

# Complexity-Optimized Low-Density Parity-Check Codes for Gallager Decoding Algorithm B

Wei Yu, Masoud Ardakani, Benjamin Smith, Frank Kschischang  
Electrical and Computer Engineering Dept.  
University of Toronto  
Toronto, Ontario M4S 3G5, Canada  
Email: {weiyu, masoud, ben, frank}@comm.utoronto.ca

**Abstract**—The complexity-rate tradeoff for error-correcting codes below the Shannon limit is a central question in coding theory. This paper makes progress in this area by presenting a joint numerical optimization of rate and decoding complexity for low-density parity-check codes. The focus of this paper is on the binary symmetric channel and on a class of decoding algorithms for which an exact extrinsic information transfer (EXIT) chart analysis is possible. This class of decoding algorithms includes the Gallager decoding Algorithm B. The main feature of the optimization method is a complexity measure based on the EXIT chart that accurately estimates the number of iterations required for the decoding algorithm to reach a target error rate. Under a fixed check-degree distribution, it is shown that the proposed complexity measure is a convex function of the variable-degree distribution in a region of interest. This allows us to numerically characterize the complexity-rate tradeoff. We show that for the Gallager B decoding algorithm on binary symmetric channels, the optimization procedure can produce complexity savings of 30–40% as compared to the conventional code design method.

## I. INTRODUCTION

The design of irregular LDPC codes, i.e., the task of finding good degree distributions, for different decoding algorithms and channel types has been studied extensively in the literature (e.g. [1], [2], [3]). The common design objective in all these approaches is to find a degree distribution that maximizes the decoding threshold for a given rate or, equivalently, to find the highest-rate decodable code for a given threshold (channel condition). However, highly optimized codes obtained from these design procedures cannot actually be used at their threshold as their decoding would require an impractical number of iterations [3]. In fact, when a uni-parametric analysis of the decoder — an extrinsic information transfer (EXIT) chart analysis for example — is an exact analysis, it can be shown that the required number of iterations for convergence approaches infinity as the rate of a decodable code increases [4]. The traditional solution to this problem is to optimize the code design for a slightly worse channel condition. This way, the convergence would be ensured in a finite number of steps when the code is used in the actual channel. Notice that the actual channel is better than the channel used for the design. This is not necessarily the best solution, as codes highly optimized for one channel condition end up being used in a different channel.

In this paper, we adopt the approach of optimizing the code parameters directly under a given channel condition and under

a fixed rate- or complexity-constraint. The central observation of this paper is that the maximization of rate subject to a given complexity constraint, or equivalently the minimization of complexity subject to a given rate constraint, is in fact numerically feasible. Such an optimization is made possible by a careful complexity analysis based on EXIT charts. We show that whenever the EXIT chart analysis is exact (e.g., for Gallager B decoding algorithm [5] [6],) an accurate measure of decoding complexity can be obtained from the EXIT chart shape directly. Further, this complexity measure is a convex function of the variable-degree distributions (for a fixed check-degree distribution and under a particular condition to be discussed later.) This observation allows us to use convex optimization techniques to find the optimal variable-degree distribution subject to a complexity constraint and to characterize the precise rate-complexity tradeoff for the Gallager decoding Algorithm B on a binary symmetric channel. In particular, we show that as compared to the conventional code design method, about 30–40% reduction in complexity can be obtained with such an optimization.

The optimization scheme presented in this paper is one step toward addressing a central issue in coding theory: the rate-complexity tradeoff below the Shannon limit. In this direction, Khandekar and McEliece [7] and Sason and Urbanke [8] addressed this issue for the binary erasure channel and for irregular repeat-accumulate codes. The current work appears to be among the first to address the same issue for the binary symmetric channel. Our approach is numerical and the analysis is carried out for a particular fixed class of decoding algorithms.

