LDPC (BITSYNC) PROJECT

NITHIN | AAYUSH | SAIRAM

INTRODUCTION TO LDPC CODES

- LDPC: Low-Density Parity-Check Codes
- Invented by Robert Gallager in the 1960s
- Inspired by limitations in Hamming Codes
- Sparse parity-check matrix allows scalable error correction
- Initially impractical due to decoding complexity; revived with modern computing

WHY LDPC?

- Why LDPC?
- Efficient error correction near Shannon limit
- Sparse matrices reduce computational load
- Applicable to large-scale data (e.g., satellite, 5G, storage)
- Better performance than traditional block codes

WHAT IS LDPC?

- The density of a source of random bits is the expected fraction of bits
- A source is sparse if its density is less than 0.5
- The overlap between 2 codechunks is the number of 1's common between them.

MATH BEHIND LDPC CODES – PART 2

- Low density parity check (LDPC) codes are codes specified by a parity check matrix H containing mostly 0's and only small number of 1's.
- Valid codewords satisfy: $H \times x^t = 0$ (MA2101)
- A regular LDPC is code of blocklength n with a m x n parity check matrix where each column contains a small fixed number, w_c >= 3 and the row contains w_r >=1.
- $w_c \geq 3$ ensures error detectability and propagation

MATH BEHIND LDPC CODES – PART 3

- Each parity check constraint involves w_r codebits and each codebit has w_c constraints.
- Low density implies that w_c <<m and w_r << n
- Always take care of H. w_c .n = w_r .m
- m >= n –k , R = 1 – (m/n). w_r > w_c.

TANNER GRAPH

- Bipartite graph : Can divided into 2 classes with no edge connecting the nodes of the same class.
- One of the class is the variable nodes and the other is the check nodes.
- A edge connects the variable node only if that bit is included in the parity check equation

GIRTHS AND CYCLES

- Cycle = closed path in Tanner graph
- Girth = length of smallest cycle
- Short cycles (e.g., 4) reduce decoding performance
- Design goals: maximize girth

- Let n be the transmitted block length of an info seq of length k and m is the number of parity check eqns.
- Construct mxn matrix with w_c 1's per col and w_r 1's per row.
- Divide the matrix into w_c (m/ w_c x n) matrices each containing single 1 in each col
- Now permutate the cols.

BLOCK CODES

1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1

3	3

Schematic Illustration of Regular Gallager Codes

Notation: An integer represents a number of permutation matrices superposed on the surrounding square.

Column Weight	Fraction of columns	Row weight	Fraction
3	1	6	1

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
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
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

Example of a regular low density code matrix; $n = 20$, $w_c = 3$, $w_r = 4$

- For an **irregular** low-density parity-check code the degrees of each set of nodes are chosen according to some distribution.
- A degree distribution $\gamma(x) = \sum_i \gamma_i x^{i-1}$ is simply a polynomial with nonnegative real coefficients satisfying $\gamma(1) = 1$.
- An irregular low-density code is a code of block-length N with a sparse parity check matrix where column distribution $\lambda(x)$ and row distribution $\rho(x)$ is respectively given by

