Construction of LDPC codes

Telecommunications Laboratory
Alex Balatsoukas-Stimming

Technical University of Crete

July 1, 2009

Outline

- Regular Codes
  1. Gallager Codes
  2. Quasi-Cyclic Codes
  3. Array Codes
  4. Random Codes
- Irregular Codes
  1. Modified Array Codes
  2. Poisson (93p)
  3. Sub-Poisson (93a)
  4. Moderately Super-Poisson (93x)
  5. Very Super-Poisson (93y)
  6. Fast Encoding Versions
- References

Regular Codes

Gallager Codes (1/4)

- Original construction used by Gallager in 1963.
- To construct a parity-check matrix $\mathbf{H}$ with column weight w_c and row weight w_r, we first construct a sub-matrix $\mathbf{H}_1$ containing a single 1 in each column and w_r 1s in each row.
- The i-th row contains 1s in columns $(i-1)w_r + 1$ to w_r.
- The other $w_c - 1$ submatrices are random permutations of $\mathbf{H}_1$.

Gallager Codes (2/4)

- The final parity-check matrix is obtained by concatenating the w_c submatrices:

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_1 \\ \mathbf{H}_2 \\ \vdots \\ \mathbf{H}_{w_c} \end{bmatrix}$$

- Block length $= nw_r$

  Parity symbols $= nw_c$

  Design rate $= \frac{nw_r - nw_c}{nw_r} = 1 - \frac{w_c}{w_r}$

Gallager Codes (3/4)

- Another way to construct Gallager codes is through superposition of permutation matrices.
- The superposed matrices are generated at random subject to the constraint that no two non-zero entries coincide.

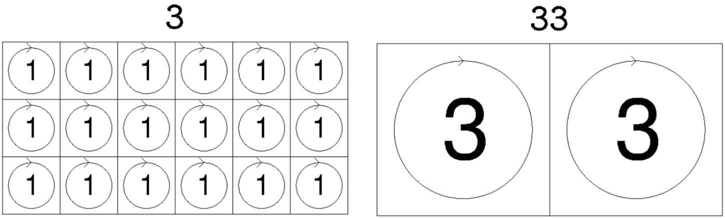

Figure: Integers denote the number of superposed permutation matrices.

- The above constructions create a rate-$\frac{1}{2}$ parity-check matrix of column weight 3 and row weight 6.

Gallager Codes (4/4)

\+ Easy construction.

\+ Good performance.

\- No structure for quick encoding.

\- Design rate may not be the actual rate, as **H** is not guaranteed to be full rank.

\- No guarantee that small cycles are not present.

Quasi-Cyclic Codes (1/3)

- QC-LDPC codes have a parity-check matrix which consists of square blocks which are either full-rank circulants, or zero matrices.
- Circulant matrix used is:

$$\mathbf{P} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ 1 & 0 & 0 & \dots & 0 \end{bmatrix}_{L \times L}$$

- Powers of the circulant matrix (i.e. right shifts of the identity matrix) and the zero matrix (denoted $\mathbf{P}^{\infty}$) are used to construct $\mathbf{H}$.

Quasi-Cyclic Codes (2/3)

- The $mL \times nL$ parity-check matrix is constructed as follows:

$$\mathbf{H} = \begin{bmatrix} \mathbf{P}^{a_{11}} & \mathbf{P}^{a_{12}} & \dots & \mathbf{P}^{a_{1n}} \\ \mathbf{P}^{a_{21}} & \mathbf{P}^{a_{22}} & \dots & \mathbf{P}^{a_{2n}} \\ \vdots & \vdots & \dots & \vdots \\ \mathbf{P}^{a_{m1}} & \mathbf{P}^{a_{m2}} & \dots & \mathbf{P}^{a_{mn}} \end{bmatrix}$$

- where $a_{ij} \in \{1, 2, \dots, L-1, \infty\}$
- Block size = nL

  Parity checks = mL

  Design rate = $\frac{nL - mL}{nL} = \frac{n-m}{n}$
- If all blocks are non-zero matrices, the code is regular with row weight n and column weight m, else, it is irregular.

