Information Theory and Channel Coding

Prof. Rodrigo C. de Lamare
CETUC, PUC-Rio, Brazil
delamare@cetuc.puc-rio.br

VIII. Low-density parity-check (LDPC) codes

A. Introduction

B. Encoding

C. Structure and design of LDPC codes

D. Decoding

A. Introduction

- LDPC codes are linear block codes that were invented by Robert Gallager in his PhD thesis at MIT in 1960.

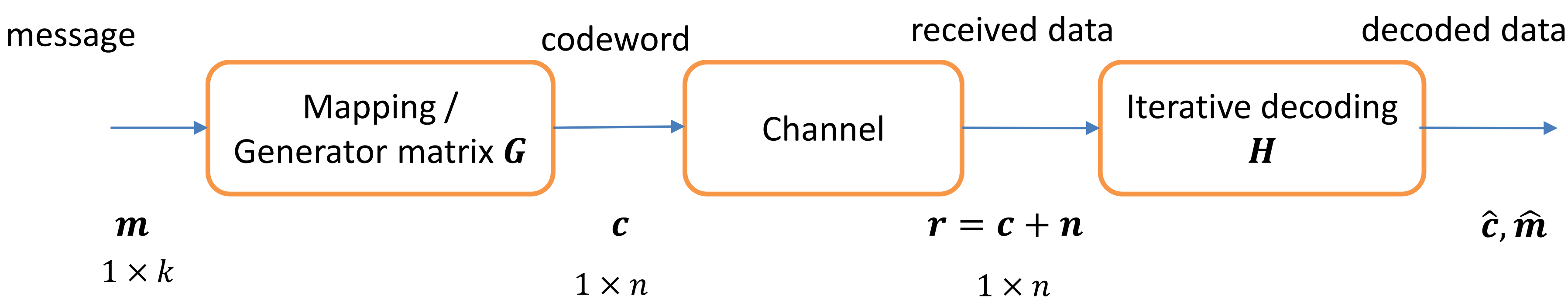
- LDPC codes can approach Shannon's theoretical limit by using sparse parity-check matrices and message passing decoding.

- The basic idea consists of designing a parity-check matrix with few ones and many zeros, which would facilitate decoding.

- Decoding for such sparse structures in LDPC codes is carried out by message passing, which is easy to implement.

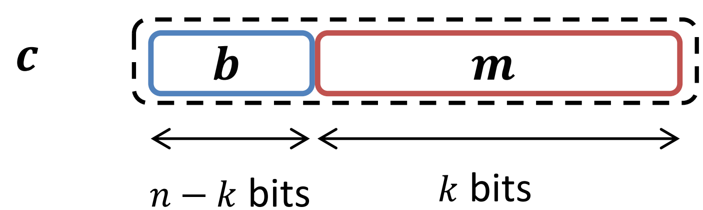
Robert G. Gallager (1963). *Low Density Parity Check Codes* (PDF). Monograph, M.I.T. Press

- Let us consider an LDPC system with the following block diagram.



- LDPC codes are denoted by $C(n, k)$ or $C(n, t_c, t_r)$ where k is the length of $\boldsymbol{m}$ in bits, n is the length of $\boldsymbol{c}$ in bits, t_c and t_r are the numbers of ones per column and row of $\boldsymbol{H}$, respectively.

- The structure of an LDPC code $\boldsymbol{c}$ in systematic form is illustrated by



- LDPC codes are often specified by
  - The parity-check matrix $\boldsymbol{H}$
  - The block length n
  - The number of ones in each column of $\boldsymbol{H}$, t_c.
  - The number of ones in each row of $\boldsymbol{H}$, t_r.

- Unlike other channel codes that focus on the design of the generator matrix $\boldsymbol{G}$, LDPC codes first design the parity-check matrix $\boldsymbol{H}$.

- The code rate of LDPC codes is given by

$$R = 1 - \frac{t_c}{t_r} = \frac{k}{n}$$

- The relation of the code rate can be obtained as follows.