$$\lambda(x) = \sum_{i>1} \lambda_i x^{i-1}$$

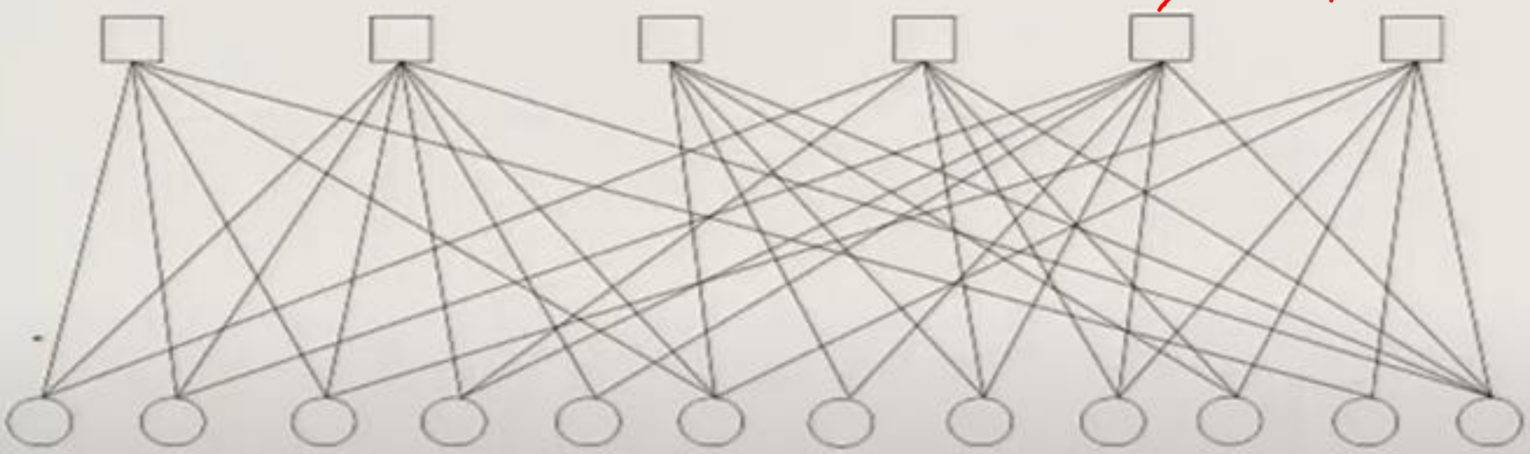
$$\rho(x) = \sum_{i>1} \rho_i x^{i-1}$$

3

2

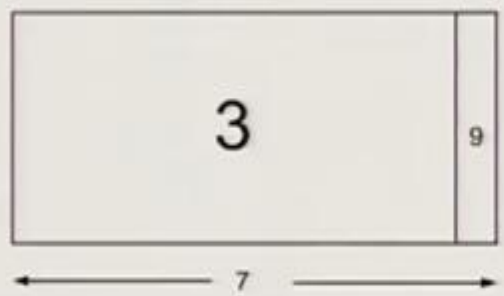
4

5



$$\lambda(x) = \frac{1}{4}x + \frac{7}{12}x^2 + \frac{1}{12}x^3 + \frac{1}{12}x^4$$
$$\rho(x) = \frac{1}{3}x^4 + \frac{1}{3}x^5 + \frac{1}{3}x^6$$

- Now generate the asked thing….



Notation: integers "3" and "9" represent the column weights.

Column Weight	Fraction of columns	Row weight	Fraction
3	11/12	7	1
9	1/12		

Notation: An integer represents a number of permutation matrices superposed on the surrounding square. Horizontal and vertical lines indicate the boundaries of the permutation blocks.

