

Performance Analysis of Low-Density Parity-Check Codes Derived from Finite Inversive Spaces

Marcus Greferath and Cornelia Roessing

*School of Mathematical Sciences
University College Dublin
IRELAND*

E-mail: `marcus.greferath@ucd.ie`, `roessing@maths.ucd.ie`

Abstract — Low Density Parity Check (LDPC) codes have been the center of numerous researches in the last ten years. The reason for this interest is due to the performance (Error ratio over SNR) that these codes can achieve keeping the complexity of the pair coding/decoding to a level lower than other codes such as Turbo codes. This project aims to continue and intensify performance simulations on LDPC codes that have been begun in a previous project on the Walton cluster granted by ICHEC. One of its main interests is to study the performance of these codes at bit-error ratios down to 10^{-13} , which can only be determined in a long number of Monte-Carlo simulations on a distributed computing cluster with high computational resources.

I CONTEXT AND OUTCOME

Reliable communications are in great demand today. Common applications desire higher bandwidth communications in devices consuming less and less energy. It is, therefore, imperative to use the transmission systems available as effectively as possible.

The science of finding efficient schemes by which information can be coded for reliable transmission through a noisy channel is called coding theory. The basic idea behind coding and error correction is to add redundant data with each transmission so that, even if errors occur, sufficient protection exists to recover the original message. In other words, the redundant data can be used to recreate information lost during transmission.

Many different types of error correcting codes have been discovered during the past 50 years and are used in telecommunications; Among these the BCH and Reed-Solomon (RS) codes are some of the most common codes used from GSM to optical transmissions.

In recent years two classes of codes that exhibit performances near the Shannon limit for noisy channels have been developed. These are Turbo codes and Low Density Parity Check codes (LDPC). Introduced by Gallager in 1963 [1] LDPC codes had been neglected until the work of MacKay in 1995[2]; these codes can yield high performances on the binary symmetric channel (BSC) as well as on the additive white Gaussian noise (AWGN) channel, and have been shown to outperform the Turbo codes in many applications. The algorithm used for decoding is called belief propagation, and one of its versions is known as the *sum-product algorithm*. This algorithm uses a graphical representation of the code, the Tanner Graph.

The decoding scheme based on belief propagation is highly efficient, and it is desirable to have the encoding process of these codes as most efficient as well. To achieve this goal various scholars have searched for systematic constructions of LDPC codes. Many good constructions are known nowadays, but only a few of them (if at all) outperform random constructions of LDPC codes. For this reason there is further demand for the systematic construction of LDPC codes with excellent performance.

II LDPC CODES FROM FINITE GEOMETRY

LDPC codes have been systematically constructed in various ways. Margulis [3] initiated the use of a Cayley graph of a group to construct a sparse bipartite graph which in turn induces an LDPC code

once one writes down the incidence matrix of the graph. Further work has been done by Rosenthal and Vontobel [4, 5] and Lafferty and Rockmore [6]. In both cases Ramanujan graphs, which are optimal relative to a certain expansion property, were constructed using the Cayley graph of a suitable group. Another interesting approach is due to Bond, Hui and Schmidt [7] and later Greferath, O’Sullivan and Smarandache [8]. In all these constructions, linear congruences are used to relate the row and column numbers of the nonzero entries of a sparse parity-check matrix. Simulation results show that the codes constructed using this method perform at least as well as the randomly generated low-density parity-check codes.

A few years ago, it turned out that also geometric approaches can be used in the construction of LDPC codes. For example, Vontobel and Tanner [9] discovered a way to use finite generalized polygons (FGPs) to construct Tanner graphs and LDPC codes. This associated graph has the property that the girth is exactly twice the diameter which is the largest possible. Another construction was proposed by Kou, Lin and Fossorier [10] and makes use of the general concept of incidence structures. In [10] the underlying incidence structures were either affine or projective spaces over the finite field \mathbb{F}_{2^s} . The approach in [9] can be viewed on the same concept, namely that the occurring points are points of a projective space and the lines form a subset of the lines in that space determined by a bilinear form. By exploiting quadratic forms in such spaces, comparable work has been done recently in papers by Mellinger and Storme [11].

In a foregoing project with ICHEC, we studied a selection of LDPC codes derived from incidence structures that are known as *circle geometries*. The basic context is what are called *chain spaces*. These spaces are incidence structures consisting of points and chains, where chains might be thought of as a generalisation of circles. Indeed, among other postulates in this structure, any three distinct points define exactly one chain of the space. In such spaces there is a notion of tangent chains, and a maximal set of chains mutually tangent in one and the same point is usually referred to as a *pencil*. Under suitable assumptions a derived incidence structure consisting of these pencils as points, and the original chains as lines can be shown to be a triangle free partial linear space, also known as $(0, 1)$ -geometry. The studies and performance simulations of the LDPC codes in the afore-mentioned ICHEC project have shown that the codes in question are of very high quality. This is illustrated in the following waterfall diagramme.

