Random-Access LDPC Codes

Eshed Ram Yuval Cassuto
Andrew and Erna Viterbi Department of Electrical Engineering
Technion – Israel Institute of Technology, Haifa 32000, Israel
E-mails: {s6eshdr@campus, ycassuto@ee}.technion.ac.il

Abstract—New types of LDPC codes motivated by practical storage applications are presented. LDPCCL codes (suffix ‘L’ stands for locality) can be decoded locally at the level of sub-blocks that are much smaller than the full code block, thus providing fast random access to the coded information. The same code can also be decoded globally using the entire code block (as usual), for increased data reliability. We present constructions of LDPCCL and spatially-coupled (SC) LDPCCL codes that enable random access, and we exemplify their benefits over ordinary LDPC codes.

1. INTRODUCTION

The growing demand for denser storage devices yield an increase in their error rates. Since every storage decoding failure implies data losses, storage application require very strong ECCs. Low-density parity-check (LDPC) codes and failure implies data losses, storage application require very increase in their error rates. Since every storage decoding symbols are decoded. For a wider and deeper study of random-access codes, please refer to [6] and [7].

In random-access codes (a.k.a multi-block codes [1]), code-blocks of length \( N \) are partitioned into \( M \) sub-blocks with \( n \) coded symbols each (i.e., \( N = Mn \)). Each sub-block is a codeword of some code, and can be locally decoded, independent from other sub-block, to extract \( k \) information symbols. Local decoding is fast and induce error-correcting capabilities which are lower than the harsh requirements. In case of catastrophic error events, the whole code-block is accessed to provide the data-reliability needed, and \( K = Mk \) information symbols are decoded. For a wider and deeper study of random-access LDPC codes, please refer to [6] and [7].

2. LDPCCL Codes

Every LDPC code can be represented by a bipartite graph, called a Tanner graph, with nodes partitioned to variable nodes (VNs) and check nodes (CNs). We define an LDPCCL code of length \( N = Mn \) through a two-sided Tanner graph. In this graph, the VNs are divided to \( M \) disjoint sets of size \( n \) each, and the CNs are divided into two sets: local CNs and joint CNs. The graph construction is constrained such that each local CN is connected only to VNs that are in the same sub-block; this constraint ensures sub-block decoding. The joint CN connections have no constraints. We denote by \( \lambda_{L,i} \) (resp. \( \rho_{L,i} \)) the fraction of local edges connected to a VN (resp CN) with local-degree (resp. degree) \( i \). We call \( \{ \lambda_{L,i}, \rho_{L,i} \} \) local-degree distributions. The joint-degree distributions \( \{ \lambda_{J,i}, \rho_{J,i} \} \) are defined similarly but with an important difference: we allow some VNs to have joint degrees 0 or 1. This turns out to be necessary to increase the code rate. We use \( P_0 \) to denote the fraction of VN that have a joint degree zero. The parameters \( M, n, \lambda_{L,i}, \rho_{L,i}, P_0 \) define a family of LDPCCL codes (an LDPCCL ensemble), with a design rate given by \( R = 1 - \int \frac{\lambda_{L,i}}{\lambda_{L,i} + \rho_{L,i}} \rho_{L,i} \rho_{J,i} (1 - P_0) \). Here we write \( \lambda_L \) (or any other degree distribution) as a notation for the generating polynomial \( \lambda_L(x) = \sum_i \lambda_{L,i} x^{i-1} \).

We suggest a decoding algorithm that operate in two modes: local mode and global mode. In the local mode, the decoder tries to decode a sub-block of length \( n \) using the corresponding local sub-graph. If the decoder meets a failure criterion, it enters the global mode where it decodes the entire code block using the complete two-sided Tanner graph.

Figure 1 compares the BER of three rate-half codes over the additive Gaussian channel with the variance \( \sigma \) as a free parameter: 1) an ordinary LDPC code with degree distributions \( \lambda(x) = 0.27684x + 0.28342x^2 + 0.43074x^8, \rho(x) = 0.01568x^5 + 0.85244x^6 + 0.13188x^7 \), and block length \( n = 4KB \); 2) an ordinary LDPC code with same degree distributions, but shorter block length \( n = 1KB \); 3) a random-access LDPCCL code with \( M = 4 \) sub-blocks of length \( n = 1KB \) each, local degree distributions \( \lambda_L(x) = 0.1575x + 0.3429x^2 + 0.0363x^5 + 0.059x^6 + 0.279x^8 + 0.1253x^9, \rho_L(x) = 0.8260x^{34} + 0.1345x^{35} + 0.0087x^{70} + 0.0302x^{71} \), joint degree distributions \( \lambda_J, \rho_J \) which are equal to those of the first two codes, and \( P_0 = 0.22 \). The asymptotic theoretical threshold for all three codes is \( \sigma^* = 0.9497 \). The random-access LDPCCL code also enables sub-block decoding with an asymptotic threshold \( \sigma^*_L = 0.5 \). The plot exemplifies the benefits of LDPCCL codes: in one hand, the LDPCCL code provides local access to 1KB sub-blocks unlike the LDPC 4KB code, and on the other hand, the LDPCCL global data-reliability is better than the LDPC 1KB code.

