On the Performance of Finite Length Irregular Repeat Accumulate Codes

Souad Guemghar and Giuseppe Caire
Institut Eurécom, Sophia-Antipolis, France, firstname.lastname@eurecom.fr

Abstract

We study the performance of finite length regular and irregular repeat accumulate (RA) codes, whose Tanner graphs are constructed semi-randomly and satisfy one of the following criteria: (a) the graph is free of cycles up to $2S$, (b) cycles of length up to $2d_{ACE}$ have at least η external edges (S, d_{ACE} and η are design parameters). We derive an upper bound on the bit error probability of random RA codes under maximum likelihood decoding. For short block lengths, it is found that the performance under belief propagation decoding of the semi-random regular RA ensemble is superior to the performance of the random ensemble under maximum likelihood decoding, and that it is comparable to the best known low density parity check codes, with comparable girth conditioning and block length. As for large block lengths, the performance of the semi-random irregular RA ensemble is superior to the performance of the random ensemble, but remains inferior to that of low density parity check codes with comparable graph conditioning and block length.

1 Introduction

In this paper, we are concerned with the performance of finite length regular and irregular Repeat Accumulate (RA) codes when used on the binary input additive white Gaussian channel (BIAWGNC). These codes are an appealing choice because they have a low encoding/decoding complexity and their performance, in terms of iterative decoding thresholds (ideal infinite length), is competitive with that of turbo codes [1] and low density parity check (LDPC) codes [2]. One possible application of RA codes is joint source-channel coding since they can be seen as an instance of the so-called low-density generator matrix (LGDM) codes [3], [4].

The performance of the random-like ensemble of systematic Irregular Repeat Accumulate (IRA) codes, in the limit of large block length, optimized in [5] for binary-input symmetric-output channels, is found to be very close to the Shannon limit. The bipartite graphs associated to these codes are free of cycles of any finite length, in the limit of infinite block length. However, finite length bipartite graphs may contain short cycles, and the belief propagation (BP) message-passing decoder is no longer optimal, since the local message independence assumption is no longer valid. It is commonly believed (based on heuristic arguments) that the removal of short cycles improves the performance of short length bipartite graph codes under BP. It is also found that maximizing the length of the shortest cycle yields a large minimum distance [6], [7], thus improving the code performance in the high signal to noise ratio (SNR) region. We therefore use the progressive edge-growth algorithm (PEG) [7] to construct regular Repeat Accumulate (regular RA) and IRA codes, whose associated bipartite graph has a shortest cycle length (girth) at least equal to $2S + 2$, where S is a design parameter.

Under BP decoding of random LDPC codes, it is found that small stopping sets [8] result in high error floors on the binary erasure channel (BEC). So, alternatively to girth conditioning, we use the method proposed in [9], to construct IRA codes whose Tanner graphs are free of small stopping sets.

It is instructive to determine the average random IRA performance on the BIAWGNC, under maximum likelihood (ML) decoding. This allows to determine whether a poor performance of the random IRA ensemble is due to the BP decoder or to the code ensemble itself. Moreover, it is interesting to compare the performance of the semi-random ensemble performance under BP against that of the random ensemble under ML decoding.

The rest of the paper is organized as follows. Section 2 presents the systematic IRA encoder. In section 3 we show how to construct the semi-random IRA code ensemble using the two methods of [7] and [9]. Section 4 shows an upper bound on the girth of the Tanner graph of IRA codes, and compares the theoretical and experimental results for regular RA codes. Section 5 shows how to compute the input-output weight enumerators (IOWE) of regular RA and IRA codes in order to determine an upper bound on the random IRA ensemble performance under ML decoding. Section 6 presents the average random and semi-random ensemble performances of regular RA codes for short block lengths and IRA codes for large block lengths. Also shown

The work of Souad Guemghar was supported in part by a PACA research grant.

are the performances of two explicit random and semi-random (regular) code constructions selected to have large minimum distances. Finally, we draw conclusions in Section 7.

2 Definitions and Background

Fig. 1 shows the block diagram of a systematic IRA encoder. A block of information bits $\mathbf{b} = (b_1, \ldots, b_k) \in \mathbb{F}_2^k$ is encoded by a repetition code of rate k/N. Each information bit is repeated r_j times, where $(r_1, \ldots, r_k)$ is a sequence of integers such that $2 \leq r_j \leq d$ and $\sum_{j=1}^{k} r_j = N$ (d is the maximum repetition degree). If $r_1 = r_2 = \ldots = r_N$, then the code is a regular RA code. Otherwise, it is an IRA code. The block of repeated symbols is interleaved, with a one-to-one permutation, and the resulting block $\mathbf{x}_1 = (x_{1,1}, \ldots, x_{1,N}) \in \mathbb{F}_2^N$ is fed into an accumulator, defined by the recursion

$$x_{2,j+1} = x_{2,j} + \sum_{i=0}^{a-1} x_{1,aj+i}, \quad j = 0, \ldots, n-1 \quad (1)$$