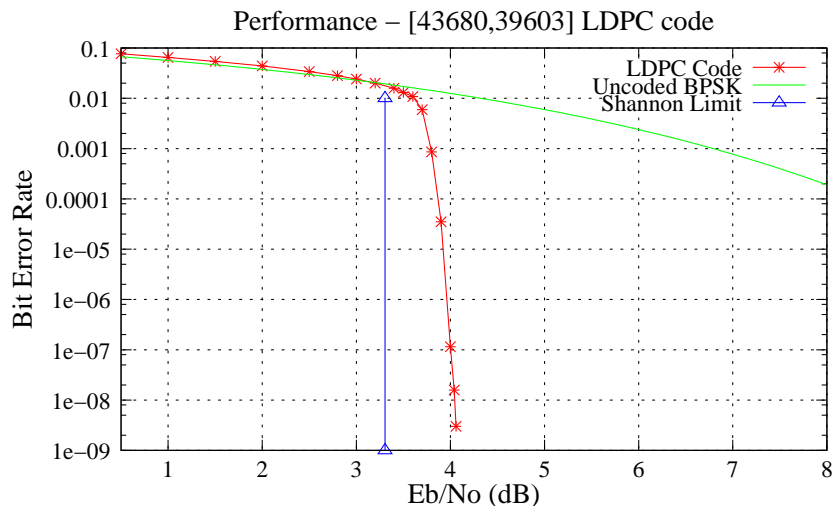


Fig. 1: Error performance of a long high-rate LDPC code derived from an inversive space of order $q = 2$.

In this type of diagramme three curves are set in relation to each other. One is a vertical line which is usually denoted by the *Shannon limit*. It marks the point on the x -axis beyond which the benefit of coding might be expected. A further curve of very moderate slope marks the behaviour of a communication system in which no coding is performed at all. The signal-to-noise ratio (SNR) of the channel is essentially mapped in a one-to-one manner to the bit-error-probability in the received word. This curve is a semilogarithmic plot of the identity and might be considered as a gauging curve. The third

curve, the actual waterfall, is measuring the performance of the given code under the assumption of what is called an AWGN channel. The steeper this curve, and the closer it is to the vertical Shannon limit, the better. In the example of the above diagramme, the horizontal distance between this curve and the Shannon limit is less than 0.76 dB and hence hints at a very good performance of the code in question.

Many LDPC codes that have been constructed exhibit what is called an *error floor* in the engineering language. This is a region in which the error probability does not approach 0 as quickly at high SNRs as it does at low SNRs. For example, in a plot of the above type, one definition of an error floor is that there is a region where the slope of the plotted curve approaches that of a horizontal line.

LDPC codes that have been constructed according to the recipe explained above have been simulated over the AWGN channel in extensive simulations supported by the foregoing ICHEC grant on the Walton cluster. They have not shown any error floor phenomenon in the range of SNRs for which simulations were feasible. Particularly for high-rate examples as the one illustrated above this suggests their use in modern and next-generation storage applications (magnetic channel). To prove their appropriateness for this type of applications we need to evaluate their performance in further simulations down to a bit-error probability of less than 10^{-13} . The goal of the project is hence, to run a very high number of parallel simulation tasks for times ranging from several hours to several days.

III CASE FOR SUPPORT

Given an LDPC code there is generally no mathematical/theoretical way to predict its performance. Although there are some parameters that can indicate how it will perform the only way to decide if it is a good code is doing an extensive simulation of its behavior in a communication channel.

The performance for LDPC codes are generally expressed in the form of the above-mentioned graph (waterfall diagramme) representing the Bit Error Ratio (BER) as function of the Signal to Noise Ratio (SNR). Obtaining this graph requires simulation of the decoding of a message using the code under test, the channel model and the decoder at the receiver.

To test a code and a decoder at a specific noise level, the simulation procedure is as follows: a message vector is randomly generated, the message is encoded using the generator matrix to produce a codeword, which is modulated to create the desired signal strength. Noise is generated randomly and added to the codeword to simulate the channel. The received corrupted codeword is then decoded using a decoding algorithm. This procedure is repeated a number of times (called trials) using different codewords for the same conditions, that is, the same code, the same noise level and the same decoder. The results are recorded and an analysis of the performance is undertaken.