3. SC-LDPCCL Codes

SC-LDPC codes [2] are LDPC codes in which the sparse parity-check matrix has a band-diagonal structure. SC-LDPC
codes were shown to have many desired properties. For example, threshold saturation [3], linear-growth of minimum distance [4] and linear-growth of minimal trapping sets of typical codes from the ensemble [5].

One way to construct a SC-LDPC code is via protographs. A protograph is a small bipartite graph \( G = (V \cup C, E) \), where \( V \), \( C \) and \( E \) are the sets of VN's, CN's and edges, respectively. A Tanner graph is generated from a protograph \( G \) by a lifting operation ("copy-and-permute") specified by a parameter \( L \) (see [4]). Let \( 1^l \times r \) be an all-ones bi-adjacency matrix with \( l \) rows and \( r \) columns representing an \((l, r)\)-regular protograph. An \((l, r)\)-regular SC-LDPC protograph is constructed by diagonally placing copies of \((B_1; B_2)\), where \( B_1, B_2 \) are non-negative \( l \times r \) matrices such that \( B_1 + B_2 = 1^l \times r \) (the results readily extend to coupling more matrices). For example, the spatially-coupled \((3, 6)\) protograph is generated by \( B_1 = \begin{pmatrix} v_1 & v_2; v_3 \end{pmatrix}, \) where \( v_1 = (1, 1, 0, 0, 0, 0), v_2 = (1, 1, 1, 1, 0, 0), v_3 = (1, 1, 1, 1, 1, 1) \).

To the best of our knowledge, existing constructions of SC-LDPC codes do not enable sub-block access (for example, the \((l, r)\) SC-LDPC ensemble from [4, Definition 3] with \( l = \text{gcd}(l, r) \)). This motivates a construction of \((l, r)\)-regular random-access SC-LDPC codes [7]:

**Construction 1.** Let \( t \in \{1, 2, \ldots, l - 2\} \), and let \( A_1 \) be the \( t \times r \) matrix illustrated in Figure 2 (a), where \( 1 \) and \( 0 \) are length-\( \lfloor \frac{r}{t+1} \rfloor \) all-one vector and length-\( \lfloor \frac{r}{t+1} \rfloor \) all-zero vector, respectively. Let \( A_2 \) be an all-ones \((l-t) \times r \) matrix. We build the \((l, r, t)\) protograph \( M \) copies of \((B_1; B_2)\) on the diagonal, where \( B_1 = (\underbrace{A_1; A_1}_t \overbrace{1^l}^{t}) \) and \( B_2 = 1^l \times r - B_1 \).

We call the parameter \( 1 \leq t \leq l-1 \) the coupling parameter. It serves as a design tool that controls the trade-off between the local and global decoding capabilities: small values of \( t \) implies high local noise resilience, and large values of \( t \) leads to high global data protection. Figure 2 (b) illustrates the \((l = 3, r = 6, t = 1)\) SC-LDPC protograph with \( M = 3 \) SBs. Each sub-block in this protograph can be decoded independently to the others.

SC-LDPC codes can be decoded in a mode in between local and global decoding, which we call semi-global (SG) decoding. SG decoding has a substantial complexity advantage over global decoding with a very small cost in data-protection. Consider a SC-LDPC protograph with \( M \) SBs, and assume that the user wants to extract SB \( m \in [1 : M] \). We call SB \( m \) the *target*. In SG decoding, the decoder uses \( d \) helper SBs to decode the target in two phases: the *helper phase*, and the *target phase*. In the former, helper SBs are decoded locally using information from other previously-decoded helper SBs, and in the latter, the target SB \( m \) is decoded using information from its neighboring helper SBs. Figure 3 exemplifies SG decoding with \( d = 4 \) helper SBs. The helper phase consists of decoding helpers \( m-2 \) and \( m+2 \) locally, and decoding helpers \( m-1 \) and \( m+1 \) using the information from helpers \( m-2 \) and \( m+2 \), respectively. In the target phase, SB \( m \) is decoded using information from both SB \( m-1 \) and \( m+1 \).

**REFERENCES**


