Concatenated Codes: Serial and Parallel

Alexander Barg, *Senior Member, IEEE,* and Gilles Zémor, *Member, IEEE*

***Abstract*—An analogy is examined between serially concatenated codes and parallel concatenations whose interleavers are described by bipartite graphs with good expanding properties. In particular, a modified expander code construction is shown to behave very much like Forney's classical concatenated codes, though with improved decoding complexity. It is proved that these new codes achieve the Zyablov bound δ_Z on the minimum distance. For these codes, a soft-decision, reliability-based, linear-time decoding algorithm is introduced, that corrects any fraction of errors up to almost $\delta_Z/2$. For the binary-symmetric channel, this algorithm's error exponent attains the Forney bound previously known only for classical (serial) concatenations.**

***Index Terms*—Bipartite-graph codes, concatenated codes, expander codes, Forney bound, min-sum algorithm, soft-decision decoding, Zyablov bound.**

I. Introduction

CODE concatenation is the method of choice for constructing code families with good asymptotic as well as practical properties. The general idea can be traced back to Elias's product code construction, developed later by Forney [1] into a simple and powerful concatenated code construction. Another important idea in the area of combining codes is due to Tanner [2]; it is now seen as generalizing Elias's product codes by allowing arbitrary permutations ("interleavers") to be used in the choice of ways for the information symbols to appear in check equations of the constituent codes. Codes associated with bipartite graphs form an instance of the general construction termed "parallel concatenation" which covers also turbo-like code constructions. The counterpart of this construction, termed "serial concatenations" came into use recently to describe concatenated codes in the sense of Forney and other similar constructions (see, e.g., [3]).

The main line of thought pursued in this paper is to examine the analogy between the two versions of concatenated codes, those by Forney and Tanner. The latter class, in particular, contains expander codes of Sipser and Spielman [4] and their later modifications. One of our goals will be to prove that, roughly speaking, "whatever can be achieved with Forney's concatenated codes and their generalizations, can be also achieved with Tanner codes and their appropriate generalizations," except that for Tanner codes the decoding complexity is lower by a factor proportional to the code length, going down from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$.

In Section II, we recall serial and parallel concatenations. While the idea of describing parallel concatenated codes in a way similar to the serial construction is implicit in numerous earlier works, we believe this is the first time it is presented explicitly. Below, we offer a description of parallel codes that strives to expose the analogy to the fullest possible degree, the difference being essentially imposed by the decoding procedure.

In Section III, we develop a modified construction of parallel concatenations. A new decoding algorithm which employs soft reliability information is presented in Section III-C. The main result of this paper is a proof, in Section III-D, that Forney's bound on the error exponent of serial codes is attainable with parallel codes whose interleavers are defined by expander graphs. This proof involves a nontrivial generalization of the link [4], [5] between the convergence of iterative decoding and the spectrum of the underlying graph. As a byproduct, it is proved in Section III-E that parallel concatenation achieves the Zyablov bound δ_Z on the minimum distance, defined in (1) below, and corrects a $(1/2)\delta_Z$ fraction of errors.

The parallel concatenation method discussed here was briefly introduced in [6, Sec. 6] where it was shown that it can actually surpass Forney's exponent, but by methods valid only in the range of code rates close to capacity, and at the expense of randomizing the encoding procedure (but keeping decoding deterministic and linear time). In concurrent research, Guruswami and Indyk [7] introduced a somewhat different construction of expander codes that they have shown to attain the Zyablov bound and correct a $(1/2)\delta_Z$ fraction of errors. While in retrospect the two constructions come close, a notable difference lies in the decoding method which relies on generalized minimum distance (GMD) decoding in [7], whereas the present work uses the min-sum version of message passing.

Let us introduce some notation. We write $W[\Delta, k, d]$ to denote a linear code of length Δ, dimension k, and distance d. The alphabet of the code can be binary or q-ary for some value of $q = 2^t$. The Hamming weight $\mathrm{w}(\boldsymbol{x})$ is the number of nonzero coordinates of the vector $\boldsymbol{x}$. By $\mathrm{d}(\boldsymbol{x}, \boldsymbol{y}) = \mathrm{w}(\boldsymbol{x} - \boldsymbol{y})$ we denote the Hamming distance.

We assume transmission over a binary-symmetric channel (BSC) with transition probability p, denoted below by $\mathrm{BSC}(p)$. Let $h(x) = -x \log x - (1-x) \log(1-x)$ denote the binary entropy function (the base of the logarithms is 2 throughout). Let $\delta_{\mathrm{GV}}(R) = h^{-1}(1 - R)$ denote the Gilbert–Varshamov (GV) relative distance for the rate R.

Manuscript received September 18, 2003; revised November 15, 2004. The work of A. Barg was supported in part by the National Science Foundation under Grant CCR-031096 and by ENST, Paris, France, during a visit in June 2003. Part of this work was done while G. Zémor was visiting DIMACS, Rutgers University, Piscataway, NJ, in August 2002. The material in this paper was presented in part at the 2003 IEEE International Symposium on Information Theory, Yokohama, Japan, June/July 2003.

A. Barg was with DIMACS, Rutgers University, Piscataway, NJ 08854 USA. He is now with the Department of Electrical and Computer Engineering, University of Maryland, College Park, MD 20742 USA (e-mail: abarg@ieee.org).

G. Zémor is with the École Nationale Supérieure des Télécommunications, 75 634 Paris 13, France (e-mail: zemor@enst.fr).

Communicated by K. A. S. Abdel-Ghaffar, Associate Editor for Coding Theory.

Digital Object Identifier 10.1109/TIT.2005.846392

Throughout the paper, we will consider probabilities of different events expressed in the form $\exp(-NE)$ where N is a large positive number and E a nonnegative function of the code parameters. In such situations, E will be called the *exponent* of the probability or the error exponent if the event we have in mind is a decoding error.

Finally, let $E_0(R,p)$ be the "random coding exponent" [8]. For $R_{\text{crit}} \le R \le \mathcal{C}$ we have

$$E_0(R,p) = E_{\text{sp}}(R,p) := D(\delta_{\text{GV}}(R)\|p)$$

where

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

$R_{\text{crit}} = 1 - h(\rho_0)$, $\rho_0 = \sqrt{p}/(\sqrt{p}+\sqrt{1-p}))$ is the critical rate, and $\mathcal{C} = 1 - h(p)$ is the channel capacity. For rates $0 \le R \le R_{\text{crit}}$ the random coding exponent has the following form:

$$E_0(R,p) = -\,\delta_{\text{GV}}(R)\log 2\sqrt{p(1-p)} \qquad (0 \le R \le R_x)$$
$$E_0(R,p) = D(\rho_0\|p) + R_{\text{crit}} - R \qquad (R_x \le R \le R_{\text{crit}})$$

where $R_x = 1 - h(2\rho_0(1-\rho_0))$. It is well known that for large code length there exist families of linear codes of rate R whose distance approaches the GV bound and the error probability of maximum-likelihood decoding behaves as $\exp(-N(E_0(R,p)-\epsilon))$, where $\epsilon > 0$ is an arbitrarily small quantity.

II. Code Concatenations: Context

A. Serial Concatenations

Let $A[\Delta, R_0\Delta, d_0]$ and $B[\Delta, R_1\Delta, d_1]$ be two additive binary codes of length Δ. The product code C is formed by taking a direct product $A \otimes B$; it therefore has length Δ^2, rate R_0R_1, and distance d_0d_1. A typical codeword $\boldsymbol{c}$ of the code C can be thought of as a $\Delta \times \Delta$ matrix in which each row is a codeword in B and each column a codeword in A. These conditions lead to an obvious form of the parity-check matrix H of C: for every $i = 1,\dots,\Delta$ the subset of coordinates $\{c_\ell : \ell = (i-1)\Delta+1,\dots,i\Delta\}$ must satisfy the parity-check equations of the code B; for every $j = 1,\dots,\Delta$ the subset of coordinates $\{c_\ell : \ell = j, \Delta+j,\dots,(\Delta-1)\Delta+j\}$ must satisfy the parity-check equations of the code A. Generally, this set will contain some dependent equations, nevertheless it is a convenient way of representing H. For definiteness we call the row code the *outer code* and the column code the *inner code*.