with initial condition $x_{2,0} = 0$, where $\mathbf{x}_2 = (x_{2,1}, \ldots, x_{2,n}) \in \mathbb{F}_2^n$ is the accumulator output block corresponding to the input $\mathbf{x}_1$, $a \geq 1$ is a given integer (referred to as *grouping factor*), and we assume that $n = N/a$ is an integer. Finally, the codeword corresponding to the information block $\mathbf{b}$ is given by $\mathbf{x} = (\mathbf{b}, \mathbf{x}_2)$ of length $m = k + n$. The code rate is therefore given by

$$R = \frac{k}{k+n} = \frac{a}{a + \frac{1}{k}\sum_{j=1}^{k} r_j} \quad (2)$$

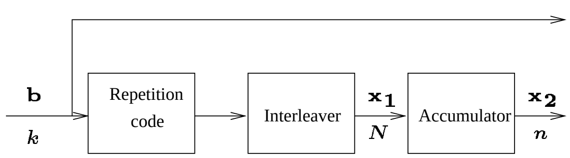

Fig. 1. IRA encoder

The Tanner graph [10] of an IRA code is shown in Fig. 2. In general, the Tanner graph of a linear code is a bipartite graph whose node set is partitioned into two subsets: the *bitnodes*, corresponding to the coded symbols, and the *checknodes*, corresponding to the parity-check equations that codewords must satisfy. The graph has an edge between bitnode α and checknode β if the symbol corresponding to α participates in the parity-check equation corresponding to β.

Since the IRA encoder is systematic (cf. Fig. 1), it is useful to further distinguish the bitnodes into two subclasses: the information bitnodes $\{v_j, j = 1, \ldots, k\}$, corresponding to information bits, and the parity bitnodes $\{p_j, j = 1, \ldots, n\}$, corresponding to the symbols output by the accumulator. Those information bits that are repeated r_j times are represented by bitnodes with degree r_j, as they participate in r_j parity-check equations. Each checknode $c_j,\ j = 1, \ldots, n$ is connected to a information bitnodes and to two parity bitnodes and represents one of the equations (1) for a particular j. The connections between checknodes and information bitnodes are determined by the interleaver. The connections between checknodes and parity bitnodes are arranged in a regular zig-zag pattern since, according to (1), every pair of consecutive parity bits are involved in one parity-check equation.

Let there be n_{deg} different repetition degrees r_j. Then, the repetition code can alternatively be defined in terms of a degree distribution $\{0 \leq f_i \leq 1, i = 1, \ldots, n_{deg}\}$ and a degree set $\{2 \leq d_i \leq d, i = 1, \ldots, n_{deg}\}$, such that $f_i k$ is the number of information bitnodes of degree d_i, and

$$\sum_{i=1}^{n_{deg}} f_i = 1 \quad (3)$$

Therefore, the graph has $N = \bar{d}k$ edges between information bitnodes and checknodes where $\bar{d}$ is the average information bitnode degree, given by:

$$\bar{d} = \sum_{j=1}^{n_{deg}} d_j f_j \quad (4)$$

The bit error rate (BER) performance of the BP message-passing decoder averaged over the IRA code ensemble can be analyzed in the limit of $k \to \infty$, by the Density Evolution (DE) technique [2], as shown in [5]. In [5], we have proposed DE approximation methods to optimize the degree sequences of the IRA code ensemble, for a broad class of binary-input symmetric-output channels. It is shown that some of the proposed methods on the binary symmetric channel (BSC) and the BIAWGNC, yield IRA codes that are competitive with respect to the best-known LDPC codes [11].

Randomly constructed IRA codes of finite length may have poor performances under BP decoding, in the following aspects:

- In the low SNR region, the BER waterfall is far from the code ensemble threshold, hence the gap from channel capacity is not as good as predicted by the infinite length DE analysis;
- In the medium to high SNR region, the BER flattens in an even larger gap from channel capacity for very low BER. This behavior is referred to as an "error floor";
- If $f_2 \neq 0$, as is the case for optimal degree sequence distributions, then the word error rate (WER) is poor.

There exists a tradeoff between the threshold SNR and the "error floor" BER of irregular versus regular codes

[12]. For short code block lengths, regular codes usually exhibit better error floor BER than their irregular counterparts. On the other hand, for large block lengths, optimized irregular codes largely outperform their regular counterparts in approaching their DE threshold.

Different approaches have been adopted for the construction of finite-length LDPC codes. [7], [13], [14] propose methods to remove short cycles in order to maximize the girth, and [9] proposes a method to remove short cycles that contribute to small stopping sets. In the following section, we summarize the PEG algorithm [7] as well as the stopping set maximizing algorithm [9] that we use to construct IRA codes.

