Finite-Length Analysis of a Capacity-Achieving Ensemble for the Binary Erasure Channel

Henry D. Pfister

EPFL, School of Computer & Communication Sciences
Lausanne, CH-1015, Switzerland
henry.pfister@epfl.ch

***Abstract*— In this paper, we consider the finite-length performance of a capacity-achieving sequence of irregular repeat-accumulate (IRA) code ensembles. We focus on a sequence of bit-regular ensembles with degree 3 which was shown to achieve capacity with bounded complexity [9]. To characterize how fast the block length of the code must grow with respect to the truncation point of the degree distribution (i.e., maximum check degree), we compute an upper bound on the average weight enumerator. Based on this analysis, we present a particular truncation sequence that could achieve a minimum distance which grows like $n^{1/3}$ even as the gap to capacity goes to zero. We also consider the performance of these codes in the waterfall region by extending the finite-length scaling law [1] from low-density parity-check codes to IRA codes. This shows that the performance near the iterative decoding threshold is well characterized by a suitably scaled Q-function for large enough block length. Numerical results are given for the scaling parameters of this ensemble sequence and for a few other IRA codes. Unfortunately, the simulation results for the capacity-achieving sequence start to match the scaling law only for very large block lengths.**

I. Introduction

The last decade has seen enormous progress in in the construction of low-complexity error-correcting codes which approach the capacity of many standard communication channels. Indeed, there are now a number of code constructions which provably achieve the capacity of the binary erasure channel (BEC) using iterative decoding. The first degree distribution (d.d.) for low-density parity-check (LDPC) codes that was proven to achieve capacity on the BEC was given by Luby et al. [6], [5]. Shokrollahi introduced another in [12] and then an infinite family of them with Oswald [7]. For irregular repeat-accumulate (IRA) codes, capacity-achieving ensembles were introduced first by Jin et al. in [4] and then improved by Sason and Urbanke in [11]. The first capacity-achieving d.d. with *bounded complexity* (i.e., whose decoding complexity per information bit is bounded as the gap to capacity vanishes) was introduced by Pfister et al. [9] using non-systematic IRA codes.

In this paper, we consider the non-systematic IRA ensemble from [9] with a fixed bit degree of 3 and an irregular check d.d. supported on the positive integers. By defining a sequence ensembles whose truncation depth depends on the the block length, we can connect the notion of gap to capacity with block length. Based on our analysis of the weight enumerator (WE) in Section III, we consider a particular truncation sequence that could achieve a minimum distance which grows like $n^{1/3}$ even as the gap to capacity goes to zero. We also extend the finite-length scaling law of Amraoui et al. [1][2] for LDPC codes to IRA codes. The performance near the iterative decoding threshold can therefore be characterized by a suitably scaled Q-function. Numerical results are given in Section IV for the scaling parameters of this ensemble sequence and for a few other IRA codes. Unfortunately, the simulation results for the capacity-achieving sequence start to match the scaling law only for very large block lengths.

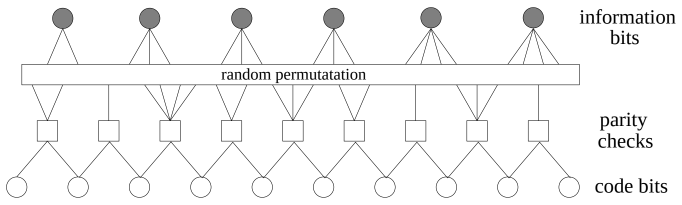

Fig. 1. Gallager-Tanner-Wiberg decoding graph for an IRA code.