Serially concatenated codes in the sense of Forney [1] generalize this construction as follows. Let $m = R_0\Delta$ be the dimension of the (inner) code A. The outer code B will now be a Q-ary $[n, k = R_1n, d_1]$ linear code with $Q = 2^m$. A typical codeword of the concatenated code C can be thought of as a binary $\Delta \times n$ matrix in which the ith column, $1 \le i \le n$, represents an encoding with the code A of the binary representation of the ith symbol of the Reed–Solomon (RS) codeword. The length of the code C equals $N = \Delta n$, the rate is $R = R_0R_1$. The minimum distance in general is not easy to compute; obviously, it is at least $D = d_0d_1$, but can be much greater than that. The number D is called the *designed distance* of the code C.

A somewhat more general version of this construction is obtained if instead of one Q-ary code we use m independent encodings with a binary outer code B of length n. This results into m codewords of the outer code, viewed as an nm bit string. Then the bits of this encoding are permuted in some way and encoded to form n codewords of the code A. (For instance, in the previous paragraph the permutation is simply $i \mapsto i(\bmod n)+1$, $i = 1,\dots,nm$.) The resulting bit string is, by definition, a codeword in the code C which is a serial concatenation of the codes A and B. Serially concatenated codes in the sense of this description were studied in the 1990s, see [3] and references therein. Later in our discussions of serial concatenations we confine ourselves to the original construction of [1].

To describe decoding of serial concatenations let $\boldsymbol{z} = (\boldsymbol{z}_1,\dots,\boldsymbol{z}_n) \in (\mathbb{F}_2^{\Delta})^n$ be the vector received from the channel. For the purposes of analysis, we assume that the all-zero vector was transmitted (this assumption will be taken throughout the paper in similar situations). First, for every j, the vector $\boldsymbol{z}_j$ is decoded with the code A using maximum-likelihood decoding. The result is a collection $\{\boldsymbol{a}_1,\dots,\boldsymbol{a}_n\}$ of codewords of A. Next, some decoding with the code B is applied to the message parts of these codewords, viewed as a received vector for the code B. If one wishes to keep the overall decoding complexity of C polynomial, the best results under this decoding procedure are obtained if B is an algebraic code (say, an RS code) decoded with the so-called GMD decoding algorithm [1] (see [9, pp. 1964–1967] for applications to serially concatenated codes). Then the decoding procedure described corrects every combination of errors of weight up to half the designed distance D. To maximize the value of D, choose A to be a linear binary code meeting the GV bound and B an RS code. Then, as N increases, the relative distance $\delta = d/N$ approaches the Zyablov bound [10]

$$\delta_Z(R) = \max_{R \le R_0 \le 1} \delta_{\text{GV}}(R_0)\frac{R_0 - R}{R_0}. \tag{1}$$

For this code family, concatenated decoding corrects a $\frac{1}{2}\delta_Z(R)$ fraction of errors. This bound is attained with a $\mathcal{O}(N^2)$ decoding complexity.

The error exponent attained under concatenated decoding satisfies Forney's bound [1]

$$E_{\text{F}}(R,p) = \max_{R \le R_0 \le \mathcal{C}} E_0(R_0,p)\frac{R_0 - R}{R_0}. \tag{2}$$

The decoding complexity is again $\mathcal{O}(N^2)$.

B. Parallel Concatenations: Bipartite-Graph Codes

Let $G = (V_0 \cup V_1, E)$ be a bipartite Δ-regular graph with the vertex set $V_0 \cup V_1$, $|V_0| = |V_1| = n$, where every edge in E has one end in V_0 and the other in V_1. Let us choose an arbitrary ordering of the $N = n\Delta$ edges of the graph which will be fixed throughout the construction. For a given vertex $v \in G$ this defines an ordering of edges $v(1), v(2),\dots,v(\Delta)$ incident to it. We denote this subset of edges by $E(v)$. For a vertex v in one part of G the set of vertices in the other part adjacent to v will be also called the *neighborhood* of the vertex v, denoted $\mathcal{N}(v)$.

Let $A[\Delta, R_0\Delta, d_0]$ and $B[\Delta, R_1\Delta, d_1]$ be two additive binary codes of length Δ. Let us define the *bipartite-graph code* $C = C(G; A, B)$ to be the binary code of length N consisting of all vectors $\boldsymbol{c}$ such that, for every vertex $v \in V_0 (v \in V_1)$, the subvector $\boldsymbol{c}(v) = (c_{v(1)}, c_{v(2)}, \dots, c_{v(\Delta)})$ belongs to code A (resp., B). The redundancy of the overall code C is not more than the sum of the redundancies of all the vertex subcodes, therefore, the rate of the code C is at least $R = R_0 + R_1 - 1$ where R_0 is the rate of A and R_1 is the rate of B.

Viewed from a somewhat different perspective, this construction simply suggests to encode the message symbols of the code into n codewords of the code A and independently encode the same message symbols into n codewords of the code B. Iterative decoding of parallel concatenations performs well if between the two encoding stages, the message stream is permuted according to some rule (called an *interleaver*) [11]. The bipartite graph G is thus a convenient way to describe this interleaver.

The above graphical construction of a code was introduced in Tanner [2] (though without necessarily requiring that G be bipartite, a feature that will be important to us for decoding). First we note that the product codes of the previous section form a particular case of it, because they can be seen as a bipartite graph code $C = C(G; A, B)$ with $G = K_{n,n}$, the complete bipartite graph on $2n$ vertices. Thus, product codes can be viewed as a seed for both serially and parallel concatenated codes.

Tanner's construction has an advantage of adding flexibility to the original product code construction: the idea of using other graphs instead of $K_{n,n}$ suggests itself naturally. In particular, Sipser and Spielman [4] associated the performance of these codes under a linear-complexity iterative decoding procedure to the additional requirement that G be an "expander graph," and suggested the name *expander codes*. That G is an expander graph means that it has a second eigenvalue λ which is small compared to the degree Δ. For example, G can be chosen to be a Ramanujan graph, for which $\lambda \leq 2\sqrt{\Delta - 1}$. The codes constructed in [4] are asymptotically good and have the following parameters.

Theorem 1: [4] Let $A[\Delta, R_0\Delta, \delta\Delta]$ be a binary linear code and let G be a Δ-regular graph with second largest eigenvalue λ. The corresponding expander code $C = C(G; A, A)$ has parameters $[N = n\Delta, K, D]$ with rate $K/N \geq 2R_0 - 1$ and relative distance $D/N \geq \epsilon$ with $\epsilon = \delta^2 - \mathcal{O}(\lambda/\Delta)$.

The first use of expander graphs to construct asymptotically good codes appeared in [12]. Expander codes were further studied in [13], [6], [7], [5] among others. In the sequel, we also will assume that the graph is Ramanujan, so that $\lambda = \mathcal{O}(\sqrt{\Delta})$.

It is often convenient to choose the component codes A and B over the alphabet of $Q = 2^t$ letters; in this case, to every edge there correspond t coordinates of the codeword of C, where t is some constant. Equivalently, each edge in the graph G is replaced by a bundle of t parallel edges. In the next sections, we will work with this version of the construction.

C. Expander Decoding

Let $\boldsymbol{y} \in \{0,1\}^N$ be a vector. A left-decoding round L consists of decoding in parallel with the code A the subvectors $\boldsymbol{y}(v)$ for every $v \in V_0$. Likewise, a right-decoding round R applies decoding with the code B to the subvectors $\boldsymbol{y}(v)$ for every $v \in V_1$. In this subsection, decoding of the component codes is assumed to be maximum likelihood.