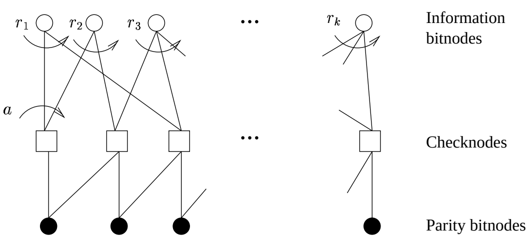

Fig. 2. Tanner graph of IRA code

3 Construction of Finite Length IRA Codes

Without loss of generality, assume that $r_1 \leq r_2 \ldots \leq r_k$, i.e., information bitnodes are indexed in the non-decreasing order of repetition degrees. Let $e_1, e_2, \ldots, e_N$ denote the edges connected to the information bitnodes $v_1, v_2, \ldots, v_k$ so that edges $e_1, e_2, \ldots, e_{r_1}$ are connected to information bitnode v_1, edges $e_{r_1+1}, e_{r_1+2}, \ldots, e_{r_1+r_2}$ are connected to information bitnode v_2, and so on. Define the mapping V as

$$\begin{aligned} V : \{1, \ldots, N\} &\rightarrow \{1, \ldots, k\} \\ j &\rightarrow i \text{ such that } e_j \text{ is connected to } v_i \end{aligned}$$

We construct the a-to-1 mapping C

$$\begin{aligned} C : \{1, \ldots, N\} &\rightarrow \{1, \ldots, n\} \\ j &\rightarrow i \text{ such that } e_j \text{ is connected to } c_i \end{aligned}$$

on an edge by edge basis. Initially, $C(N)$ can take any value in the list

$$\mathcal{L}_N = \{1, 1, \ldots 1, 2, 2 \ldots 2, \ldots, n, n, \ldots, n\}$$

which is formed by repeating every element of the set $\{1, 2, \ldots, n\}$ a times. Once $C(N)$ is selected, the list $\mathcal{L}_{N-1}$ from which $C(N-1)$ can be selected is obtained by removing $C(N)$ from $\mathcal{L}_N$. Likewise, $\mathcal{L}_j$ is obtained by removing $C(j+1)$ from the list $\mathcal{L}_{j+1}$. We construct the mapping C using either one of the following two methods.

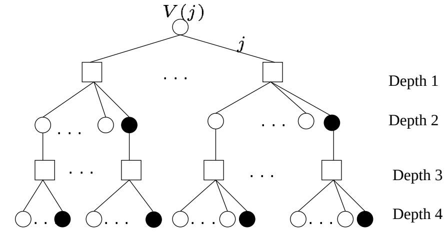

Fig. 3. Local neighborhood expanded on 4 levels

A Progressive Edge Growth Algorithm [7]: The PEG algorithm allows to construct a graph free of cycles of length $4, 6, \ldots, 2S$. Assuming that $C(N), C(N-1), \ldots, C(j+2), C(j+1)$ have all been determined, and that there are no cycles of length smaller than $2S+2$, then $C(j)$ is determined as follows. We first randomly select $C(j)$ from $\mathcal{L}_j$. Then, a local neighborhood originating at $V(j)$ is expanded up to depth S (see Fig. 3 for $S = 4$). If there is no cycle in the local neighborhood then let $\mathcal{L}_{j-1} = \mathcal{L}_j - \{C(j)\}$ and proceed to edge e_{j-1}, otherwise select another $C(j)$ and redo the previous step.

Edges e_j are assigned in the decreasing order $j = N, N-1, \ldots, 1$, i.e., in the non-increasing order of repetition degrees $r_{V(N)}, r_{V(N-1)}, \ldots, r_{V(1)}$, because it is easier to assign edges connected to high-degree information bitnodes under a girth constraint at the beginning of the algorithm than toward the end.

B Stopping Set Maximization (SSMAX) [9]: The algorithm proposed in [9] attempts to maximize the size of stopping sets in the graph, by ensuring that cycles of length $4, 6 \ldots, 2d_{ACE}$ have $ACE \geq \eta$, where ACE is the number of external edges connected to a cycle, and is given as

$$ACE \triangleq \sum_i (d_i - 2)$$

where d_i is the degree of the i^{th} (information or parity) bitnode in the cycle. Assuming that $C(N), C(N-1), \ldots, C(j+2), C(j+1)$ have all been determined, and that all cycles of length up to $2d_{ACE}$ have $ACE \geq \eta$, then $C(j)$ is determined as follows. We first randomly select $C(j)$ from $\mathcal{L}_j$. Then, a local neighborhood originating at $V(j)$ is expanded up to depth S. If all cycles in the local neighborhood, of length up to $2d_{ACE}$, have $ACE \geq \eta$, then let $\mathcal{L}_{j-1} = \mathcal{L}_j - \{C(j)\}$ and proceed to edge e_{j-1}, otherwise select another $C(j)$ and redo the previous step.

The following proposition shows two simple conditions, that if satisfied, ensure that the resulting IRA graph is free of cycles of length 4.

Proposition 1: An IRA graph satisfying the following two conditions

1)

$$\forall j, j' \in \{1, \ldots, N\},\ j' \neq j$$
$$\text{if } V(j') = V(j) \Rightarrow |C(j) - C(j')| \geq 2 \quad (5)$$

2)

$$\forall j, j', j_1, j_1' \in \{1, \ldots, N\},\ j' \neq j,\ j_1 \neq j',\ j_1' \neq j_1$$

$$\text{if} \left.\begin{array}{ll} V(j') & = V(j) \\ C(j_1) & = C(j) \\ V(j_1') & = V(j_1)\} \end{array}\right\} \Rightarrow |C(j') - C(j_1')| \geq 1 \tag{6}$$

is free of cycles of length 4.

