Irregular Codes from Regular Graphs

Pascal O. Vontobel[1]
ISI / ETH Zentrum
CH-8092 Zürich, Switzerland
e-mail: `vontobel@isi.ee.ethz.ch`

Hans-Andrea Loeliger
ISI / ETH Zentrum
CH-8092 Zürich, Switzerland
e-mail: `loeliger@isi.ee.ethz.ch`

Abstract — **It has been recognized that appropriatly chosen irregular low-density parity-check codes (LDPCCs) have a remarkable performance advantage compared to regular LDPCCs and that their threshold is much better [1]. Starting with regular graphs, we construct codes consisting of more sophisticated subcodes (compared to simple XORs) and show that performance gains can be achieved by optimizing the proportions of the high-degree bit nodes and the different subcodes being used.**

I. Introduction and Motivation

Let "strict-sense LDPC codes" (SS-LDPCCs) be LDPCCs where all check nodes are simple XORs and let "wide-sense LDPC codes" (WS-LDPCCs) consist of more complicated subcodes. In 1962, Gallager defined the class of SS-LDPCCs where additionally all bit nodes have the same degree and all check nodes have the same degree (regular SS-LDPCCs). Later, Tanner generalized this idea to WS-LDPCCs. In the second half of the last decade, it was recognized that a sensitive choice of bit and check degrees of SS-LDPCCs can improve the performance quite markedly (see, e.g., [1]); the tool of density evolution for calculating performance thresholds proved to be very valuable. The aim of this paper is to show that irregularity helps also WS-LDPCCs; we will design codes that have optimized proportions of the high-degree bit nodes and the subcodes being used, respectively.

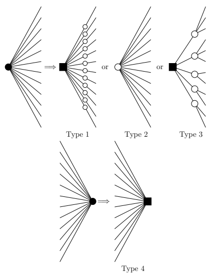
A second motivation was the fact that most graph families studied by mathematicians are regular (i.e. the degrees of all vertices are the same) or biregular. One approach to construct a regular SS-LDPCC is to take a biregular bipartite graph where the nodes of each class have the same degree. Other approaches to construct regular and irregular LDPC codes modify, and/or extend, and/or prune the underlying graph while maintaining at least some desirable properties of the graph (see, e.g., [2]). In this paper, we try to keep the whole structure of the underlying graph as far as possible.

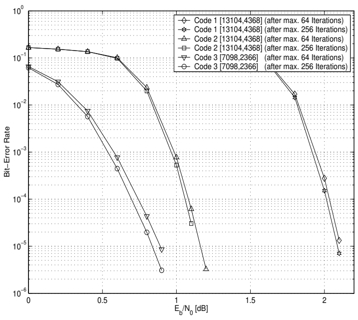
II. Description of the Codes and Examples

To be explicit, we give an exemplary construction starting from a regular bipartite graph, where we call the vertices of the two classes the left and the right vertices, respectively. To get a factor graph representing a code, we apply the following procedure: the left vertices and the outgoing edges are replaced by parts of a factor graph as shown in the top-left corner of the figure, whereas the right vertices and outgoing edges are replaced by parts of a factor graph as shown in the bottom-left corner of the figure (shown here for a 12-regular underlying graph; one can come up with other replacements). Because of their large-girth property, we will start with Ramanujan graphs $\mathrm{RG}_{\mathrm{LPS}}(q,p)$ as defined in [3], where the explicit construction for any distinct odd primes q and p was given in [4].

The bit-error-rate plot shows simulation results of three different rate-1/3 codes over the binary-input AWGN channel with sum-product decoding. Code 1 is a randomly generated $(4,6)$-regular SS-LDPCC with four-cycles removed (this family has threshold 1.67 dB). Code 2 is a WS-LDPCC based on a 12-regular (bipartite) $\mathrm{RG}_{\mathrm{LPS}}(13,11)$ where as subcodes we have uniformly taken $[12,8,3]$ subcodes (similar in spirit to a code in [5]). Finally, Code 3 is a WS-LDPCC based on the same (bipartite) $\mathrm{RG}_{\mathrm{LPS}}(13,11)$. But now, we used the following replacements and subcodes: on the left half, 50% type-1 with a $[12,8,3]$ subcode, 50% type-2 with degree-12 bit-node; on the right half we use type-4 replacements where 10% have a $[12,8,3]$ subcode, 40% have a $[12,9,2]$ subcode, 23% have a $[12,10,2]$ subcode, and 27% have a $[12,11,2]$ subcode (i.e., a simple XOR). (Certainly, with more time and effort, one can find better proportions.) Observe that Code 3 is by 46% shorter than the Codes 1 and 2.

We conclude with a few general remarks. The subcodes must be placed so as to avoid small-weight codewords; one can optimize such codes by calculating thresholds; codes with a better threshold usually also have a lower encoding complexity; if the underlying graph is a Cayley graph one can place the subcodes according to the generators.





References

[1] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of Capacity-Approaching Irregular Low-Density Parity-Check Codes," *IEEE Trans. on Inform. Theory*, vol. 47, no. 2, pp. 619–637, 2001.

[2] J. Rosenthal and P. O. Vontobel, "Construction of Regular and Irregular LDPC Codes Using Ramanujan Graphs and Ideas from Margulis," in *Proc. IEEE Intern. Symp. on Inform. Theory*, (Washington, D.C., USA), p. 4, June 24–29 2001.

[3] A. Lubotzky, R. Phillips, and P. Sarnak, "Ramanujan Graphs," *Combinatorica*, vol. 8, no. 3, pp. 261–277, 1988.

[4] P. O. Vontobel, "On the Construction of Turbo Code Interleavers Based on Graphs with Large Girth," in *Proc. IEEE Int. Conf. Communications*, (New York, NY, USA), Apr. 28-May 2 2002.

[5] J. Lafferty and D. Rockmore, "Codes and Iterative Decoding on Algebraic Expander Graphs," in *Proc. of ISITA 2000*, (Hawaii, USA), 2000.

[1] This work was supported by ETH-Grant TH-16./99-3.