The rest of this paper is organized as follows. In Section II, we review Gallager’s decoding algorithm B and present an analysis of its decoding complexity per information bit per iteration. A description of EXIT chart analysis, in particular for Gallager B decoding algorithm, is presented, and the measure of complexity per information bit is defined. In Section III, we derive an approximate formula for the number of required iterations for convergence to a target error rate in a given EXIT chart and prove its convexity properties. In Section IV, the optimization problem is formulated, and the numerically optimized degree distributions and the complexity-performance tradeoff curves are presented. We conclude the paper in Section V.

## II. BACKGROUND

An ensemble of irregular LDPC codes is defined by its variable-degree distribution  $\{\lambda_2, \lambda_3, \dots\}$  and its check-degree distribution  $\{\rho_2, \rho_3, \dots\}$ , where  $\lambda_i$  denotes the fraction of edges incident on variable nodes of degree  $i$  and  $\rho_j$  denotes the fraction of edges incident on check nodes of degree  $j$ . Given the degree distribution of an LDPC code, it is easy to see that the *design* rate of the code is

$$R = 1 - \frac{\sum_i \frac{\rho_i}{i}}{\sum_i \frac{\lambda_i}{i}}. \quad (1)$$

The actual rate might be slightly higher, if some of the parity check constraints are linearly dependent.

There are many decoding algorithms available for LDPC codes. In this work we focus on Gallager's decoding algorithm B, which will be referred to as Algorithm B in the remainder of this paper. Algorithm B is the focus of this paper because it has the property that its EXIT chart analysis is exact. As shown by Gallager [5], the convergence behavior of Algorithm B can be visualized by plotting the error rate of the extrinsic messages in one iteration as a function of the error rate in the previous iteration. This is almost equivalent to the EXIT chart analysis of [9] and exactly equivalent to the EXIT chart analysis of [4]. The main difference between these two types of analysis is that in [9] and many other works, EXIT charts track the evolution of mutual information, whereas the generalized EXIT charts of [4] track the error rate.

EXIT chart analysis based on error rate has a key advantage in that the EXIT chart of an irregular code can be decomposed as a linear combination of EXIT charts of regular codes (i.e., elementary EXIT charts [4]), whereas for mutual information EXIT charts, this decomposition is not always possible. This observation is central to the formulation of the optimization problem as will be seen in the next section.

### A. Gallager decoding algorithm B

In Algorithm B, introduced by Gallager [5], the message alphabet is  $\{0, 1\}$ . No soft information is used. The update rule at a check node  $c$  is

$$m_{c \rightarrow v} = \bigoplus_{y \in n(c) - \{v\}} m_{y \rightarrow c}, \quad (2)$$

where  $\bigoplus$  represents modulo-two sum of binary messages,  $m_{y \rightarrow c}$  represents a message sent from variable node  $y$  to the check node  $c$  and  $n(c)$  represents the set of the neighbors of check node  $c$ .

At a variable node  $v$  the outgoing message  $m_{v \rightarrow c}$  is

$$m_{v \rightarrow c} = \begin{cases} \overline{m}_0 & \text{if } \exists y_1, y_2, \dots, y_b \in n(v) - \{c\} \\ & \text{such that } m_{y_1 \rightarrow v} = \dots = m_{y_b \rightarrow v} = \overline{m}_0; \\ m_0 & \text{otherwise,} \end{cases} \quad (3)$$

where  $b$  is an integer in the range  $\lfloor \frac{d_v - 1}{2} \rfloor < b < d_v$ . Here, the outgoing message of a variable node is the same as the intrinsic message, unless at least  $b$  of the extrinsic messages disagree. The value of  $b$  may change from one iteration to another. The optimum value of  $b$  for a regular  $(d_v, d_c)$  LDPC