Let $\boldsymbol{y} = \boldsymbol{y}_0 \in \{0,1\}^N$ be the vector received from the channel. The *expander decoding* scheme of [5] consists of performing successive decoding steps of the form $\boldsymbol{y}_{i+1} = R(L(\boldsymbol{y}_i))$, $i = 0, 1, \dots$. Decoding terminates by either encountering a fixed point or performing $\mathcal{O}(\log n)$ decoding steps.

To study asymptotic properties of expander codes and of this decoding we assume throughout that Δ is a suitably chosen constant and the number of vertices $n \to \infty$. Then the overall decoding procedure can be performed by a circuit of size $\mathcal{O}(n \log n)$ or by an $\mathcal{O}(n)$ sequential algorithm.

A sufficient condition for convergence of expander decoding was established in [5]. Slightly reworded as in [13] it reads as follows.

Lemma 2: [5] Let S be the subset of vertices decoded incorrectly in any even-numbered (odd-numbered) iteration. If $|S| \leq \beta n(\delta/2 - \lambda/\Delta)$, where $\delta = \min(\delta_0, \delta_1) > 3\lambda$, $0 < \beta < 1$, then expander decoding converges to the right decision in $\mathcal{O}(\log n)$ additional iterations.

The study of the error probability of expander decoding was initiated in [13] where it was shown that this probability decreases exponentially for all code rates below the capacity $\mathcal{C}$ of the BSC; this is the first known result of this kind for low-density parity-check codes and simple iterative decoding procedures. The behavior of the error exponent was further analyzed in [6] for code rates R approaching $\mathcal{C}$, which is also the region that receives most attention in coding theory. Improved estimates of the exponent obtained in that paper required nontrivial modifications of the code construction and expander decoding.

The analysis of the properties and decoding performance of expander codes relies in part on the following lemma. For a vertex v of G and a subset S of vertices, denote by $\deg_S(v)$ the number of edges incident to v and to a vertex of S.

Lemma 3 [6]: Let $G = (V_0 \cup V_1, E)$ be a Δ-regular bipartite graph, $|V_0| = |V_1| = n$, with second eigenvalue λ. Let $S \subset V_0$, $|S| = \sigma n$. Let $\alpha > \alpha_\sigma$, where $\alpha_\sigma = \lambda/2\sigma\Delta$. Let $T \subset V_1$ be defined by $T = \{v \in V_1 : \deg_S(v) \geq (1+\alpha)\sigma\Delta\}$, then

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

In particular, if $\lambda \approx \sqrt{\Delta}$ and σ and α are fixed, then

$$|T| \leq \mathcal{O}\left(\frac{n}{\sqrt{\Delta}}\right).$$

Suppose, for instance, that we are interested in the error probability of the basic version of expander decoding described above. Let $\boldsymbol{z}_0$ be the received vector, and $\boldsymbol{z}_1$ the vector obtained after the first (left) decoding iteration. By choosing a sufficiently large Δ, the error probability for a left vertex to be incorrectly decoded in the first round can be made arbitrarily close to $\exp(-\Delta E_0(R_0, p))$. In the second iteration, for a right vertex to be decoded incorrectly it needs to have at least $d_1/2$ incorrect edges adjacent to it. By Lemma 3, if the subset S of left vertices incorrectly decoded after the first iteration is of size $|S| \le n\delta_1/2$ then the size of the set T of right vertices whose S-degree exceeds $(1+\alpha)n\Delta\delta_1/2$ is small. Hence, most subvectors $\boldsymbol{z}_1(v)$, $v \in V_1$ will be decoded correctly in the second iteration. That the remaining errors will be removed in subsequent iterations is guaranteed by Lemma 2. The value of the error probability of expander decoding established by this analysis is (arbitrarily close to) [13]

$$\exp(-NE_0(R_0, p)\delta_1/2).$$

Apart from the error probability, another parameter of interest is the fraction of errors correctable by expander decoding. The original result of [4] gave a bound close to $\epsilon/48$ where ϵ is the bound on the relative distance of the expander code given in Theorem 1. In [5], this estimate was improved to $\epsilon/4$ by an application of expander decoding. An improved bound on the distance of the codes (but not on fraction of correctable errors) was given in [13]. Recently, the GMD algorithm was used for expander codes [14] to show that they can correct close to $(1/2)\epsilon$ fraction of errors, where ϵ is again the quantity in Theorem 1.

In Section III, we present a somewhat different parallel concatenated construction and prove that under (modified) expander decoding it provides an exponentially smaller error probability. Without increasing the order of the decoding complexity from the basic expander construction we will show that the modified construction attains the Zyablov bound (1) on the relative distance and the Forney bound (2) on the error exponent. We also prove that the decoding algorithm introduced in Section III-C below corrects close to a $\delta_Z/2$ fraction of errors.

III. Modified Parallel Concatenated Block Codes

A. The Construction

Let $G=(V,E)$ be a bipartite graph whose parts are V_0 (the left vertices) and $V_1 \cup V_2$ (the right vertices), where $|V_i| = n$ for $i = 0,1,2$. The degree of the left vertices is Δ, the degree of the vertices in V_1 is Δ_1, and the degree of vertices in V_2 is $\Delta_2 = \Delta - \Delta_1$. For a given vertex $v \in V_0$ we denote by $E(v)$ the set of all edges incident to it and by $E_i(v) \subset E(v)$, $i=1,2$ the subset of edges of the form (v,w), where $w \in V_i$. The ordering of the edges on v defines an ordering on $E_i(v)$. Note that both subgraphs $G_i = (V_0 \cup V_i, E_i)$, $i=1,2$ can be chosen to be regular, of degrees Δ_1 and Δ_2, respectively. We require that the first subgraph $G_1 = (V_0 \cup V_1, E_1)$ be Ramanujan.

Let A be a $[t\Delta, R_0t\Delta, d_0 = t\Delta\delta_0]$ linear binary code of rate $R_0 = \Delta_1/\Delta$. The code A can also be seen as a q-ary additive $[\Delta, R_0\Delta]$ code, $q = 2^t$. Let B be a q-ary $[\Delta_1, R_1\Delta_1, d_1 = \Delta_1\delta_1]$ additive code. These codes will play a role analogous to the inner and outer code of the serial concatenated construction. Apart from them, we will need an auxiliary q-ary code A_{aux} of length Δ_1. The code C is defined as the set of vectors $\boldsymbol{x} = \{x_1, \dots, x_N\}$ such that (see Fig. 1)

1) for every vertex $v \in V_0$, the subvector $(x_j)_{j\in E(v)}$ is a (q-ary) codeword of A and the set of coordinates $E_1(v)$ is an information set for the code A;

2) for every vertex $v \in V_1$, the subvector $(x_j)_{j\in E(v)}$ is a codeword of B;

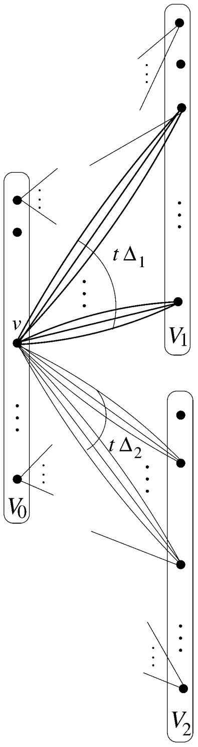

Fig. 1. The code construction.

3) for every vertex $v \in V_0$, the subvector $(x_j)_{j\in E_1(v)}$ is a codeword of A_{aux}.

We will choose the minimum distance $d_{\mathrm{aux}} = \delta_{\mathrm{aux}}\Delta_1$ of the code A_{aux} so as to make the quantity λ/d_{aux} arbitrarily small, where λ is the second eigenvalue of G_1. By choosing Δ_1 large enough, the rate R_{aux} of A_{aux} can be thought of as a quantity such that $1 - R_{\mathrm{aux}}$ is almost $\mathcal{O}(1/\sqrt{\Delta})$. The role of this code A_{aux} will become apparent when we describe and analyze the decoding procedure.

