Lowering the Error Floors of Irregular High-Rate LDPC Codes by Graph Conditioning

Wen-Yen Weng, Aditya Ramamoorthy and Richard D. Wesel
Electrical Engineering Department, UCLA,
Los Angeles, CA, 90095-1594.

Abstract— **This paper applies the graph conditioning algorithm called the approximate cycle extrinsic message degree (ACE) algorithm to design high-rate ($R \geq 1/2$) irregular LDPC codes. The algorithm was shown to be an effective tool to lower the error floors of lower-rate ($R \leq 1/2$) LDPC codes. However, for high-rate LDPC codes, due to the large number of degree-2 variable nodes in the optimal degree distribution, the error floor is high and it is more difficult to condition the graph. By constraining the number of degree-2 nodes, we found that the ACE algorithm can dramatically lower the error floor with little compromise of the threshold. A rate-3/4, length-10688 LDPC code is proposed whose AWGN channel performance is within 0.67 dB of the Shannon limit at $BER = 10^{-5}$ and its error floor is lower than 10^{-7}. Compared to existing semi-regular codes which lower the floor by adopting non-optimal degree distributions, our graph-conditioned codes provides 0.38 dB of performance improvement at $BER = 10^{-5}$. The same design criteria also apply well to the medium-length LDPC code design and are suitable for rate-compatible applications using the information-nulling technique. The rate-compatible scheme has consistently good thresholds and low error floors for $1/2 \leq R \leq 8/9$.**

I. Introduction

Low density parity check (LDPC) codes have generated much research interest because of their capacity approaching performance. Most of the work in the literature focuses on low-rate and very long codes. The goal of this paper is to design high-rate LDPC codes with low encoding complexity and low error floors. Yang *et al.* [1] proposed the specialized class of LDPC codes called the extended irregular repeat-accumulate (IRA) codes that have low complexity encoders. Richardson *et al.* [2] created the density evolution technique to optimize the degree distributions in cycle-free bipartite graphs as the block length and the number of iterations go to infinity. For finite block length, Tian *et al.* [3] proposed an efficient graph conditioning algorithm called the approximate cycle extrinsic message degree (ACE) algorithm to lower the error floor by avoiding harmful short cycles. The ACE algorithm is effective for the construction of low-rate LDPC codes. In this paper, we explore the effectiveness of ACE algorithm for high-rate ($R \geq 1/2$) code designs.

Following the common notation in the literature, k and n denote the input block length and codeword length respectively. Thus the code rate $R = k/n$. The parity check matrix, H, is an $(n-k) \times n$ sparse matrix which can be divided into two submatrices, H_1 and H_2, with k and $n-k$ columns respectively as defined in [1].

$$H = [H_1 \mid H_2] \tag{1}$$

The H_2 matrix must have $n-k-1$ weight-2 columns plus one weight-1 column with a bi-diagonal structure [1]

$$H_2 = \begin{bmatrix} 1 & & & & & \\ 1 & 1 & & & & \\ & 1 & 1 & & & \\ & & 1 & \ddots & & \\ & & & \ddots & 1 & \\ & & & & 1 & 1 \end{bmatrix} \tag{2}$$

Then H_2^{-T} is a matrix with all the upper-diagonal elements equal 1. Its generating function is $\frac{1}{1+D}$ and can be easily implemented using an accumulator. Therefore, the generator matrix, G, of the extended IRA codes can be written as

$$G = [I \mid H_1 H_2^{-T}] \tag{3}$$

which only requires a sparse matrix multiplication followed by an accumulator and yields low encoding complexity. $\lambda(x) = \sum_i \lambda_i x^{i-1}$ represents the variable node degree distribution from the edge's perspective and $\rho(x) = \sum_i \rho_i x^{i-1}$ represents the check node distribution. $N_v(l)$ denotes the number of degree-l variable nodes in the Tanner graph. In general, the weight-1 column should be avoided during the degree distribution optimization. So, throughout the paper, there is no degree-1 node in the degree distributions. One of the degree-2 nodes is converted to degree-1 when the parity check matrices of the extended IRA codes are constructed.

II. Code Design Criteria

A. Existing Criteria

Major criteria in the literature [1], [2] are summarized as follows:

(1) Optimize the degree distributions using density evolution.
(2) Forbid cycles involving only degree-2 variable nodes and avoid length-4 cycles.
(3) Low-degree variable nodes are made to correspond to non-systematic bits.