For a given code this process of coding, transmitting and decoding must be repeated a massive number of time. An idea of how many times the process is repeated can be understood if we consider that for a good code the performance graph must be drawn for values of BER that range from 1 to 10^{-7} , 10^{-8} and—particularly in the project at hand—even less, namely down to 10^{-13} . This wide range of value is required because particular features of codes like the error floor mentioned earlier appear only around these values.

It is important to note that values of BER around 10^{-8} means that the decoder makes an error (decides for a message different that the transmitted one) on average every one hundred thousand messages received. Moreover the variance associated with every point of the graph depends on how many errors have been found for that particular value of SNR, so to have an realistic and precise representation of the performance of a code billions of simulations must be carry out.

Therefore, the direct simulation process described above is a lengthy and computationally intensive method of testing a code and a decoder at a specific SNR. It is easy to understand how the speed of the simulation process can significantly increased with the use of computer cluster where the computations are distributed among different nodes. The particular challenge for the codes at hand is to show that an error floor occurs only at very low bit-error probabilities. This will prove the codes usefulness for industrial applications that have also been envisaged in an ongoing patent application with UCD-Nova.

IV JUSTIFICATION, METHODOLOGY AND WORKPLAN

The package developed for the characterization of LDPC codes consists of a collection of small C routines and the GNU-Octave package that manage the creation of the code plus the core program that performs the necessary simulations. This core program is written in C with use of the MPI library for splitting the job through the various nodes. The root node takes care of sending instructions, such as, how many trials to do and for which values of SNR to the nodes. Moreover it receives the information from the nodes and decides if a sufficient number of trials have been executed for the performance graph to be accurate.

The codes have been tested during the foregoing grant support in the Walton Cluster and work smoothly. During the previous simulation phase also time testing has been carried out. The results of these suggest that a moderate length code takes up to 512 cpu-hours and the a longer ones up to 2048 cpu-hours. The simulations for lower length codes typically run on 64 nodes for 4 hours and for long codes typically on 256 nodes for up to 8 hours. The rate of the simulations should be almost continuous since the codes have already been generated and are ready to be simulated. The goal of the project is hence, to run a very high number of parallel simulation tasks (512 nodes and more) for times ranging from several hours to several days.

The memory requirement of the program is quite low and the simulations will run exclusively on the Walton Cluster. Also the storage requirement is moderate, even when considering the storage of the long codes eight Gigabytes should be more than sufficient.

REFERENCES

- [1] R.G.Gallager, "Low density parity check codes," 1963. PhD Thesis, MIT press, Cambridge ,MA, 1963.
- [2] D.J.C.MacKay and R.M.Neal, "Good codes base on very sparse matrices," in *Cryptography and Coding, 5th IMA Conference*, pp. 100–111, Dec. 1995.
- [3] G. A. Margulis, "Explicit constructions of graphs without short cycles and low density codes," *Combinatorica*, vol. 2, no. 1, pp. 71–78, 1982.
- [4] J. Rosenthal and P. O. Vontobel, "Constructions of LDPC codes using Ramanujan graphs and ideas from Margulis," in *Proc. of the 38-th Allerton Conference on Communication, Control, and Computing*, pp. 248–257, 2000.
- [5] J. Rosenthal and P. O. Vontobel, "Constructions of regular and irregular LDPC codes using Ramanujan graphs and ideas from Margulis," in *Proceedings of the 2001 IEEE International Symposium on Information Theory*, p. 4, 2001.
- [6] J. D. Lafferty and D. N. Rockmore, "Codes and iterative decoding on algebraic expander graphs," in *Proceedings of the ISIT-A 2000*, 2000.
- [7] J. Bond, S. Hui, and H. Schmidt, "Linear-congruence constructions of low-density parity-check codes," in *Codes, systems, and graphical models (Minneapolis, MN, 1999)*, vol. 123 of *IMA Vol. Math. Appl.*, pp. 83–100, New York: Springer, 2001.
- [8] M. O'Sullivan, M. Greferath, and R. Smarandache, "Construction of ldpc codes from affine permutation matrices," in *Proceedings of the 40th Annual Allerton Conference on Communication, Control and Computing*, 2002.
- [9] P. O. Vontobel and R. M. Tanner, "Construction of codes based on finite generalized quadrangles for iterative decoding," in *Proc. IEEE Intern. Symp. on Inform. Theory, Washington, D.C., USA*, p. 223, 2001.
- [10] Y. Kou, S. Lin, and M. P. C. Fossorier, "Low density parity check codes based on finite geometries: A rediscovery and new results." Preprint, 2001.
- [11] K. E. Mellinger, "LDPC codes from triangle-free line sets," *Des. Codes Cryptogr.*, vol. 32, no. 1-3, pp. 341–350, 2004.