II. Code Ensemble and Iterative Decoding

Consider the ensemble of IRA codes described in [9]. The structure of the ensemble is defined by the degree distribution functions, $L(x) = \sum_{i\geq 1} L_i x^i$ and $R(x) = \sum_{i\geq 1} R_i x^i$, where L_i is the fraction of information bits which are repeated i times and R_j is the fraction of parity checks with degree $j+2$. The decoding graph is shown in Figure 1, and the code rate is computed by counting edges in this graph. Doing this with k information bits shows that there are $n = k\left(\gamma + L'(1)/R'(1)\right)$ code bits, where $L'(1)$ is the average bit degree, $R'(1)+2$ is the average check degree, and γ is the fraction of systematic bits transmitted. Therefore, the code rate is given by $r = \left(\gamma + L'(1)/R'(1)\right)^{-1}$.

Suppose that a random code is chosen from this ensemble and used to communicate over a BEC with erasure probability p. The performance of iterative decoding can be analyzed by tracking the average fraction of erasure messages passed during the lth iteration. This technique was introduced in [10] and is known as density evolution (DE). Let $x_0^{(l)}$ be the message erasure rate from information bits to parity checks during the kth step. Assuming that a fraction γ of the systematic bits are transmitted, the update equation (see

[4]) for $x_0^{(l+1)}$ is given

$$x_0^{(l+1)} = p_0\lambda\left(1 - \frac{(1-p)^2\rho\left(1-x_0^{(l)}\right)}{\left(1-pR\left(1-x_0^{(l)}\right)\right)^2}\right), \qquad (1)$$

where $p_0 = 1-\gamma(1-p)$, $\lambda(x) = L'(x)/L'(1)$, and $\rho(x) = R'(x)/R'(1)$. Successful decoding requires that the erasure probability vanishes as $l \to \infty$, and this is guaranteed if $x_0^{(l+1)} < x_0^{(l)}$ whenever $x_0^{(l)} > 0$.

In this paper, we focus on the bit-regular ensemble of degree three (i.e., $\lambda(x) = x^2$) whose check d.d. is defined by

$$\rho(x) = \frac{1-(1-x)^{1/2}}{\left(1-p\left(1-3x+2\left(1-(1-x)^{3/2}\right)\right)\right)^2}. \qquad (2)$$

It was shown in [9] that this d.d. pair satisfies (1) with equality and that $\rho(x)$ has a non-negative power series expansion for $p \in (0, \frac{1}{13}]$. Therefore, we can define a sequence of codes by truncating of the power series of $\rho(x)$ which achieves capacity in the limit. Consider the sequence of truncated d.d. given by

$$\rho_M(x) = \sum_{i=2}^{M-1} \rho_i x^{i-1} + \varepsilon_M x^{M-1},$$

where $\rho_i = [x^{i-1}]\rho(x)$ and $\varepsilon_M = \sum_{i=M}^{\infty} \rho_i$. A small fraction γ_M of the systematic bits must be sent to get decoding started. Empirically, we find that $\gamma_M \sim M^{-1}$ and that the gap to capacity also decays like M^{-1}. If the block length, n, goes to infinity for each fixed M, then the concentration theorem of [10] shows that the probability of decoding failure goes to zero as well. If the block length is chosen instead as a function of M, then it is much more difficult to say how this ensemble behaves. Understanding the relationship between block length and gap to capacity requires this type of limit though. In [9], the asymptotic decay of the R_k coefficients was determined to be $O\left(k^{-5/2}\right)$. This implies that

$$R''(1) = \sum_{k=1}^{M} k(k-1)O\left(k^{-5/2}\right) = O\left(M^{1/2}\right). \qquad (3)$$

We also note that the decay rate of the degree distribution implies fractional check nodes[1] unless $nR_M = \Omega(1)$ or equivalently $n = \Omega\left(M^{5/2}\right)$.