B. ACE Algorithm

Definition 1: A variable node set is called a stopping set if all its neighbors are connected to it at least twice.

The size of the smallest stopping set determines the error floor behavior. However, it is hard to find the smallest stopping set because the complexity is too high.

0-7803-8521-7/04/$20.00 © 2004 IEEE

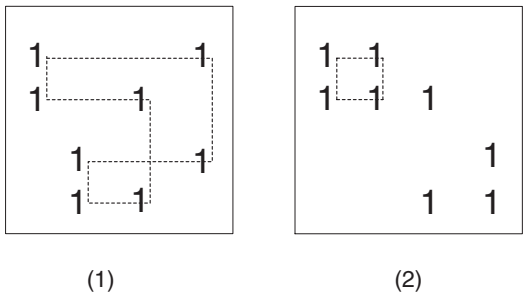

Fig. 1. An example of the number of degree-2 nodes equals to $n-k$, where n-k=4. There must exist cycles between these nodes and every cycle consisting of all degree-2 nodes is a stopping set. Example (1) has a length-8 cycle and example (2) has a length-4 cycle.

Definition 2: The approximate cycle extrinsic message degree (ACE) of a length $2d$ cycle is $\sum_i (d_i - 2)$, where d_i is the degree of the i^{th} variable node in the cycle.

An LDPC code has property (d_{ACE}, η) if all the cycles whose length is $2d_{ACE}$ or less have ACE values of at least η. The ACE algorithm is an efficient Viterbi-like linear complexity algorithm proposed in [3] to detect and avoid harmful short cycles during code construction. Given the degree distribution, $\lambda(x)$, columns of the parity check matrix are generated one at a time starting from low-degree nodes. The edges of every new column are generated randomly and the ACE algorithm checks whether the (d_{ACE}, η) requirement is met. If not, this column must be generated again. This step repeats until the whole parity check matrix is generated.

It is more difficult to apply the ACE algorithm to high-rate codes than to low-rate codes because with the same block length, high-rate codes have fewer check nodes, i.e. columns are shorter in H, and the number of cycles will increase which makes it harder to guarantee certain (d_{ACE}, η) values. For high-rate LDPC codes, the optimal degree distributions usually have more than $n-k$ degree-2 variable nodes and there always exist cycles between only degree-2 nodes [4]. Figure 1 gives two example of four length-4 degree-2 columns. The first matrix has a length-8 cycle which is the longest cycle possible while the second matrix has a length-4 cycle. Note that if $N_v(2) \geq n-k$, any combination of $n-k$ degree-2 columns forms cycle(s) and these cycles are all stopping sets. Figure 2 shows the curves of the optimal ratio of degree-2 nodes and the ratio of parity bits. The two curves cross each other at approximately $R = 1/2$. Therefore, if we don't constrain the number of degree-2 nodes and choose the optimal degree distributions, the ACE algorithm can only help to lower the error floor a little, but it is still high since too many loops with small ACE values.

To lower the error floor further, we must decrease the number of degree-2 variable nodes and adopt a degree distribution optimized with the constraint $N_v(2) < n-k$. The error floor is lowered at the cost of a small increase in the threshold SNR. This tradeoff between threshold and error floor is also observed in many LDPC and turbo-code

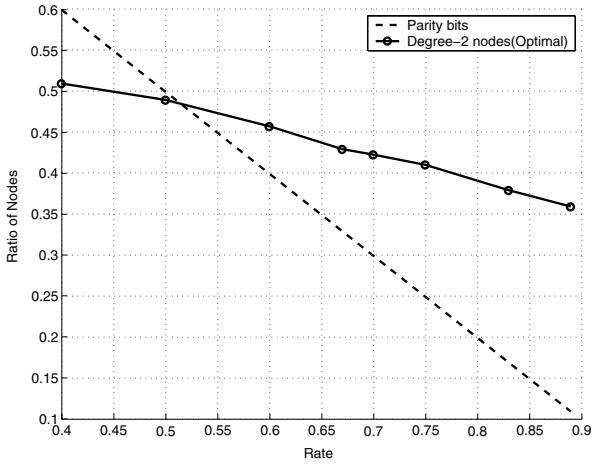

Fig. 2. The ratio of degree-2 nodes, $\lambda_v(2)$, in the optimal degree distribution as a function of the code rate. Also plotted is the ratio of parity check bits which equals $1-R$. Note that for approximately $R \geq 1/2$, $\lambda_v(2)$ is greater than $1-R$ which results in loops between only degree-2 nodes.