Quasi-Cyclic Codes (3/3)

+ Low hardware cost encoding with feedback shift registers with spatial complexity linearly proportional to mL.

+ Low memory requirement for storing **H** due to structure.

- Design rate may not be the actual rate, as **H** is not guaranteed to be full rank.

- No guarantee that small cycles are not present.

Array Codes (1/2)

- For a prime $L = q$ ($\mathbf{P}$ is defined as in QC-LDPC codes) and the integers $j \leq k \leq q$ we create the matrix:

$$\mathbf{H} = \begin{bmatrix} \mathbf{I} & \mathbf{I} & \dots & \mathbf{I} & \dots & \mathbf{I} \\ \mathbf{I} & \mathbf{P}^1 & \dots & \mathbf{P}^{j-1} & \dots & \mathbf{P}^{k-1} \\ \mathbf{I} & \mathbf{P}^2 & \dots & \mathbf{P}^{2(j-1)} & \dots & \mathbf{P}^{2(k-1)} \\ \vdots & \vdots & & \vdots & & \vdots \\ \mathbf{I} & \mathbf{P}^{(j-1)} & \dots & \mathbf{P}^{(j-1)(j-1)} & \dots & \mathbf{P}^{(j-1)(k-1)} \end{bmatrix}$$

- Block length = kq

  Parity checks = jq

  Design rate $= \frac{kq-jq}{kL} = \frac{k-j}{k}$

Array Codes (2/2)

+ For $j > 3$, no length-4 cycles exist.
+ Design rate equals actual rate, since $\mathbf{H}$ is full rank.

- No structure for quick encoding.
- Worse performance than Gallager codes.

Random Codes (1/2)

- For a regular code, $nw_c = mw_r$ must hold.
- We can construct a rate-$\frac{n-m}{n}$ code by following these steps:
- Begin with the zero matrix $\mathbf{H}_0$
  1. At step i, choose a random $m \times 1$ column of weight w_c, which is not already being used in $\mathbf{H}_{i-1}$ or rejected in previous steps, and add it to $\mathbf{H}_{i-1}$.
  2. Check whether the added column has more than one 1-component in common with any column in $\mathbf{H}_{i-1}$. If not, go to next step. Else reject the column and go back to step 1.
  3. If all rows have weight less than w_r, save $\mathbf{H}_i$ and continue to next round. Else, reject the column and go back to step 1.
- When n columns have been added, stop and return the finished matrix $\mathbf{H_n}$.

Random Codes (2/2)

+ No length-4 cycles exist (due to step 2 of the process).
+ Code ensemble has been proven to have good performance.

- Design rate may not be the actual rate, as **H** is not guaranteed to be full rank.
- No structure for quick encoding.
- Construction can become computationally very expensive, especially for large column weights and code lengths.

Irregular Codes

Modified Array Codes (1/2)

- For a prime $L = q$ ($\mathbf{P}$ is defined as in QC-LDPC codes) and the integers $j \leq k \leq q$ we create the matrix:

$$\mathbf{H} = \begin{bmatrix} \mathbf{I} & \mathbf{I} & \mathbf{I} & \dots & \mathbf{I} & \dots & \mathbf{I} \\ \mathbf{0} & \mathbf{I} & \mathbf{P}^1 & \dots & \mathbf{P}^{j-2} & \dots & \mathbf{P}^{k-2} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \dots & \mathbf{P}^{2(j-3)} & \dots & \mathbf{P}^{2(k-3)} \\ \vdots & \vdots & \vdots & & \vdots & & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{I} & \dots & \mathbf{P}^{(j-1)(k-j)} \end{bmatrix}$$

- Block length $=$ kq

  Parity checks $=$ jq

  Design rate $= \frac{kq-jq}{kL} = \frac{k-j}{k}$

Modified Array Codes (2/2)

+ For $j > 3$, no length-4 cycles exist.

+ Design rate equals actual rate, since $\mathbf{H}$ is full rank.

+ Structure allows for quick encoding using Richardson and Urbanke's method with time complexity linearly proportional to block length.

