Low-density parity-check (LDPC) codes

- Performance similar to turbo codes
- Do not require long interleaver to achieve good performance
- Better block error performance
- Error floor occurs at lower BER
- Decoding is not based on trellises
- Iterative decoding
  - More iterations than turbo decoding
  - Simpler operations in each iteration step
- Not covered by patents!

Introduction

- A binary full rank parity check matrix $\mathbf{H}_{J\times n}$ is given
- The null space of $\mathbf{H} = \{\mathbf{v} \in GF(2)^n : \mathbf{v}\mathbf{H}^T = \mathbf{0}\}$ is a binary linear code
- Regular LDPC code: The null space of a matrix **H** which has the following properties:
  - Its J rows are not necessarily linearly independent
  - There are ρ 1s in each row
  - There are γ 1s in each column
  - No pair of columns has more than one common 1
  - ρ and γ are small compared to the number of rows (J) and columns (n) in **H**
- Density $r = \rho/n = \gamma/J$
- Irregular LDPC code: The number of 1s in the rows/columns may vary

Example

- $J = n = 15$
- $\rho = \gamma = 4$

$$\mathbf{H} = \begin{bmatrix}
0&0&0&0&0&0&0&1&1&0&1&0&0&0&1\\
1&0&0&0&0&0&0&0&1&1&0&1&0&0&0\\
0&1&0&0&0&0&0&0&0&1&1&0&1&0&0\\
0&0&1&0&0&0&0&0&0&0&1&1&0&1&0\\
0&0&0&1&0&0&0&0&0&0&0&1&1&0&1\\
1&0&0&0&1&0&0&0&0&0&0&0&1&1&0\\
0&1&0&0&0&1&0&0&0&0&0&0&0&1&1\\
1&0&1&0&0&0&1&0&0&0&0&0&0&0&1\\
1&1&0&1&0&0&0&1&0&0&0&0&0&0&0\\
0&1&1&0&1&0&0&0&1&0&0&0&0&0&0\\
0&0&1&1&0&1&0&0&0&1&0&0&0&0&0\\
0&0&0&1&1&0&1&0&0&0&1&0&0&0&0\\
0&0&0&0&1&1&0&1&0&0&0&1&0&0&0\\
0&0&0&0&0&1&1&0&1&0&0&0&1&0&0\\
0&0&0&0&0&0&1&1&0&1&0&0&0&1&0
\end{bmatrix}$$

Gallager's LDPC codes

- Select ρ and γ
- Form a $k\gamma \times k\rho$ matrix $\mathbf{H}$ consisting of γ submatrices $\mathbf{H}_1$, ..., $\mathbf{H}_\gamma$, each of dimension $k \times k\rho$
- $\mathbf{H}_1$ is a matrix such that its ith row has its ρ 1s confined to columns $(i\text{-}1)\rho+1$ through $i\rho$, $i = 1, \ldots, k$
- The other submatrices are column permutations of $\mathbf{H}_1$
- The actual column permutations give the properties of the code

Gallager's LDPC codes: Example

- Select ρ = 4, γ = 3, and k = 5
- Column permutations for $\mathbf{H}_2$ and $\mathbf{H}_3$ are selected so that no two columns have more than one 1 in common
- This is a (20,7,6) code

$$\mathbf{H} = \left[\begin{array}{cccccccccccccccccccc}
1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&1&1&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1\\
\hline
1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&0&0&0&0\\
0&1&0&0&0&1&0&0&0&1&0&0&0&0&0&0&1&0&0&0\\
0&0&1&0&0&0&1&0&0&0&0&0&0&1&0&0&0&1&0&0\\
0&0&0&1&0&0&0&0&0&0&1&0&0&0&1&0&0&0&1&0\\
0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1\\
\hline
1&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&1&0&0\\
0&1&0&0&0&0&1&0&0&0&1&0&0&0&0&1&0&0&0&0\\
0&0&1&0&0&0&0&1&0&0&0&0&1&0&0&0&0&0&1&0\\
0&0&0&1&0&0&0&0&1&0&0&0&0&1&0&0&1&0&0&0\\
0&0&0&0&1&0&0&0&0&1&0&0&0&0&1&0&0&0&0&1
\end{array}\right]$$