- Consider ρ as the density of ones in $\boldsymbol{H}$, then if we set

$$t_c = \rho \underbrace{(n-k)}_{\text{parity bits}}$$

and

$$t_r = \rho n$$

- We then divide t_c by t_r to obtain

$$\frac{t_c}{t_r} = \frac{\rho(n-k)}{\rho n} = 1 - \frac{k}{n} = 1 - R \;\;\rightarrow\;\; R = 1 - \frac{t_c}{t_r}$$

- Note that the parity-check matrix $\boldsymbol{H}$ is often not systematic and requires a procedure in its design to ensure a sytematic structure for it.

B. Encoding

- Let us now describe the encoding procedure of LDPC codes and focus on the systematic form for the sake of simplicity.

- The parity bits can be described in matrix form by

$$\boldsymbol{b} = \boldsymbol{m}\,\boldsymbol{P},$$

where $\boldsymbol{P} = \begin{bmatrix} p_{0,0} & \cdots & p_{0,n-k-1} \\ \vdots & \ddots & \vdots \\ p_{k-1,0} & \cdots & p_{k-1,n-k-1} \end{bmatrix} \in \mathrm{F}^{k\times(n-k)}$ is the parity matrix.

- The codeword is described by

$$\begin{aligned} \boldsymbol{c} &= [\boldsymbol{b} \mid \boldsymbol{m}] \\ &= [\boldsymbol{mP} \mid \boldsymbol{m}] = \boldsymbol{m}\,\underbrace{[\boldsymbol{P} \mid \boldsymbol{I}_k]}_{\boldsymbol{G}} = \boldsymbol{mG}, \end{aligned}$$

where $\boldsymbol{G} \in \mathrm{F}^{k\times n}$ is the generator matrix.

- The parity-check matrix $\boldsymbol{H} \in \mathrm{F}^{n-k\times n}$ of LDPC codes is key for both design and decoding, and should be sparse and structured as follows:

$$\boldsymbol{H} = [\boldsymbol{I}_{n-k} |\, \boldsymbol{P}^T] \in \mathrm{F}^{n-k\times n}$$
$$= \Big[\underbrace{\boldsymbol{H}_1^T}_{\mathrm{n-k}\,\times\,\mathrm{n-k}} \quad | \quad \underbrace{\boldsymbol{H}_2^T}_{\mathrm{n-k}\,\times\,\mathrm{k}}\Big]$$

- With the above partitioning and by imposing the constraint $\boldsymbol{cH}^T = \boldsymbol{0}$, we have

$$\boldsymbol{cH}^T = [\boldsymbol{b} \,|\, \boldsymbol{m}] \begin{bmatrix} \boldsymbol{H}_1 \\ \text{—} \\ \boldsymbol{H}_2 \end{bmatrix} = \boldsymbol{0}$$
$$\boldsymbol{bH}_1 + \boldsymbol{mH}_2 = \boldsymbol{0}$$
$$\boldsymbol{mPH}_1 + \boldsymbol{mH}_2 = \boldsymbol{0}$$
$$\boldsymbol{m}(\boldsymbol{PH}_1 + \boldsymbol{H}_2) = \boldsymbol{0}$$

- The non trivial solution is given by

$$\mathbf{P} = \boldsymbol{H}_2\boldsymbol{H}_1^{-1}$$

- Therefore, the generator matrix $\boldsymbol{G}$ for LDPC codes is given by

$$\begin{aligned}\boldsymbol{G} &= [\boldsymbol{P} \mid \boldsymbol{I}_k] \\ &= [\boldsymbol{H}_2\boldsymbol{H}_1^{-1} \mid \boldsymbol{I}_k] \in \mathrm{F}^{k\times n}\end{aligned}$$

- In the design of the generator matrix $\boldsymbol{G}$ for LDPC codes we need to take care with $\boldsymbol{H}_1$ so that it is non singular and its inverse exists.