This construction in a different form appears briefly in [6]. It has been recast here to stress the analogy with serial concatenations. The rate of the code C is given by the following proposition.

Proposition 4: The code C has length $N = tn\Delta$ and the rate $R \ge R_0R_1 - R_0(1 - R_{\mathrm{aux}})$. Hence, for any $\epsilon > 0$ there exists a certain value of Δ, independent of n, such that

$$R = R(C) \ge R_0R_1 - \epsilon.$$

Proof: To estimate $R(C)$ let us calculate the number of parity-check equations of the code C. We obtain

$$\begin{aligned}&tn\Delta(1 - R(C))\\ &\qquad \le tn\Delta(1 - R_0) + tn\Delta_1(1 - R_1) + tn\Delta_1(1 - R_{\mathrm{aux}}).\end{aligned}$$

This implies the claimed estimate of R. □

The key issue about the preceding construction is defining an appropriate decoding procedure and evaluating its properties in terms of the number of errors corrected and the error probability of decoding.

First, let us estimate the minimum distance of C.

B. Minimum Distance

We recall the following lemma from [5].

Lemma 5: Let $S \subset V_0, T \subset V_1$. The average degree $\overline{d}_{ST}$ of the subgraph of G_1 induced by S and T satisfies

$$\overline{d}_{ST} \leq \frac{2|S||T|}{|S|+|T|}\frac{\Delta_1}{n} + \lambda - \frac{\lambda}{n}\frac{|S|^2+|T|^2}{|S|+|T|}.$$

Theorem 6: The minimum distance D of C satisfies

$$D \geq \delta_0\delta_1\left(1-\frac{\lambda}{d_{\text{aux}}}\right)\left(1-\frac{\lambda}{2d_1}\right)N.$$

Proof: Let $\boldsymbol{c} = (c_1 \ldots c_N)$ be a nonzero codeword of C, let S and T be, respectively, the set of vertices v of V_0 (resp., V_1) for which the subvector $(\boldsymbol{c}_j)_{j\in E(v)}$ is nonzero.

By definition of C, every vertex of S has degree at least d_{aux} in G_1 and every vertex of T has degree at least d_1, so that Lemma 5 implies

$$\frac{d_{\text{aux}}|S| + d_1|T|}{|S|+|T|} \leq \frac{2|S||T|}{|S|+|T|}\frac{\Delta_1}{n} + \lambda$$
$$d_{\text{aux}}|S| + d_1|T| \leq 2|S||T|\frac{\Delta_1}{n} + \lambda(|S|+|T|)$$

which is rewritten as

$$\frac{d_{\text{aux}}-\lambda}{|T|} + \frac{d_1-\lambda}{|S|} \leq 2\frac{\Delta_1}{n} \tag{3}$$

and because $d_{\text{aux}} - \lambda$ and $d_1 - \lambda$ are assumed to be positive quantities, (3) yields

$$|S| \geq \frac{d_1-\lambda}{2\Delta_1}n \tag{4}$$
$$|T| \geq \frac{d_{\text{aux}}-\lambda}{2\Delta_1}n. \tag{5}$$

Let $\sigma = |S|/n$, let $\alpha = d_1/\sigma\Delta_1 - 1$, and let $\alpha_\sigma = \lambda/2\sigma\Delta_1$. If $\alpha \leq \alpha_\sigma$, then $(d_1 - \lambda/2) \leq \sigma\Delta_1$ which means

$$\delta_1\left(1-\frac{\lambda}{2d_1}\right)n \leq |S|$$

and $D \geq |S|d_0$ implies

$$D \geq \delta_0\delta_1\left(1-\frac{\lambda}{2d_1}\right)N$$

which proves (slightly better than) the theorem. We may therefore assume that $\alpha > \alpha_\sigma$ and Lemma 3 applies and gives

$$|T| \leq \frac{\lambda/2\sigma\Delta_1}{d_1/\sigma\Delta_1 - 1 - \lambda/2\sigma\Delta_1}|S|$$
$$|T| \leq \frac{\lambda}{2d_1 - 2\sigma\Delta_1 - \lambda}|S|$$

which together with (5) yields

$$\frac{d_{\text{aux}}-\lambda}{2\Delta_1}n \leq \frac{\lambda}{2d_1 - 2\sigma\Delta_1 - \lambda}|S|$$

and

$$\frac{d_{\text{aux}}-\lambda}{\Delta_1}(d_1 - \sigma\Delta_1 - \lambda/2)n \leq \lambda|S|$$
$$\frac{d_{\text{aux}}-\lambda}{\Delta_1}(d_1 - \lambda/2)n \leq d_{\text{aux}}|S|$$
$$\left(1-\frac{\lambda}{d_{\text{aux}}}\right)\left(1-\frac{\lambda}{2d_1}\right)\delta_1 n \leq |S|.$$

Together with $D \geq |S|d_0$ this proves the theorem. □

Proposition 4 and Theorem 6 together show that the family of (modified) expander codes studied in this section reached the Zyablov asymptotic bound (1) on the code parameters. Indeed, for any given $\epsilon_0 > 0$ it is possible to choose Δ large enough (but independent of n) so that the relative distance δ_0 satisfies $\delta_0 \geq \delta_{\text{GV}}(R_0) - \epsilon_0$. Similarly, by choosing Δ large enough it is possible to make the q-ary relative distance δ_1 of the code B arbitrarily close to the q-ary GV distance defined as the (smaller) root of

$$R_1 = 1 + \delta\log_q\frac{\delta}{q-1} + (1-\delta)\log_q(1-\delta).$$

Moreover, it is also possible to choose t large enough so that the q-ary GV distance is arbitratily close to $1 - R_1$. To summarize, for any ϵ_1, there exist large but fixed Δ and t such that δ_1 satisfies $\delta_1 \geq (1 - R_1) - \epsilon_1$. Then the relative distance D/N of Theorem 6 comes close to δ_Z.

C. Decoding of the Code C

Recall that in the case of serially concatenated codes, a naive decoding algorithm (decoding the inner codes up to half the minimum distance, then the outer codes by the same procedure) guarantees correction of about $1/4$ of the designed distance of codes $d = d_0d_1$. Similarly, by performing maximum-likelihood decoding of the inner codes and then algebraic decoding of the outer codes, we can attain the exponent of the error probability equal to $E_{\text{F}}/2$, where E_{F} is given by (2). By taking into account the reliability information obtained by the decoder of the inner code A and using it in the GMD algorithm for the code B we can double the attainable error exponent making it close to E_{F}. Accordingly, there are two versions of iterative decoding of C: a "hard" version and a "soft" version.

1) Hard-Decision Decoding: Decoding is defined as an iterative procedure. In the first iteration, every subvector $\boldsymbol{z}(v)$, $v \in V_0$ of the vector $\boldsymbol{z}$ received from the channel is decoded with the binary code A using maximum-likelihood decoding. Denote by $\boldsymbol{z}_0$ the result of this iteration. For the rest of the iterative procedure we forget about the symbols indexed by E_2 and shorten $\boldsymbol{z}_0$ to the set of coordinates indexed by E_1. We next submit the resulting vector to expander decoding (Section II-C) with the expander code $C(G_1; A_{\text{aux}}, B)$ defined by the subgraph G_1 and the q-ary codes A_{aux} (the left code) and B (the right code). If, upon terminating, this decoding produces a codeword of the code $C(G_1; A_{\text{aux}}, B)$, then by conditions 1) and 3) of construction III-A for every vertex $v \in V_1$ we have found information symbols of the left codewords, which enables us to reconstruct the transmitted codeword.

The hard-decision version of decoding can be analyzed pretty much the same way as before: namely, the proportion of correctable errors can be seen to be arbitrarily close to $\delta_0\delta_1/4$, as in [5] and [13] and the error exponent arbitrarily close to $E_0(R_0, p)\delta_1/2$. Note, however, that the rate of the overall code C is improved with the modified construction.

