Proto-Graph Based LDPC Code Ensembles

Jeremy Thorpe

June 12, 2003

Abstract

We introduce a new class of LDPC code ensembles based on a proto-graph. While this class is not as general as the multi-edge-type framework developed by Richardson, it has desirable properties in code structure and density evolution analysis. Proto-graphs can be effectively optimized by the simulated annealing algorithm.

1 Introduction

1.1 Historical Perspective

Low-Density Parity Check (LDPC) Codes have emerged as one of the top contenders for near-channel-capacity error correction.

1.2 Engineering Overview

an LDPC code over F_2 is described by its (sparse) parity-check matrix H. matrices can be efficiently represented by a bipartite (Tanner) graph G.

2 Proto-Graph LDPC Codes

2.1 Overview

A proto-graph $G^p = (V^p, C^p, E^p)$ is a finite-sized Tanner graph which serves as a blueprint for constructing Tanner graphs of arbitrary size. The construction generates a graph $G = (V, C, E)$ that is t times bigger than $G^{(p)}$. Specifically, we let $V = V^p \times \{1, 2, ...t\}$, $C = C^p \times \{1, 2, ...t\}$, and $E = \{(v, c) : e_{v^p, c^p} \in E^p, i \in \{1, 2, ...t\}\}$, where π_e is an arbitrary permutation for each edge e. If we let the $\pi_e = I$, (I is the identity permutation), then G is composed of t disjoint copies of G^p.

2.2 Local Non-randomness

This construction has the property that if a neighborhood within ratius r of a variable node v in G is tree-like, its structure is completely determined by $G^{(p)}$ ($v^{(p)}$ need not be tree-like within radius r). By contrast, in a code generated

from an irregular, or multi-edge-type degree profile, the structure of such a neighborhood is in general random.

2.3 Proto-Graph Ensembles

Having given a description of an arbitrary graph G based on (G^p, t), we can define an ensemble of codes by assigning a probability to each. The ensemble that we shall consider is that obtained by generating π_e independently and uniformly over all permutations of length t.

2.4 Density Evolution

Density evolution (DE) analysis can easily be carried out on a proto-graph to determine whether or not decoding will yield small bit error probability on a large graph. As in the DE analysis of other ensembles, the standard assumption that the $\overrightarrow{0}$ codeword has been transmitted is made. The messages which are defined in the message passing algorithm are treated as random variables and their distribution is computed.

For proto-graph based codes, a distribution $P_{\overrightarrow{e^p}}^{(i)}$ on the messages from the c^p to v^p in and a distribution $P_{\overleftarrow{e^p}}^{(i)}$ is computed for each iteration i for every edge e^p in E^p. Because of the local non-randomness property, the message distribution $P_{\overrightarrow{e^p}}^{(i)}$ computed for the protograph edge e^p in G^p will be the exact distribution on *any* corresponding edge in G.

2.5 Density Evolution Threshold

For a given channel, the message densities will either evolve such that at least one message for any variable will have arbitrary small probability of error, or they will not. For a channel parameterized by parameter θ (where channel fidelity increases with θ), the threshold θ^* is the minimum value such that the message densities evolve in this way.

3 Optimization

We define an algorithm which optimizes a proto-graph for DE threshold via a randomized iterative algorithm called Simulated Annealing (SA). This algorithm requires a search space S, a random permutation $p : S \to S$, an energy function $E : S \to \mathbb{R}$, and a temperature profile $T(i)$. SA The algorithm begins with an arbitrary solution $s^{(0)}$, and at each iteration i applies a random perturbation $p\left(s^{(i)}\right)$ to generate a new solution $s^{(i)\prime}$. If $E(s') < E(s)$ then $s^{(i+1)} = s^{(i)\prime}$, otherwise $s^{(i+1)} = s^{(i)\prime}$ w.p. $e^{-\frac{E(s')-E(s)}{T(i)}}$ and $s^{(i)}$ otherwise.

3.1 Optimization Space

The algorithm optimizes over proto-graphs with a fixed number n^p of transmitted variable nodes, a fixed number u^p of untransmitted variable nodes, and a fixed number k^p of check nodes. Since the rate of the code can be computed in terms of these parameters as $R = \frac{n^p - u^p}{k^p}$, the rate of the code is constant throughout the optimization space.

3.1.1 Untransmitted Nodes

Variable nodes that are defined in the code but not transmitted over the channel can improve the performance of proto-graph based LDPC codes. These variables are decoded by the decoder (in the same way as if the channel had yielded an "erasure"), and thus contribute to the rate of the code.

3.2 Energy and Temperature

As an ideal, we choose the Energy Function E to be the threshold computed by DE. However, this is slightly impractical since we will have to run DE at least a few times in order to determine the threshold with any accuracy. A more practical solution is to let $E(s)$ be the number of iterations required to drive the output error probability below some small constant ε.

We choose a temperature profile that goes from pretty high to pretty low.

3.3 Approximation to Density Evolution

The algorithm would still not be practical if it weren't for a single-parameter approximation to DE. Here we use the reciprocal-channel approximation introduced by Chung.

4 Results

There are some results. I'll have them tomorrow.

5 Implementation

Code can be found at http://www.systems.caltech.edu/~jeremy/research/software/LDPC.zip. This code compiles Microsoft Visual Studio .NET and runs under the .NET framework.

6 Reference

References

[1] "The cpacity of low-density parity check codes under message-passing decoding," IEEE Trans. Ifrom. Theory, vol. 47, pp 599-618, Feb. 2000.

[2] "Concentrate!"

[3] "On the Capacity of Low-Density Parity-Check Codes"