Lecture 13. Error Floor Analysis

① Dominating errors at the error floor regions are small weight errors that has certain structures

① BEC & Peeling decoder : Stopping Sets

Can be strictly analyzed without Monte-Carlo Simulation

② BMS & BP decoder

- I. Channel dependent ← MR, MLC, BSC, BIAWGN ?
- II. Decoder dependent ← quantized ?
- III. Code dependent ← degrees ?

neighborhood expansion ?

Normally these structures are observed after MC simulation despite finally we have some nicely formulated & mathematical definitions of them.

Takeaway: Needs to do simulation and look into the error profile to figure out those structures if you encounter some new channels and slightly different variation of BP decoders.

② Peeling Decoder & Stopping Sets

Definition: A stopping set (SS) S is a subset of VNs V, s.t. all neighbors of S connect to at least two VNs in S.

Q: are codewords SS ?

are zero vector SS ?

are SS codewords ?

Properties of SS:

I. If S_1, S_2 are SS, so is $S_1 \cup S_2$.

II. Every subset of V contains a maximum SS.

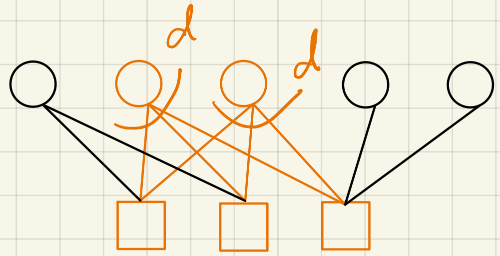
III. Every error pattern (set of erased VNs) will have a maximum SS remaining after peeling decoder.

Q: Try to prove them.

Removing short cycles can remove small SS.

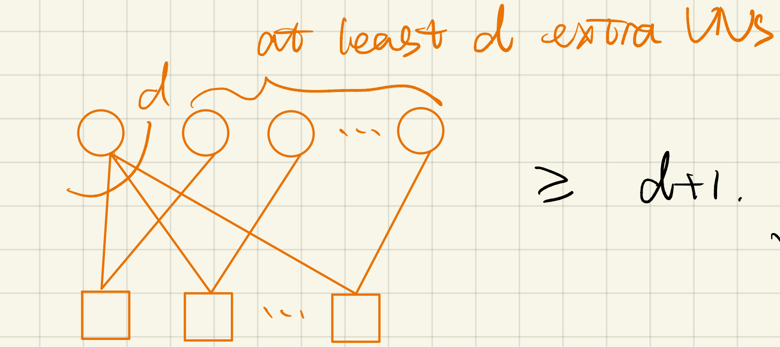
Example: Consider a code with VN degree d and girth g. what is the minimum size of a SS in it?

$g=4$.



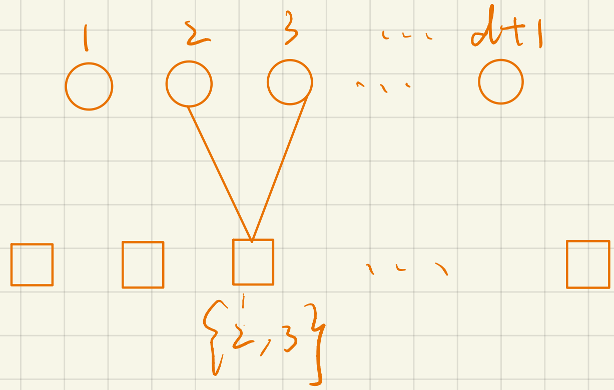
Is this a SS ?