- In the design of the parity-check matrix $\boldsymbol{H}$ we often take care with the existence of the inverse of $\boldsymbol{H}_1$ before we proceed to obtain $\boldsymbol{G}$.

Example 1

Consider an LDPC code with $n = 10, t_c = 3$ and $t_r = 5$ and the following parity-check matrix:

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

a) Compute the rate of the LDPC code
b) Compute the partitioned matrices $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$
c) Determine the generator matrix $\boldsymbol{G}$
d) Compute the codeword for $\boldsymbol{m} = [1\ 0\ 0 1]$

Solution:

a) The rate of the LDPC code is given by

$$R = 1 - \frac{t_c}{t_r} = 1 - \frac{3}{5} = 0.4$$

b) The partitioned matrices are

$$\boldsymbol{H}_1^T = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \text{and } \boldsymbol{H}_2^T = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

c) The generator matrix can be obtained by

$$
\begin{aligned}
\boldsymbol{G} &= [\boldsymbol{P} \mid \boldsymbol{I}_k] \\
&= [\boldsymbol{H}_2\boldsymbol{H}_1^{-1} \mid \boldsymbol{I}_k]
\end{aligned}
$$

Substituting the values of $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$, we obtain

$$
\boldsymbol{G} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}
$$

d) The codeword produced by $\boldsymbol{m} = [1\ 0\ 0 1]$ is given by

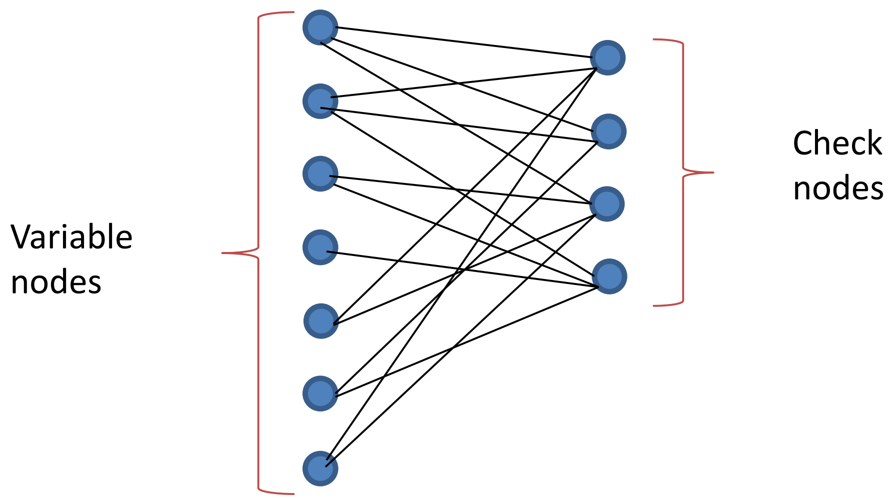
$$
\boldsymbol{c} = \boldsymbol{m}\boldsymbol{G} = [\ 1\ 1\ 0\ 0\ 0\ 0\ \mid 1\ 0\ 0 1]
$$

C. Structure and design of LDPC codes

- LDPC codes can be described by bipartite graphs known as Tanner graphs.
- The basic idea is to employ a bipartite graph to describe the parity-check matrix of an LDPC code.
- According to the Tanner graph the variable nodes correspond to the elements of the codeword.
- The check nodes correspond to the parity-check constraints of the LDPC code.

R. M. Tanner, “A recursive approach to low-complexity codes,” IEEE Trans. Inform. Theory, vol. 27, pp. 53S547, Sept. 1981

