the number σn of left vertices that are wrongly decoded by the first iteration, we know that the typical resulting error vector $y^{(1)}$ satisfies :

1. for almost every right vertex w, the right subvector $y_w^{(1)}$ has q-ary weight very close to $\sigma\Delta$. This is an application of Lemma 3.

2. almost every edge corresponds to a binary t-tuple which is either all-zero or of weight very close to $\delta_E(R_0)t$. This is an application of Proposition 1.

Given this information on the right subvectors $y_w^{(1)}$, we want to use a modified maximum-likelihood decoder for the code C_1 and estimate the probability that an error will occur at the second decoding iteration. However, the difficulty we face is that even if we have a typical error pattern for almost every $y_w^{(1)}$, we have little control over its probability distribution. To tackle this problem we will introduce partial randomization of the overall code construction and determine an error distribution for almost every right subvector $y_w^{(1)}$.

5.3 Refined technique : details

We modify somewhat the code family and the decoding algorithm. The codes are the same as in Section 4.3, but we will need the full power of Lemma 3, it is important that we choose bipartite graphs G with the smallest possible λ, e.g. Ramanujan graphs [11] with $\lambda = O(\sqrt{\Delta})$.

Furthermore, we introduce a small amount of randomization: namely, we consider an ensemble of expander codes obtained by choosing randomly and independently the coordinate ordering of every vertex subcode. We take the code C_0 to satisfy the condition of Prop. 1 and the code C_1 to satisfy Lemma 9 below. The decoding algorithm also consists of alternating left and right decodings. The first (left) decoding, $L(y^{(0)})$, is the same as in (4). The second (right) decoding uses a modified "max-likelihood" decoding adapted to a non-symmetric additive q-ary channel, which we describe in Section 5.4. Every subsequent decoding step uses standard decoding for the q-ary symmetric channel, described in Sect. 5.1.

Our first result, proved in the remainder of Sect. 5, is :

Theorem 7 *For a given rate R, there exists a polynomial time constructible family of replicated expander codes of length N, defined up to the orderings of the coordinates of the con-*

stituent codes, such that, given a random set ω of orderings of the constituent codes,

$$P_e(C,p,\omega) \le 2^{-NF(R,p)(1-\varepsilon(R,p))},$$

where $\varepsilon(R,p)$ is a function depending only on R and p such that $\varepsilon(R,p) < 1$ when $R < \mathcal{C}$ and $\varepsilon(R,p) \to 0$ when $R \to \mathcal{C}$ and where

$$F(R,p) = \max_{R<R_0<\mathcal{C}} E_0(R_0,p)\frac{R_0 - R}{H(\delta_{\mathrm{E}}(R_0))}.$$

The decoding complexity of these codes is the same as in Theorem 4.

NOTATION. Two relative minimum distances are used for C_0. The *binary* relative distance $\delta_0 = d_0/(t\Delta)$ where d_0 is the binary minimum distance of C_0, and the *q-ary* relative distance $\delta_0^q = d_0^q/\Delta$ where d_0^q is the q-ary minimum distance of C_0. By choosing Δ and t large enough (but fixed) we ensure that both distances are close to the GV bounds: $\delta_0 \approx \delta_{\mathrm{GV}}(R_0)$ and $\delta_0^q \approx \delta_{\mathrm{GV}}^{(q)}(R_0)$, and that the error probability of max-likelihood decoding of C_0 behaves as

$$P_e(C_0) \approx 2^{-t\Delta E_0(R_0,p)} \quad (R \to \mathcal{C}).$$

For the code C_1 we only consider its q-ary distance $D_1 = \delta_1^q\Delta$ of which we assume $\delta_1^q \approx \delta_{GV}^{(q)}(R_1)$.

Let γ_0 be the typical fraction of bits in error after decoding a corrupted codeword of C_0, given that a wrong codeword is output. By Proposition 1 we have $\gamma_0 = \delta_E(R_0)$. Let us denote by S the set of vertices of A in error after the first iteration and by T the set of vertices of B in error after the second iteration. Note that

$$\Pr[|S| = \sigma n] \lesssim \binom{n}{\sigma n} 2^{-t\Delta\sigma n E_0(R_0,p)} \lesssim \exp[-N\sigma E_0(R_0,p)]$$

since by choosing t and Δ sufficiently large the term $N^{-1}\log\binom{n}{\sigma n}$ can be made arbitrarily small. Here and henceforth the probability $\Pr[\cdot]$ is computed with respect to the random vector z received from the channel.