Rows and parity checks

- Each row of **H** forms a parity check
- Let A_l be the set of rows $\{\mathbf{h}_1^{(l)}, \ldots, \mathbf{h}_\gamma^{(l)}\}$ that have a one in position l
- Since no pair of columns has more than one common 1, any code bit other than bit l is checked by at most one of the rows in A_l
- Thus, there are γ rows orthogonal on l, and any error pattern of weight $\leq$ floor(γ/2) can be decoded by one-step majority logic (MLG) decoding. However, (especially when γ is small) this gives poor performance
- Gallager proposed two iterative decoding algorithms, one hard decision algorithm and one soft decision algorithm

Brief overview of graph theoretic notations

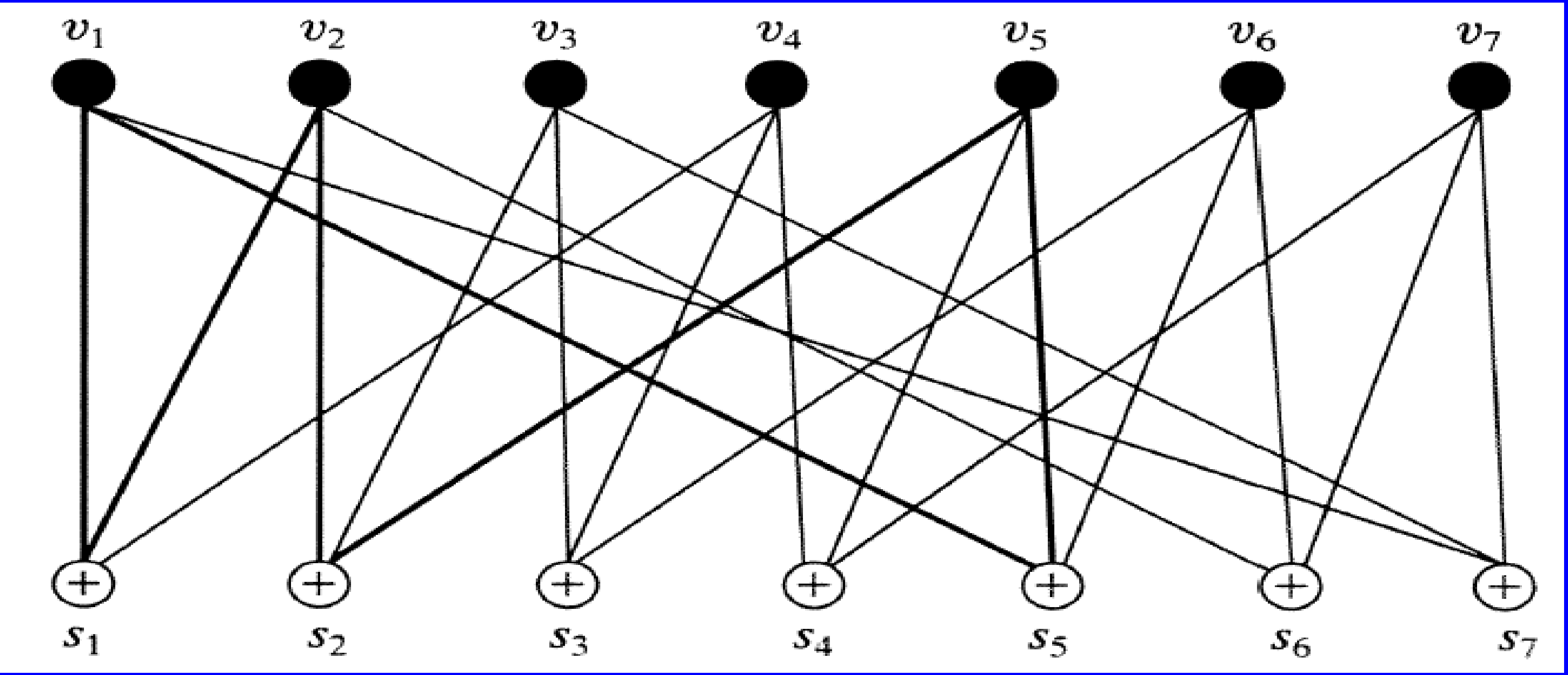
- An undirected graph is denoted by $G(V,E)$
- The degree of a vertex/node is the number of edges adjacent to it
- Two edges are adjacent if they are connected to the same vertex
- A path in a graph is an alternating sequence of vertices and edges (starting and ending in a vertex), where all vertices are distinct except for maybe the first and the last
- A tree is a graph without cycles
- If there is a path between any pair of vertices, the graph is connected
- If the set of vertices can be partitioned into two disjoint parts; so that each edge goes from a vertex in one part to a vertex in the other, then the graph is bipartite

