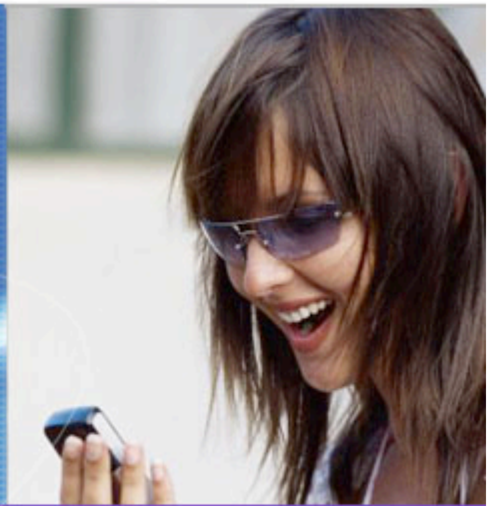
C30-20070108-051

Design of flexible rate-compatible LDPC codes

Noah Jacobsen

jacobsen@alcatel-lucent.com



Introduction

- Irregular LDPC codes exhibit performance advantage over most known turbo designs
- Sparseness of parity matrix lends itself to parallel processing, high-throughput
- Rate-compatible design is proposed
- Flexible rate and block length configurations
- Support for various information lengths with algebraic expansion

Summary

- Rate-compatible design based on daughter code with extensions to lower rates
- Complexity benefits: redundancy generated as needed, decoding performed on corresponding sub-graph
- Girth conditioning employed for good graphs at moderate block length: PEG algorithm
- Parity splitting for concentrating the parity degree

Notation

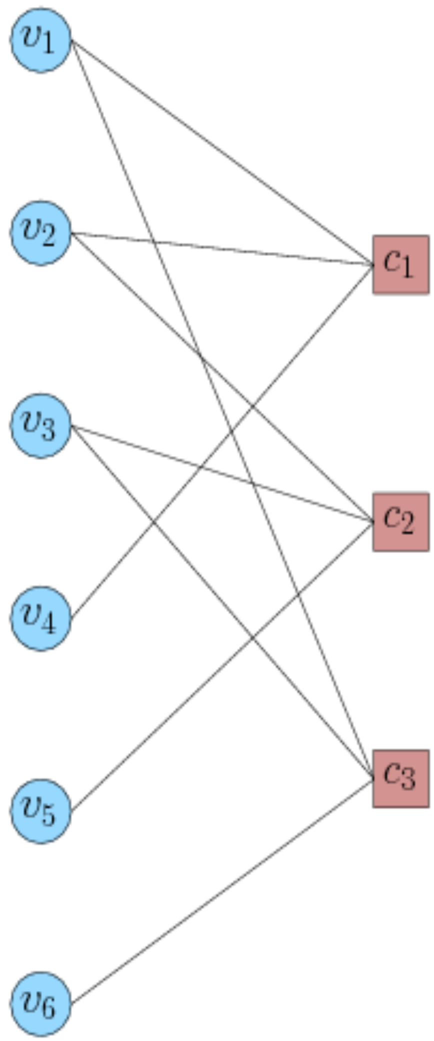
- A binary (n, k) linear block code maps k information bits to an n bit codeword

$$\mathrm{c} = \mathrm{uG}$$

- Dual representation: rows of H span the null space of G
- Valid codewords satisfy parity equations