Proof: Suppose conditions 1 and 2 are satisfied, and that the graph contains a cycle of length 4. There are only two ways in which this cycle forms in the graph of an IRA:

1) The cycle is composed of one information bitnode and one parity bitnode (cf. Fig. 4(a)). Because of the zigzag pattern of the graph of the accumulator, the two checknodes in the length-4 cycle are adjacent to each other, therefore the distance between them is exactly 1. There remain two edges connecting the information bitnode to the two checknodes. Then, condition 1 is violated.
2) The cycle is composed of two information bitnodes sharing two checknodes (cf. Fig. 4(b)). Denoting the four edges of the cycle as j, j_1, j', j_1', and letting $V(j) = V(j')$ and $V(j_1) = V(j_1')$, then because condition 1 is met, $|C(j) - C(j')| \geq 2$ and $|C(j_1) - C(j_1')| \geq 2$. Then because the cycle is of length 4, $C(j) = C(j_1)$ and $C(j') = C(j_1')$. This is a contradiction with condition 2, which requires one of the two distances $|C(j) - C(j_1)|$ and $|C(j') - C(j_1')|$ to be at least equal to 1.

■

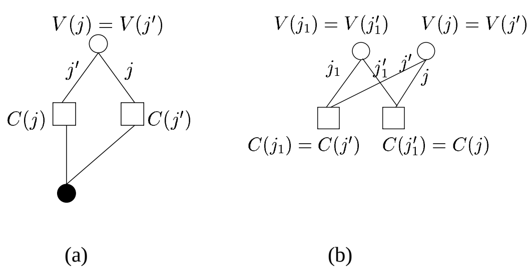

Fig. 4. Length-4 cycles

4 Upper Bound on the Girth of IRA Graphs

An upper bound on the girth of IRA codes can be obtained using results in [15], which were used in [14] to derive an upper bound on the girth of LDPC codes. The average variable node degree λ and the average check node degree μ are given as:

$$\begin{array}{ll} \lambda & = \frac{\bar{d}k(2+a)-a}{k(a+\bar{d})} \\ \mu & = a + 2 - \frac{1}{n} \end{array} \tag{7}$$

which can be approximated, for practical information and parity block lengths k and n, as

$$\begin{array}{ll} \lambda & \approx \frac{\bar{d}(2+a)}{a+\bar{d}} \\ \mu & \approx a + 2 \end{array} \tag{8}$$

Let g denote the girth of the graph. Then, from [15], the following upper bound can be derived:

$$g \leq 4 \frac{\log\left[\frac{k+n}{\mu}((\mu-1)(\lambda-1)-1)+1\right]}{\log\left[(\mu-1)(\lambda-1)\right]} \tag{9}$$

then, using (8) and $n = k\bar{d}/a$, we get:

$$g \leq 4 \frac{\log\left[k(\bar{d}-1)+1\right]}{\log\left[\frac{a+1}{a+\bar{d}}(\bar{d}a + \bar{d} - a)\right]} \tag{10}$$

Using the adjacency matrix of the parity check matrix associated to a Tanner graph [14], we can evaluate the girth of graphs generated by the methods described in the previous section. Let us consider regular RA codes with parameters $a = 4$ and $d = 4$ generated using the PEG method. The following table shows the minimum k required to obtain a graph with girth g, as well as the corresponding theoretical girth upper bound.

k	g_{bound}	g
21	7.2	6
190	11.0	8
1930	15.0	10

TABLE I

GIRTH OF REGULAR RA GRAPHS

5 Maximum Likelihood Decoding

An upper bound on the BER and WER of the random ensemble of IRA codes, under ML decoding on the BIAWGNC, can be determined using the tangential sphere bound (TSB) [16], [17], [18]. The computation of the TSB requires the knowledge of the input-output weight enumerators (IOWE) of the random IRA ensemble $A_{w,h}$, $w = 0, 1, \ldots, k$ and $h = 0, 1, \ldots, m$. $A_{w,h}$ is the number of codewords with weight h, generated by information words of weight w. Using the uniform interleaver technique [19], we can calculate the average $A_{w,h}$, by assuming that the encoder of Fig. 1 has a uniform interleaver at the output of the repetition encoder, and a second uniform interleaver between the grouping and the accumulator (cf. Fig. 5). Then

$$A_{w,h+w} = \sum_{p=1}^{N} \frac{A^R_{w,p}}{\binom{N}{p}} \sum_{l=0}^{n} \frac{A^G_{p,l} A^A_{l,h}}{\binom{n}{l}} \tag{11}$$
$$\text{for } w = 0, 1, \cdots, k \text{ and } h = 0, 1, \cdots, n$$

where $A^R_{w,p}, A^G_{p,l}, A^A_{l,h}$ are the IOWE of, respectively, the repetition code, the grouping and the accumulator.

The following IOWE computation is the extension of the results in [20] to the case $a > 1$ and irregular repetition.