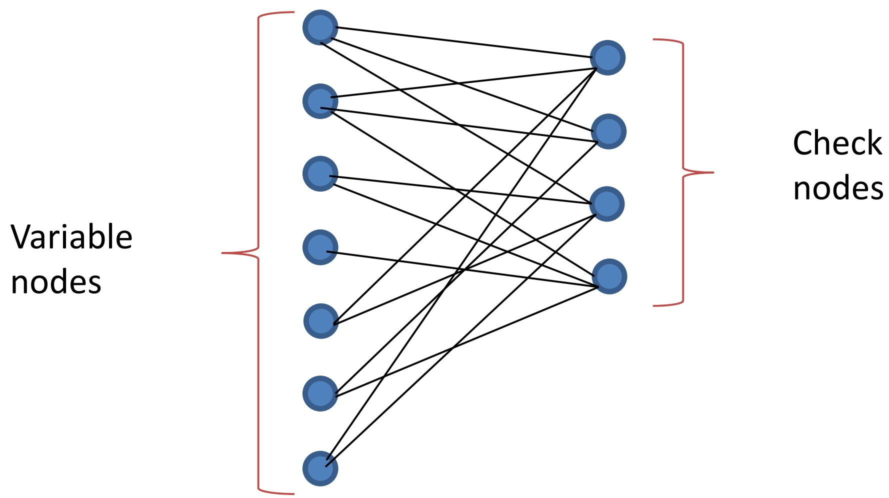
- A Tanner graph can be illustrated by



- A check node j is connected to variable node i when the element of $\boldsymbol{H}$ is a one.

- The $m = n - k$ rows of $\boldsymbol{H}$ specify the m check node connections.

- The n columns of $\boldsymbol{H}$ specify the n variable node connections.



- Cycles: a path comprising ν edges which closes back to itself.

- Girth (γ): it is the minimum cycle length of the Tanner graph.

- Note: the shortest possible girth is 4 -> $\boldsymbol{H} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}$

Example 2

Consider a (n, t_c, t_r) LDPC code with the following parity-check matrix

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

a) Determine the parameters of the code such as rate, t_c and t_r.

b) Draw the bipartite graph, also known as as the Tanner graph.

c) Compute the girth of the code.

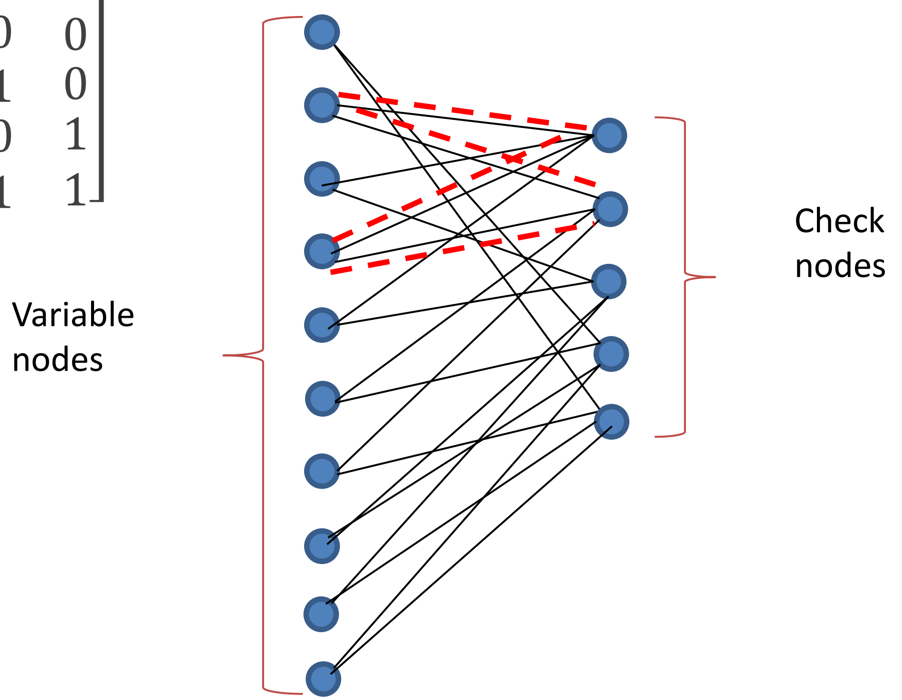
Solution:

a) The parameters of the code are

$t_c = 2$, $t_r = 4$ and $R = 1 - \frac{t_c}{t_r} = 0.5$

b) The Tanner graph corresponding to this parity-check matrix is

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



c) The girth of the graph is $\gamma = 4$

Design strategies

- The design of LDPC codes is based on the generation of a parity-check matrix $\boldsymbol{H}$ with a given specification.