Lemma 3 implies that the event $|T| < \delta_0^q n/4$ is sufficient to ensure convergence of the decoding algorithm. This is because the number of vertices in error at the next iteration will have an order of magnitude than can be made as small as $(\mathrm{const}\cdot|T|/\sqrt{\Delta})$, which we can make smaller than $\delta_1^q n/2$. Under this condition the convergence of the algorithm after $O(\log n)$ iterations follows by [2, Prop. 2].

The error probability P_e under the iterative decoding algorithm is upperbounded by the probability that the convergence conditions do not hold. Let $\mathcal{F}$ be the set of multiple edges (or q-ary symbols), that are "heavy" after the first decoding iteration, i.e. have binary weight at least $(1+\alpha)^2\gamma_0 t$. We can therefore claim that for any ε, $\alpha > 0$,

$$P_e \leq \sum_{s \geq D_1/2} [P_0(\varepsilon, \alpha, s) + P_1(\varepsilon, \alpha, s)] \tag{6}$$

where

$$\begin{aligned} P_1(\varepsilon, \alpha, s) &= \Pr\Big[|S| = s, |\mathcal{F}| > \varepsilon s \Delta\Big] \\ P_0(\varepsilon, \alpha, s) &= \Pr[A_{s,\varepsilon}] \Pr\Big[|T| \geq \frac{\delta_0^q}{2} n \mid A_{s,\varepsilon}\Big], \end{aligned}$$

and where $A_{s,\varepsilon}$ stands for the event $A_{s,\varepsilon} = \{|S| = s, |\mathcal{F}| \leq \varepsilon s \Delta\}$.

The rest of the proof is upperbounding P_e. We first obtain a rough estimate of P_1. Denote $\sigma = s/n$. An edge may be "heavy" for two reasons. It may be incident to the set $S_{\mathcal{F}} \subset S$ made up of those vertices for which left decoding of the first iteration outputs a (wrong) codeword of C_0 of binary weight at least $(1+\alpha)\gamma_0 t\Delta$. Being incident to $S_{\mathcal{F}}$ is a rare event but given that, a high proportion of its edges are likely to be heavy. Or, it may be incident to $S \setminus S_{\mathcal{F}}$ but then being heavy is in itself a rare event. We write

$$P_1 \leq P_2 + P_3,$$

where

$$\begin{aligned} P_2 &= \Pr\Big[|S_{\mathcal{F}}| > \frac{\varepsilon \sigma n}{2} \mid |S| = \sigma n\Big] \\ P_3 &= \Pr\Big[|\mathcal{F}| \geq \varepsilon \sigma n \Delta \;\Big|\; |S_{\mathcal{F}}| \leq \frac{\varepsilon \sigma n}{2}, |S| = \sigma n\Big]. \end{aligned}$$

Lemma 8 *We have $P_2 \leq \Pr[|S| = s] 2^{-f_2(\delta_0,\alpha)\varepsilon\sigma N}$ and $P_3 \leq \Pr[|S| = s] 2^{-f_3(\delta_0,\alpha)\varepsilon\sigma N}$ where $f_2(\delta_0, \alpha)$ and $f_3(\delta_0, \alpha)$, both positive, stay bounded away from zero as $R \to \mathcal{C}$.*

PROOF. We have

$$P_2 \leq \Pr[|S| = s] \binom{\sigma n}{\varepsilon \sigma n/2} \Pr\Big[|\gamma_0 - \frac{|y|}{t\Delta}| \geq \alpha \gamma_0\Big]^{\varepsilon \sigma n/2}.$$

By Proposition 1 we can write $-\log \Pr[\cdot]^{\varepsilon \sigma n/2} \geq c(\alpha\gamma_0)\varepsilon\sigma N/2$. Compared to this, the binomial coefficient $\binom{\sigma n}{\varepsilon \sigma n/2}$ can be made to have a negligeable exponent by choosing $t\Delta$ large enough. This gives an exponent $f_2(\delta_0, \alpha) \approx c(\alpha\gamma_0)/2$.