III. Weight Enumerator Analysis

The stability of our ensemble sequence can be understood by considering the weight enumerator (WE) as a function of M and n. While each ensemble in the sequence is unconditionally stable (since there are no degree 2 information bits) for finite M, there is no guarantee this remains true when $n = f(M)$. We start by deriving an upper bound on the input output weight enumerator IOWE of the ensemble. We do this by viewing the bit regular non-systematic IRA codes as the serial concatenation of a repeat code, an irregular single parity check (SPC) code, and an accumulate code. First, the IOWE of each code is given and then they are combined to get the final answer. Let n be the block length code and also the number of parity check nodes. Connecting all edges implies that the number of information bits is therefore $nR'(1)/3$.

[1]In practice, the number of parity checks must be integer so gaps of zeros will appear between non-zero elements.

The IOWE, $A_{w,h}$, of a code equals the number of input patterns of weight w which are mapped to output patterns of weight h. Since the repeat code is bit-regular with degree 3, its IOWE is given by

$$A_{p,s}^{(rep)} = [x^p y^s]\left(1+xy^3\right)^{nR'(1)/3} = \binom{nR'(1)/3}{p}\delta_{s,3p},$$

where $\delta_{i,j}$ is the Kronecker delta function. The irregular (SPC) code is mixture of SPC codes where a fraction R_k of the codes take k inputs and all codes output the modulo-2 sum (i.e. parity) of their inputs. Its IOWE can be upper bounded by

$$A_{s,q}^{(par)} = [x^s y^q]\prod_{i=1}^{M}\left[\sum_j \binom{i}{2j}x^{2j} + \sum_j \binom{i}{2j+1}x^{2j+1}\right]^{nR_i}$$
$$\le \binom{n}{q}(R'(1))^q \frac{\left(\frac{1}{2}nR''(1)\right)^k}{k!}\delta_{s-2k,q},$$

and the bound is asymptotically tight for $s = o(\sqrt{n})$. The cumulative IOWE of the accumulate code can written and upper bounded with

$$A_{q,\le w}^{(acc)} = \sum_{i=1}^{w}\binom{n-i}{\lfloor q/2\rfloor}\binom{i-1}{\lceil q/2\rceil - 1} \le \binom{n}{\lfloor q/2\rfloor}\frac{w^{\lceil q/2\rceil}}{\lceil q/2\rceil!}.$$

Combining these, we can upper bound the average number of codewords of with information weight p and code weight not more than w by

$$\overline{A}_{p,\le w}^{(IRA)} = \sum_{s,q} A_{p,s}^{(rep)}\frac{A_{s,q}^{(par)}}{\binom{nR'(1)}{s}}\frac{A_{q,\le w}^{(acc)}}{\binom{n}{q}},$$

which can be upper bounded by

$$\frac{\binom{nR'(1)/3}{p}}{\binom{nR'(1)}{3p}}\sum_{k=0}^{\lfloor 3p/2\rfloor}\binom{n}{\lfloor 3p/2-k\rfloor}\frac{w^{\lceil 3p/2-k\rceil}}{\lceil 3p/2-k\rceil!}\frac{R'(1)^{3p}}{R'(1)^{2k}}\frac{(nR''(1))^k}{k!2^k}.$$

To get a better understanding how this behaves with large n, we assume that $w = O(n^\alpha)$, $R''(1) = O\left(n^\beta\right)$, and compute

$$\begin{aligned}\lim_{n\to\infty}\log_n \overline{A}_{p,\le w}^{(IRA)} &= \\ &\le p - 3p + \lfloor 3p/2\rfloor + \lceil 3p/2\rceil\alpha - k\alpha + k\beta \\ &\overset{(a)}{=} p - \lceil 3p/2\rceil + \lceil 3p/2\rceil\alpha + k(\beta-\alpha) \\ &\overset{(b)}{\le} p - \lceil 3p/2\rceil(1-\beta) \\ &\overset{(c)}{\le} -\lceil 3p/2\rceil\varepsilon,\end{aligned}$$

where (a) uses $-3p + \lfloor 3p/2\rfloor = p - \lceil 3p/2\rceil$, (b) assumes $\alpha \le \beta$ and upper bounds with $k = \lceil 3p/2\rceil$, and (c) assumes $\beta \le 1/3 - \varepsilon$. This shows that the probability an input of fixed weight p generates a codeword of weight less than $n^{1/3}$ goes to