$g=6.$



$\geq d+1.$

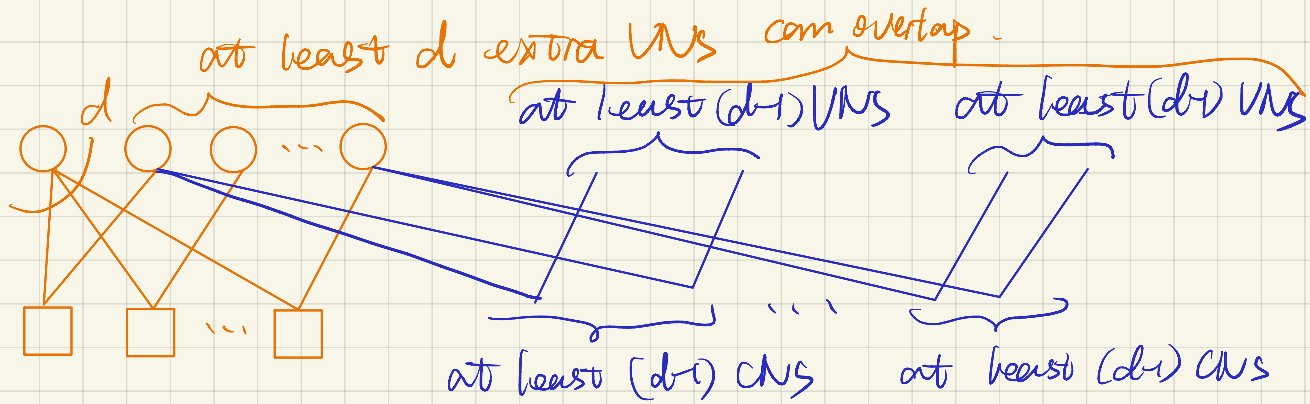
achievable.



$i \neq j \leftarrow \binom{d+1}{2}$

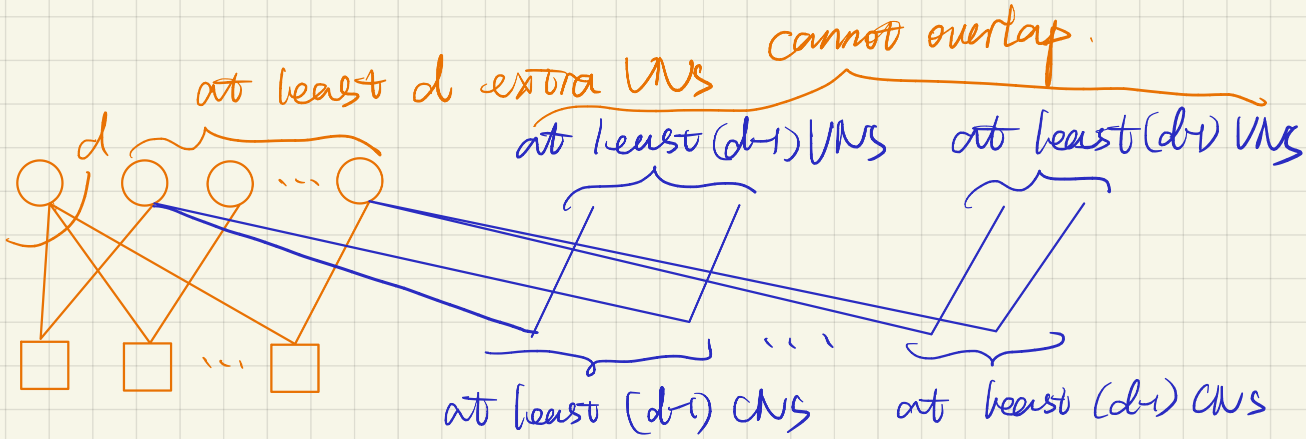
- CN $\{i,j\}$ connects to VNs i, j.
- each VN connects to d CNs. and there is no cycle 4.

$g=8$



$\geq 1+d+(d-1)=2d.$

$g=10$



$\geq 1+d+d(d-1)$

③ Absorbing Sets & Quantized BP over BMS

(BSC, BIAWGN, MLC, MR)

<u>Definition</u> An (a,b)-absorbing set (AS) is defined by a subset $\mathcal{A}$ of VNs V, where each node in $\mathcal{A}$ is connected to strictly more satisfied CNs than unsatisfied CNs.