- In particular, the parameters (n, k, t_c, t_r) used in the specification play a major role in the performance of LDPC codes.

- LDPC codes can also be either regular or irregular depending on the patterns of ones in $\boldsymbol{H}$.

- Regular codes: the number of ones in each column or row is constant

- Irregular codes: the number of ones in each column or row is variable.

- Specifically, in irregular codes t_c and t_r are functions of the column numbers -> use of degree distributions.

- The placement of ones is carried out via an optimization procedure known as density evolution.

- The degree distribution polynomials are:

    - The variable node polynomial $\lambda(x)$

    - The check node polynomial $\rho(x)$

- The variable node polynomial is

$$\lambda(x) = \sum_{d=1}^{d_v} \lambda_d x^{d-1},$$

where λ_d is the fraction of all edges connected to degree d variable nodes and d_v denotes the maximum node degree.

- The check node polynomial is

$$\rho(x) = \sum_{d=1}^{d_c} \rho_d x^{d-1},$$

where $\rho_d$is the fraction of all edges connected to degree d check nodes and d_c denotes the maximum node degree.

Example 3

Consider the design of LDPC codes based on degree polynomials.

a) Write down the polynomials $\lambda(x)$ and $\rho(x)$ for a regular code with $t_c = 2$ and $t_r = 4$.

b) Write down and discuss the meaning of an irregular code with $\lambda_1 = 0.00015$, $\lambda_2 = 0.30235$, $\lambda_3 = 0.27132$ and $\lambda_7 = 0{,}42618$, and $\rho_6 = 0.35559$ and $\rho_7 = 0.64445$.

Solution:

a) By noticing that $t_c = d_v = 2$ and $t_r = d_c = 4$, we have

$$\lambda(x) = \sum_{d=1}^{d_v} \lambda_d x^{d-1} = x$$

and

$$\rho(x) = \sum_{d=1}^{d_c} \rho_d x^{d-1} = x^3$$

b) By substituting $\lambda_1 = 0.00015$, $\lambda_2 = 0.30235$, $\lambda_3 = 0.27132$ and $\lambda_7 = 0{,}42618$, and $\rho_6 = 0.35559$ and $\rho_7 = 0.64445$ into the expressions of the degree polynomials, we obtain

$$\lambda(x) = \sum_{d=1}^{d_v} \lambda_d x^{d-1} = 0.00015 + 0.30235x + 0.27132x^2 + 0.42618x^6$$

and

$$\rho(x) = \sum_{d=1}^{d_c} \rho_d x^{d-1} = 0.35559x^5 + 0.64445x^6$$

This means that about 42.6% of edges are connected to degree 6 variable degrees, 27,1 % of edges are connected to degree 2 variable degrees and so on.

Gallager codes

a) The original design approach to LDPC codes is known as Gallager codes, which rely on

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{H}_1 \\ \boldsymbol{H}_2 \\ \vdots \\ \boldsymbol{H}_{t_c} \end{bmatrix}$$

The submatrices $\boldsymbol{H}_d$ are computed as follows.

For any μ and t_r greater than 1, each $\boldsymbol{H}_d$ has dimensions $\mu \times \mu t_r$ with row weight w_r and column weight 1.

The submatrix $\boldsymbol{H}_1$ has the form

For $i = 1,2, \dots, \mu$

The ith row contains all t_r ones in columns $(i-1)t_r$+1 to it_r

end

The remaining submatrices are permutations of $\boldsymbol{H}_1$ as illustrated by

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{H}_1 \\ \boldsymbol{H}_2 \\ \vdots \\ \boldsymbol{H}_{t_c} \end{bmatrix} = \begin{bmatrix} \boldsymbol{H}_1 \\ \boldsymbol{\phi}_1(\boldsymbol{H}_1) \\ \vdots \\ \boldsymbol{\phi}_{t_c-1}(\boldsymbol{H}_1) \end{bmatrix}$$

where $\boldsymbol{\phi}\,(.\,)$ is a permutation operation.