Let us now evaluate P_3. Take any vertex v of $S \setminus S_{\mathcal{F}}$ and let γ be the proportion of bits in error of the corresponding output codeword of the code C_0. By definition of $S_{\mathcal{F}}$ we have $\gamma \leq (1+\alpha)\gamma_0$. Consider any multiple edge: the random grouping of the $t\Delta$ coordinate positions of C_0 into Δ multiple edges will give $\binom{t\Delta}{t}$ possible choices for the t edges that will form the multiple edge. The probability that the multiple edge contains more than $(1+\alpha)\gamma t$ bits in error is therefore

$$\pi = \sum_{t \geq k > (1+\alpha)\gamma t} \frac{\binom{\gamma t\Delta}{k}\binom{t\Delta - \gamma t\Delta}{t-k}}{\binom{t\Delta}{t}}. \tag{7}$$

The dominating term in this last sum occurs for the smallest k; after rearranging and neglecting terms non-exponential in t it is fairly routine to obtain that

$$\pi \approx \gamma^{(1+\alpha)\gamma t}(1-\gamma)^{(1-(1+\alpha)\gamma)t}\binom{t}{(1+\alpha)\gamma t}$$

which gives, for small α,

$$\pi \approx e^{-\frac{\gamma}{2(1-\gamma)}\alpha^2 t},$$

so that for fixed α and γ we obtain that π is exponentially small in t in a way that does not depend on Δ.

Next observe that if we compute the probability π' that a given multiple edge has weight larger than $(1+\alpha)\gamma$, given that i other given edges have weight larger than $(1+\alpha)\gamma$, then we will obtain the formula (7) with Δ replaced by $\Delta' = \Delta - i$ and γ replaced by $\gamma' < \gamma$. We will therefore have $\pi' \leq \pi$. With this in mind we write that

$$P_3 \leq \Pr[|S| = s]\binom{\sigma n\Delta}{\varepsilon\sigma n\Delta/2}\pi^{\varepsilon\sigma n\Delta/2}.$$

Again, by choosing t large enough, we obtain that $\binom{\sigma n\Delta}{\varepsilon\sigma n\Delta/2}$ has negligeable exponent and obtain the desired upper bound for P_3 with

$$f_3(\delta_0, \alpha) \approx \frac{1}{4\ln 2}\frac{\gamma_0}{1-\gamma_0}\alpha^2$$

for small α. □

We see that the exponents of P_2 and P_3, however small, stay larger than a constant times σN as $R \to \mathcal{C}$. It will therefore be apparent that when $R \to \mathcal{C}$, the probabilities P_2 and P_3 and hence P_1 are negligeable compared to P_0. Next, we upperbound P_0 in (6) by writing:

$$P_0 \leq \Pr[|S| = \sigma n]\Pr\left[|T| \geq \frac{\delta_0^q}{2}n \mid |S| = \sigma n, |S_{\mathcal{F}}| \leq \varepsilon\sigma n\right].$$

Assume that $P_0 = 2^{-NE(R)}$. Since we are only interested in exponential behavior of the sum in (6), we simply need to find the term P_0 with the smallest exponent. Therefore, letting

$$\Pr\left[|T| \geq \frac{\delta_0^q}{2}n \mid A_{s,\varepsilon}\right] = 2^{-NE_T}$$

and switching to exponents, we obtain

$$E(R) \geq \min_{\sigma \geq \delta_1^q/2}(E_0(R_0, p)\sigma + E_T). \tag{8}$$

Next we evaluate the exponent E_T: this is were the random choice of coordinate orderings comes in. Because the coordinate orderings of the right subcodes have been chosen randomly and are independent of the error vector (they might very well have been chosen *after* the first decoding step), we can argue that each right subcode is, independently of the others, submitted to an error vector chosen equiprobably among a set of error vectors with a given weight pattern. We therefore need to estimate the typical weight pattern of the error vector on a right vertex after the first decoding step. By Lemma 3, all but a negligeable fraction of right vertices have more that $(1+\alpha)\sigma\Delta$ edges incident to S, (recall that α_σ in Lemma 3 can be made as small as we want because $\lambda = O(\sqrt{\Delta})$ and Δ can be taken to be arbitrarily large). This means that almost every right vertex w has an error vector of weight at most $(1+\alpha)\sigma\Delta$. Let us establish a further property of these error vectors.