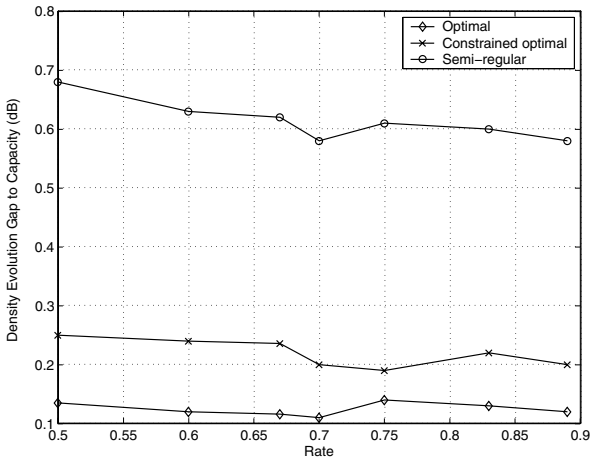

Fig. 3. The gap to capacity for the optimal, constrained optimal, and semi-regular degree distributions found using density evolution technique as a function of the code rate.

papers [5], [6], [3].

An exciting discovery is that for this constrained degree distribution, the graph conditioning can be carried out successfully and lower the error floor dramatically. The semi-regular LDPC codes in [1] also constrain the number of degree-2 nodes but use a regular H_1 to guarantee low error floors. Figure 3 shows the theoretical gap to BPSK capacity for the three types of codes. The semi-regular design trades about 0.5 dB of threshold SNR for low error floors while our design only requires 0.1 dB increase of threshold SNR to have a low error floor because of the effective graph-conditioning algorithm.

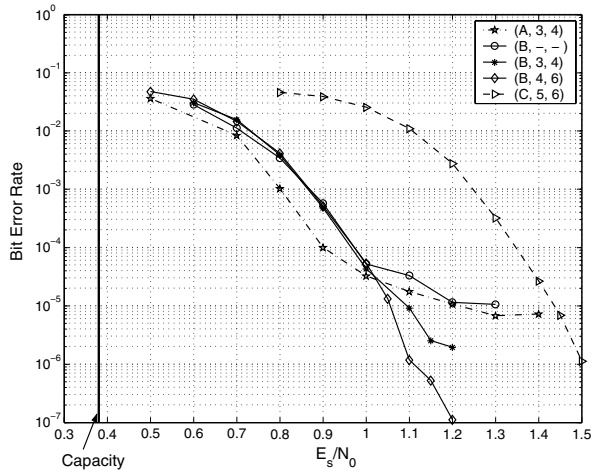

Fig. 4. Simulation results for (8016,10688) codes for 200 iterations. The BPSK capacity at R=3/4 is E_s/N_o=0.38 dB. Codes are labeled by (Scheme, d_{ACE}, η).

TABLE I

THE THREE DEGREE DISTRIBUTIONS USED IN THE LONG LDPC CODE DESIGN EXAMPLE. SCHEME-A IS THE OPTIMAL DEGREE DISTRIBUTION, SCHEME-B IS OPTIMAL WITH THE CONSTRAINT $N_v(2) \leq n-k$, AND SCHEME-C IS THE SEMI-REGULAR CODE WITH $n-k$ DEGREE-2 VARIABLE NODES AND k DEGREE-5 VARIABLE NODES.

	Scheme-A (optimal)	Scheme-B (constrained optimal)	Scheme-C (semi-regular)
λ_2	0.1970	0.1250	0.1176
λ_3	0.0801	0.4460	
λ_4	0.2410		
λ_5	0.0082		0.8824
λ_{11}		0.4078	
λ_{12}		0.0213	
λ_{14}	0.4736		
ρ_{16}		1.0000	
ρ_{17}			1.0000
ρ_{18}	0.700		
ρ_{19}	0.300		

III. Code Design Examples and Simulation Results

A. Long Block Length High-Rate LDPC Codes

Figure 4 includes design examples of rate-3/4 LDPC codes with block length n=10688 simulated on AWGN channel using BPSK modulation for a maximuum of 200 iterations. Three different degree distributions are simulated here (See Table I). Scheme-A is the optimal degree distribution without any constraint. Scheme-B is the optimal degree distribution with the constraint $N_v(2) < n-k$ while scheme-C is the semi-regular code with a regular H_1 of column weight 5.

