Complexity-Optimized Low-Density Parity-Check Codes

Masoud Ardakani
Department of Electrical & Computer Engineering
University of Alberta, ardakani@ece.ualberta.ca

Benjamin Smith, Wei Yu, Frank R. Kschischang
Department of Electrical & Computer Engineering
University of Toronto, {ben, weiyu, frank}@comm.utoronto.ca

Abstract

Using a numerical approach, tradeoffs between code rate and decoding complexity are studied for long-block-length irregular low-density parity-check codes decoded using the sum-product algorithm under the usual parallel-update message-passing schedule. The channel is an additive white Gaussian noise channel and the modulation format is binary antipodal signalling, although the methodology can be extended to any other channels for which a density evolution analysis may be carried out. A measure is introduced that incorporates two factors that contribute to the decoding complexity. One factor, which scales linearly with the number of edges in the code's factor graph, measures the number of operations required to carry out a single decoding iteration. The other factor is an estimate of the number of iterations required to reduce the the bit-error probability from that given by the channel to a desired target. The decoding complexity measure is obtained from a density-evolution analysis of the code, which is used to relate decoding complexity with the code's degree distribution and code rate. One natural optimization problem that arises in this context is to maximize code rate for a given channel subject to a constraint on decoding complexity. At one extreme (no constraint on decoding complexity) one obtains the "threshold-optimized" LDPC codes that have been the focus of much attention in recent years. Such codes themselves represent one possible means of trading decoding complexity for rate, as such codes can be applied in channels better than the one for which they are designed, achieving the benefit of a reduced decoding complexity. However, it is found that the codes optimized using the methods described in this paper provide a better tradeoff, often achieving the same code rate with approximately 1/3 the decoding complexity of the threshold-optimized codes.

1 Introduction

The problem of designing capacity-approaching irregular low-density parity-check (LDPC) codes under different decoding algorithms and channel models has been studied extensively (e.g., as a starting point, see [1, 2, 3]). The usual design objective is to find a code degree distribution that maximizes the decoding threshold (e.g., the largest noise variance for which successful decoding is possible) for a given rate or, equivalently, to find a degree distribution that maximizes the code rate for a given decoding threshold. In practice, however, such codes would require an impractically large number of decoding iterations. In fact, for decoding algorithms with the property that message distributions can be described by a single parameter, it is proved that, in the limit, the required number of iterations for convergence approaches infinity as the rate of the code increases [4]. As a result, a threshold-optimized code in practice must be used over a channel different (better) than the one for which the code is designed.

The problem with the threshold-optimization approach is that decoding complexity is not considered in the process of code design. Put another way, if one wishes to design a practically-decodable code, it may be best to further trade the decoding threshold for a more desirable decoding trajectory. If decoding complexity could be incorporated in the design process, one could find the highest-rate code for a certain affordable level of complexity on a given channel condition or, the code with the minimum decoding complexity for a required rate and a given channel condition. Clearly, these design objectives better reflect the requirements of practical communication-system design. Towards this end, we show in this paper how the decoding complexity of an irregular LDPC code can be related to its degree distribution. We then formulate a methodology that is capable of finding low-complexity degree distributions for a given code rate and channel.

Understanding the performance/complexity tradeoff has always been a central issue in coding theory. In recent years, particularly since the discovery of capacity-approaching codes, "performance" has come to mean the achievable code rate. For the binary erasure channel (BEC), the rate/complexity issue is addressed by Shokrollahi [5] for LDPC codes, and by Khandekar and McEliece [6] and also Sason and Urbanke [7] for irregular repeat-accumulate codes. Decoding the output of an erasure channel is quite different than decoding the output of a noisy channel, as the decoding procedure operates via an edge-deletion process, whose complexity scales linearly with the number of edges in the code's factor graph. However, for other channel models, decoding is an iterative process, and so a decoding complexity measure must also scale with the required number of iterations. This observation has previously been made in the work of Richardson and Urbanke [2], where a numerical design procedure for irregular LDPC codes involving a measure of the number of iterations is proposed. Recently, in [8], we studied this problem for LDPC coding under Gallager's decoding Algorithm B [9] over the binary symmetric channel. The results of [8] rely on the one-dimensional nature of decoding Algorithm B. This paper addresses complexity-based designs for the general binary-input symmetric-output channels, and symmetric decoding. That is to say, as long as a density evolution analysis of the decoder is possible, our method of complexity optimization is applicable.