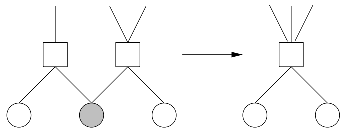

Fig. 2. Graph reduction sums adjacent parity checks to remove erased bits.

zero as $n \to \infty$. The requirement is that $R''(1) = O\left(n^{1/3-\varepsilon}\right)$ which, using (3), we see is equivalent to $n = \Omega\left(M^{3/2+\varepsilon}\right)$. Finally, we believe the difficulty in extending this result to the minimum distance (i.e., that it grows like $n^{1/3}$) is not related to the weakness of our bound but instead to the weakness of our bound (not shown) on the sum over k and p.

IV. Finite Length Scaling for IRA Codes

There are two distinct approaches to the finite-length analysis of graph based codes on the BEC. The first is based on counting static erasure patterns, known as *stopping sets*, which cause the iterative decoder to fail. This approach was taken by Di et al. in [3] to compute the average probability of iterative decoding failure for the ensemble of LDPC codes. The second is based on the dynamics of the residual graph process defined by Luby et al. in [6]. This involves tracking the number of bits and checks in residual graph as decoding progresses. Decoding fails only if the number of degree 1 checks becomes zero before decoding is complete. Recently, this approach was taken by Amraoui et al. in [1] to characterize probability of iterative decoding failure near the threshold as a function of the block length. Their result is that the probability of block erasure P_B for a code of length n is given by

$$P_B = Q\left(\frac{\sqrt{n}(p^* - \beta n^{-2/3} - p)}{\alpha}\right),$$

in the limit as $n \to \infty$ with $\sqrt{n}\,(p^* - p) = \Theta(1)$. The parameter α is related to variance of the number of degree 1 checks at the critical point and the parameter β is related to width of parabola at the critical point. With the addition of preprocessing step we call *graph reduction*, we find that it is possible to reduce an IRA code to an LDPC code [8]. Since this preprocessing step is amenable to the same type of analysis used for scaling, and we can combine the two and get the scaling result for IRA codes. In this section, we describe that preprocessing step and give some results.

The finite-length scaling analysis of [1] works by breaking the decoding into a large number of small steps. This allows one to prove that the state converges weakly to Gaussian random vector as the system size goes to infinity. Therefore, one need only track the mean and covariance of the state. This can be done with a set of differential equations defined solely in terms of the small steps. We introduce a simple preprocessing step which has this same property (i.e., it is composed of many small steps) which allows one to decode an IRA code on the BEC by first reducing it to an LDPC code. The concentration and weak convergence to a Gaussian follows immediately from [1].

The decoding process of a randomly chosen IRA code with a fixed degree distribution can be broken into four stages. The first two stages comprise *graph reduction*, while the last two stages correspond to the LDPC decoding process from [6]. The whole process is as follows. Figure 2 shows step (1c).

1) For each code bit erasure
   a) Pick a code bit randomly
   b) Attach its two edges to distinct parity checks
   c) Sum the two parity checks to eliminate the bit
2) Since all remaining code bits are now known
   a) Pass their values and remove their edges
3) For each information bit received correctly
   a) Pick an information bit randomly
   b) Attach its edges to random check edges
   c) Pass its bit value to the checks
   d) Remove the information bit and its edges
4) While there are still degree 1 checks left
   a) Pick a random information bit edge
   b) Attach it to the degree 1 check
   c) Attach its other edges to random check edges
   d) Pass the degree 1 check value to all these edges
   e) Remove the degree 1 check, the bit node, all edges

This decoding approach naturally averages over all code graphs. For example, one might worry about the possibility edge cancellation when the two parity checks are summed. But this possibility is accounted for implicitly by the fact that the LDPC code ensemble allows multiple edges.