Our results show that for scheme-A, the error floor is about 10^{-3} without graph conditioning. The ACE algorithm can only achieve $(d_{ACE}, \eta) = (3, 4)$ to improve the error floor to a level between 10^{-4} and 10^{-5}, which is still high. Scheme-B without graph conditioning, (B, -, -), has an error floor around $BER = 10^{-5}$. Scheme-C uses the same design proposed in [1] which adopts a non-optimal degree distribution and trades some threshold for a lower error floor. Note that we were able to construct a (C, 5, 6) code but its performance is almost identical to the (C, -, -) code at the bit error rate above 10^{-7}. This is because graph conditioning plays the role of lowering the error floor but the error floor of scheme-C is lower than 10^{-7} even without any graph conditioning. The results of (B, 3, 4) and (B, 4, 6) are exciting because with proper graph conditioning, the error floor can be lowered from 10^{-5} to at least 10^{-7} with little compromise (less than 0.1 dB) of the threshold of convergence. As a result, at $BER = 10^{-5}$, our best code, (B, 4, 6), performs within 0.67 dB of the Shannon limit and is 0.38 dB better than the semi-regular code.

B. Medium Block Length High-Rate LDPC Codes

For applications requiring high throughput such as the wireless local area network (WLAN), high-rate LDPC codes are considered a good candidate channel coding scheme. The decoder complexity and delay constraint limit the code length to be less than a few thousand bits and the number of decoding iterations to be about 10-20. Unlike the previous example which adopts a long, length-10688, LDPC code, here we will limit the codeword length to 2000 bits and design LDPC codes using the afore mentioned criteria.

We designed rate-1/2, 2/3, 3/4, 4/5, 5/6 and 8/9 LDPC codes separately, all with $n = 2000$. Figure 5 are the simulation results for these codes. Note that the achievable (d_{ACE}, η) region is still an open problem for a given block length and the degree distribution. Because of the linear complexity of the ACE algorithm, it is not too computationally intensive to run the algorithm starting from small (d_{ACE}, η) values and increase the constraint until such codes cannot be constructed. Then several candidate codes with largest η for each d_{ACE} value are simulated to determine the best code.

Note for the same rate (rate-3/4) and the same $d_{ACE} = 4$, the largest achievable ACE value, η, in the previous example is 6 while it decreases to 3 as the code length is reduced from 10688 to 2000. Although the achievable (d_{ACE}, η) is smaller especially for the highest-rate code (rate-8/9), the graph conditioning algorithm can still effectively lower the error floors. All the codes in Figure 5 have error floors lower than 10^{-6}. Figure 6 plots the gap to capacity for density evolution threshold, individually designed codes and the rate-compatible code which will be discusd later. SNRs are measured at the $BER = 10^{-5}$. Also observe that the gap to capacity is larger at the low-rate than at the high-rate. This is because for the same block length, low-rate codes have more check nodes and each check node has a lower degree such that it requires more iterations to converge. When the number of iterations is large enough, all the codes will converge to their limits and the gap-to-capacity curve will become flatter.

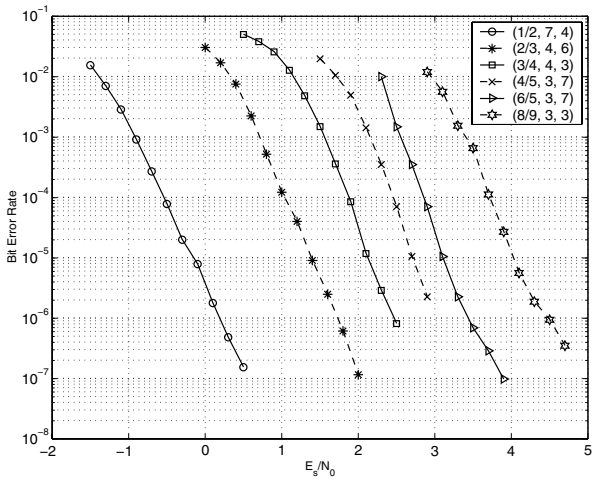

Fig. 5. For medium block length LDPC codes, the ACE algorithm still applies well and the proposed codes all have error floors lower than $BER = 10^{-6}$. All codes are length-2000, simulated for 10 iterations on AWGN channel. Codes are labeled as (R, d_{ACE}, η)