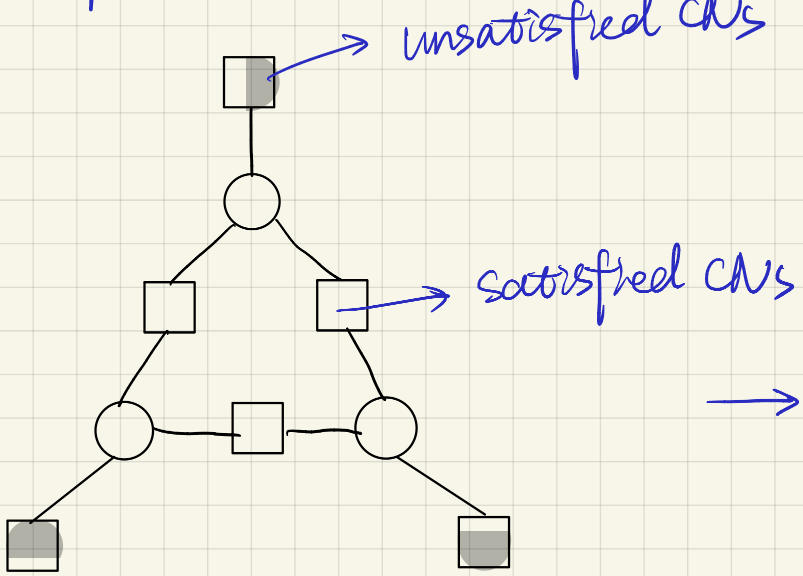
CNs whose syndrome value is $\underline{0}$ if there are bit flips on $\mathcal{A}$.

1.

In binary codes, those are CNs connects to even/odd number of VNs in $\mathcal{A}$.

where $|\mathcal{A}| = a$, and the number of CNs that only connects to 1 node in $\mathcal{A}$ is b.

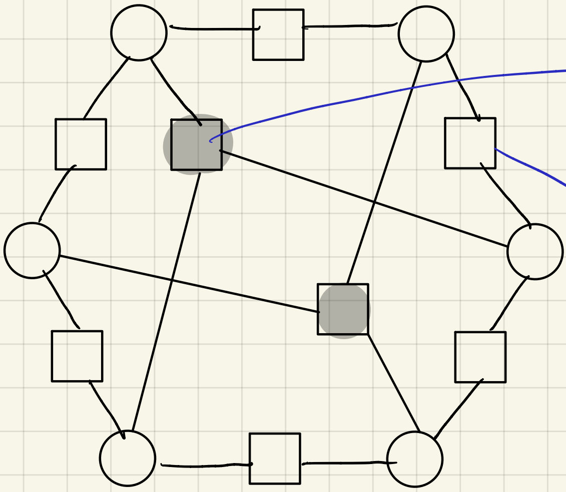
Examples:



Note: Unlike SSs, ASs are not the configurations that guarantees a decoding failure. It depends on its neighbors.

→ a $(3,3)$-AS. in degree 3 codes

Q: is cycle-6 an AS in degree 4 codes?



unsatisfied CNs

satisfied CNs.

→ a (6,0)—AS. in degree 3 codes.

Q: is this still an AS in degree 4 codes?

* Higher node degree can decrease the size of minimum AS.

Q: are codewords ASs?

④ Finite-length optimization of LDPC codes.

Objective: remove the most detrimental objects.