McKay codes

b) The design of McKay codes relies on the partitioning of the parity-check matrix

$$\boldsymbol{H} = \left[\boldsymbol{H}_1^T \,|\boldsymbol{H}_2^T\right]$$

The design algorithm for regular codes has the following steps:

1. $\boldsymbol{H}$ is generated randomly using columns with column weights t_c and uniform row weights.
2. $\boldsymbol{H}$ is constrained such that no columns have overlap greater than one.
3. $\boldsymbol{H}$ is constrained to avoid short cycles.
4. $\boldsymbol{H}$ is constructed such that $\boldsymbol{H}_1$ is invertible.

McKay codes lack structure for low-complexity encoding.

The encoding is performed as follows:

We obtain

$$\begin{aligned} \boldsymbol{G} &= [\boldsymbol{P} \mid \boldsymbol{I}_k] \\ &= [\boldsymbol{H}_2\boldsymbol{H}_1^{-1} \mid \boldsymbol{I}_k] \in \mathrm{F}^{k\times n} \end{aligned}$$

using Gauss-Jordan elimination.

Other designs

c) Irregular LDPC designs

- Optimization of $\lambda(x)$ and $\rho(x)$ using density evolution.
- Adjustment by other methods.

d) Progressive edge growth designs

- Optimization of connections or edges between variable and check nodes in a progressive fashion.

e) Repeat accummulate (RA) and irregular RA (IRA) designs

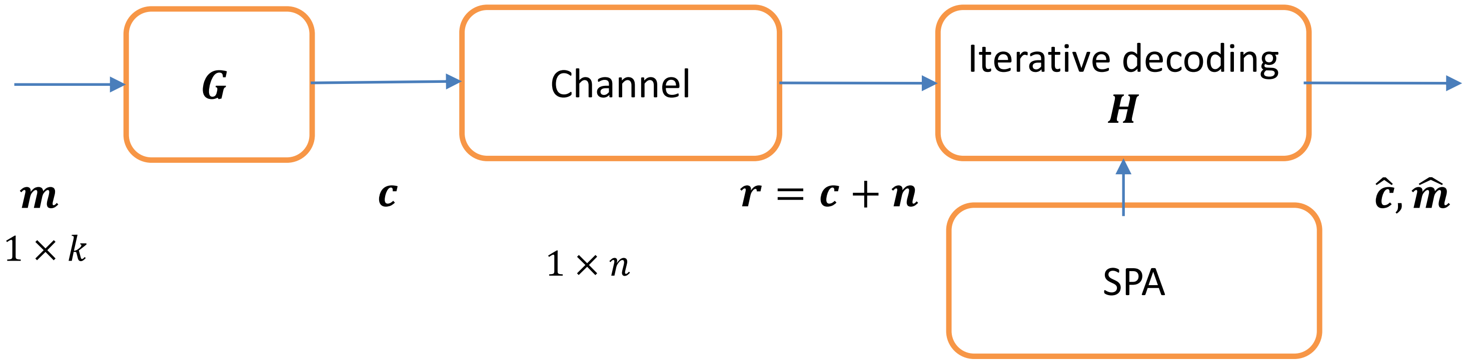
- Low-complexity designs that are competitive for low code rates.

f) Quasi-cyclic and finite geometry designs

- Low-complexity encoding -> reduction from $O(n^2)$ to $O(\mathrm{n}\log_{10} n)$

Xiao-Yu Hu, E. Eleftheriou and D. M. Arnold, "Regular and irregular progressive edge-growth tanner graphs," in IEEE Transactions on Information Theory, vol. 51, no. 1, pp. 386-398, Jan. 2005

D. Decoding



- Given the received data vector $\boldsymbol{r}$, the decoder must find the most probable $\hat{\boldsymbol{c}}$ that satisfies

$$\hat{\boldsymbol{c}}\boldsymbol{H}^T = \boldsymbol{0}$$

- According to the MAP decoding rule, we have the log likelihood ratios (LLRs)

$$l(c_j) = l(c_j|\boldsymbol{r}) = \log\left(\frac{P(c_j = +1|\boldsymbol{r})}{P(c_j = -1|\boldsymbol{r})}\right) \underset{<}{\geq} 0$$

- A message passing strategy known as sum-product algorithm (SPA) is employed to compute the LLRs.