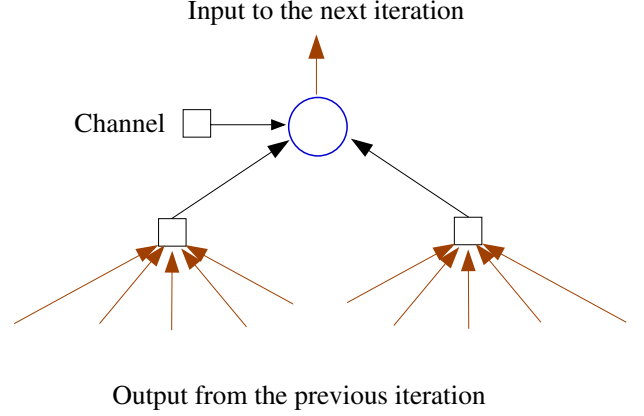


Fig. 1. One iteration of decoding for a regular (3, 6) code

code is computed by Gallager [5] and is the smallest integer  $b$  for which

$$\frac{1 - p_0}{p_0} \leq \left[ \frac{1 + (1 - 2p)^{d_c - 1}}{1 - (1 - 2p)^{d_c - 1}} \right]^{2b - d_v + 1}, \quad (4)$$

where  $p_0$  and  $p$  are channel crossover probability (intrinsic message error rate) and extrinsic message error rate, respectively. For irregular codes, one can use Algorithm B for nodes of different degree separately.

### B. EXIT chart analysis for Algorithm B

Using (2), (3) and (4), for a given check- and variable-degree distribution and a given channel condition, one can easily produce a  $p_{in}$  vs.  $p_{out}$  EXIT chart. Here,  $p_{in}$  is the error rate before an iteration (error rate of messages sent from variable nodes to check nodes) and  $p_{out}$  is the error rate after an iteration, i.e., after updating the messages at the check nodes and back at the variable nodes (see Fig. 1). Fig. 2 shows such an EXIT chart for a regular (4, 8) code.

For an irregular code, at each  $p_{in}$ , the  $p_{out}$  at the output of variable nodes is computed by first computing the error rate  $p_c$  at the output of the check nodes using Baye's rule, then similarly updating the  $p_{out}$  at the output of the variable nodes. The resulting  $p_{out}$  can be equivalently computed by linearly combining the corresponding  $p_{out}$ 's for regular codes. Consequently, when the check degree distribution is fixed, the EXIT chart of an irregular code is simply a linear combination of EXIT charts of regular codes weighted by the variable degree distribution.

The ability to linearly combine EXIT charts of regular codes to produce the EXIT chart of an irregular code is an important feature for EXIT charts based on tracking the probability of error [4]. This feature allows the optimization problem to be formulated easily. The EXIT chart corresponding for a fixed variable-degree  $i$  is called the *elementary* EXIT chart of degree  $i$ . It is denoted as  $f_i(p)$  in the rest of the paper. For example  $f_3(p)$  is the EXIT chart of a code whose variable nodes are all degree three. With a variable-degree distribution  $\lambda_i$ , the EXIT

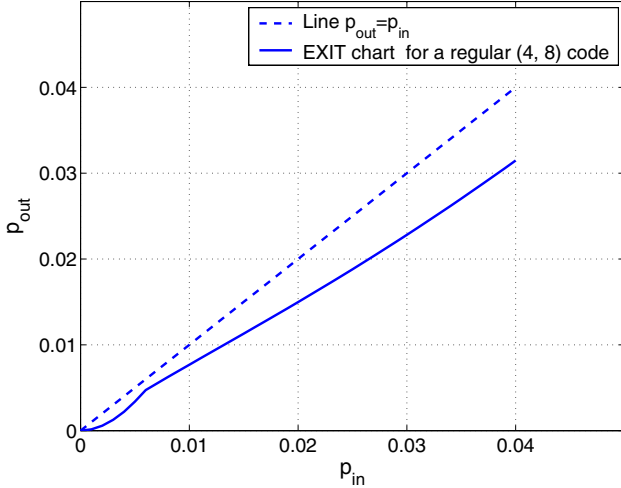


Fig. 2. Shows the Algorithm B EXIT chart for a regular (4, 8) code

chart of the irregular code becomes

$$f(p) = \sum_i \lambda_i f_i(p) \quad (5)$$

The elementary EXIT charts can be pre-computed. The code design variables are  $\lambda_i$  with  $\sum_i \lambda_i = 1$  and  $\lambda_i \geq 0$ .