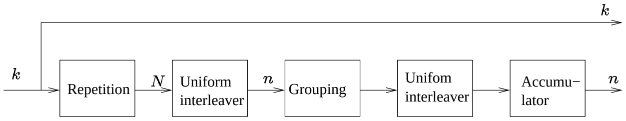

Fig. 5. Modified IRA encoder

A IOWE of Repetition Code: The IOWE of the repetition code is given by

$$A^R_{w,h} = \sum_{\mathbf{w}} A^R_{\mathbf{w},h} \qquad (12)$$

where

$$A^R_{\mathbf{w},h} = \begin{cases} \prod_{j=1}^{n_{deg}} \binom{f_j k}{w_j} & \text{if} \quad h = \sum_{j=1}^{n_{deg}} d_j w_j \\ 0 & \text{otherwise} \end{cases} \qquad (13)$$

and the summation is carried out over all ordered integer partitions $\mathbf{w} = [w_1, \ldots, w_{n_{deg}}]$ of w into at most n_{deg} parts, i.e., $\sum_{j=1}^{n_{deg}} w_j = w$ and $w_j = 0, 1, \ldots, w$. The number of these partitions grows at least as $\Theta(w^{n_{deg}-1})$ [21], making the computation of the irregular repetition IOWE intractable for information block lengths $k > 200$.

If $n_{deg} = 1$, the repetition is regular with repetition degree d, and the resulting IOWE is [20]

$$A^R_{w,h} = \begin{cases} \binom{k}{w} & \text{if} \quad h = dw \\ 0 & \text{otherwise} \end{cases} \qquad (14)$$

B IOWE of Grouping: $A^G_{w,h}$ is the coefficient of x^w in $G(x)$, denoted by $\lfloor G(x) \rfloor_w$,where

$$G(x) = \binom{n}{h} 2^{-n} \left((1+x)^a - (1-x)^a\right)^h$$
$$\left((1+x)^a + (1-x)^a\right)^{n-h} \qquad (15)$$

For $a = 2$, the grouping IOWE is

$$\lfloor G(x) \rfloor_w = \begin{cases} 2^h \binom{n}{h} \binom{n-h}{\frac{w-h}{2}} & \text{if } w-h \text{ is even} \\ 0 & \text{if } w-h \text{ is odd} \end{cases} \qquad (16)$$

and the IOWE of grouping with $a = 4$ is

$$\lfloor G(x) \rfloor_w = \begin{cases} 2^h \quad \binom{n}{h} \sum_{l=0}^{l_{max}} \binom{n-h}{l} 2^{2l} \binom{2n-2l-h}{\frac{w-h}{2} - l} \\ \quad \text{if } w-h \text{ is even} \\ 0 \quad \text{if } w-h \text{ is odd} \end{cases} \qquad (17)$$

where

$$l_{max} = \min\left(2n - \frac{w+h}{2}, n-h, \frac{w-h}{2}\right) \qquad (18)$$

For $a \neq 2$ and $a \neq 4$, the grouping IOWE cannot be expressed in closed-form. However, noticing that the coefficients of $G(x)$ are positive, the following result [22] provides an upper bound on the grouping IOWE:

$$\lfloor G(x) \rfloor_w \leq \inf_{x>0} \frac{G(x)}{x^w} \qquad (19)$$

C IOWE of Accumulator (without grouping): As stated in [23], the IOWE of the accumulator is given as:

$$A^A_{w,h} = \binom{n-h}{\lfloor w/2 \rfloor} \binom{h-1}{\lceil w/2 \rceil - 1} \qquad (20)$$

D Regular RA Code with Grouping Factor $a = 2$ and $a = 4$: Consider a regular RA code with repetition degree d. Using (16) and (17), the IOWE of the RA code with grouping factor $a = 2$ is

$$A_{w,w+h} = \frac{\binom{k}{w}}{\binom{dk}{dw}} \sum_t 2^t \binom{n-t}{\frac{dw-t}{2}} \binom{n-h}{\lfloor t/2 \rfloor} \binom{h-1}{\lceil t/2 \rceil - 1} \qquad (21)$$

and the IOWE with grouping factor $a = 4$ is

$$A_{w,w+h} = \frac{\binom{k}{w}}{\binom{dk}{dw}} \sum_t 2^{2t} \binom{n-h}{\lfloor t/2 \rfloor} \binom{h-1}{\lceil t/2 \rceil - 1}$$
$$\left(\sum_{l=0}^{l_{max}} 2^{2l} \binom{n-t}{l} \binom{2n-2l-t}{\frac{dw-t}{2} - l} \right) \qquad (22)$$

where

$$\begin{aligned} w &= 0, \ldots, k \\ h &= 0, \ldots, n \\ l_{max} &= \min(n-t, \tfrac{dw-t}{2}) \end{aligned}$$

and summations are on all integers t such that $dw - t$ is even and $t \leq \min(n, dw)$.

6 Simulation Results

6.1 Regular RA codes

Figs. 6, 7 and 8 show the average performances of randomly and semi-randomly PEG constructed regular RA codes, with $S = 2$ and $S = 3$ (graphs are free of cycles of length up to $2S$), on the BIAWGNC under BP decoding with 25 decoder iterations. Also shown is the TSB on the BER and WER of the random regular RA ensemble under ML decoding. The code rate is $R = 1/2$, and the information block lengths are respectively $k = 150$, $k = 256$ and $k = 512$. The code parameters $d = 4$ and $a = 4$ were selected because they correspond to the best DE threshold of the random-like regular RA ensemble at rate $R = 1/2$.