A. Density and Covariance Evolution for Graph Reduction

Readers who are familiar with the results of [1][2] should realize immediately that the graph reduction process satisfies the conditions of the covariance evolution result. This type of process converges to fluid limit which is characterized completely by its *drift* and *local covariance*. The resulting differential equations give the trajectory in the density and covariance evolution for the graph reduction process.

Using notation similar to that of [1], we consider a system that start with n check nodes and let our state vector be $X_{n,t}$ where $X_{n,i}^{(j)}$ is the number of check nodes of degree j after $\lfloor i \rfloor$ decoding steps (i.e., check nodes removed). After i decoding steps, the probability that a randomly chosen check node will be of degree j is given by $X_{n,i}^{(j)}/(n-i)$. Therefore, the probability that a check node of degree j and a check node of degree of degree k will be replaced by a check node of degree $j+k$ is given by

$$P(j,k \to j+k) = \begin{cases} \frac{X_{n,i}^{(j)} X_{n,i}^{(k)}}{(n-i)(n-i-1)} & \text{if } j \neq k \\ \frac{X_{n,i}^{(j)}\left(X_{n,i}^{(j)}-1\right)}{(n-i)(n-i-1)} & \text{if } j = k \end{cases}.$$

The trajectory of the mean is given by the vector $\overline{z}(t) = \left(z^{(1)}(t), \ldots, z^{(d)}(t)\right)$ with

$$z^{(i)}(t) = \lim \frac{1}{n} X_{n,nt}^{(i)}.$$

The Markov kernel has a smooth limit because

$$P(j,k \to j+k) = \frac{X_{n,i}^{(j)} X_{n,i}^{(k)}}{(n-i)(n-i)} + O\left(\frac{1}{n}\right),$$

and therefore we have the differential equation

$$\frac{d}{dt} z^{(i)}(t) = f_t^{(i)} \left(z^{(1)}, z^{(2)}, \ldots, z^{(d)} \right), \tag{4}$$

where $f_t^{(i)}(\overline{x})$ is the state dependent drift given by

$$f_t^{(i)}(\overline{x}) = \begin{cases} \frac{-2x^{(i)}}{(1-t)} + \sum_{k=1}^{i-1} \frac{x^{(i-k)}x^{(k)}}{(1-t)^2} & \text{if } i < d \\ \frac{-2x^{(d)}}{(1-t)} + \sum_{j=d}^{2d} \sum_{k=1}^{j-1} \frac{x^{(j-k)}x^{(k)}}{(1-t)^2} & \text{if } i = d \end{cases}.$$

Notice that we have introduced a maximum check degree of d. This truncation implies that $X_{n,i}^{(d)}$ really tracks the fraction of check of degree d or larger.

This differential equation actually has an algebraic formulation if there is no maximum degree. If we let

$$Z(t, x) = \sum_i z^{(i)}(t) x^i,$$

then we can write (4) compactly as

$$\frac{\partial}{\partial t} Z(t, x) = -\frac{2Z(t, x)}{1-t} + \left(\frac{Z(t, x)}{1-t} \right)^2,$$

where $Z(0, x) = R(x)$ is the initial check node degree distribution. It is easy to verify that its solution is

$$Z(t, x) = \frac{(1-t)^2 R(x)}{1 - tR(x)}$$

so that $z^{(i)}(t) = [x^i] Z(t, x)$.