Throughout this paper, the check-degree distribution is assumed to be fixed. We do not expect this assumption to result in significant performance degradation. Notice that with fixed check-degrees, the capacity of the binary erasure channel (BEC) can be achieved [10] and a performance very close to the Shannon limit on the Gaussian channel has been reported [11]. Our current work can be extended, and further optimization of check-degree distribution is possible. But, we do not do so here.

### C. Decoding complexity per iteration

From the update rules of Algorithm B, it can be shown that at a variable node of degree  $d_v$ ,  $3d_v$  operations are needed to compute all the output messages. This is computed as follows. At each variable node, one can add all  $d_v + 1$  input messages in  $d_v$  operations. To compute each outgoing message, an additional subtraction and an additional comparison (with  $b$ ) are required. As  $d_v$  outgoing messages need to be computed in each variable node, the total number of operations (addition, subtraction and comparison) per iteration for each variable node is  $d_v + 2d_v$ . Thus, the total number of operations summed over all variable nodes is  $3 \sum_v d_v$ , or equivalently  $3E$ , where  $E$  is the number of edges in the graph.

Similarly, at a check node of degree  $d_c$ , a total of  $2d_c - 1$  operations is needed. The number of operations can be counted as follows: first  $(d_c - 1)$  modulo-two additions need to be performed to find the total sum modulo-two, then another modulo-two addition is needed for every outgoing message. As there are  $d_c$  outgoing messages, the total number of operations is  $2d_c - 1$ . Summing across all check nodes, the total number of operations at the check side of the code is therefore  $2E - C$ , where  $C$  is the number of check nodes.

The total complexity is then  $5E - C$  per iteration. Since the number of edges in the graph is much larger than the number of check nodes, i.e.,  $5E \gg C$ , the complexity can be approximated by  $5E$ . Finally, as the complexity per iteration is implementation-dependent, it is reasonable to assume that the complexity is simply proportional to  $E$ .

The above complexity computation is for the whole codeword on a per-iteration basis. The complexity per information bit per iteration is then proportional to

$$\frac{E}{Rn},$$

where  $R$  is the code rate and  $n$  is the block length. As a result, complexity per information bit,  $K$ , is

$$K = \frac{N}{\sum_i \lambda_i / i - \sum_i \rho_i / i}, \quad (6)$$

where  $N$  is the number of iterations. To estimate the total complexity, the crucial parameter is therefore the total number of iterations.

### III. ANALYSIS OF NUMBER OF ITERATIONS

One of the main points of this paper is that the total number of iterations can be accurately estimated based on the shape of the EXIT chart  $f(p)$ . In the following, we present an approximate formula for the number of iterations needed to go from some initial probability  $p$  to some target probability of error  $p_t$  as a function of  $f(p)$ .

Consider first an EXIT chart which is a straight line,  $f(p) = \alpha p$ . In this case, the number of iterations needed to go from  $p$  to  $p_t$  can be computed exactly:

$$N = \left\lceil \frac{\log(p_t) - \log(p)}{\log(\alpha)} \right\rceil. \quad (7)$$

For a moment, let's ignore the fact that  $N$  has to be an integer. We can compute the incremental increase in  $N$  as a function of the incremental change in  $p$ :

$$\frac{dN}{dp} = \frac{-1}{p \log(\alpha)}. \quad (8)$$

Now, we may argue that for an arbitrary  $f(p)$  which is not necessarily a linear function, the local behavior of  $N$  as a function of  $p$  depends on an equivalent  $\alpha$ , which is  $\frac{f(p)}{p}$ . Thus, to estimate the total number of iterations from  $p_0$  to  $p_t$ , we may integrate the above:

$$N = \int_{p_t}^{p_0} \frac{dp}{p \log\left(\frac{p}{f(p)}\right)}. \quad (9)$$

This formula is a surprisingly accurate estimate of the number of iterations for a wide range of  $f(p)$ 's. Table I verifies the accuracy of the complexity measure by listing the actual number of iterations and that computed by the above formula for several realistic EXIT charts.

Perhaps the most important property of the measure of complexity (9) is that it is a convex function of the degree distribution.