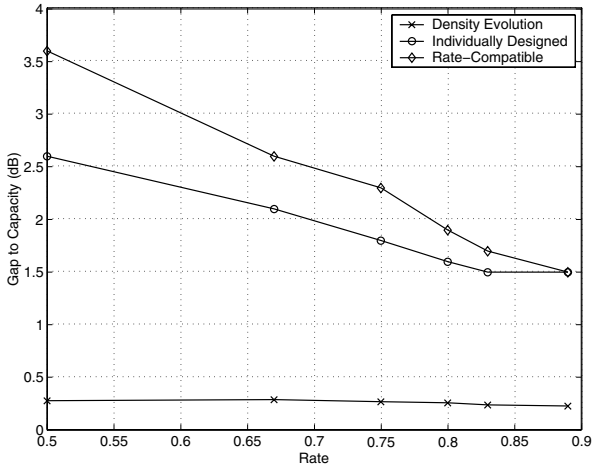

Fig. 6. When the allowed variable node degrees are only 2, 3 and 10, the density evolution threshold of the constrained optimal code is consistently good. The other two curves are the gap to capacity at $BER = 10^{-5}$ for the individually designed codes and the rate-compatible code. The individually designed codes and the mother code of the rate-compatible code have block length 2000. All codes simulated for only 10 iterations on AWGN channel.

C. Rate-Compatible LDPC Codes Using Information-Nulling

In [7], Tian proposed an algorithm that combines the graph-conditioning and information nulling to design rate-compatible LDPC codes. Information-nulling uses the highest-rate code as its mother code and lower its code rate by inserting zeros into the information block. Tian showed that for the low-rate range, $0.1 \leq R \leq 0.5$, a rate-1/2 mother can be constructed such that all of the shortened codes are still ensured certain ACE constraint. Here we will apply the same technique to design rate-compatible LDPC codes for high-rate application.

A key requirement of the algorithm to work is the consistency of degree distributions over the rate range. The codeword length at rate-R' becomes

$$n' = n_m(\frac{1-R_m}{1-R'}) \tag{4}$$

given a rate-R_m, lenth-n_m mother code. Let the constrained optimal variable node degree distribution from the node's perspective at rate-R for degree-l be $\lambda_v(R,l)$, the number of variable nodes of degree-l at rate-R' with code length-n' is

$$N_v(R',l) = n'\lambda_v(R',l) = n_m(\frac{1-R_m}{1-R'})\lambda_v(R',l) \tag{5}$$

Figure 7 plots the the constrained-optimal number of variable nodes of degree-2, 3, and 10 at each rate with $n_m = 2000$. Similar to [7], only degree-2, 3, and 10 variable nodes are allowed for design simplicity. Since $N_v(R,l)$ is a non-decreasing function of the rate for each degree, we can use the following algorithm to construct a rate-compatible LDPC code and the lower the code rate using information nulling.

(1) Specify the corresponding code length at each rate of operation.
(2) Starting from the lowest-rate code, the process of generating the columns from the current code to the next-rate code is called a stage. Calculate the number of columns of each degree to be generated at each stage.
(3) Fix the d_{ACE} value and apply appropriate η values at each stage. Note that low-rate codes in general has higher achievable η which means the ACE constraint is also consistent.

The designed rate-compatible codes have not only constrained-optimal degree distributions but also good ACE constraint. Figure 8 shows the performance of the rate-compatible code. Note that none of the codes suffers from high error floor which means the graph-conditioning works effectively. Hence the shortened code is as good as a specifically designed constrained-optimal LDPC code with the same length.

Compared to the same rate-compatible scheme but using a semi-regular codes with degree-2 and degree-4 nodes as the mother code, the semi-regular codes with no graph-conditioning have error floors even higher than the constrained-optimal graph-conditioned codes and are 0.2 to 0.4 dB worse at $BER = 10^{-5}$. We can definitely apply the ACE algorithm to the semi-regular codes to lower their error floors but the threshold SNR will remain the same, which means the semi-regular codes are still inferior to the constrained-optimal codes. Reading from Figure 6, a 1.0 dB loss is observed at rate-1/2 when compared to the individually designed LDPC codes with a fixed code length due to this shortened block length.