Graphical description of LDPC codes

- Tanner graph: A bipartite graph $G(V,E)$ where $V = V_1 \cup V_2$ and
  - V_1 is a set of n code bit nodes $v_0, \ldots, v_{n-1}$, each representing one of the n code bits
  - V_2 is a set of J check nodes $s_0, \ldots, s_{J-1}$, each representing one of the J parity checks
  - There is an edge between $v \in V_1$ and $s \in V_2$ iff. the code bit corresponding to v is contained in the parity check corresponding to s
- No cycles of length 4 in the Tanner graph of an LDPC code

Example

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$



Decoding of LDPC codes

- Several decoding techniques are available. Listed in order of increasing complexity (and improving performance):
  - Majority logic (MLG) decoding
  - Bit flipping (BF) decoding
  - Weighted bit flipping (WBF) decoding
  - Iterative decoding based on belief propagation (IDBP), also known as the sum-product algorithm (SPA)
  - A posteriori probability (APP) decoding

Decoding notations

- Assume BPSK modulation and an AWGN channel
- Codeword: $\mathbf{v} = (v_0, \ldots, v_{n\text{-}1})$
- Channel output: $\mathbf{y} = (y_0, \ldots, y_{n\text{-}1})$
- Hard decisions: $\mathbf{z} = (z_0, \ldots, z_{n\text{-}1})$ where $z_j = 1$ (0) if $y_j >$ (<) 0
- Rows of $\mathbf{H}$: $\mathbf{h}_1, \ldots, \mathbf{h}_J$
- Syndrome: $\mathbf{s} = (s_1, \ldots, s_J) = \mathbf{z} \cdot \mathbf{H}^T$ where $s_j = \mathbf{z} \cdot \mathbf{h}_j = \sum_{i=0\ldots n\text{-}1} z_i h_{j,i}$
- Parity failure: At least one syndrome bit is nonzero
- Error pattern: $\mathbf{e} = \mathbf{v} + \mathbf{z}$
- Syndrome: $\mathbf{s} = (s_1, \ldots, s_J) = \mathbf{e} \cdot \mathbf{H}^T$ where $s_j = \mathbf{e} \cdot \mathbf{h}_j = \sum_{i=0\ldots n\text{-}1} e_i h_{j,i}$

MLG decoding

- One-step MLG decoding is described in Chapter 8*
- Let A_l be the set of rows $\{\mathbf{h}_1^{(l)}, \ldots, \mathbf{h}_\gamma^{(l)}\}$ that have a one in position l
- A set of γ parity check sums orthogonal on code bit l is

  $S_l = \{s_j^{(l)} = \mathbf{z} \cdot \mathbf{h}_j^{(l)} = \mathbf{e} \cdot \mathbf{h}_j^{(l)} = \sum_{i=0\ldots n-1} e_i h_{j,i}^{(l)} : \mathbf{h}_j^{(l)} \in A_l\}$
- If a majority of the parity check sums on bit l are satisfied; conclude that $e_l = 0$. Otherwise, conclude that $e_l = 1$
- This decoding is guaranteed to result in a correct decoding if at most floor(γ/2) errors occur
- Problem: γ is by assumption a small number

BF decoding

- Introduced by Gallager in 1961
- First form the hard decision vector $\mathbf{z} = (z_0, \ldots, z_{n\text{-}1})$
- Compute the syndrome vector
  $\mathbf{s} = (s_1, \ldots, s_J) = \mathbf{z} \cdot \mathbf{H}^T$ where $s_j = \mathbf{z} \cdot \mathbf{h}_j = \sum_{i=0\ldots n\text{-}1} z_i h_{j,i}$
- For each bit l, find f_l = the number of failed parity checks for bit l
- Let S = the set of bits for which f_l is large
- Flip the values of the bits in S
- Repeat until all parity check sums are satisfied, or a maximum number of iterations have been reached

BF decoding (cont.)