$$\mathrm{Hc}^T = \mathbf{0}$$

Graphical interpretation

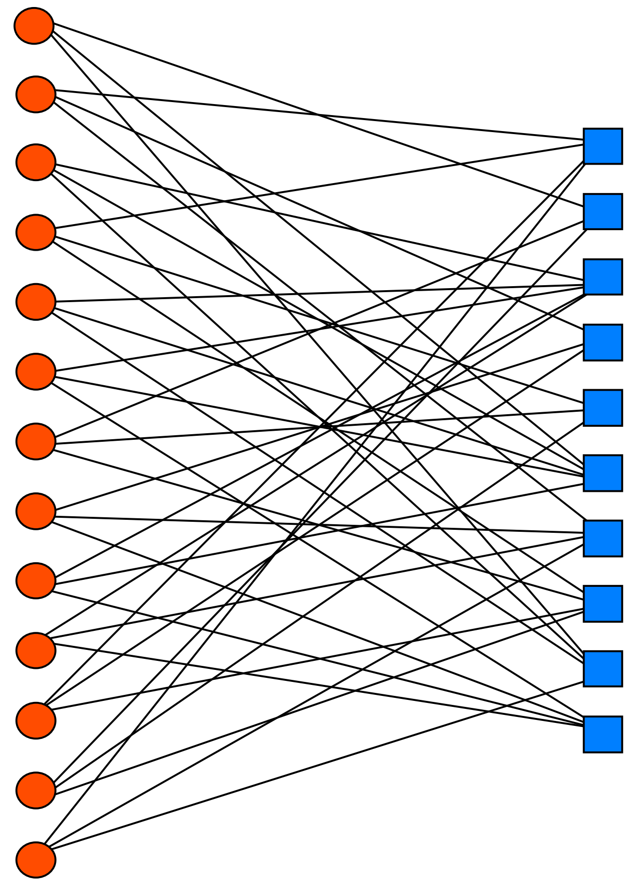


$$H = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

- Bipartite graph: check nodes and variable nodes
- Variables represented coded symbols
- Checks enforce parity constraint
- Edge in graph means variable participates in check

LDPC codes

- Random linear block code with large n
- Sparse distribution of ones in parity matrix
- For large block lengths, cycles typically have large girth
- Similar to random interleaving in turbo-codes, but no trellis processing



Alcatel·Lucent

Irregular LDPC codes

- Irregular LDPC codes are allowed to vary the degrees of the check nodes and variable nodes
- The degree distribution