2) Soft-Decision Decoding: This is a refinement of the "hard version" based on the following idea. The code A is used only once in the first step of the hard decoding algorithm to take a decision about the binary vectors transmitted in the neighborhood $\mathcal{N}(v)$ of each vertex $v \in V_0$. Instead, we propose to use this code to gather information about the reliability of the q-ary symbols transmitted on the edges $e \in E_1$ and pass that information to the vertices in V_1. A vertex w receives this information from the adjacent vertices in V_0 and then performs decoding of the code B that outputs a codeword with the maximum total reliability of its q-ary symbols. In other words, the algorithm described in this section performs, in its first stage, depth-two soft decoding of the code B, and then proceeds with regular expander decoding of the code $C(G_1; A_{\text{aux}}, B)$. A formal description follows.

The first iteration of decoding is performed in parallel for every vertex $v \in V_0$ and the corresponding received subvector $\boldsymbol{z}(v)$. The decoder computes, for every symbol b of the q-ary alphabet, and for every (multiple) edge $e \in E_1$ incident to v, the weight of the edge with respect to b as follows:

$$d_{e,b}(\boldsymbol{z}) = \min_{\boldsymbol{a}\in A: a(e)=b} \mathrm{d}(\boldsymbol{a}, \boldsymbol{z}(v))$$

where by $a(e)$ we denote the q-ary coordinate of the codeword $\boldsymbol{a}$ that corresponds to the edge e, and where $\mathrm{d}(\cdot,\cdot)$ is the *binary* Hamming distance. This information is passed along the edge e to the corresponding right decoder. No changes in the received vector $\boldsymbol{z}$ are made up to this point. In the second iteration, for every vertex $w \in V_1$, the right decoder associated to it finds a q-ary codeword $\boldsymbol{b} = (b_1, \dots, b_{\Delta_1}) \in B$ that satisfies

$$\boldsymbol{b} = \operatorname*{argmin}_{\boldsymbol{x}=(x_1,\dots,x_{\Delta_1})\in B} \sum_{i=1}^{\Delta_1} d_{w(i),x_i}(\boldsymbol{z}) \tag{6}$$

and passes b_i along the edge $w(i)$, $i = 1, \dots, \Delta_1$ to a vertex in V_0. Then decoding reverts to hard decisions, i.e., as before alternates between left and right q-ary maximum-likelihood decoding of the expander code $C(G_1; A_{\text{aux}}, B)$.

Let us provide some more detail about the first two stages of the described procedure. Consider a right vertex $w \in V_1$. The $\Delta_1 t$ edges incident to it connect w with some left vertices $v_1, \dots, v_{\Delta_1}$. The subset $E(v_i)$ of edges incident to v_i, $i = 1, \dots, \Delta_1$ consists of Δ (multiple) edges one of which goes into w and the other $\Delta - 1$ connect v_i with some right vertices in V_1 or V_2 (see Fig. 2, where these right vertices are replicated to the left of the part V_0). Let us consider the projection L of the code C on the coordinates in the depth-two neighborhood of w, i.e., the coordinates that correspond to the edges in the subset $E(v_1) \cup E(v_2) \cup \cdots \cup E(v_{\Delta_1})$. The projected code (which is simply the punctured code C) can be also thought of as a serial concatenation of the codes B and A. It is a binary linear code L of length $\Delta_1\Delta t$ and rate $R(L) \approx R_0$.[1] After some thought it becomes clear that the algorithm described in the previous paragraph performs a version of maximum-likelihood decoding of the code L restricted to the coordinates incident to the vertex w. We also note that this procedure is an instance of a general procedure known in iterative decoding as the min-sum algorithm [2], [15].

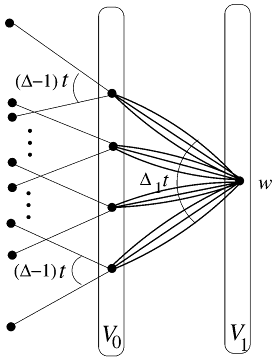

Fig. 2. The depth-two code (part V_2 of the graph G not shown).

D. Error Exponents

Let us turn to the analysis of the error probability for the soft-decision version. Our main result is the following theorem. It improves the exponent obtainable with the hard-decision algorithm by a factor of 2 and shows that iterative decoding of 1-level parallel concatenated schemes achieves Forney's exponent E_{F}.

Theorem 7: For any $\epsilon > 0$ there exist sufficiently large but finite values of Δ, Δ_1, and t, independent of n, and a bipartite graph $G = (V, E)$ such that the error probability of decoding of a parallel concatenated code defined by the graph G and additive codes A, B, A_{aux} behaves as $2^{-N(E_{\mathrm{F}}-\epsilon)}$.

The remainder of this section is devoted to the proof of this theorem. We will prove that the overall error rate of the algorithm is determined by the frequency of error events in the second iteration.

Assume that the all-zero vector was transmitted and let $\boldsymbol{z}$ be the vector received from the channel. Observe that, since after two iterations we revert to hard decisions, the iterative decoding algorithm will only be in error if many right vertices are incorrectly decoded at the second iteration. With this in mind, we will analyze the transition from the first to the second decoding iteration.

For $v \in V_0$, consider the quantity (the *reliability* or *cost* of the vertex v)

$$r_v := \frac{1}{t}(\mathrm{d}(\boldsymbol{z}(v), \boldsymbol{0}) - \mathrm{d}(\boldsymbol{z}(v), \boldsymbol{c}))$$

where $\boldsymbol{c}$ is a nonzero vector of A closest to $\boldsymbol{z}(v)$. Note that standard maximum-likelihood decoding of the left code at vertex v can succeed only when $r_v \le 0$.

In the next series of lemmas we derive sufficient conditions for correct decoding. Our goal will be to formulate them in

[1] More precisely, $R(L) \ge R_0 - (1 - R_{\text{aux}}) - \Delta^{-1}(1 - R_1)$.

global terms, i.e., as properties of some subsets of vertices of the graph G. We start with local conditions and use the spectral properties of the expander graph to "globalize" them. The starting observation is given by the following statement.

Lemma 8: A vertex $w \in V_1$ is wrongly decoded at the second iteration only if there exists a set $U \subset V_0$ of neighbors of w such that

$$|U| \ge d_1 \tag{7}$$

$$\sum_{v\in U} r_v \ge 0. \tag{8}$$

Proof: Suppose a vertex $w \in V_1$ is decoded incorrectly. By (6), the algorithm finds a codeword $\boldsymbol{b} = (b_1, \dots, b_{\Delta_1}) \in B\backslash\{\boldsymbol{0}\}$ more reliable than the all-zero vector. Let $U \subset V_0$, $|U| \ge d_1$ be the subset of vertices that correspond to nonzero coordinates of $\boldsymbol{b}$, i.e., the vertices of V_0 incident to edges $w(i)$ for i such that $b_i \neq 0$. By definition of the decoding procedure, condition (8) must be true with respect to U. □

For a subset $S \in V_0$ and a vertex $x \in V_1$ define

$$r_S(x) = \sum_{s\in S\cap\mathcal{N}(x)} r_s.$$

From the previous lemma we obtain the following set of sufficient conditions for correct decoding of $w \in V_1$.

Lemma 9: Let $w \in V_1$ and $S \subset V_0$ satisfy

1) $r_v \le 0$ and $r_v \le r_s$ for every $s \in S$ and every $v \in V_0 \backslash S$;
2) $|S \cap \mathcal{N}(w)| \le d_1$;
3) $r_S(w) \le 0$.

Then w will be decoded correctly after iteration 2.

To make this condition manageable we will need a combinatorial lemma which is a generalization of Lemmas 3 and 5 from "simple" to weighted degrees of vertices.