Number of Iterations:	Actual	Estimated
EXIT Chart 1	192	193.41
EXIT Chart 2	180	181.03
EXIT Chart 3	126	126.62
EXIT Chart 4	83	82.45
EXIT Chart 5	62	61.36

TABLE I

ESTIMATE OF NUMBER OF ITERATIONS,  $p_0 = 0.05$ ,  $p_t = 10^{-6}$

*Theorem 1:* Let  $f(p) = \sum_i \lambda_i f_i(p)$  where  $f_i(p)$  is the elementary EXIT chart corresponding to degree- $i$  variable nodes. The measure of number of iterations as expressed in (9) is a convex function of  $\lambda_i$  in the region where  $f(p) \geq e^{-2}p$ .

*Proof:* To show the convexity of an integral, we only need to show the convexity of the integrand. Further, to show the convexity as a function of  $\lambda_i$ , we only need to show convexity along any line in the  $\lambda$ -plane. Let  $\lambda_i = \gamma_i \lambda + \beta_i$  for some arbitrary  $\gamma_i$  and  $\beta_i$ . The convexity of the integrand as a function of  $\lambda$  can be verified directly by taking its second derivative. The integrand is of the form:

$$g(\lambda) = \frac{1/p}{C - \log(A\lambda + B)} \quad (10)$$

where  $C = \log(p)$ ,  $A = \sum_i \gamma_i f_i(p)$  and  $B = \sum_i \beta_i f_i(p)$ . Using the fact that  $C \geq \log(A\lambda + B)$ , a direct verification reveals that the second derivative is always positive if  $C - \log(A\lambda + B) \leq 2$ , which is equivalent to  $f(p) \geq e^{-2}p$ .  $\square$

The condition  $f(p) \geq e^{-2}p$  is a fairly mild one at rates close to the decoding threshold. Numerically, the curve  $f(p) = e^{-2}p = 0.1353p$  is substantially below the line  $f(p) = p$ . At rates close to the decoding threshold, an optimized code generally has an EXIT chart that is between  $f(p) = p$  and  $f(p) = 0.1353p$ .

#### IV. OPTIMIZATION PROBLEM

##### A. Formulation

We are now ready to formulate the problem of minimizing the complexity of a code subject to a rate constraint.

$$\begin{aligned} & \text{minimize} && K \\ & \text{subject to} && R \geq R_0. \end{aligned} \quad (11)$$

The optimization variables are the variable-degree distribution parameters  $\lambda_i$ . Implicitly, we must have  $\sum_i \lambda_i = 1$  and  $\lambda_i \geq 0$ . It is easy to see from (1) that if the check-degree distribution  $\rho_i$  is fixed, the code rate  $R$  is simply a function of  $\lambda_i$ . More specifically, the rate constraint becomes a linear constraint:

$$\sum_i \lambda_i/i \geq \frac{1}{1-R_0} \sum_i \rho_i/i. \quad (12)$$

Clearly the above constraint would be met with equality for the minimal complexity code. In this case, the complexity measure  $K$  is directly proportional to the number of iterations  $N$ . As  $N$  is a convex function of  $\lambda_i$  and the constraints are linear,

this is a standard convex programming problem for which efficient numerical algorithms are available. For the sake of completeness, the optimization problem is:

$$\begin{aligned} & \text{minimize} && \left( \frac{1-R_0}{R_0 \sum_i \rho_i/i} \right) \int_{p_t}^{p_0} \frac{dp}{p \log \left( \frac{p}{\sum_i \lambda_i f_i(p)} \right)} \\ & \text{subject to} && \sum_i \lambda_i/i \geq \frac{1}{1-R_0} \sum_i \rho_i/i \\ & && \sum_i \lambda_i = 1 \\ & && \lambda_i \geq 0 \end{aligned} \quad (13)$$

Here,  $R_0$ ,  $\rho_i$  and  $f_i(p)$  are fixed parameters.  $\lambda_i$  is the optimization variable. In practice, the integral is numerically evaluated as a Riemann sum.