Let Q' be the subset of Q defined by the set of those q-ary symbols whose binary representation is of weight at most $(1+\alpha)^2\gamma_0 t$. We now argue that almost all of subvectors $y_w^{(1)}$ have most of their symbols in the subset Q'.

Indeed, $\mathcal{F}$ is exactly the set of symbols that do not belong to Q' after the first decoding iteration. By Lemma 8 we can assume that $|\mathcal{F}| \leq \varepsilon\sigma n\Delta$ because the opposite event occurs with an exponent that would be much greater that the others (and thus negligeable probability) when $R \to \mathcal{C}$.

By the Markov inequality, the number of right vertices w that have more than $\beta\sigma\Delta$ incident edges that correspond to a symbol in $Q \setminus Q'$ is less than $\varepsilon n/\beta$. Now we can simultaneously choose β and ε such that β, ε, and $\lambda = \varepsilon/\beta$ are all small. Summarizing, we obtain that almost all vertices of B, (i.e. $|B|(1-\eta)$ of them) have an error vector of weight at most $(1+\alpha)\sigma\Delta$ and such that all but $\beta\sigma\Delta$ of its nonzero symbols belong to Q', where β is again arbitrarily small. We now claim that

$$E_T \geq (\delta_0^q/4 - \eta)E_1, \tag{9}$$

where E_1 is an error exponent for the right decoder, given that the error vector has the above pattern. To see this, assume the worst, namely that all $\eta|B|$ vertices with their error vector of the wrong pattern will be wrongly decoded. The claim now consists of saying that if not more than $n\delta_0^q/4$ right vertices are in error then the subsequent decoding steps must converge correctly. This in turn follows from Lemma 3 which implies that the number of left vertices that have more than $\Delta\delta_0^q/2$ edges incident to T (the only ones that can be wrongly decoded at the third iteration) can, by choosing λ/Δ small enough, be made sufficiently small (smaller than $n\delta_1^q/4$ for example) so that the number of vertices in error will shrink geometrically at each iteration as in [15, 2] or section 5.1. The choice of the fraction 1/4 is arbitrary and can be replaced by any number less than 1/2.

We next evaluate E_1.

5.4 Decoding C_1

The right decoder assumes that the q-ary error vector has weight not more than $\sigma'\Delta$ with $\sigma' = \sigma(1+\alpha)$, and that among its nonzero symbols, not more than $\beta\sigma\Delta$ do not belong to the subset Q'. If it does not find any codeword that fits this hypothesis it returns an arbitrary codeword (say, a random one). We have

$$|Q| = q = 2^t \quad \text{and} \quad |Q'| = \sum_{k \le \gamma t} \binom{t}{k} \approx 2^{H(\gamma)t}$$

where $\gamma = (1+\alpha)^2\gamma_0$.

Lemma 9 *There exists a $[\Delta, R_1\Delta]$ linear q-ary code C_1 such that for sufficiently large Δ and t and any $\beta > 0$*

$$E_1 \ge \delta_1^q - \sigma' H(\gamma) - 2\beta\sigma.$$

PROOF. A nonzero vector falls in a random q-ary linear $[\Delta, R_1\Delta]$ code C_1 with probability $q^{-\Delta}(q^{\Delta R_1} - 1)$. Hence the average number of codewords of weight $i > 0$ in C_1 with at most $\beta\sigma\Delta$ nonzero symbols in $Q \setminus Q'$ equals

$$\mathsf{E} A_{i,\beta} \le q^{-\Delta(1-R_1)} \binom{\Delta}{i} \sum_{j=0}^{\beta\sigma\Delta} \binom{i}{j} |Q'|^{i-j} (q-1)^j.$$

When t is large, by (1) we have $1 - R_1 \approx \delta_1^q$, and the exponents of both binomial coefficients are small. Therefore the above inequality can be rewritten as

$$\mathsf{E} A_{i,\beta} \lesssim q^{\Delta(-\delta_1^q + H(\gamma)i + \beta\sigma)}.$$