however, it is hard to do so

I. How detrimental an object is depends on neighborhood expansions ← not all small AS are equally detrimental

II. ASs typically have intricate structures and each node can be included in various ASs.

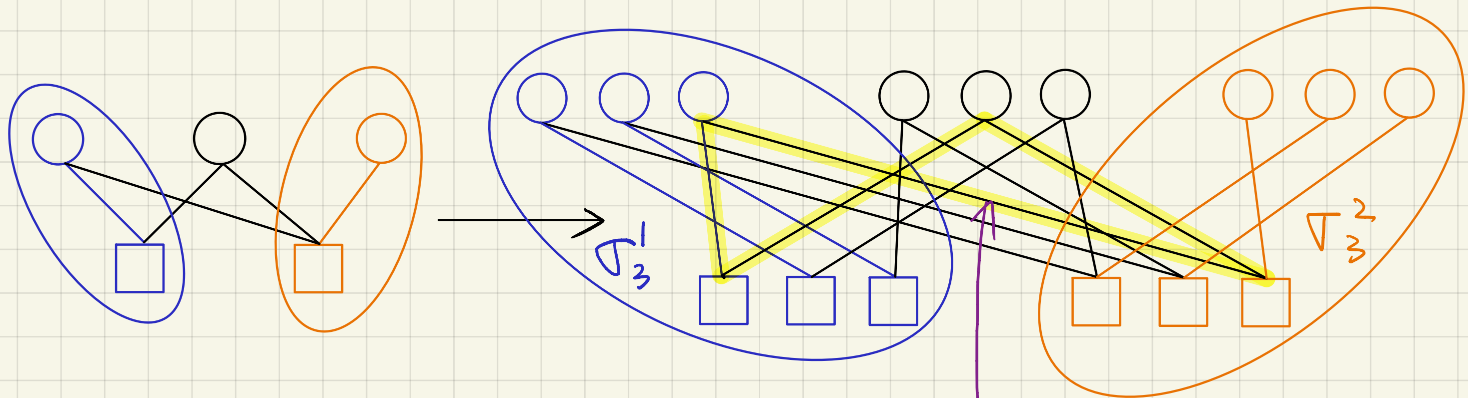
simplification →

remove small cycles { smallest common structures of ASs; detrimental to waterfall region

Note: we can do better by removing concatenated cycles, will be shown in SC-LDPC codes later.

Quasi-Cycle LDPC Codes

A quasi-cyclic (QC) LDPC codes is constructed from lifting a protograph:



$$H^P = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \longrightarrow \left[\begin{array}{ccc|ccc|ccc} 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \hline 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \end{array}\right]$$

protograph

$1-2+1-0=0$

Lifting matrix

$$L = \begin{bmatrix} 1 & 2 & -1 \\ 0 & 1 & 2 \end{bmatrix}$$

A placeholder indicating there is an "0" in the protograph

cycle-4 with alternating sum $1-2+1-0=0$

A QC code can be represented by replacing each "1" in H^P (parity-check matrix of the protograph) with a power of cyclic-shift matrix σ_z:

$$\sigma_z = \begin{bmatrix} 0 & \cdots & 0 & 1 \\ 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix} = \left[\begin{array}{c|c} 0_{1\times(z-1)} & 1 \\ \hline I_{(z-1)} & 0_{(z-1)\times 1} \end{array}\right], \quad \sigma_z^k = \left[\begin{array}{c|c} 0_{k\times(z-k)} & I_k \\ \hline I_{(z-k)} & 0_{(z-k)\times k} \end{array}\right]$$

Given the lifting matrix L (size the same to H^P), the QC code is defined as:

$$H[(i-1)z+1 : iz, (j-1)z+1 : jz] = \begin{cases} 0_{z\times z}, & \text{if } H^P_{i,j}=0 \\ \sigma_z^{L_{i,j}}, & \text{if } H^P_{i,j}=1 \end{cases}$$

Cycle Optimization

→ Assign values in the lifting matrix to minimize the number of cycles.

Array-based Codes

$$L_{i,j} = ij \mod z.$$

$$\begin{aligned} & L_{i_1,j_1} - L_{i_1,j_2} + L_{i_2,j_1} - L_{i_2,j_2} \\ = & \; i_1 j_1 - i_1 j_2 + i_2 j_1 - i_2 j_2 \\ = & \; (i_1 - i_2)(j_1 - j_2) \mod z \end{aligned}$$

If z is prime, size of $L < z$. ⇒ no cycles-4.

Using large z can always remove small cycles.

- QC codes with the same protograph can have close waterfall performance, but different error floors.
- Typical optimization procedure: do greedy algorithm)
  1. remove short cycles (can start with AB codes and
  2. list small ASs in the code, do relocations (change

the lifting parameters in $\mathcal{L}$) to remove them