In this work, although the decoding complexity of an LDPC code is related to its degree distribution via a one-dimensional representation of the decoder, the results are not based on a one-dimensional analysis. We use density evolution for the analysis, yet represent the convergence behavior in a one-dimensional format that we call an extrinsic information transfer (EXIT) chart (despite the fact that it tracks the message error rate rather than the mutual information, and the fact that it is based on a density evolution analysis instead of a Gaussian approximation). A one-dimensional representation plays a central role in the formulation of the code-design problem [10] as well as in the formulation of complexity in terms of the code degree distribution [8].

The rest of this paper is organized as follows. In Sections 2 and 3, we briefly review elementary EXIT charts and their role in irregular LDPC code design as well as a method for obtaining elementary EXIT charts through density evolution. In Section 4 we define the decoding complexity and we present a formula which relates the degree distribution of the code to its decoding complexity for achieving a certain target error rate. In Section 5, the optimization problem is formulated and the optimization methodology is described. The numerically optimized degree distributions and the complexity-performance tradeoff curves for a Gaussian channel are presented in Section 6 and the paper is concluded in Section 7.

2 Preliminaries

Following the notation of [1], we define an ensemble of irregular LDPC codes by its variable-degree distribution $\{\lambda_2, \lambda_3, \ldots\}$ and its check-degree distribution $\{\rho_2, \rho_3, \ldots\}$, where λ_i denotes the fraction of edges incident on variable nodes of degree i and ρ_j denotes the fraction of edges incident on check nodes of degree j. If all the parity constraints are linearly independent, it is easy to see that the rate of an irregular LDPC code is related to its degree distribution by

$$R(\lambda, \rho) = 1 - \frac{\sum_i \frac{\rho_i}{i}}{\sum_i \frac{\lambda_i}{i}}. \tag{1}$$

If, due to the random construction of the code, some of the parity check constraints are linearly dependent, the actual rate of the code will be slightly higher.

There are many decoding algorithms available for LDPC codes. If the decoding algorithm and the channel satisfy some symmetry properties [11], performance of a given code can be studied by density evolution. The inputs to the density evolution algorithm [11] are the probability density function (pdf) of channel log-likelihood ratio (LLR) messages[1] and the pdf of the extrinsic LLR messages from the previous iteration. The output is the pdf of the extrinsic LLR messages at the current iterations. This density will be used as input for finding the message density in the next iteration. The negative tail of the LLR density is the message error rate. If decoding is successful, this tail vanishes as the number of iterations tends to infinity.

In the above discussion, one iteration is defined as one round of message updates at both the check nodes and the variable nodes. In other words, the input to one iteration is the messages sent from variable nodes to check nodes and the output is the same set of messages after being updated at check nodes and variable nodes. We assume that these updates occur in parallel, even though it is possible that a serial updating schedule may result in fewer iterations.

3 EXIT chart representation of density evolution

An EXIT chart based on message error rate is motivated by Gallager's decoding Algorithm B [9]. Using Algorithm B to decode a regular LDPC code, it is easy to establish a recursive equation that provides an exact description of the message error rate after a given number of iterations. In this case, a p_{in} vs. p_{out} EXIT chart is precisely a graphical representation of this equation.