We now compute the error probability for the right decoder under the condition that the input vector is a random vector of the required pattern. It can be bounded above by the probability that such a vector covers at least half the symbols of a given vector of weight $i > 0$ and of the above pattern, which is not more than

$$q^{\Delta(-H(\gamma)i/2 + \beta\sigma + \varepsilon(\Delta))},$$

where $\varepsilon(\Delta) > 0$ can be made smaller than any given number by an appropriate choice of Δ. The probability that the random error vector covers half a codeword of weight i is not more than

$$A_{i,\beta} q^{\Delta(-H(\gamma)i/2 + \beta\sigma + \varepsilon(\Delta))}. \tag{10}$$

As usual, we can choose a code C_1 such that every $A_{i,\beta}$ is not more than $\mathsf{E} A_{i,\beta}$ times a polynomial in n. We obtain therefore that the maximum of (10) is obtained when i is as large as possible, namely $i = 2\sigma'\Delta$.

Switching to exponents we obtain

$$E_1 \ge \delta_1^q - \sigma' H(\gamma) - 2\beta\sigma.$$

$\square$

As $R \to \mathcal{C}$, the first term in (8) tends to zero while the second remains bounded away from zero. Together with (9) this enables us to claim that the lower bound (8) is minimized for

$$\sigma \to \frac{\delta_1^q}{H(\gamma_0)} = \frac{1 - R_1}{H(\gamma_0)} = \frac{R_0 - R}{H(\gamma_0)}. \tag{11}$$

Substituting this value of σ in (8), we obtain the exponent $F(R, p)$ of the theorem. □

6 A further improvement of the exponent. Borrowing from turbo-codes

In this section we show how to modify slightly the code construction to improve Theorem 7 to

Theorem 10 *For a given rate R, there exists a polynomial time family of (replicated, generalized) expander codes of length N, defined up to the orderings of the coordinates of the constituent codes, such that, given a random set ω of orderings of the constituent codes,*

$$P_e(C, p, \omega) \leq 2^{-NF(R,p)(1-\varepsilon(R,p))},$$

where $\varepsilon(R, p)$ is a function depending only on R and p such that $\varepsilon(R, p) < 1$ when $R < 1 - H(p)$ and $\varepsilon(R, p) \to 0$ when $R \to 1 - H(p)$ and where

$$F(R, p) = \max_{R < R_0 < \mathcal{C}} E_0(R_0, p) \frac{1 - R/R_0}{H(\delta_E(R_0))}.$$

The decoding complexity of these codes is the same as in Theorem 4.

The idea is based on the following remark. Every bit of an expander code belongs to *two* constituent codes, C_0 and C_1. Turbo-codes, on the other hand have the property that only the information bits belong to two constituent codes, while the redundancy bits belong to a single one.

To mimic this structure in the expander code context, let us modify the code construction that we have used in the following way : we keep the same expander graph G as before but append some extra edges, namely we add $at\Delta$ "dangling" edges to every one of the left vertices of A. By "dangling" we mean that we introduce as many vertices (of degree 1) as we introduce edges. The right side of the graph is left untouched, so is the constituent code C_1. The left constituent code C_0 is modified only inasmuch as it is now a randomly chosen code of rate R_0 and length $t\Delta(1 + a)$ (instead of $t\Delta$). We now assign coordinate positions to C_0 in such a way that all information bits are assigned edges of the bipartite graph G : in other words all the "dangling" edges must correspond to redundancy bits. For this to be possible the number of additional edges must be such that $R_0 t\Delta(1 + a) \leq t\Delta$: to this end we choose $a = R_0^{-1}(1 - R_0 - \varepsilon)$ so that $R_0 t\Delta(1 + a) = t\Delta(1 - \varepsilon)$.

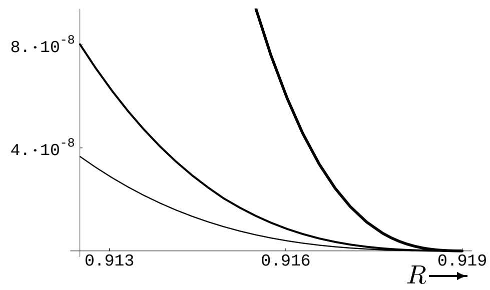

Figure 1: Error exponents of expander and concatenated codes in the neighborhood of capacity of a BSC with $p = 0.01, \mathcal{C} \approx 0.919$. Bottom to top: basic construction of expander codes, Theorem 6; the exponent E_F of concatenated codes; improved bound of Theorem 10.