##### B. Numerical Results

We used an interior-point method to perform the numerical minimization of the code complexity subject to rate constraints on a binary symmetric channel with crossover probability of 0.05 and a target probability of error  $10^{-6}$ . The check degree is fixed to 14 and variable degrees of up to 20 are used.

The optimized complexity rate tradeoff result is plotted in Fig. 3. Also plotted on the same figure is the complexity of the codes designed using conventional methods. In the conventional method, the code is designed for a slightly worse channel (with a higher crossover probability) but tested in the actual channel. Our convex optimization approach significantly lowers the complexity. In many cases, the complexity savings are about 30–40%.

The tradeoff figure also illustrates clearly that Algorithm B is capable of decoding at a rate up to 0.575. As expected, the complexity-rate tradeoff curve is the steepest in that regime, and it tapers off rapidly as the rate decreases below 0.56.

Fig. 4 and Fig. 5 show the EXIT charts of the optimized codes in the linear scale and log scale, respectively. As expected, the EXIT chart opens more widely as rate decreases and complexity decreases. In a log-scale, the EXIT charts at low probabilities essentially behave like  $f(p) = \alpha p$  for different values of the coefficient. All EXIT charts are within the convexity region stated in Theorem 1. Table II lists the optimized degree distributions at various complexity levels.

#### V. CONCLUDING REMARKS

The paper illustrates that the conventional LDPC code-design in which the design rate is maximized subject to successful decoding is not necessarily the best approach. The central observation of this paper is that rate maximization can be done more directly subject to a complexity constraint. Our numerical method relies on an accurate EXIT chart analysis of the decoding algorithm based on the probability of error, and on the fact that the complexity in terms of the number of decoding iterations is a convex function of the degree distribution (which is true at rates close to the decoding threshold.) When the EXIT chart analysis is exact (as is the case for Algorithm B), this method produces an accurate

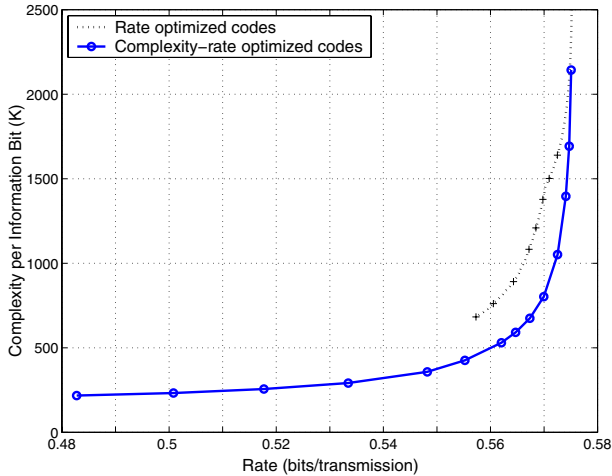


Fig. 3. Complexity-rate tradeoff on a binary symmetric channel with crossover probability 0.05 for an irregular LDPC code with check degree 14 and with Gallager's Algorithm B decoding. The lower curve is the complexity-rate optimized code design. The upper curve is produced by designing the highest rate codes for a slightly worse channel crossover probability.

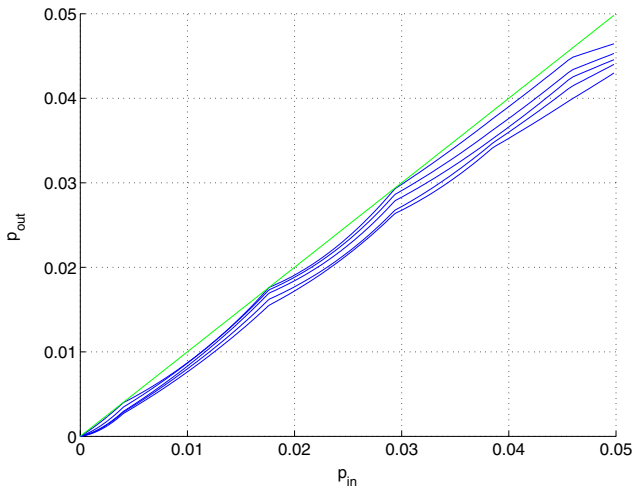


Fig. 4. EXIT charts for complexity-rate optimized codes in linear scale.

numerical characterization of the complexity-rate tradeoff for optimized codes.