With this in mind, from density evolution we may obtain a p_{in} vs. p_{out} EXIT chart for an arbitrary channel and decoder pair as follows. After performing N iterations of

[1] Under the assumption that the all-zero codeword is transmitted.

density evolution, one can visualize the convergence behavior of the decoder by plotting the extrinsic message error rate of iteration $m, m \in \{1, 2, \cdots, N\}$ (let us call this $p^{(m)}$) vs. the extrinsic message error rate of iteration $m-1$, i.e., $p^{(m-1)}$. This resembles the EXIT chart analysis of [12], hence we call it an EXIT chart. This can also be represented as a mapping (a function)

$$p^{(m)} = f(p^{(m-1)}, \lambda, \rho), \ m \in \{1, 2, \cdots, N\}. \tag{2}$$

Notice that the above defined method results in an EXIT chart which is defined on N discrete points. Also notice that f as well as its domain and range are influenced by the code parameters λ and ρ. Nevertheless, using Bayes' rule we have

$$p^{(m)} = \sum_i \lambda_i \cdot p_i^{(m)}, \tag{3}$$

where $p_i^{(m)}$ is the message error rate at the output of degree i variable nodes at iteration m. Equation (3) can be rewritten as

$$f(p, \lambda, \rho) = \sum_i \lambda_i \cdot f_i(p, \lambda, \rho), \tag{4}$$

where $f_i(p, \lambda, \rho)$ can be thought as an EXIT chart associated with degree i nodes. This is similar to elementary EXIT charts of [4], where the EXIT chart of an irregular code decomposes as a linear combination of EXIT charts of left-regular codes. This reduces the problem of code design to the problem of shaping an EXIT chart out of some pre-computed elementary EXIT charts. It is evident here that elementary EXIT charts are central to the formulation of the optimization problem of interest.

However, as the elementary EXIT charts of (4) are functions of λ, the EXIT chart of the code is affected by λ_k in two ways. One is through the linear combination of (4) as a multiplying factor and the other one is the effect of λ on $f_i(p, \lambda, \rho)$'s, i.e.,

$$\frac{\partial f(p, \lambda, \rho)}{\partial \lambda_k} = f_k(p, \lambda, \rho) + \sum_i \lambda_i \frac{\partial f_i(p, \lambda, \rho)}{\partial \lambda_k}. \tag{5}$$

In practice, the first term on the right hand of (5) is much larger than the second term. Therefore, as long as λ undergoes a small change, we may disregard the dependency of elementary EXIT charts on λ. Fixing ρ, (4) can be simplified to

$$f(p) = \sum_i \lambda_i \cdot f_i(p), \tag{6}$$

which is equivalent to the formulation of [4]. However, due to the dependency of elementary EXIT charts on λ, they have to be updated when λ undergoes a large change.

Another technical issue (which arises here but not in the case of Algorithm B) is that the elementary EXIT charts cannot be computed individually from left-regular codes. There are two reasons for this. First of all, the probability distribution associated with Algorithm B can be described by a single parameter. Therefore, a particular value of p_{in} maps to a unique input distribution. However, in the case of the Gaussian channel, the input distribution at a particular value of p_{in} is dependent on λ, and can only be obtained by performing full density evolution on the entire code at once. Secondly, analysis of Algorithm B provides an explicit formula for the left-regular elementary EXIT charts.

Therefore, a component code that may not converge on its own during density evolution, say a code which has exclusively degree-two variable nodes, nevertheless has an explicit representation that allows one to construct a p_{in} vs. p_{out} curve over its entire domain. For the case of the Gaussian channel, if one conducts density evolution on an elementary degree-two variable code, the process would fail to converge for most cases of interest, preventing us from obtaining an elementary EXIT chart defined over the full domain of p_{in}.

To overcome these problems, we obtain elementary EXIT charts by running density evolution for the irregular code. At each iteration, we find the LLR message pdf at the output of variable nodes of different degrees, and we extract the LLR message error rate (negative tail of pdf) for each variable node degree. This forms $f_i(p, \lambda, \rho)$. Here, it is assumed that the irregular code will converge to the specified target error rate, which guarantees that the elementary EXIT charts are defined over a discrete set of p such that $\min(p) \leq p_t$, where p_t is the target error rate specified in the code design problem. This allows for shaping an EXIT chart of desired properties over the range of interest. By interpolating, one can acquire a continuous version of the elementary EXIT charts over the interval $[p_t, p_0]$, where p_0 is the initial message error rate (from the channel messages) and p_t is the target error rate.

4 Decoding Complexity Analysis

In this section, we first study the decoding complexity per iteration, and then we analyze the required number of iterations for a target error rate.