The decoding procedure is hardly modified: the first decoding step is again max-likelihood decoding of C_0. The second decoding step is unchanged. The third decoding step and subsequent unevenly numbered decoding steps are slightly modified in the sense that q-ary max-likelihood decoding is again applied, not to the code C_0 however, but to the shortened code C_0^S obtained from C_0 by throwing away the redundancy bits corresponding to the dangling edges. The code C_0^S has q-ary length Δ and the same dimension as C_0 i.e. rate $R_0(1+a) = 1-\varepsilon$. Throughout the rest of the decoding procedure the dangling edges are ignored and their value is recovered only at the very end if the algorithm has converged (and has therefore recovered all the information bits). An examination of the convergence conditions of Section 5.3 shows that everything holds in the modified case, with the exception that δ_0^q must now be understood to refer to the q-ary relative minimum distance, not of C_0 but of the shortened code C_0^S. We will be careful to choose ε to be sufficiently big so that we can again claim that the first term in (8) dominates the second term. Nevertheless this does not stop us from having $\varepsilon \to 0$ when $R \to \mathcal{C}$.

As a result, the conditions for convergence are essentially unchanged but now the overall rate R of the code is higher. Indeed, the new length is now $N = nt\Delta(1+a)$ and the new redundancy is

$$(1-R)N = n(t\Delta(1-R_1) + t\Delta(1+a)(1-R_0))$$

which gives, after some rearranging, the expression for the new rate

$$R = R_0R_1 - \frac{\varepsilon}{1-\varepsilon}R_0(1-R_1).$$

Together with (8)-(11) this implies Theorem 10.

We include a sketch of the exponents discussed in the paper in Fig 1 for code rate R in the neighborhood of capacity.

7 Concluding comments.

For code rates close to capacity the result of Theorem 10 improves substantially on our earlier result [2]. Interestingly, for $R \to \mathcal{C}$ we also have $F(R,p) > E_{\mathrm{F}}(R)$, i.e., expander codes have an exponentially smaller error rate than (quadratic-time-decodable) Forney's concatenated codes. A common point that these two families share is the use of two constituent codes; however in the expander code construction both codes are binary, while in the concatenated scheme the q-ary code C_1 has a strong algebraic structure.

We note that in the study of minimum distance of codes there is a substantial difference between explicit families of codes and randomized ensembles. This difference does not play such a prominent role when we study decoding error exponents. Indeed, in this context the focus is on decoding performance and the goal is to estimate the error probability of some explicit decoding algorithm. It hardly matters whether this probability is computed for a fixed code and a random error vector, or the product of a random code and a random error vector.

Comparing Theorem 10 with the error exponent of multilevel concatenations (3), we notice that $E_i(R) > F(R,p)$ starting from some finite (not too large) value of i that depends on the channel crossover probability p.

The following research problem suggests itself rather naturally: is it possible to improve the error exponents of expander codes by using several constituent codes similarly to the improvement (3) by Blokh and Zyablov of Forney's exponent $E_{\mathrm{F}}(R)$?

More generally, now that concatenated codes finally have a rival, it should be very interesting to see whether expander codes (or some enlarged family) ultimately have the potential to provide the best constructive exponents. An even more general question is whether there exist polynomial-time decodable code families with error exponent $E(R)$ such that the quotient $E_0(R,p)/E(R)$ stays bounded as $R \to \mathcal{C}$.

Finally, we observe that the study of exponents provides an interesting theoretical framework for trying to discern which properties of codes are relevant for iterative decoding. Note that criteria such as minimum distance could not have lead to the construction of Section 6.

Appendix: Proof of Prop. 1

We will study properties of an ensemble $\mathcal{A}$ of binary linear codes defined by $(N-K)\times N$ parity-check matrices whose elements are chosen independently with $P(0) = P(1) = 1/2$.

Let $K = RN$. We write $f(N) \cong g(N)$ if $\lim_{N\to\infty} \frac{1}{N} \log \frac{f(N)}{g(N)} = 0$.

Let $C \in \mathcal{A}$ be a linear code and A_w be the number of vectors of weight w in it. It is easy to see that

$$\mathsf{E}A_w = \binom{N}{w}(2^K - 1)2^{-N}.$$

By the Markov inequality there exist codes with

$$A_w \leq n\binom{N}{w}2^{K-N} \quad (w = 1, 2, \ldots, N). \tag{12}$$

We continue with a technical lemma.