Column Weight	Fraction of columns	Row weight	Fraction
3	11/12	7	1
9	1/12		

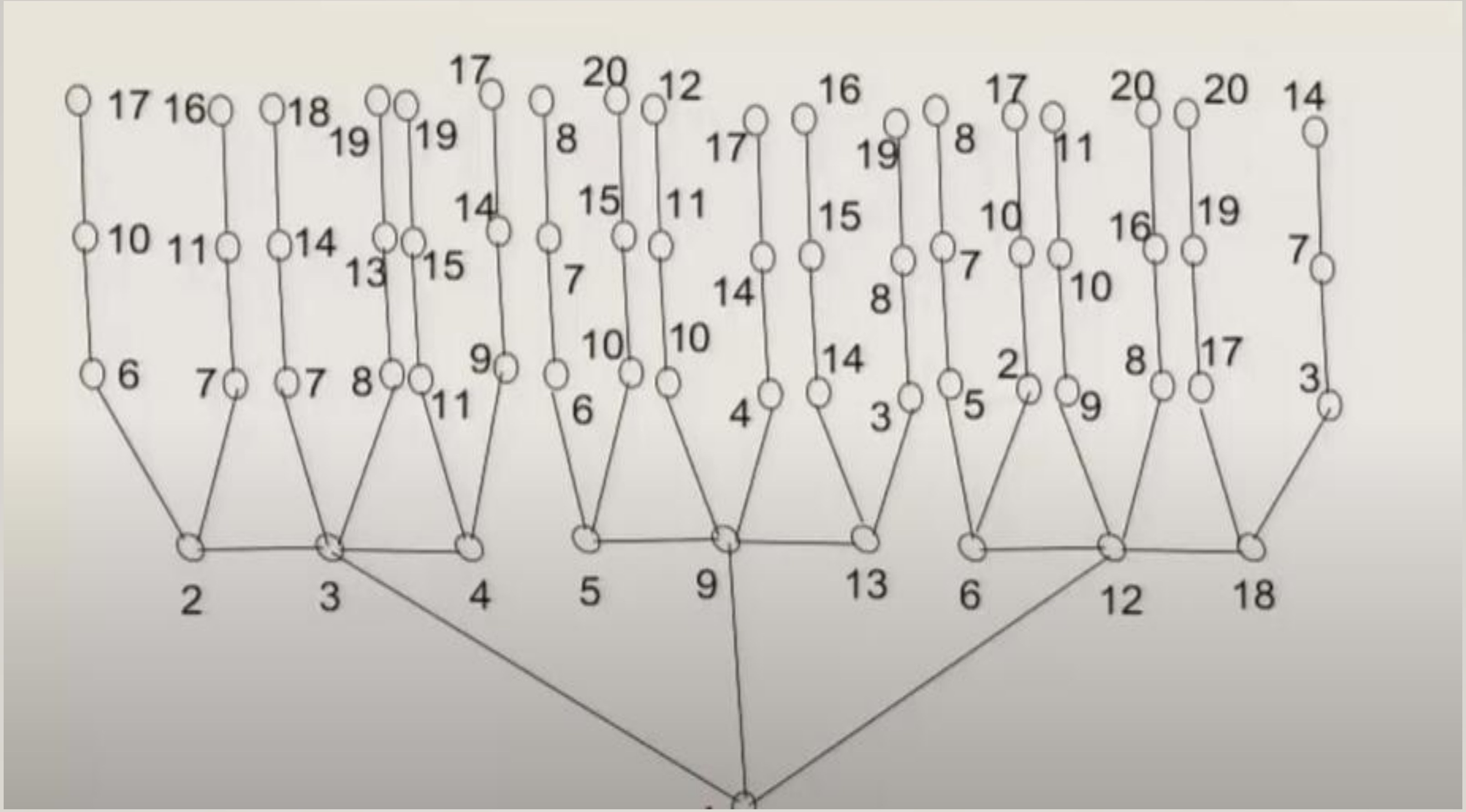
DECODING METHODS

- Decoding Methods
- Hard Decision: Bit-Flip decoding
- Soft Decision: Belief Propagation (BP)
- Choice depends on application complexity and error rate

BIT FLIP METHOD

- Bit Flip Method
- Simple iterative algorithm
- Flip bit if involved in more unsatisfied than satisfied checks
- Fast but less accurate than BP

- The set of bits contained in a parity-check equation constitutes a **parity check set**.
- **Parity check set tree** is a representation of parity check set in a tree structure.
  - An arbitrary bit d is represented by the node of the base of the tree.
  - Each line rising from this node represents one of the parity-check sets containing d.
  - The other nodes bits in these parity-check sets are represented by the nodes on the first tier of the tree.


2
3
4
5
9
13
6
12
18

Example 1: Single transmission error case

Transmitted bits= {0,0,0,0,0,0,1,1,1,1,1,1,1,0,1,0,1,1,0,0}
Received bits={1,0,0,0,0,0,1,1,1,1,1,1,1,0,1,0,1,1,0,0}

- The first bit is received in error.

BELIEF PROPAGATION METHOD

- Belief Propagation Method
- Iterative probabilistic message passing
- Uses soft information (log-likelihood ratios)
- Converges to most likely codeword under noise

BELIEF UPDATE AND DECISION MAKING

- Belief Update and Decision Making
- Update = Channel LLR + Sum of incoming check messages
- If belief > 0 → decide 0; if < 0 → decide 1
- Iterations continue until syndrome = 0 or max loops reached

MATH BEHIND BELIEF PROPAGATION

- Math Behind Belief Propagation
- Messages use the Box-Plus operator for combining LLRs
- $m(V \rightarrow C)$ = LLR + sum of other incoming check messages
- Sums and products done over log-domain to maintain numerical stability

SYNDROME CHECK AND DECODE COMPLETION

- Syndrome Check and Decode Completion
- Check if $H \times x^t = 0$ after each iteration
- If satisfied → decoding success
- Else → continue or declare decoding failure

MACKAY'S IRREGULAR GALLAGER METHOD

- MacKay's Irregular Gallager Method
- Allows variable and check node degrees to vary
- Improves convergence and decoding success rates
- Adapted to optimize performance for specific channels

ADVANTAGES

- Math Behind Irregular Advantages – Part 1
- Irregular graphs have better error-floor properties
- High-degree nodes provide strong influence early in decoding
- Performance studied via density evolution

ADVANTAGES

- Math Behind Irregular Advantages – Part 2
- EXIT charts visualize iterative decoding thresholds
- Optimization of degree distribution using linear programming
- Better approximation to Shannon capacity

APPLICATIONS OF LDPC

- Applications of LDPC
- Wi-Fi (802.11n/ac/ax)
- 5G NR (physical layer)
- Satellite communication (DVB-S2)
- Solid-state drives (error correction)
- Deep-space telemetry (NASA missions)

CONCLUSION

- Conclusion
- LDPC codes are powerful, capacity-approaching error correction tools
- Belief Propagation leverages graphical structure for decoding
- Irregular structures enhance flexibility and performance
- Key technology in modern digital communication