4.1 Decoding complexity per iteration

For a message-passing decoder, it is not hard to see that the decoding complexity per iteration scales roughly linearly with the number of edges. This is because each edge carries a message, and each message has to be updated at each iteration, and such an update requires essentially constant computational effort. A detailed study of this complexity for the sum-product decoding (which can be extended to general message-passing rules) is as follows.

At a variable node of degree d_v, a maximum of $2d_v$ operations is enough to compute all the output messages. Notice that at each variable node, one can add all $d_v + 1$ input LLR messages in d_v operations. To compute each outgoing message, one subtraction is required. Since d_v outgoing messages should be computed, the total number of operations (addition and subtraction) per iteration at the variable nodes is $2\sum_v d_v$, or equivalently $2E$, 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 $(d_c - 1)$ products (of tanh) need to be performed, then one division per outgoing message (taking tanh and atanh is not considered as an operation). Since there are d_c outgoing messages, the total number of operations is $2d_c - 1$, which results in a total of $2E - C$ operations, where C is the number of check nodes.

The overall complexity per iteration is then $4E - C$, which is roughly proportional to E, since usually $4E \gg C$. Therefore, in this work we assume that the complexity per iteration is simply proportional to E. This complexity has to be normalized per information bit to be meaningful. The complexity per information bit per iteration is then proportional to

$$\frac{E}{Rn}, \tag{7}$$

where R is the code rate and n is the block length.

4.2 Analysis of Number of Iterations

As mentioned earlier, for the binary erasure channel, since decoding is an edge deletion process, the total complexity is measured as the number of edges. However, in all other message-passing decoding algorithms, the number of iterations directly affects the total decoding complexity. The complexity discussion in the previous section was only for one iteration. Considering N decoding iterations, from (7) and (1), the overall complexity per information bit, X, can be measured as

$$X = \frac{N}{\sum_i \lambda_i/i - \sum_i \rho_i/i}. \tag{8}$$

Therefore, to find the total complexity, the required number of iterations, N, should be estimated.

The number of iteration strongly depends on the shape of the EXIT chart $f(p)$ and the target message error rate. It is shown in [8] that the number of required iterations can be closely approximated as

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

This formula is a very accurate estimate of the number of iterations for a wide range of $f(p)$'s. Some examples are provided in [8].

We finish this section by rewriting (8), explicitly in terms of the code degree distribution. That is

$$X(\lambda,\rho) = \frac{1}{\sum_i \lambda_i/i - \sum_i \rho_i/i} \int_{p_t}^{p_0} \frac{dp}{p \log\left(\frac{p}{\sum_i \lambda_i f_i(p)}\right)}. \tag{10}$$

It is also shown in [8] that for a fixed ρ this complexity measure is a convex function of λ over the set

$$\left\{\lambda : \forall p \left(\frac{1}{e^2} \leq \frac{\sum \lambda_i f_i(p)}{p} \leq 1\right)\right\}.$$

5 Optimization Methodology

Following the approach of [8], we minimize the complexity of a code subject to a rate constraint. By varying the target rate and solving the sequence of optimization problems, we obtain the complexity-rate tradeoff curve.

For simplicity, let us assume that ρ is fixed. There exists convincing evidence that conventional rate/threshold optimization techniques are not very sensitive to ρ. For instance, it is shown that codes with regular check degree can achieve the capacity of the binary erasure channel [5] and can perform very close to the Shannon limit on the Gaussian channel [10]. Recall that the fixed ρ assumption, together with the assumption of keeping λ in a small region of space, results in (6), which defines a linear relation between the design parameters, i.e., λ's, and the shape of the EXIT chart of the irregular code. However, it should be noted that for a given target rate, varying ρ affects both the number of decoding iterations as well as the number of edges in the code's graph. Therefore, evidence indicating that concentrated check degrees are sufficient for threshold optimized codes may not extend to complexity optimized codes, for which we can further trade decoding iterations for the density of edges. A joint optimization of both ρ and λ may be beneficial, but we focus on the optimization of λ herein.