Lemma 10: Let $G = (V_0 \cup V_1, E(G))$ be a regular bipartite graph of degree Δ, $|V_0| = |V_1| = n$ with second eigenvalue λ. Let $S \subset V_0$, $|S| = \sigma n$, let $\alpha_\sigma = \lambda/(2\sigma\Delta)$, and let α satisfy $2\alpha_\sigma < \alpha \le 1$. Suppose for every vertex $s \in S$ there is a number r_s, $-\Delta \le r_s \le \Delta$. Let

$$T = \{u \in V_1, \quad r_S(u) \ge 0\}.$$

If $\sum_{s\in S} r_s \le -\alpha\Delta|S|$ then

$$|T| \le \frac{5\alpha_\sigma}{\alpha - 2\alpha_\sigma}|S|.$$

Proof: Let $r = |S|^{-1}\sum_{s\in S} r_s$. Let $\boldsymbol{A} = (a_{ij})$ be the $2n \times 2n$ adjacency matrix of the bipartite graph G, i.e., $a_{ij} = 1$ if the vertex indexed by i is adjacent to the vertex indexed by j and $a_{ij} = 0$ otherwise: a fixed ordering of the vertices is assumed. Let $\boldsymbol{X}$ be the column vector of length $2n$ such that every coordinate indexed by a vertex s of S equals $(r_s - r)/\Delta$, every coordinate indexed by a vertex u of T equals 1, and all the other coordinates equal 0.

We have, denoting by ${}^t\boldsymbol{X}$ the transpose of $\boldsymbol{X}$

$$\begin{aligned} {}^t\boldsymbol{X}\boldsymbol{A}\boldsymbol{X} &= \frac{2}{\Delta}\sum_{u\in T}\sum_{s\in S\cap\mathcal{N}(u)}(r_s - r) \\ &= \frac{2}{\Delta}\sum_{u\in T}(r_S(u) - \deg_S(u)r). \end{aligned} \tag{9}$$

Now let $\boldsymbol{j}$ be the all-one vector and let $\boldsymbol{k}$ be the vector such that every coordinate indexed by a vertex of V_0 equals 1 and every coordinate indexed by a vertex of V_1 equals -1. The vectors $\boldsymbol{j}$ and $\boldsymbol{k}$ are eigenvectors of $\boldsymbol{A}$ associated to the eigenvalues Δ and $-\Delta$, respectively. Now define the vector $\boldsymbol{Y}$ by

$$\boldsymbol{X} = \frac{|T|}{2n}\boldsymbol{j} - \frac{|T|}{2n}\boldsymbol{k} + \boldsymbol{Y}.$$

The coordinates of the vector $\boldsymbol{Y}$ are

$$y_v = \begin{cases} (r_v - r)/\Delta, & v \in S \\ 0, & v \in V_0 \setminus S \\ 1 - \frac{|T|}{n}, & v \in T \\ -\frac{|T|}{n}, & v \in V_1 \setminus T \end{cases}$$

so that $\langle \boldsymbol{j}, \boldsymbol{Y}\rangle = \langle \boldsymbol{k}, \boldsymbol{Y}\rangle = 0$. Because the eigenspaces of $\boldsymbol{A}$ are orthogonal we have

$$\begin{aligned} {}^t\boldsymbol{X}\boldsymbol{A}\boldsymbol{X} &= \left(\frac{|T|}{2n}\right)^2 \boldsymbol{j}^2\Delta - \left(\frac{|T|}{2n}\right)^2 \boldsymbol{k}^2\Delta +^t \boldsymbol{Y}\boldsymbol{A}\boldsymbol{Y} \\ {}^t\boldsymbol{X}\boldsymbol{A}\boldsymbol{X} &= {}^t\boldsymbol{Y}\boldsymbol{A}\boldsymbol{Y}. \end{aligned} \tag{10}$$

Now we have

$$\begin{aligned} \|\boldsymbol{Y}\|^2 &= \sum_{s\in S}(r_s/\Delta - r/\Delta)^2 + |T|(1 - |T|/n)^2 \\ &\quad + (n - |T|)(|T|/n)^2 \\ &\le 4|S| + |T| \end{aligned}$$

since $|r_s - r| \le 2\Delta$. We have therefore by (10)

$${}^t\boldsymbol{X}\boldsymbol{A}\boldsymbol{X} \le \lambda(4|S| + |T|). \tag{11}$$

We now proceed to lower bound ${}^t\boldsymbol{X}\boldsymbol{A}\boldsymbol{X}$. By definition of T we have $\sum_{u\in T} r_S(u) \ge 0$ and since $-r \ge \alpha\Delta$, (9) gives

$$2\alpha\sum_{u\in T}\deg_S(u) \le {}^t\boldsymbol{X}\boldsymbol{A}\boldsymbol{X}. \tag{12}$$

Now denote $\overline{S} = V_0 \setminus S$ and consider the subgraph of (V_0, V_1) induced by $\overline{S}$ and T. Apply Lemma 5 to it to obtain

$$\begin{aligned} &(|\overline{S}| + |T|)\overline{d}_{\overline{S}T} \\ &\quad \le 2|\overline{S}||T|\frac{\Delta}{n} + \lambda(|\overline{S}| + |T|) - \frac{\lambda}{n}(|\overline{S}|^2 + |T|^2). \end{aligned} \tag{13}$$

Notice that the left-hand side of (13) is twice the number of edges in the graph $(\overline{S}, T)$ and therefore equals

$$\begin{aligned} 2\sum_{u\in T}\deg_{\overline{S}}(u) &= 2\sum_{u\in T}(\Delta - \deg_S(u)) \\ &= 2|T|\Delta - 2\sum_{u\in T}\deg_S(u). \end{aligned}$$

Substituting in (13) and writing $|\overline{S}| = n - |S|$ we obtain, after rearranging

$$-2\sum_{u\in T}\deg_S(u) \le -2|S||T|\frac{\Delta}{n} + \lambda(|S|+|T|) - \frac{\lambda}{n}(|S|^2+|T|^2).$$

Throw away the $-\frac{\lambda}{n}(|S|^2+|T|^2)$ term and multiply by (-1) so that inequality (12) now becomes

$$2\alpha\sigma\Delta|T| - \alpha\lambda(|S|+|T|) \le {}^t\boldsymbol{XAX}$$

which together with (11) gives

$$2\alpha\sigma\Delta|T| \le (5|S|+2|T|)\lambda$$

and the result after rearranging. □

Note that Lemma 9 gave a set of local conditions sufficient for correct decoding of a given vertex. The next lemma replaces condition 2) of the previous lemma by a global condition easier to work with. The resulting set of conditions is sufficient for the local conditions to hold for almost every vertex of V_1.

Lemma 11: Let $2\alpha_\sigma < \alpha < 1$. Suppose there exists a subset $S \subset V_0$ satisfying

1) $r_v \le 0$ and $r_v \le r_s$ for every $s \in S$ and every $v \in V_0 \setminus S$;
2) $|S| = (1+\alpha)^{-1}\delta_1 n$;
3) $\sum_{s\in S} r_s \le -\alpha\Delta|S|$.

Then the number of wrongly decoded codewords of V_1 in iteration 2 is not more than

$$\frac{6\alpha_\sigma}{\alpha - 2\alpha_\sigma}|S|.$$

Proof: Let $w \in V_1$. One possibility for w to be decoded incorrectly is that it is adjacent to a number of vertices in S greater than $(1+\alpha)\Delta|S|/n$. An upper bound on the number of such vertices is given by Lemma 3. If w is not such a vertex then it has at most d_1 neighbors in S and therefore is incorrectly decoded only if

$$r_S(w) \ge 0. \tag{14}$$

Indeed, if this inequality does not hold, the total cost of any set of at least d_1 neighbors of w will be negative by condition 1) of the lemma, and so w will be decoded correctly. The maximum number of vertices that satisfy (14) is given by Lemma 10. Summing and upper-bounding the estimates given by Lemmas 3 and 10 proves the result. □

The conditions of Lemma 11 imply that the subvectors $\boldsymbol{z}_1(w)$ for all vertices in $w \in V_1$, except for a small fraction of them, will be correct after the second iteration. By properly choosing d_{aux} (somewhat above $\sqrt{\Delta_1}$) we impose that the proportion of remaining vertices in error is less than approximately $\delta_{\text{aux}}/2$ and the overall convergence of the decoding procedure follows by Lemma 2. Let us state this more formally.