Once you can track the evolution of the mean, the next step is tracking the evolution of the covariance

$$W^{(ij)}(t) = \lim_{n \to \infty} \frac{1}{n} \left(E\left[X_{n,nt}^{(i)} X_{n,nt}^{(j)} \right] - E\left[X_{n,nt}^{(i)} \right] E\left[X_{n,nt}^{(i)} \right] \right).$$

The differential equation for $W^{(ij)}(t)$ (see [1]) is given by

$$\frac{d}{dt} W^{(ij)}(t) = f_t^{(ij)}(\overline{x}) + \sum_{k=0}^{d} W^{(ik)}(t) J_t^{(jk)}(\overline{x}) + W^{(kj)}(t) J_t^{(ik)}(\overline{x})$$

evaluated at $\overline{x} = \overline{z}(t)$, where $f_t^{(ij)}(\overline{x})$ is the local covariance and $J_t^{(ij)}(\overline{x}) = \frac{\partial f_t^{(i)}}{\partial x^{(j)}}(\overline{x})$ is the Jacobian of $f_t(\overline{x})$. The local covariance is given by $f_t^{(ij)}(\overline{x}) = g_t^{(ij)}(\overline{x}) - f_t^{(i)}(\overline{x}) f_t^{(j)}(\overline{x})$ where

$$g_t^{(kl)}(\overline{x}) = \sum_{i=1}^{d} \sum_{j=1}^{d} \frac{x^{(i)}x^{(j)}}{(1-t)^2} Q(i,j,k) Q(i,j,l),$$

and

$$Q(i,j,k) = \begin{cases} -\delta_{i,k} - \delta_{j,k} + \delta_{i+j,k} & \text{if } k < d \\ -\delta_{i,k} - \delta_{j,k} + \sum_{m=d}^{2d} \delta_{i+j,m} & \text{if } k = d \end{cases}.$$

B. Putting It All Together

The state of the differential system at time t gives the degree distribution and covariance of the IRA code after exactly nt code bits have been erased. Before one can start LDPC decoding on the reduced graph, one must change variables from the node perspective to the edge perspective. Fortunately, this is linear transformation of the state, so it easy to calculate the new mean and covariance. The mean and convariance of the fraction of edges attached to a degree i bit after graph reduction is given by

$$y^{(i)} = \frac{i\, z^{(i)}(p)}{R'(1)} \qquad V^{(ij)} = \frac{ij\, W^{(ij)}(p)}{R'(1)^2}.$$

We note that both quantities are now implicitly scaled by the number of edges in the graph rather than the number of parity checks. The initial condition of the covariance evolution for irregular LDPC code decoding (see [2]) is now given by the variables $\overline{y}$ and $\overline{V}$.

The scaling law also depends critically on the derivative of the number of degree 1 checks with respect to the erasure probability at the critical point. For IRA codes, the fraction of degree 1 checks is given by

$$r_1(y, p) = 1 - \gamma(1-p)\lambda(y) \left(y - 1 + \frac{(1-p)^2 \rho(1-y)}{(1 - pR(1-y))^2} \right),$$

where y is erasure probability at the output of the check nodes (see [6]). The iterative decoding threshold p^* is defined to be the largest p such that $\min_{y \in [0,1]} r_1(y, p) \geq 0$ and the critical point y^* is point where $r_1(y, p^*)$ just touches zero. The expression for the derivative of $r_1(y, p)$with respect to p at the critical point is given by

$$\left. \frac{\partial r_1}{\partial p} \right|_* = -\frac{2p_0 \lambda(y)(1-y)(1 - R(1 - p_0\lambda(y)))}{(1-p)(1 - pR(1 - p_0\lambda(y)))} - \frac{\lambda(y)^2}{\lambda'(y)}.$$

To get the correct scaling parameters, we must also normalize block lengths between families. This will make it easier to produce fair comparisons. The easiest way to do this is to realize that differential system for decoding given in [2] is implicitly scaled by the number of edges in the graph[2]. Let v be the variance of the number of degree 1 checks at the critical point. The scaling parameter α for an LDPC code is given by

$$\alpha_{LDPC} = \left(\frac{\partial r_1}{\partial p} \right)_{LDPC}^{-1} \sqrt{\frac{v^2}{L'(1)} + p^*(1-p^*)}.$$