- SPA has two alternating steps:
  - The horizontal step runs along the rows of $\boldsymbol{H}$.
  - The vertical step runs along the columns of $\boldsymbol{H}$.

- The variable or bit nodes are elements of $\boldsymbol{r}$

- The check nodes are the rows of $\boldsymbol{H}$.

- $\mathcal{J}(i)$ is the set of bits that participate in check node i.

- $\mathcal{J}(j)$ is the set of nodes in which variable node j participates.

- $\mathcal{J}(i)/j$ is the set $\mathcal{J}(i)$ that excludes variable node j.

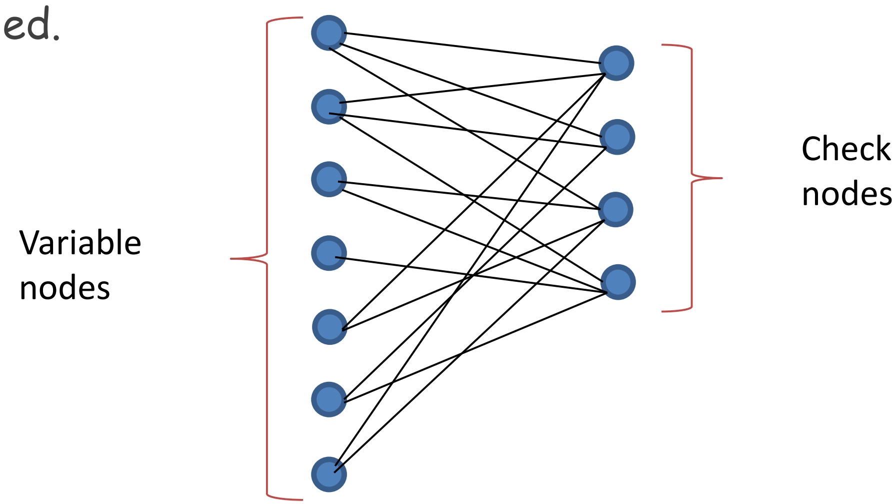
- $\mathcal{J}(j)/i$ is the set $\mathcal{J}(j)$ that excludes check node i.

The SPA decoding strategy is as follows:

- To exchange LLRs associated with the ones of $\boldsymbol{H}$ in an alternating way.

- We compute the LLR of check node i $l(q_{ij})$ (horizontal step) and send it to variable node j $l(r_{ij})$ $\rightarrow$ This checks the probability that bit j is $+1$ or -1.

- We compute the LLR of variable node j $l(r_{ij})$ (vertical step) and send it to check node i $l(q_{ij})$ $\rightarrow$ This checks the probability that check node i is satisfied.



SPA

- The SPA first considers the LLR

$$l(c_j) = l(c_j|\boldsymbol{r}) = \log\left(\frac{P(c_j = +1|\boldsymbol{r})}{P(c_j = -1|\boldsymbol{r})}\right),$$

where c_j is the jth element of the codeword $\boldsymbol{c}$.

- We compute the LLR of check node i and send it to variable node j as described by

$$l(r_{ij}) = 2\tanh^{-1}\left\{\prod_{i \in \mathcal{J}(j)/i} \tanh\left[\frac{1}{2} l(q_{ij})\right]\right\},$$

where $i = 1,2, \ldots, n-k$ and $j = 1, \ldots, n$.