Lemma 12: Suppose α and d_{aux} satisfy

$$1 > \alpha > \frac{2\lambda}{d_1} \tag{15}$$

$$d_{\text{aux}} > 2\lambda\left(1 + \frac{6}{\alpha - 2\lambda/d_1}\right). \tag{16}$$

Suppose furthermore that there exists $S \subset V_0$ satisfying the conditions of Lemma 11. Then the decoding algorithm converges to the right decision in $\mathcal{O}(\log n)$ iterations.

Proof: By Lemma 2, expander decoding of the code $C(G_1; A_{\text{aux}}, B)$ will converge to a correct decision if the number of incorrectly decoded vertices at the second iteration does not exceed $\beta n\left(\frac{1}{2}\delta_{\text{aux}} - \lambda/\Delta_1\right)$, $\beta < 1$. Take $\beta = 1/2$. Thus, by Lemma 11, we will have convergence if

$$\frac{1}{2}n\left(\frac{1}{2}\delta_{\text{aux}} - \frac{\lambda}{\Delta_1}\right) > \frac{6\alpha_\sigma}{\alpha - 2\alpha_\sigma}|S| = \frac{6\alpha_\sigma}{\alpha - 2\alpha_\sigma}\frac{n\delta_1}{(1+\alpha)} \tag{17}$$

where $\alpha > 2\alpha_\sigma$, $\alpha_\sigma = \lambda/(2\sigma\Delta)$, and $\sigma = (1+\alpha)^{-1}\delta_1$. Condition (17) gets rewritten as

$$\frac{1}{2}\delta_{\text{aux}} - \frac{\lambda}{\Delta_1} > \frac{6\lambda/\Delta_1}{\alpha - \frac{\lambda(1+\alpha)}{d_1}}$$

which is easily seen to be implied by condition (16). Condition (15) implies $\alpha > 2\alpha_\sigma$. □

Next, we need to estimate the probability of the event $\mathcal{E}$ that there does not exist a $S \subset V_0$ which satisfies Lemma 11. Let

$$\mathcal{E} = \left\{\text{for all } S \in V_0 \text{ such that } |S| = n\delta_1(1+\alpha)^1, \quad \sum_{s\in S} r_s \ge -\alpha\Delta|S|\right\}.$$

Lemma 13: The probability $\Pr[\mathcal{E}]$ satisfies

$$\Pr[\mathcal{E}] \le \exp\left[\frac{-n\Delta t\delta_1}{1+\alpha}(E_0(R_0,p) - M\alpha)(1-o(1))\right]$$

where M is a quantity that depends only on the values of R_0 and p.

Proof: We recall a result about the probability distribution of the random variables $t \cdot r_s$ for a random linear code. Let $W[\ell, \kappa\ell]$ be such a code used for transmission over a BSC with crossover probability p. Suppose that the transmitted codeword is the all-zero one and that $\boldsymbol{w}$ is the nonzero codevector closest to the received vector $\boldsymbol{z}$ by the Hamming distance. Let

$$X(\boldsymbol{z}) = \mathrm{d}(\boldsymbol{z}, \boldsymbol{0}) - \mathrm{d}(\boldsymbol{z}, \boldsymbol{w}).$$

Let $\rho \in [-1, 1]$ be some number. Denote by $E_1(\kappa, p, \rho)$ the exponent of the probability $\Pr[X \ge \rho\ell]$, i.e.,

$$\Pr[X \ge \rho\ell] = \exp[-\ell E_1(\kappa, p, \rho)(1-o(1))].$$

As shown in [16] for the ensemble of all codes and in [17] for linear codes

$$E_1(\kappa, p, \rho) \ge E_0(\kappa, p) + M(\kappa, p)\rho.$$

Here

$$M(\kappa, p) = \frac{1}{2}\log\frac{1-p}{p}, \qquad \text{if } \kappa \le R_{\text{crit}}$$

$$M(\kappa, p) = \log\frac{\delta_{\text{GV}}(\kappa)(1-p)}{(1-\delta_{\text{GV}}(\kappa))p}, \qquad \text{if } \kappa \ge R_{\text{crit}}.$$

(A simple proof of this bound together with an improvement of the results of [17] is given in [18].) In our case, each left code has length $\ell = t\Delta$ and rate $\kappa = R_0$, and for every left vertex s the quantity $t \cdot r_s$ has the distribution of the above random variable X. Let

$$\rho_s \in \left\{-1, -\frac{t\Delta - 1}{t\Delta}, \dots, 0, \dots, \frac{t\Delta - 1}{t\Delta}, 1\right\}$$

be the value of the threshold ρ associated with the vertex s. We write, omitting the $o(1)$ terms

$$\begin{aligned}\Pr[\mathcal{E}] &\le \binom{n}{|S|} \sum_{(\rho_s)} \prod_{s \in S} \Pr[r_s \ge \rho_s \Delta] \\ &\le \binom{n}{|S|} \sum_{(\rho_s)} \prod_{s \in S} e^{-t\Delta(E_0(R_0,p) + M(R_0,p)\rho_s)}\end{aligned}$$

where the sums on the right-hand side run over all the choices of $(\rho_s)_{s\in S}$ for which $\sum \rho_s \ge -\alpha|S|$. For every s there are at most $2t\Delta + 1$ possible values of r_s and of ρ_s so that we get

$$\Pr[\mathcal{E}] \le \binom{n}{|S|} (2t\Delta + 1)^{|S|} e^{-\Delta t |S| (E_0(R_0,p) - M(R_0,p)\alpha)}.$$

The proof is completed by computing logarithms, choosing Δ large enough, and taking account of the fact that the binomial coefficient $\binom{n}{|S|}$ is only exponential in n but not in Δ. □

We are now ready to complete the proof of Theorem 7. We have just shown that, whenever α and d_{aux} satisfy conditions (16) and (15), then the decoding algorithm can be in error only when event $\mathcal{E}$ happens. By choosing d_1 to be linear in Δ_1 and a large enough Δ_1, we see that when the graph G_1 is Ramanujan ($\lambda \le 2\sqrt{\Delta_1 - 1}$), we can satisfy both (16) and (15) with simultaneously an arbitrarily small α and an auxiliary distance d_{aux} sublinear in Δ_1.

Because α is arbitrarily small, Lemma 13 gives an error exponent for the decoding algorithm arbitrarily close to $\delta_1 E_0(R_0, p)$. By taking t sufficiently large and choosing the q-ary additive code B, $q = 2^t$, to meet the q-ary GV bound we can make the value δ_1 be arbitrarily close to $1 - R_1$. Together with Proposition 4, we obtain that the error exponent is at least a quantity arbitrarily close to

$$E_0(R_0, p)\left[1 - \frac{R}{R_0} + (1 - R_{\text{aux}})\right]$$

which, because d_{aux} is sublinear in Δ_1, can in turn be made arbitrarily close to $E_{\text{F}}(R, p)$. This concludes the proof of Theorem 7.

E. Number of Correctable Errors

In this subsection, the ideas of the previous subsection are used to establish a lower bound on the number of errors correctable under expander decoding. We will prove that this number is close to half the designed distance, i.e., to half the Zyablov bound $N\delta_Z(R)/2$.

The main result of this section is given by the following theorem.

Theorem 14: Let C be a modified expander code of rate R defined by a graph $G = (V, E)$ of left degree Δ and additive codes $A[t\Delta, R_0 t\Delta, d_0 = \delta_0 t\Delta]$, $B[\Delta_1, R_1\Delta_1, d_1 = \delta_1\Delta_1]$, A_{aux}. For any $\epsilon > 0$, there exists infinitely many choices of the graph for which $R > R_0 R_1 - \epsilon$ and such that soft-decision decoding of the code C corrects every combination of errors of weight not exceeding $N(\delta_0\delta_1/2 - \epsilon)$.