The division by $L'(1)$ maps the number of edges to the block length. This is the same formula used in [2], noting that $\int_0^1 \lambda(x)dx = 1/L'(1)$. For IRA codes, the block length is $\frac{\gamma}{L'(1)} + \frac{1}{R'(1)}$ times the number of edges, so we have

$$\alpha_{IRA} = \left(\frac{\partial r_1}{\partial p} \right)_{IRA}^{-1} \sqrt{\frac{\gamma v^2}{L'(1)} + \frac{v^2}{R'(1)} + p^*(1-p^*)}.$$

[2]The analysis starts with a edge degree distribution which sums to one and we should multiply by n to get back to the original system.

Code	γ	Rate	p^*	α	β
SIRA (3,3)	1	.5	.4448	.5959	.8387
SIRA (4,4)	1	.5	.4451	.5710	.7667
SIRA (3,9)	1	.75	.2139	.4788	.6590
IRA M=20	.0019	.9126	.0754	.4122	2.938
IRA M=30	.0021	.9173	.0754	.4842	4.079
IRA M=40	.0020	.9194	.0754	.5684	5.462
IRA M=50	.0019	.9206	.0753	.6577	7.017
IRA M=60	.0017	.9214	.0753	.7491	8.737

TABLE I
SCALING RESULTS FOR VARIOUS IRA CODES.

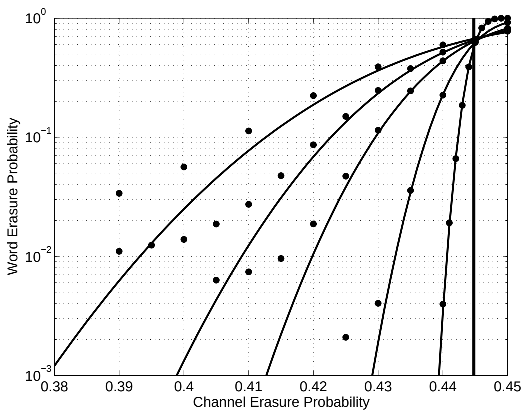

Fig. 3. Finite-length scaling (●) versus simulation (−) for a regular (3,3) SIRA code. The block lengths from left to right are 1024, 2048, 4096, 16384, and 131072.

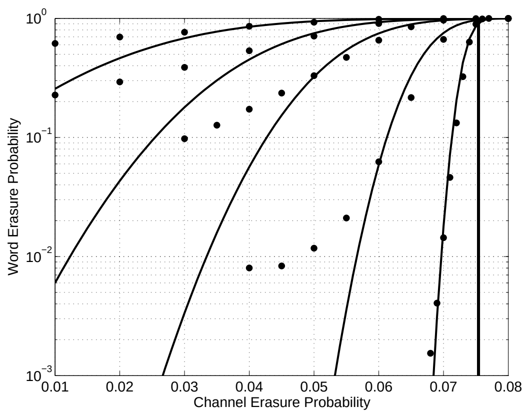

Fig. 4. Finite-length scaling (●) versus simulation (−) for a capacity achieving ensemble with $M = 40$. The block lengths from left to right are 1024, 2048, 4096, 16384, and 131072.

The shift parameter must also be properly mapped from LDPC codes to IRA codes. Since the block length of our LDPC code after graph reduction is equal to the number of information bits, we can simply use the derivation in [2] for β_{LDPC} and write $\beta_{IRA} = \beta_{LDPC} \left(\gamma + L'(1)/R'(1)\right)^{2/3}$.

C. Results

Table I gives the scaling parameters for a number of IRA codes. The first three codes listed, notated SIRA (j, k), correspond to regular systematic IRA codes with bit degree j and check degree k. Figure 3 compares simulation results with the finite-length scaling law for the SIRA (3,3) code. Notice that the curves match quite well near the threshold. The pronounced error floor is due to the fact that we are averaging over all graphs and some are quite bad.