- We then compute the LLR of the variable node j updated and send it to check node i as follows:

$$l(q_{ij}) = l(c_j) + \sum_{j \in \mathcal{J}(i)/j} l(r_{ij})$$

- At the end of each iteration, variable node j computes the total LLR given by

$$l(Q_j) = l(c_j) + \sum_{j \in \mathcal{J}(i)} l(r_{ij})$$

- The decisions about the bits are obtained by

$$\hat{c}_j = \begin{cases} 1, & \text{if } l(Q_j) \geq 0 \\ -1, & \text{otherwise} \end{cases}$$

Summary of SPA

Initialization: $l(r_{ij}) = l(c_j)$ (equal probabilities)

Goal: to compute $l(c_j) = l(c_j|\boldsymbol{r}) = \log\left(\frac{P(c_j=+1|\boldsymbol{r})}{P(c_j=-1|\boldsymbol{r})}\right)$

For each iteration, update

1. $l(r_{ij}) = 2\tanh^{-1}\left\{\prod_{i\,\in\,\mathcal{J}(j)/i} \tanh\left[\frac{1}{2} l(q_{ij})\right]\right\}$,

2. $l(q_{ij}) = l(c_j) + \sum_{j\,\in\,\mathcal{J}(i)/j} l(r_{ij})$

3. $l(Q_j) = l(c_j) + \sum_{j\,\in\,\mathcal{J}(i)} l(r_{ij})$

Stop if $\hat{\boldsymbol{c}}\boldsymbol{H}^T = \boldsymbol{0}$ or if the maximum number of iterations is reached

Make decision

$$\hat{c}_j = \begin{cases} 1, & \text{if } l(Q_j) \geq 0 \\ -1, & \text{otherwise} \end{cases}, j = 1,2,\dots,n$$

$$l(c_j) = l(c_j|\boldsymbol{r}) = \log\left(\frac{P(c_j=+1|\boldsymbol{r})}{P(c_j=-1|\boldsymbol{r})}\right),$$

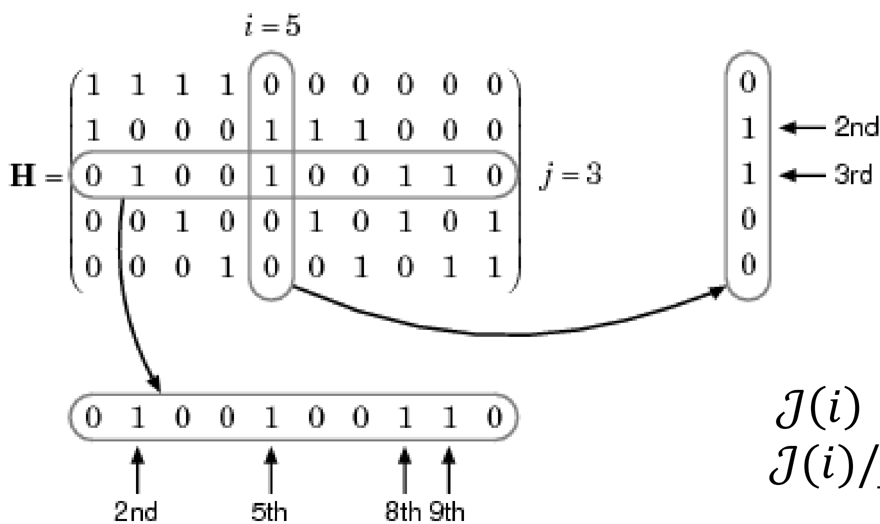
Example 4

Illustrate the SPA decoding of the following parity-check matrix

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

Solution:

For $i = 5$ and $j = 3$, the index sets would be



$\mathcal{J}(i) = \{2,3\}$

$\mathcal{J}(i)/j$={2}

$\mathcal{J}(j) = \{2,5,8,9\}$

$\mathcal{J}(j)/i$={2,8,9}

At the end of each iteration, $l(Q_j)$ provides an updated estimate of the a posteriori log-likelihood ratio for the transmitted bit .