#### REFERENCES

- [1] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, "Improved low-density parity-check codes using irregular graphs," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 585–598, Feb. 2001.
- [2] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.
- [3] S.-Y. Chung, G. D. Forney Jr., T. J. Richardson, and R. L. Urbanke, "On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit," *IEEE Commun. Lett.*, vol. 5, no. 2, pp. 58–60, Feb. 2001.
- [4] M. Ardakani, T. H. Chan, and F. R. Kschischang, "EXIT-Chart properties of the highest-rate LDPC code with desired convergence behavior," *IEEE Commun. Lett.*, vol. 9, no. 1, pp. 52–54, Jan. 2005.
- [5] R. G. Gallager, *Low-Density Parity-Check Codes*, MIT Press, Cambridge, MA, 1963.

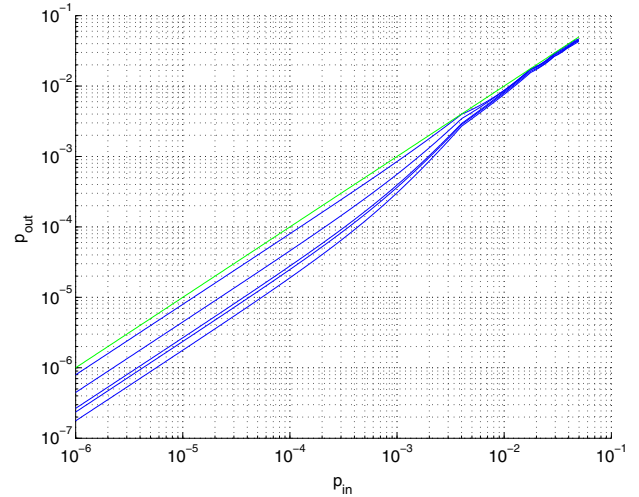


Fig. 5. EXIT charts for complexity-rate optimized codes in log scale.

Degrees	K=1639	K=530	K=232
2	0.039	0.015	0.000
3	0.254	0.209	0.104
4	0.030	0.176	0.222
5	0.000	0.000	0.071
6	0.000	0.000	0.060
7	0.000	0.000	0.001
8	0.000	0.000	0.001
9	0.000	0.000	0.001
10	0.000	0.001	0.001
11	0.000	0.000	0.001
12	0.668	0.002	0.001
13	0.000	0.001	0.001
14	0.003	0.514	0.002
15	0.004	0.019	0.003
16	0.002	0.027	0.004
17	0.000	0.031	0.007
18	0.000	0.002	0.009
19	0.000	0.002	0.399
20	0.000	0.001	0.112

TABLE II

DEGREE DISTRIBUTION FOR RATE-COMPLEXITY OPTIMIZED CODES AT VARIOUS COMPLEXITY LEVELS

- [6] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.
- [7] A. Khandekar and R. J. McEliece, "On the complexity of reliable communication on the erasure channel," in *Proc. IEEE Int. Symp. Information Theory (ISIT)*, 2001, p. 1.
- [8] I. Sason and R. Urbanke, "Complexity versus performance of capacity-achieving irregular repeat-accumulate codes on the binary erasure channel," *IEEE Trans. Inform. Theory*, vol. 50, no. 6, pp. 1247–1256, June 2004.
- [9] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Trans. Commun.*, vol. 49, pp. 1727–1737, Oct. 2001.
- [10] A. Shokrollahi, "New sequence of linear time erasure codes approaching the channel capacity," in *Proc. the International Symposium on Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, Lecture Notes in Computer Science, 1999, number 1719, pp. 65–67.
- [11] M. Ardakani and F. R. Kschischang, "A more accurate one-dimensional analysis and design of LDPC codes," *IEEE Trans. Commun.*, vol. 52, no. 12, pp. 2106–2114, Dec. 2004.