To solve the optimization problem, we suggest solving a sequence of problems of the following form:

$$\begin{aligned} \text{minimize} \quad & \left(\frac{1-R_0}{R_0 \sum_i \rho_i/i}\right) \int_{p_t}^{p_0} \frac{dp}{p \log\left(\frac{p}{\sum_i \lambda_i f_i(p)}\right)} \\ \text{subject to} \quad & \sum_i \lambda_i/i \geq \frac{1}{1-R_0} \sum_i \rho_i/i \\ & \sum_i \lambda_i = 1 \\ & \lambda_i \geq 0 \\ & \|\lambda - \bar{\lambda}\|_2 \leq \epsilon \end{aligned} \tag{11}$$

Here, ρ_i is fixed, ϵ is the radius of a search sphere, R_0 is some target rate, $\bar{\lambda}$ is the initial value of λ, and λ is the optimization variable.

We begin with $\bar{\lambda}$ set to the variable degree distribution of the maximum rate code for the desired channel condition, with its corresponding elementary EXIT charts $f_i(p)$. Such a distribution can be found by an iterative linear programming formulation that similarly considers elementary EXIT charts to be invariant under small changes in λ. Next, we set R_0 to our target rate, and solve a sequence of optimization problems (11) iteratively until we obtain a minimal complexity code. As shown in [8], with a fixed set of $f_i(p)$, the optimization problem (11) is convex in λ_i. Thus, a global optimum solution may be found efficiently at each iteration step. At the end of each iteration, which is typically limited in progress due to ϵ, we set $\bar{\lambda}$ to the most recently computed λ, and recompute the elementary EXIT charts. In practice, we choose a larger ϵ in earlier stages and a smaller one in later stages.

Our preceeding analysis suggests that ϵ should be quite small in order to provide valid results, thus requiring frequent calls to the density evolution routine. However, a slightly different and empirically more efficient optimization approach is also possible. Consider a modified version of the above optimization problem that omits the search sphere constraint on λ, but that solves a series of problems with a rate constraint R_0 that is successively decreased between iterations, say by 1%, until a target rate is reached. Note that a small change in rate does not necessarily imply a small change in λ, and furthermore that the computed minimum will generally not be equal to the true minimum. Nevertheless, the results achieved using such an approach are strikingly similar to those achieved with a restrictive search sphere constraint. Therefore as long as the change in rate between iterations is relatively small, this more efficient method generates nearly equivalent results. In both cases, we find that the complexity metric imposes an aversion to low-degree variable nodes.

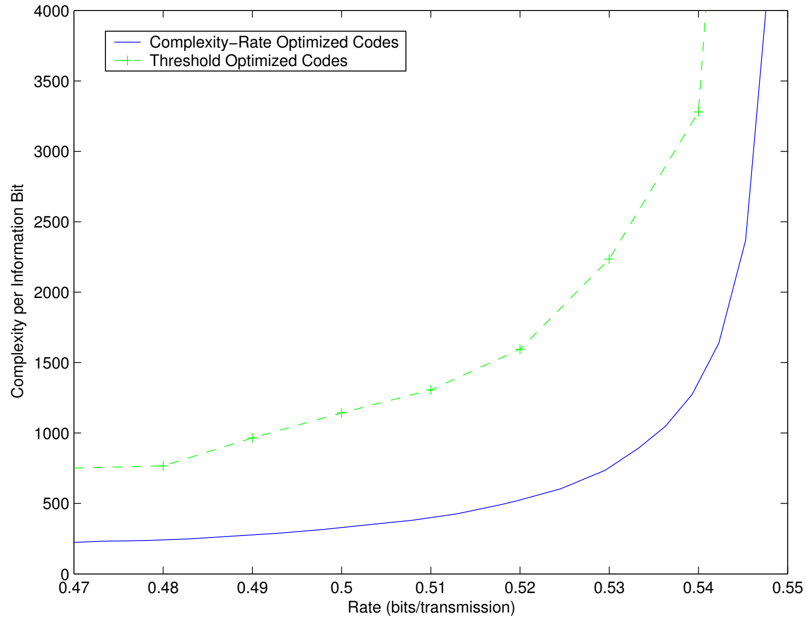


Figure 1: Complexity-rate tradeoff on an AWGN channel with noise $\sigma = 0.9$ for an irregular LDPC code with check degree 9 and sum-product decoding. The lower curve is the complexity-rate optimized code design. The upper curve is produced by designing the highest threshold code for each rate.