- Worse performance than Gallager codes.

Poisson Construction (1/2)

- All examples below use the following profile:

$$\lambda(x) = \frac{11}{12}x^2 + \frac{1}{12}x^8$$

$$\rho(x) = x^6$$

- The variable nodes connected to 9 check nodes will be called "elite bits".
- In the Poisson construction, most checks connect to one or two elite bits, but a fraction of them will connect to more than two, and some will connect to none.

Poisson Construction (2/2)

- The construction is as follows:

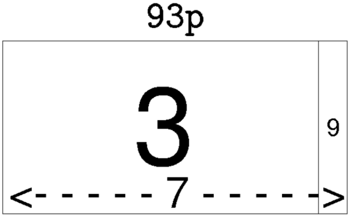

Figure: Horizontal line emphasizes constant row weight.

Sub-Poisson Construction

- This construction allocates exactly one or two elite bits to each check.
- The construction is as follows:

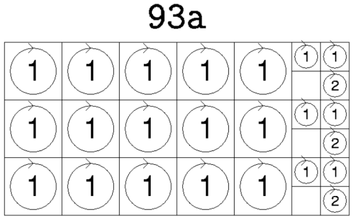

Figure: Integers denote the number of superposed permutation matrices.

Moderately Super-Poisson Construction

- In this construction, one third of the checks are connected to one elite bit, one third are connected to none, one sixth are connected to 3, and one sixth are connected to 4.
- The construction is as follows:

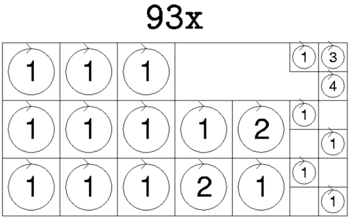

Figure: Integers denote the number of superposed permutation matrices.

Very Super-Poisson Construction

- In this construction, one third of the checks are connected to four elite bits, one third are connected to one, ad one third are connected to none.
- The construction is as follows:

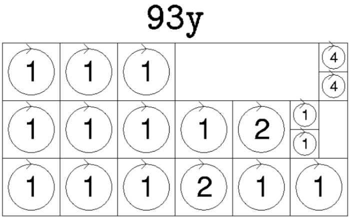

Figure: Integers denote the number of superposed permutation matrices.

x-Poisson Constructions

+ Better performance than regular codes even though we have no guarantee for the absence of small cycles.

+ Decoding in fewer iterations.

- No structure for quick encoding.

- Each decoding round requires more operations, so total decoding time is about the same as for regular codes.

Fast Encoding

- Some fast encoding constructions based on the above follow:

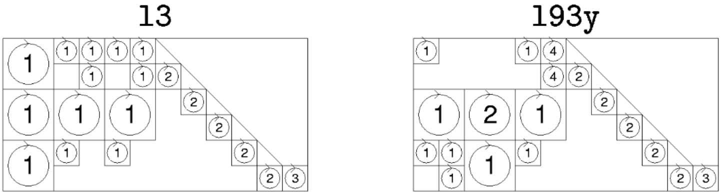

Figure: Integers denote the number of superposed permutation matrices. Diagonal line denotes line of 1s.

- Due to the approximate upper-triangular form of the matrices, fast encoding is possible with complexity $O(n+g^2)$, where g denotes the size of the non-triangular part of the matrix.
- No significant performance loss.

References

(1) Shu Lin, Daniel J. Costello Jr., "Error Control Coding", Second Edition, 2004

(2) Tamoaki Ohtsuki, "LDPC Codes in Communications and Broadcasting", IEICE Trans. Commun., Vol. E90-B, No. 3 March 2007

(3) Seho Myung, Kyeongcheol Yang, and Jaeyoel Kim, "Quasi-Cyclic LDPC Codes for Fast Encoding", IEEE Trans. Inf. Theory, Vol. 51, No. 8, August 2005

(4) David J. C. MacKay, Simon T. Wilson, and Matthew C. Davey, "Comparison of Constructions of Irregular Gallager Codes", IEEE Trans. Commun., Vol. 47, No. 10, pp. 1449-1454, October 1999