- Let S = the set of bits for which f_l is large
- For example:
  - S = the set of bits for which f_l exceeds some threshold δ that depends on the code parameters ρ, γ, the minimum distance, and the channel SNR. If decoding fails, try again with a reduced value of δ
  - Simple alternative: S = the set of bits for which f_l is maximum
- Comments:
  - If few errors occur, decoding should commence in a few iterations
  - On a poor channel, the number of iterations should be allowed to grow large
  - Improvement: Adaptive thresholds

Weighted MLG and BF decoding

- Use weighting to include soft decision/reliability information in the decoding decision
- First form the hard decision vector $\mathbf{z} = (z_0, \ldots, z_{n-1})$
- Weight: $|y_j|^{(l)}_{\min} = \min\{|y_i|: 0 \le i \le n\text{-}1,\ \mathrm{h}_{j,i} = 1,\ \mathbf{h}_j \in A_l\}$
  - The reliability of the jth parity check on l
- Weighted MLG decoding: Hard decision on
  - $E_l = \sum_{s_j^{(l)} \in S_l} (2s_j^{(l)} - 1)\cdot|y_j|^{(l)}_{\min}$
  - where $S_l = \{s_j^{(l)} = \mathbf{z} \cdot \mathbf{h}_j^{(l)} = \mathbf{e} \cdot \mathbf{h}_j^{(l)} = \sum_{i=0\ldots n-1} e_i h_{j,i}^{(l)}: \mathbf{h}_j^{(l)} \in A_l\}$
- Weighted BF decoding:
- Compute the syndrome vector $\mathbf{s} = (s_1, \ldots, s_J) = \mathbf{z} \cdot \mathbf{H}^T$
- For each bit l, compute E_l
- Flip the value of the bit for which E_l is the largest
- Repeat until all parity checks are satisfied, or a maximum number of iterations have been reached

SPA decoding

- Iterative decoding based on belief propagation
- Symbol-by-symbol SISO: The goal is to compute
  - $P(v_l |\, \mathbf{y})$
  - $L(v_l) = \log\,(P(v_l = 1|\, \mathbf{y})/P(v_l = 0|\, \mathbf{y}))$
- A priori probabilities: $p_l^x = P(v_l = x),\, x \in \{0,1\}$
- $q_{j,l}^{x,(i)} = P(v_l = x|$ check sums $\in A_l \setminus \{\mathbf{h}_j\}$ at the ith iteration$\}$
- $\mathbf{h}_j = (h_{j,0}, h_{j,1}, \ldots, h_{j,n\text{-}1})$; support of $\mathbf{h}_j = B(\mathbf{h}_j) = \{l\!: h_{j,l} = 1\}$
- $\sigma_{j,l}^{x,(i)} = \Sigma_{\{v_t:\, t \in B(\mathbf{h}_j)\setminus\{l\}\}} P(s_j = 0|\, v_l = x, \{v_t\!: t \in B(\mathbf{h}_j)\setminus\{l\}\}) \cdot \prod_{t \in B(\mathbf{h}_j)\setminus\{l\}} q_{j,t}^{v_t,(i)}$
- $q_{j,l}^{x,(i+1)} = \alpha_{j,l}^{(i+1)}\, p_l^x \prod_{\mathbf{h}_t \in A_l \setminus \{\mathbf{h}_j\}} \sigma_{t,l}^{x,(i)}$
  - $\alpha_{j,l}^{(i+1)}$ chosen so that $q_{j,l}^{0,(i+1)} + q_{j,l}^{1,(i+1)} = 1$
- $P^{(i)}(v_l = x|\, \mathbf{y}) = \beta_l^{(i)}\, p_l^x \prod_{\mathbf{h}_j \in A_l} \sigma_{j,l}^{x,(i\text{-}1)}$
  - $\beta_l^{(i)}$ chosen so that $P^{(i)}(v_l = 0|\, \mathbf{y}) + P^{(i)}(v_l = 1|\, \mathbf{y}) = 1$

SPA decoding on the Tanner graph

- $q_{j,l}^{x,(i)} = P(v_l = x \mid \text{check sums} \in A_l \setminus \{\mathbf{h}_j\} \text{ at the } i\text{th iteration}\}$
- $\sigma_{j,l}^{x,(i)} = \sum_{\{v_t:\, t \in B(\mathbf{h}_j)\setminus\{l\}\}} P(s_j = 0 \mid v_l = x, \{v_t:\, t \in B(\mathbf{h}_j)\setminus\{l\}\}) \cdot \prod_{t \in B(\mathbf{h}_j)\setminus\{l\}} q_{j,t}^{v_t,(i)}$
- $q_{j,l}^{x,(i+1)} = \alpha_{j,l}^{(i+1)}\, p_l^x \prod_{\mathbf{h}_t \in A_l \setminus \{\mathbf{h}_j\}} \sigma_{t,l}^{x,(i)}$