Lemma 11 *Let* $S_j = \{x \in \mathcal{H}_2^N : |x| = j\}$, $r \leq w \leq N/2$,

$$M(w,r) := \sum_{c \in S_w} \big|\{y \in S_r : \mathrm{d}(y,c) \leq r\}\big|.$$

Then for $N \to \infty$, $r = \rho N$, $M(w,r) \lesssim \binom{n}{r}^2$ *with equality*

$$M(w,r) \cong \binom{n}{r}^2$$

only for $\frac{w}{N} \sim 2\rho(1-\rho)$.

PROOF.

$$M(w,r) = \binom{N}{w}\sum_{i=w/2}^{r}\binom{w}{i}\binom{N-w}{r-i}.$$

The unconstrained maximum of the summation term is attained when $i = wr/N < w/2$. Hence we write

$$\begin{aligned}
M(w,r) &\cong \binom{N}{w}\binom{w}{w/2}\binom{N-w}{r-w/2} \\
&= \binom{N}{r}\binom{N-r}{w/2}\binom{r}{w/2} \qquad \text{(rewriting the multinomial coefficient)} \\
&\lesssim \binom{N}{r}\binom{N-r}{r(1-\rho)}\binom{r}{\rho(N-r)}.
\end{aligned}$$

Finding the maximum on w in the last step calls for some explanation. For a fixed vector y of weight r we count the number of weight-w vectors c with $\mathrm{d}(y,c) = r = \rho N$. This number is maximized if c is a typical vector obtained after n independent drawings from the binomial probability distribution given by $P[y+c=1] = \rho$, $P[y+c=0] = 1-\rho$. Hence the maximizing argument is given by $w/2 = r(1-\rho)$. Substituting it, we obtain

$$\begin{aligned}
M(r(1-\rho),r) &\cong \exp[N(H(\rho) + (1-\rho)H(1-\rho) + \rho H(1-\rho))] = \exp(2NH(\rho)) \\
&\cong \binom{N}{r}^2.
\end{aligned}$$

□

Now let us compute the error exponent of maximum likelihood decoding for a code C of rate R with weight distribution as in (12). Suppose that $y \in \mathcal{H}^N$ is a vector received from the channel BSC(p) and that the transmitted vector is the all-zero one. Let $d = \delta_{\rm GV}(R)N$. The probability of decoding error can be written as

$$P_e \le \Sigma_1 + \Sigma_2, \tag{13}$$

where

$$\Sigma_1 = \sum_{w=d}^{2d} \sum_{r=d/2}^{d} A_w \Pr[\mathcal{E}_w \mid |y| = r], \quad \Sigma_2 = \Pr[|y| \ge d];$$

d is the distance of C and where $\mathcal{E}_w$ denotes the event that y is decoded incorrectly to a given codeword of weight w. We have

$$\Pr[\mathcal{E}_w \mid |y| = r] = \binom{N}{r}^{-1} \sum_{i=w/2}^{w} \binom{w}{i}\binom{N-w}{r-i}.$$

As above, the sum on w is dominated by the first term, and we have

$$\Pr[\mathcal{E}_w \mid |y| = r] \cong \frac{\binom{w}{w/2}\binom{N-w}{r-w/2}}{\binom{N}{r}}.$$

Hence

$$\Sigma_1 \cong 2^{-n(1-R)} \max_{d/2 \le r \le d} \max_{d \le w \le 2d} \binom{N}{w}\binom{w}{w/2}\binom{N-w}{r-w/2} p^r (1-p)^{N-r}.$$

Let $w = \omega n, r = \rho n$. By Lemma 11 the unconstrained maximum on $w = \omega N$ is attained for $\omega = 2\rho(1-\rho)$, and then

$$\begin{aligned} \Sigma_1 &\cong 2^{-N(1-R)} \max_{d/2 \le r \le d} \binom{n}{r}^2 p^r (1-p)^{N-r} \\ &= \max_{\delta_{\rm GV}(R)/2 \le \rho \le \delta_{\rm GV}(R)} \exp[-N(D(\rho \| p) + (1-R) - H(\rho)]. \end{aligned}$$

The unconstrained maximum on ρ on the right-hand side (the minimum of the exponent) is attained for $\rho = \rho_0$ (2). We are interested in the case of $R \ge R_{\rm crit}$, i.e., $\rho_0 \ge \delta_{\rm GV}(R)$. Then both Σ_1 and Σ_2 behave (in the $\cong$ sense) as $\exp(-ND(\delta_{\rm GV}(R)\|p))$. Moreover, the exponent $D(\delta_{\rm GV}(R)\|p)$ is attained for $\rho = \delta_{\rm GV}(R)$ and hence for $\omega = \delta_E(R)$.