6 Numerical Results and Discussion

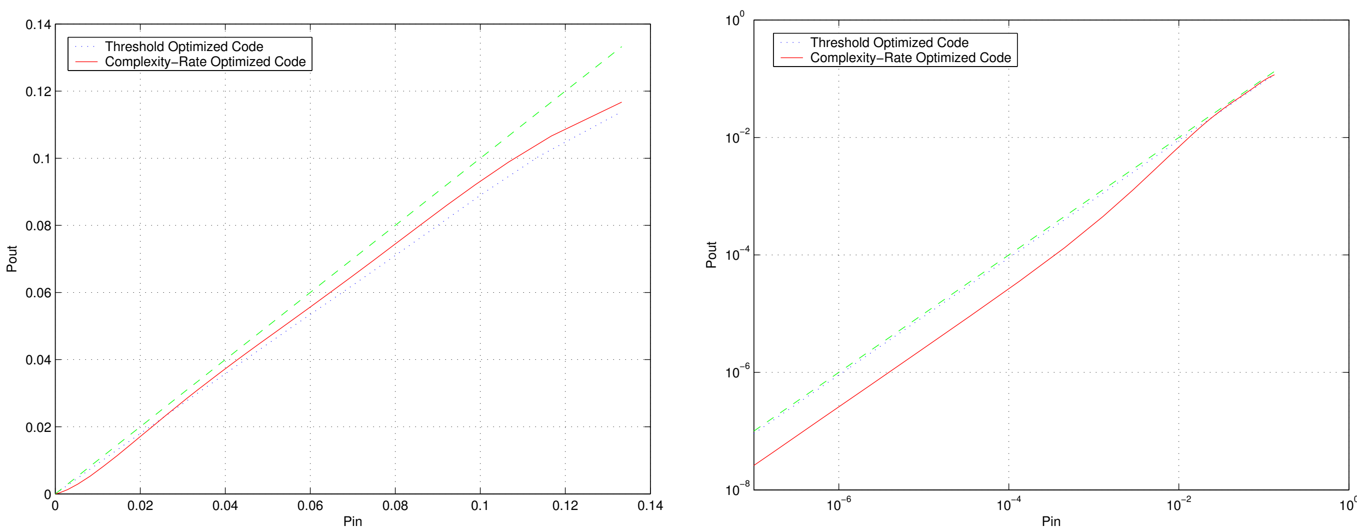
In this section, we present the results of our numerical optimization on a binary-input additive white Gaussian noise channel with noise $\sigma = 0.9$. The target error rate is set to 10^{-7}. To allow for a fair comparison between our design method and conventional design methods, threshold optimized codes of different rates, as produced by `LdpcOpt` [13], are tested on the channel of study and their rate is plotted versus their complexity in Fig. 1. The results of the rate-complexity tradeoff after performing our optimization are also plotted on the same figure. As can be seen, the new code-design technique results in significantly more efficient codes. In many cases, the decoding complexity is reduced by a factor of three or more. All the codes are limited to a maximum variable node degree of 30. Additionally, regular check degrees of degree 8, 9 and 10 were used. The plotted results consist of the best of the three codes at each value of the rate. Table 1 compares the degree distribution of a rate half code found through the new optimization technique for the above channel, with a rate half code found through conventional design methods. Both codes have a regular check degree of 9.