The next five codes listed in Table I correspond to the capacity achieving sequence discussed in Section II. Notice that the fraction of systematic bits is quite small and that the β parameter grows rapidly with the sequence. This is not surprising because the β parameter is related to the width of the parabola near the critical point of $r_1(y, p)$, and the capacity achieving sequence is trying to make this curve completely flat. Figure 4 compares simulation results with the finite-length scaling law for the $M = 40$ code. Notice that the scaling law is quite bad for small block lengths, but starts to look pretty good around $n = 16384$.

V. Conclusion

The goal of this research was to improve our understanding of the finite-length performance for sequence of capacity achieving codes. The WE and the finite-length scaling law was considered for a particular code sequence. Although neither analysis gave definitive answers about the achievable trade-off between block length and distance from capacity, there are a few things worth noting. First, it appears that the finite-length scaling law has some weaknesses when it comes to capacity achieving sequences. Based on what we learned, it seems likely that the β parameter will go to infinity as the gap to capacity goes to zero. Secondly, even though our WE analysis implied this sequence could have a minimum distance growing like $n^{1/3}$ as the gap to capacity vanishes, it appears from the simulations that this capacity achieving sequence has severe error flooring for short block lengths.

Acknowledgement

The author would like to thank Abdelaziz Amraoui and Rüediger Urbanke for helpful discussions and a preprint of their paper.

References

[1] A. Amraoui, A. Montanari, T. Richardson, and R. Urbanke. Finite-length scaling for iteratively decoded LDPC ensembles. submitted to *IEEE Trans. on Inform. Theory*.

[2] A. Amraoui, A. Montanari, and R. Urbanke. Finite-length scaling of irregular LDPC code ensembles. In *Proc. IEEE Inform. Theory Workshop*, New Zealand, Sept. 2005.

[3] C. Di, D. Proietti, E. Telatar, T. J. Richardson, and R. Urbanke. Finite-length analysis of low-density parity-check codes on the binary erasure channel. *IEEE Trans. Inform. Theory*, 48(6):1570–1579, June 2002.

[4] H. Jin, A. Khandekar, and R. J. McEliece. Irregular repeat-accumulate codes. In *Proc. Int. Symp. on Turbo Codes & Related Topics*, Brest, France, Sept. 2000.

[5] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman. Efficient erasure correcting codes. *IEEE Trans. Inform. Theory*, 47(2):569–584, Feb. 2001.

[6] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, D. A. Spielman, and V. Stemann. Practical loss-resilient codes. In *Proc. of the 29th Annual ACM Symp. on Theory of Comp.*, pages 150–159, 1997.

[7] P. Oswald and M. A. Shokrollahi. Capacity-achieving sequences for the erasure channel. *IEEE Trans. Inform. Theory*, 48(12):3017–3028, Dec. 2002.

[8] H. D. Pfister and I. Sason. Accumulate-repeat-accumulate codes: Systematic codes achieving the binary erasure channel capacity with bounded complexity. to be submitted to *IEEE Trans. on Inform. Theory*.

[9] H. D. Pfister, I. Sason, and R. Urbanke. Capacity-achieving ensembles for the binary erasure channel with bounded complexity. *IEEE Trans. Inform. Theory*, 51(7):2352–2379, July 2005.

[10] T. J. Richardson and R. L. Urbanke. The capacity of low-density parity check codes under message-passing decoding. *IEEE Trans. Inform. Theory*, 47(2):599–618, Feb. 2001.

[11] I. Sason and R. Urbanke. Complexity versus performance of capacity-achieving irregular repeat-accumulate codes on the binary erasure channel. *IEEE Trans. Inform. Theory*, 50(6):1247–1256, June 2004.

[12] M. A. Shokrollahi. New sequences of linear time erasure codes approaching the channel capacity. In *Applicable Algebra in Eng., Commun. Comp.*, pages 65–76, 1999.