To prove that $\delta_E(R)$ is indeed the typical relative weight of incorrectly decoded codewords it is now enough to argue that $P_e \cong \Sigma_1$, i.e., that the estimate (13) is in fact an asymptotic equality. This is the so-called sphere-packing bound [10, p. 164] which states that for *any* code with the same rate R we must have $P_e \ge \Sigma_2$.

Thus, for $R_{\rm crit} < R < \mathcal{C}$ the error vectors that lead to a decoding error will be of weight $N(\delta_{\rm GV}(R) \pm \varepsilon_1)$ in all cases of decoding error except a $2^{-g(\varepsilon_1)N}$ fraction of them (here $g(\cdot)$ is some positive function, not necessarily the same in different expressions). Moreover, the weight of incorrectly decoded codewords will be equal to $N(\delta_E(R) \pm \varepsilon_2)$ in all cases of decoding error except a $2^{-g(\varepsilon_2)N}$ fraction of them. Since all error vectors of any given weight are equiprobable we can now finish the proof by a law-of-large-numbers-type argument.

References

[1] A. Barg, "Complexity issues in coding theory," in *Handbook of Coding Theory* (V. Pless and W.C. Huffman, Eds.), Vol.I, Amsterdam: Elsevier Science (1998), pp. 649–754.

[2] A. Barg and G. Zémor, "Error exponents of expander codes," *IEEE Trans. Inform. Theory* **48**, no.6 (2002), pp. 1725–1729.

[3] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes", in *Proc. of Int. Com. Conf.* (ICC'93), pp. 1064–1070, 1993.

[4] E. L. Blokh and V. V. Zyablov, *Linear Concatenated Codes*, Moscow: Nauka Publishers (1982), in Russian.

[5] L. Decreusefond and G. Zémor, "On the error-correcting capabilities of cycle codes of graphs," *Combinatorics, Prob. and Comp.*, vol. 6 (1997), 27–38.

[6] I. Dumer, "Concatenated codes and their multilevel generalizations," in *Handbook of Coding Theory* (V. Pless and W.C. Huffman, Eds.), Vol.II, Amsterdam: Elsevier Science (1998), pp. 1911-1988.

[7] P. Elias, *Coding for noisy channels*, IRE Conv. Rec., Mar. 1955, pp. 37–46. Reprinted in D. Slepian, Ed., Key papers in the development of information theory, IEEE Press, 1974, pp. 102–111.

[8] G. D. Forney, Jr., *Concatenated Codes*, Cambridge, MA: MIT Press (1966).

[9] R. G. Gallager, *Low-Density Parity-Check Codes.* Cambridge, MA: MIT Press (1963).

[10] R. G. Gallager, *Information Theory and Reliable Communication*, New York: J. Wiley & Sons (1968).

[11] A. Lubotsky, R. Philips and P. Sarnak, "Ramanujan graphs," *Combinatorica*, **8** (3) (1988) 261–277.

[12] G. A. Margulis, "Explicit constructions of graphs without short cycles and low density codes," *Combinatorica*, **2** (1) (1982), 71–78.

[13] M. Sipser and D. A. Spielman, "Expander codes", *IEEE Trans. on Inf. Theory*, **IT-42**, no 6, pp. 1710–1722, 1996.

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

[15] G. Zémor, "On expander codes" *IEEE Trans. Inform. Theory*, **IT-47**, pp. 835–837, 2001.

[16] V. V. Zyablov and M. S. Pinsker, "Estimation of the error-correcting complexity of Gallager low-density codes," *Problems Inform. Transm.*, **11** (1) (1975), 26–36 (in Russian) and 18–28 (English Translation).