The semi-random regular RA ensemble under BP decoding outperforms its random counterpart under both BP decoding and ML decoding, in the error floor region. We notice that, as expected for such short code lengths, the BER waterfall is far from the DE evolution threshold $\left(\frac{E_b}{N_0}\right)^* = 0.2108$ dB.

Comparing the semi-random regular RA codes with the regular LDPC codes proposed in [14] of the same conditioning level (girth 6 and 8) and the same information block lengths, we note the following. For $k = 150$, the girth 6 semi-random regular RA code outperforms the regular LDPC code by 0.3 dB at a BER of 2×10^{-6}. For $k = 512$, the girth 6 and 8 regular RA codes outperform the LDPC code in the waterfall region (at $E_b/N_0 = 3$ dB, BER of semi-random regular RA code is 5×10^{-7} while the BER of the LDPC code is 5×10^{-6}). In the error floor region, BER performances of RA codes of girth 6 and 8 are comparable to those of LDPC codes(at $E_b/N_0 = 3.5$ dB, the BER of the girth 8 regular RA code is 2×10^{-8} while the BER of the girth 8 LDPC code is 10^{-8}). Note that the complexity of the PEG algorithm (as well as that of the SSMAX) is exponential in the girth and linear in the block length, whereas that of the algorithm proposed in [14] is linear in the girth and polynomial in the block length. Therefore, for short block length and small girth, it is preferable to use the PEG algorithm for girth conditioning.

Comparing the semi-random regular RA codes of girth 6 with the regular LDPC codes proposed by MacKay in [24] of information block length around $k = 256$, we note that their BER performances are similar. At $E_b/N_0 = 3.5$ dB, the BER of the RA code is 2×10^{-6}, and that of the MacKay LDPC code is 10^{-6}. The regular RA codes of girth 8 are found to outperform both MacKay regular LDPC codes and PEG-constructed LDPC codes of girth 8 [7] for information block length $k = 256$. In fact, at $E_b/N_0 = 3.25$ dB, the BER of the RA code is 5×10^{-7}, while the BER of of the McKay LDPC code is 4×10^{-6} and that of the PEG LDPC code is 2×10^{-6}.

Using the error impulse method proposed in [25], we compute the minimum distances (d_{min}) of 200 realizations of randomly and PEG constructed regular RA codes. Table II shows the minimum, maximum and average d_{min} thus obtained, and indicates that as the girth of the bipartite graphs increases from 4 to 6, so does the minimum distance of the associated regular RA codes. Fig. 9 shows the performance of rate 1/2 random and PEG codes with respective minimum distances $d_{min} = 9$ and $d_{min} = 11$, for information block length $k = 512$.

6.2 Irregular RA codes

Fig. 10 compares the average BER performance of randomly and semi-randomly (PEG and SSMAX) constructed IRA codes to that of an LDPC code of the same rate, on the BIAWGNC under BP decoding with 25 decoder iterations. The code rate is $R = 0.5$, and the information and parity block lengths are respectively $k = 5020$ and $n = 4940$. The code degree sequences and grouping factor are obtained by optimizing the code with method 1 in [5], and have a maximum repetition degree of 20.

Although the maximum achievable girth of the considered IRA graph is 12.4, the maximum girth obtained using the PEG algorithm is 6. The curves labeled (d_{cyc}, d_{ACE}, η) in fig. 10 represent the average performances of codes constructed with the SSMAX method. (d_{cyc}, d_{ACE}, η) means that the IRA code is free of cycles of length up to $2d_{cyc}$, and all cycles of length

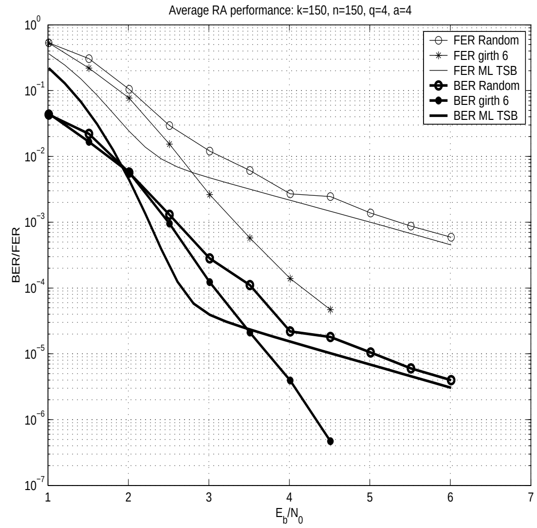

Fig. 6. Average RA performance with $k = 150$, $n = 150$, $d = 4$, $a = 4$

up to d_{ACE} have $ACE \geq \eta$.

The LDPC code has rate 1/2, a code block length of 10000, and a degree sequence optimized in [11]. It is generated randomly, but the degree-2 nodes are arranged in a single cycle.

Fig. 10 shows that the SSMAX method yields a better error floor than the PEG algorithm with $S = 2$. Comparing the $(1, 9, 3)$ code with the randomly constructed LDPC code ensemble, we note that it outperforms the random LDPC in the error floor region. But, its performance is inferior to that of the $(1, 9, 4)$ LDPC code of [9].