IV. Conclusion

In this paper, we applied the graph conditioning algorithm (ACE) to high-rate irregular extended IRA LDPC codes and showed that it can effectively lower the error floor even though the graph-conditioning becomes more difficult

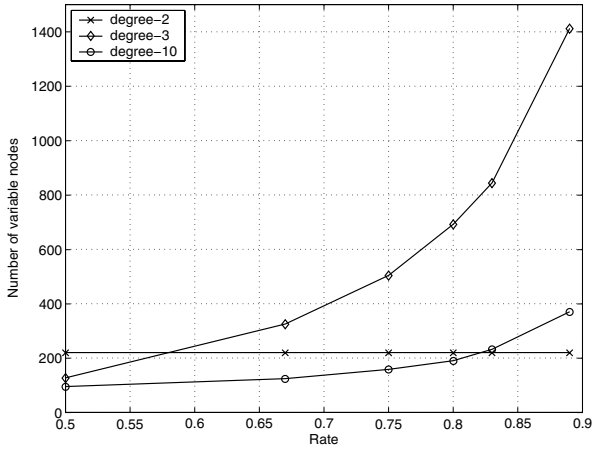

Fig. 7. The constrained optimal number of variable nodes of degree-2, 3, and 10 considering the code-shortening effect of information nulling. The non-decreasing curves of each degree enables the construction of a rate-compatible code with optimal thresholds at each rate.

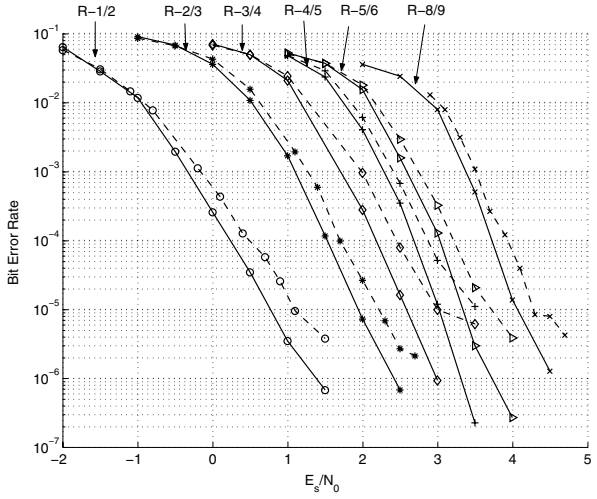

Fig. 8. With a single length-2000, rate-8/9 mother code, the rate-compatibility are achieved by information-nulling. Because graph conditioning is guaranteed, high error floors are not observed for BER higher than 10^{-6} (solid lines). All codes simulated for 10 iterations on AWGN channel. Compared to the semi-regular codes without graph-conditioning (dotted lines), the constrained optimal codes not only have smaller thresholds but their error floors are also lower.

as the rate increases. For code rate greater than 1/2, the optimal degree distribution has more than $n - k$ degree-2 nodes and it results in high error floors. The proposed LDPC codes with constrained optimal degree distributions can trade only 0.1 dB of threshold SNR for error floors which are several orders lower. This indicates that degree distributions near the optimal degree distribution are still quite good in terms of threshold SNR and may be better than the optimal degree distribution for practical block lengths. Code design examples from short block length (a few hundred bits) to long block length (around 10000) all perform well with both low SNR thresholds and low error floors. In addition, the proposed LDPC code is suitable for the construction of rate-compatible codes that maintain constrained optimal degree distribution and reasonably good ACE constraint throughout the high-rate range.

References

[1] M. Yang and W. E. Ryan. Lowering the Error-Rate Floors of Moderate-Length High-Rate Irregular LDPC Codes. In *IEEE International Symposium on Information Theory, Yokohama, Japan*, June 2003.

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

[3] T. Tian, C. Jones, J. Villasenor, and R. D. Wesel. Constructions of irregular ldpc codes with low error floors. In *ICC 2003, Alaska*, May 2003.

[4] M. Yang, W. E. Ryan, and Y. Li. Design of efficiently encodable moderate-length high-rate irregular LDPC codes. *IEEE Trans. on Comm.*, 52(4):564–571, Apr. 2004.

[5] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara. Serial Concatenation of Interleaved Codes: Performace Analysis, Design and Iterative Decoding. *IEEE Trans. on Inform. Theory*, 44(3):909–926, May 1998.

[6] D. J. C. MacKay, S. T. Wilson, and M. C. Davey. Comparison of Constructions of Irregular Gallager Codes. *IEEE Trans. on Comm.*, 47(10):1449–1454, Oct. 1999.

[7] T. Tian, C. Jones, and J. Villasenor. Rate-compatible low-density parity-check codes. In *ISIT 2004, Chicago*, July 2004.