We again rely on the modified expander code construction of Section III-A together with the soft-decision decoding algorithm defined above. Let $\boldsymbol{z}$ be the vector received from the channel (the error vector). Recall the subset $S \in V_0$ of approximately d_1 least reliable vertices from Lemma 11. The key to the proof of Theorem 14 is in connecting the weight of $\boldsymbol{z}$ to the average cost $r = |S|^{-1} \sum_{s \in S} r_s$ of a vertex in the set S. Roughly speaking, $r > 0$ only if $w(\boldsymbol{z}) \ge N d_0 d_1 / 2$. If, on the other hand, $r < 0$ then by Lemma 11, the number of right vertices decoded incorrectly in iteration 2 is small, and errors introduced by them will be removed by regular expander decoding with the code B on the right and A_{aux} on the left.

This argument is made more precise in the following lemma which also implies Theorem 14.

Lemma 15: Suppose α and d_{aux} satisfy conditions (15) and (16) and let

$$w(\boldsymbol{z}) \le (1 + \alpha)^{-1} N \frac{\delta_1}{2}(\delta_0 - \alpha). \tag{18}$$

Then the vector $\boldsymbol{z}$ will be decoded to the all-zero vector by soft-decision expander decoding.

Proof: Consider the subset $S \subset V_0$ such that $|S| = (1 + \alpha)^{-1} n\delta_1$ formed of the least reliable vertices: $r_v \le r_s$ for every $v \in V_0 \backslash S$ and every $s \in S$.

First observe that the inequality $\sum_{s \in S} r_s > -\alpha\Delta|S|$ implies that $w(\boldsymbol{z})$ does not satisfy (18). Indeed, recall that we have

$$r_s = \frac{1}{t}(\mathrm{d}(\boldsymbol{z}(s), \boldsymbol{0}) - \mathrm{d}(\boldsymbol{z}(s), \boldsymbol{c}_s))$$

where $\boldsymbol{c}_s \in A$ is the nearest nonzero codeword to $\boldsymbol{z}(s)$. With this we have

$$\begin{aligned}\sum_{s \in S} \mathrm{d}(\boldsymbol{z}(s), \boldsymbol{0}) &> \sum_{s \in S} \mathrm{d}(\boldsymbol{z}(s), \boldsymbol{c}_s) - \alpha t\Delta|S| \\ &\ge \sum_{s \in S} (\mathrm{d}(\boldsymbol{0}, \boldsymbol{c}_s) - \mathrm{d}(\boldsymbol{z}(s), \boldsymbol{0})) - \alpha t\Delta|S| \\ &\ge \sum_{s \in S} (d_0 - \mathrm{d}(\boldsymbol{z}(s), \boldsymbol{0})) - \alpha t\Delta|S| \\ &= |S| d_0 - \sum_{s \in S} \mathrm{d}(\boldsymbol{z}(s), \boldsymbol{0})) - \alpha t\Delta|S|\end{aligned}$$

so

$$\begin{aligned}w(\boldsymbol{z}) \ge \sum_{s \in S} \mathrm{d}(\boldsymbol{z}(s), \boldsymbol{0}) &> \frac{1}{2}(d_0 - \alpha t\Delta)|S| \\ &= (1 + \alpha)^{-1} N \frac{\delta_1}{2}(\delta_0 - \alpha).\end{aligned}$$

Hence, if $w(\boldsymbol{z})$ satisfies (18), then $\sum_{s \in S} r_s \le -\alpha\Delta|S|$ so that S satisfies the conditions of Lemma 11. The result follows from Lemma 12. □

Recall from [5] that hard-decision decoding of expander codes corrects close to a fraction $d/4$ of errors, where d is the code's designed distance. Theorem 14 shows that, similarly to serial concatenations, introducing soft information improves the guaranteed weight of correctable errors of hard-decision decoding by a factor of two.

Finally, recall that we have proved the existence of sequences of expander codes of rate R whose relative distance approaches the Zyablov bound $\delta_Z(R)$. By the results in this section, submitting these codes to soft-decision expander decoding will correct a proportion of errors close to $(1/2)\delta_Z$.

IV. Conclusion

The main idea of this paper was to show that parallel and serial concatenations form very closely related classes of codes. In particular, parallel concatenations with interleavers given by expander graphs attain the well-known estimates of the relative distance (the Zyablov bound) and the error probability of decoding (the Forney bound) of serially concatenated codes. The decoding complexity of the codes grows linearly with the code length.

Though this paper has focused on the similarities between both versions of concatenation, we believe there are a number of potential advantages to the parallel version, besides the improvement to decoding complexity. As was mentioned in the Introduction, it was already shown in [6] that Forney's exponent can sometimes be surpassed, though with some restrictions. In recent work [19], we prove that the present expander code variant can be made to constructively surpass the Zyablov bound on minimum distances.

References

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

[2] M. Tanner, "A recursive approach to low-complexity codes," *IEEE Trans. Inf.. Theory*, vol. IT-27, no. 5, pp. 1710–1722, Sep. 1981.

[3] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Serial concatenation of interleaved codes: Performance analysis, design, and iterative decoding," *IEEE Trans. Inf. Theory*, vol. 44, no. 3, pp. 909–926, May 1998.

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

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

[6] A. Barg and G. Zémor, "Error exponents of expander codes under linear-complexity decoding," *SIAM J. Discr. Math.*, vol. 17, no. 2, pp. 426–445, 2004.

[7] V. Guruswami and P. Indyk, "Near-optimal linear-time codes for unique decoding and new list-decodable codes over smaller alphabets," in *Proc. ACM Symp. Theory of Computing (STOC)*, Montreal, QC, Canada, May 2002, pp. 812–821.

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

[9] I. Dumer, "Concatenated codes and their multilevel generalizations," in *Handbook of Coding Theory*, V. Pless and W. C. Huffman, Eds. Amsterdam, The Netherlands: Elsevier Science, 1998, vol. 2, pp. 1911–1988.

[10] V. V. Zyablov, "An estimate of complexity of constructing binary linear cascade codes," *Probl. Pered. Inf.*, vol. 7, no. 1, pp. 3–10, 1971.

[11] S. Benedetto and G. Montorsi, "Unveiling turbo codes: Some results on parallel concatenated coding schemes," *IEEE Trans. Inf. Theory*, vol. 42, no. 2, pp. 409–428, Mar. 1996.

[12] N. Alon, J. Bruck, J. Naor, M. Naor, and R. M. Roth, "Construction of asymptotically good low-rate error correcting codes through pseudo-random graphs," *IEEE Trans. Inf. Theory*, vol. 38, no. 2, pp. 509–516, Mar. 1992.

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

[14] V. Skachek and R. Roth, "Generalized minimum distance decoding of expander codes," in *Proc. IEEE Information Theory Workshop*, Paris, France, Mar. 2003, pp. 245–248.

[15] G. D. Forney Jr, "On iterative decoding and the two-way algorithm," in *Proc. Int. Symp. Turbo Codes and Related Topics*, Brest, France, 1997, pp. 12–25.

[16] ——, "Exponential error bounds for erasure, list, and decision feedback schemes," *IEEE Trans. Inf. Theory*, vol. IT-14, no. 2, pp. 206–220, Mar. 1968.

[17] E. L. Blokh and V. V. Zyablov, *Linear Concatenated Codes* (in Russian). Moscow, U.S.S.R.: Nauka, 1982.

[18] A. Barg, "Improved error bounds for the erasure/list scheme: The binary and spherical cases," *IEEE Trans. Inf. Theory*, vol. 50, no. 10, pp. 2503–2511, Oct. 2004.

[19] A. Barg and G. Zémor. Distance Properties of Expander Codes. [Online]. Available: http://arxiv.org/abs/cs.IT/0409010