7 Conclusion

We have presented a comparative study of the performance of finite length regular and irregular repeat accumulate codes, constructed according to two criteria: girth maximization and stopping set maximization. Our simulations show that girth conditioning yields an improvement in the error floor region of short-length regular RA codes in the BER and WER, as compared to the random regular RA ensemble. The codes thus designed perform as well as the best known LDPC codes [7], [13], [14], [9]. Hence, the performance-complexity tradeoff of the constructed regular RA codes is very advantageous.

Large block length irregular RA codes exhibit a better error floor using the stopping set maximization method, as compared to the random and girth-conditioned IRA ensembles. But the average IRA code performance remains inferior to that of LPDC codes with comparable graph conditioning and block length.

Code			Random			PEG $S=2$		
k	d	a	min. d_{min}	max. d_{min}	av. d_{min}	min. d_{min}	max. d_{min}	av. d_{min}
150	4	4	2	6	4.11	5	8	6.54
256	4	4	2	8	5.27	5	9	7.07
512	4	4	2	9	6.09	5	11	8.59

TABLE II

MINIMUM DISTANCE OF REGULAR RA CODES

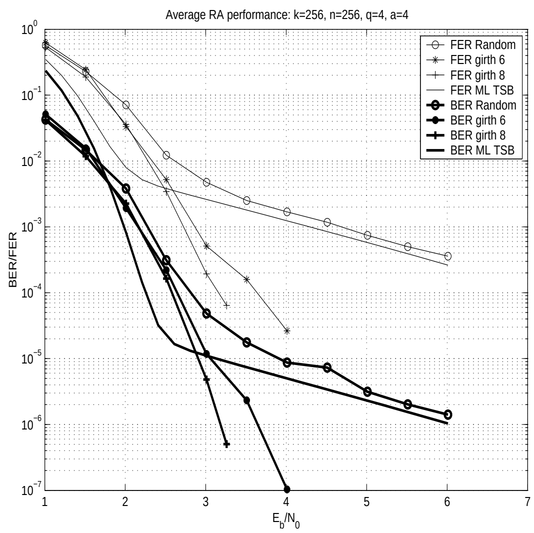

Fig. 7. Average RA performance with $k=256$, $n=256$, $d=4$, $a=4$

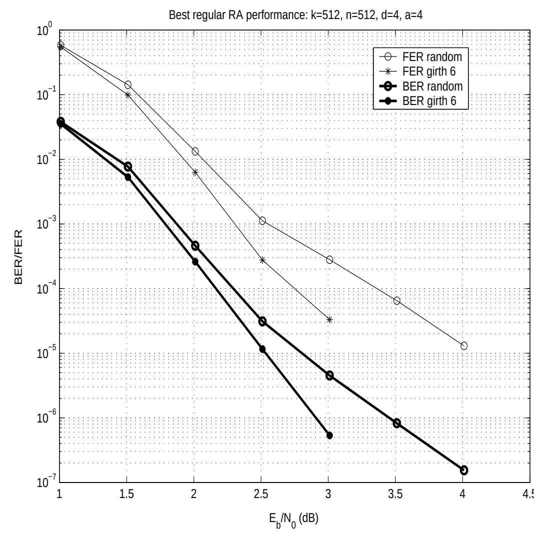

Fig. 9. Best RA performance with $k=512$, $n=512$, $d=4$, $a=4$

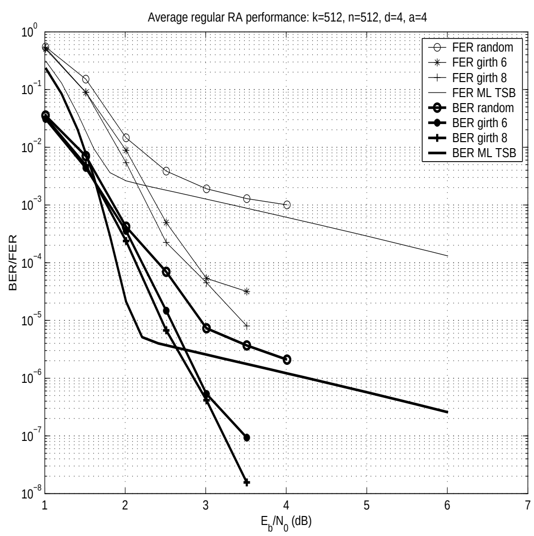

Fig. 8. Average RA performance with $k=512$, $n=512$,$d=4$, $a=4$

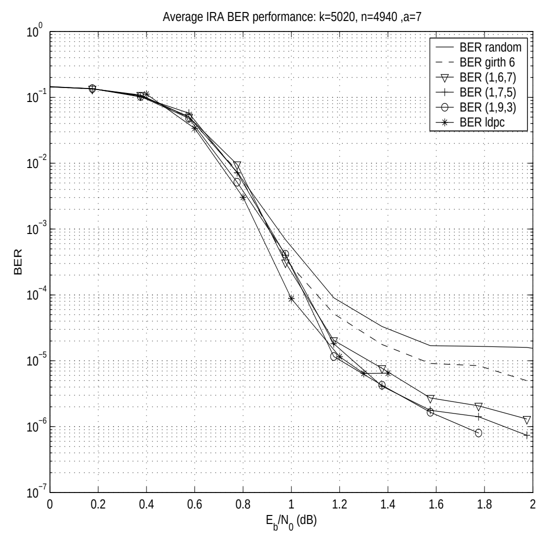

Fig. 10. Average IRA performance with $k=5020$, $n=4940$, $a=7$

References

[1] C. Berrou, A. Glavieux, and P. Thitimajshima. Near Shannon limit error-correcting and decoding: turbo codes. In *Proceedings of IEEE International Conference on Communication ICC*, pages 1064–1070, Geneva, May 1993.