Fig. 2(a) compares the EXIT charts of the codes presented in Table 1. Fig. 2(b) shows the same EXIT charts but in log scale. Observe that the rate-complexity optimized code has a tighter EXIT chart in the earlier iterations and a wider EXIT chart at the later iterations. It is proved in [4] that, for a wide class of decoding algorithms, when the EXIT chart of two codes, $f^{(1)}(p)$ and $f^{(2)}(p)$ satisfy $f^{(1)}(p) \geq f^{(2)}(p), \forall p \in (0, p_0]$ then $f^{(1)}(p)$ corresponds to a higher rate. Also in [14] it is proved that for the binary erasure channel, the area underneath the EXIT chart scales with the code rate. In this example, since both codes have the same rate, one EXIT chart cannot dominate the other one everywhere, but the optimization program carefully trades the area underneath the EXIT chart for the complexity. By opening up the EXIT chart close to the origin, we do not expect a

Deg.	TO	RCO	Deg.	TO	RCO	Deg.	TO	RCO
2	0.2124	0.0623	11	0.0000	0.0173	19	0.0000	0.0255
3	0.1985	0.4948	12	0.0000	0.0260	20	0.0003	0.0217
5	0.0084	0.0000	13	0.0000	0.0313	21	0.0000	0.0176
6	0.0747	0.0000	14	0.0000	0.0340	22	0.0000	0.0132
7	0.0142	0.0000	15	0.0000	0.0346	23	0.0000	0.0086
8	0.1665	0.0118	16	0.0000	0.0337	24	0.0000	0.0038
9	0.0091	0.0000	17	0.0000	0.0317	25	0.0000	0.0008
10	0.0200	0.0048	18	0.0000	0.0288	30	0.2959	0.0975

Table 1: Degree distributions of two rate-1/2 codes, one optimized for threshold (TO) and the other optimized for rate and complexity (RCO). Total decoding complexity per information bit is estimated at 1143 (with 127 iterations) for the former and 342 (with 38 iterations) for the latter.

significant rate loss, yet we obtain a significant complexity gain. The small rate loss is then compensated for by slightly tightening up the EXIT chart in early iterations (which have no considerable impact on the complexity).



(a) (b)

Figure 2: EXIT charts for complexity-rate optimized codes: (a) linear scale, (b) log scale.

7 Concluding Remarks

This paper proposes a new LDPC code-design objective based on a joint optimization of rate and complexity. The paper argues that the conventional LDPC code-design which maximizes the code rate alone is not the best approach for practical purposes. The central observation is that, through a one-dimensional representation of convergence behavior, the decoding complexity of an irregular LDPC code can be related to its degree distribution in a closed form, hence a joint optimization of rate and complexity is possible. Our technique is applicable to all binary-input symmetric-output channels, where a density evolution analysis is possible. Numerical results on a Gaussian channel show substantial complexity reduction using the new optimization technique.

References

[1] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, "Improved low-density parity-check codes using irregular graphs," *IEEE Transactions on Information 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 Transactions on Information 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 Communications Letters*, 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 Communications Letters*, vol. 9, no. 1, pp. 52–54, Jan. 2005.

[5] 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*, no. 1719, Lecture Notes in Computer Science, 1999, pp. 65–67.

[6] A. Khandekar and R. J. McEliece, "On the complexity of reliable communication on the erasure channel," in *Proc. IEEE International Symposium on Information Theory*, 2001, p. 1.

[7] I. Sason and R. Urbanke, "Complexity versus performance of capacity-achieving irregular repeat-accumulate codes on the binary erasure channel," *IEEE Transactions on Information Theory*, vol. 50, no. 6, pp. 1247–1256, June 2004.

[8] W. Yu, M. Ardakani, B. Smith, and F. R. Kschischang, "Complexity-optimized LDPC codes for Gallager decoding algorithm B," in *Proc. IEEE International Symposium on Information Theory*, Adelaide, Australia, Sept. 2005.

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

[10] M. Ardakani and F. R. Kschischang, "A more accurate one-dimensional analysis and design of LDPC codes," *IEEE Transactions on Communications*, vol. 52, no. 12, pp. 2106–2114, Dec. 2004.

[11] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.

[12] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Transactions on Communications*, vol. 49, pp. 1727–1737, Oct. 2001.

[13] R. L. Urbanke, "LdpcOpt: a fast and accurate degree distribution optimizer for LDPC code ensembles," online: `lthcwww.epfs.ch/research/ldpcopt`.

[14] A. Ashikhmin, G. Kramer, and S. ten Brink, "Extrinsic information transfer functions: a model and two properties," in *Proc. Conference on Information Sciences and Systems*, Princeton, USA, Mar. 2002, pp. 742–747.