[2] T.J. Richardson, M.A. Shokrollahi, and R.L. Urbanke. Design of capacity-approaching irregular low-density parity-check codes. *IEEE Transactions on Information Theory*, 47:619 – 637, Feb 2001.

[3] J. Garcia-Frias and W. Zhong. Approaching Shannon performance by iterative decoding of linear codes with low-density generator matrix. *IEEE Communications Letters*, 7(6):619–637, Feb. 2001.

[4] J. Garcia-Frias, W. Zhong, and Y. Zhao. Iterative decoding of source and joint source-channel coding of correlated sources. In *Proceedings of Asilomar Conference on Signals, Systems and Computers*, volume 1, pages 250–256, Nov. 2002.

[5] A. Roumy, S. Guemghar, G. Caire, and S. Verdu. Design methods for irregular repeat accumulate codes. In *Proceedings of IEEE International Symposium on Information Theory ISIT*, Jun. 29^{th} - Jul. 4^{th} 2003.

[6] R. M. Tanner. Minimum distance bounds by graph analysis. *IEEE Transactions on Information Theory*, 47:808–821, Feb. 2001.

[7] X. Hu, E. Eleftheriou, and D. Arnold. Progressive edge-growth Tanner graphs. In *Proceedings of IEEE Global Telecommunication Conference, Globecom*, volume 2, pages 995–1001, 25-29 Nov. 2001.

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

[9] T. Tian, C. Jones, J. D. Villasenor, and R. D. Wesel. Construction of irregular ldpc codes with low error floors. In *Proceedings of IEEE International Conference on Communication ICC*, volume 5, pages 3125–3129, 2003.

[10] R. M. Tanner. A recursive approach to low complexity codes. *IEEE Transactions on Information Theory*, IT-27:533–547, 1981.

[11] R. Urbanke et al. Web page. http://lthcwww.epfl.ch/research/ldpcopt/, 2002.

[12] D.J.C. MacKay, S.T. Wilson, and M.C. Davey. Comparison of constructions of irregular Gallager codes. In *Proceedings of Allerton Conference Comm. Control and Comput.*, September 1998.

[13] Y. Mao and A. H. Banihashemi. A heuristic for good low-density parity-check codes at short block lengths. In *Proceedings of IEEE International Conference on Communication ICC*, volume 1, pages 41–44, 11-14 Jun. 2001.

[14] J. A. McGowan and R. C. Williamson. Loop removal from ldpc codes. In *Proceedings of Information Theory Workshop ITW*, pages 230–233, Paris, France, March 31 - April 4 2003.

[15] S. Hoory. The size of bipartite graphs with a given girth. *Journal of Combinatorial Theory - Series B*, 86(2):215–220, 2002.

[16] H. Herzberg and G. Poltyrev. The error probability of m-ary psk block coded modulation schemes. *IEEE Transactions on Communications*, 44(4):427–433, April 1996.

[17] G. Poltyrev. Bounds on the decoding error probability of binary linear codes via their spectra. *IEEE Transactions on Information Theory*, 40(4):1284–1292, July 1994.

[18] I. Sason and S. Shamai. Improved upper bounds on the ml decoding error probability of parallel and serial concatenated turbo codes via their ensemble distance spectrum. *IEEE Transactions on Information Theory*, 46(1):24–47, January 2000.

[19] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara. Serial concatenation of interleaved codes: performance analysis, design, and iterative decoding. *IEEE Transactions on Information Theory*, 44(3):909–926, May 1998.

[20] H. Jin, A. Khandekar, and R. McEliece. Irregular repeat-accumulate codes. In *Proceedings of International Symposium on Turbo Codes*, pages 1–8, Brest-France, September 2000.

[21] D. L. Kreher and D. R. Stinson. *Combinatorial Algorithms-Generation, Enumeration and Search*. CRC Press, 1999.

[22] D. Burshtein and G. Miller. Asymptotic enumeration methods for analyzing ldpc codes. Submitted to IEEE Trans. on Information Theory, 2002.

[23] D. Divsalar, H. Jin, and R. J. McEliece. Coding theorems for "turbo-like" codes. In *Proceedings of Allerton Conference Comm. Control and Comput.*, pages 201–210, Urbana-Champaign, 1998.

[24] D.J.C. MacKay. Online database of low-density parity check codes. http://wol.ra.phy.cam.uk/mackay/codes/data.html.

[25] C. Berrou, S. Vaton, M. Jezequel, and C. Douillard. Computing the minimum distance of linear codes by the error impulse method. In *Proceedings of IEEE International Symposium on Information Theory ISIT*, Lausanne, Switzerland, July 2002.