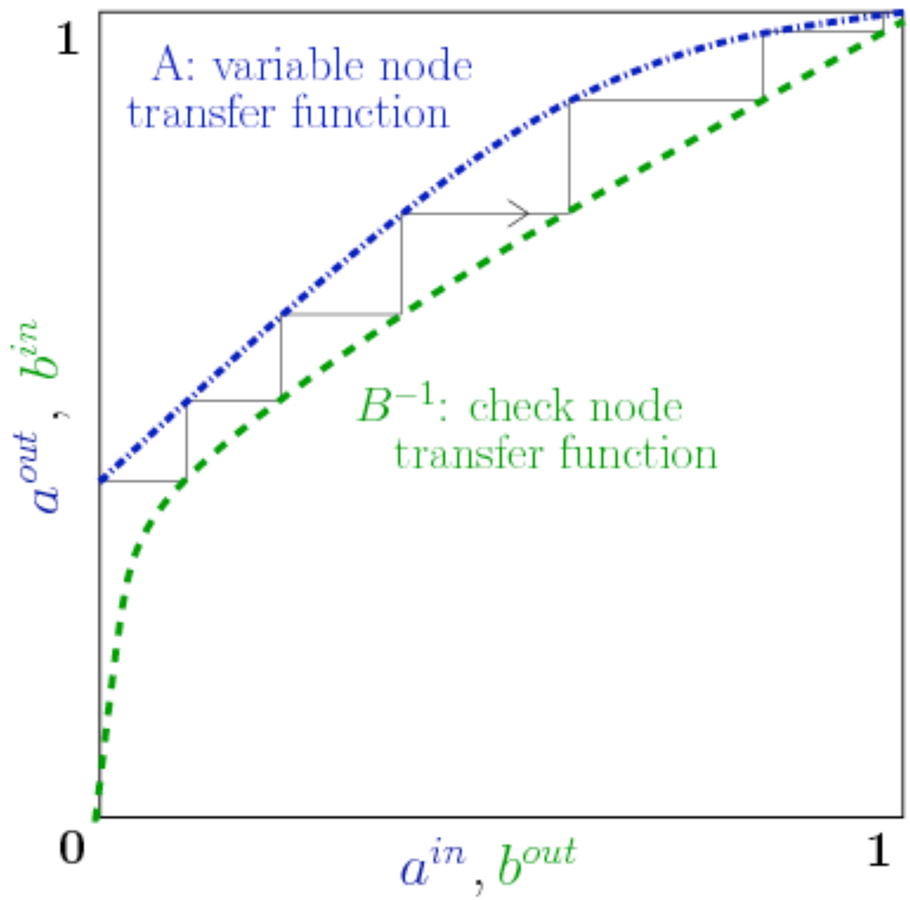
$$\rho(x) = \sum_k \rho_k x^{k-1}$$

  represents the fraction of edges ρ_k connected to a node of degree k
- An irregular (λ, ρ) LDPC has variable and check degrees distributed according to $\lambda(x)$ and $\rho(x)$ respectively
- Rate of code

$$R = \frac{k}{n} = 1 - \frac{\int_0^1 \lambda(x)dx}{\int_0^1 \rho(x)dx}$$

EXIT analysis

- One parameter analysis of message densities
- Track the mutual information of decoder messages (LLRs) and code-bits
- Scalable framework for rate-compatible optimizations



EXIT charts of code mixtures

- Given a variable node degree distribution $\lambda(x)$, the mixture EXIT function is given by

$$A(x) = \sum_{k \in \mathcal{D}_V} \lambda_k A_k(x),$$

where A_k denotes the EXIT function of a degree k variable node.

- Check node degree distribution $\rho(x)$ is optimized such that:

$$B^{-1} < A,$$

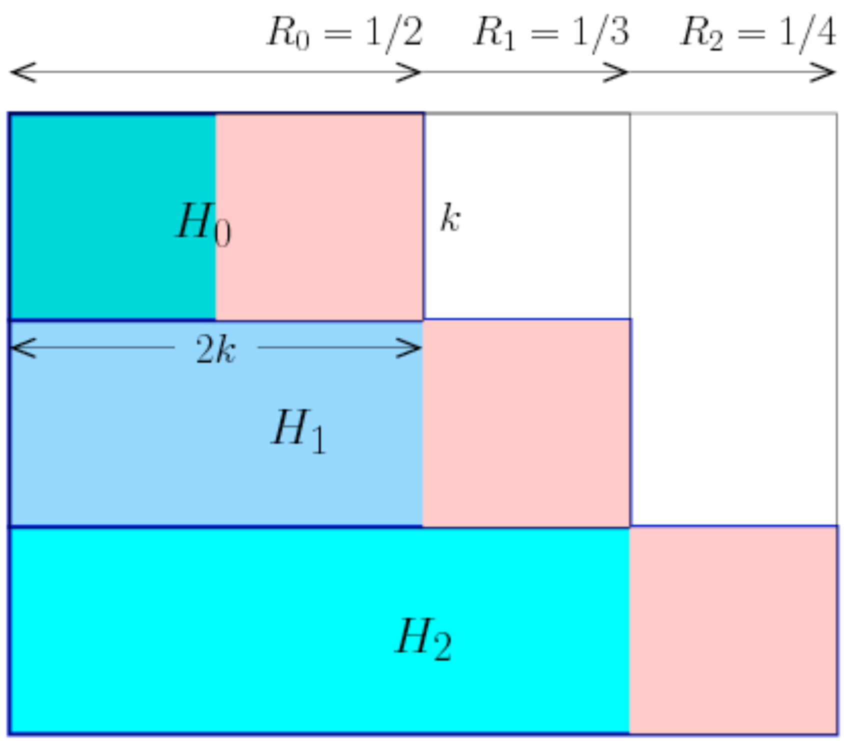
$$R = 1 - \frac{\int_0^1 \lambda(x)dx}{\int_0^1 \rho(x)dx},$$

where

$$B(x) = \sum_{k \in \mathcal{D}_C} \rho_k B_k(x),$$

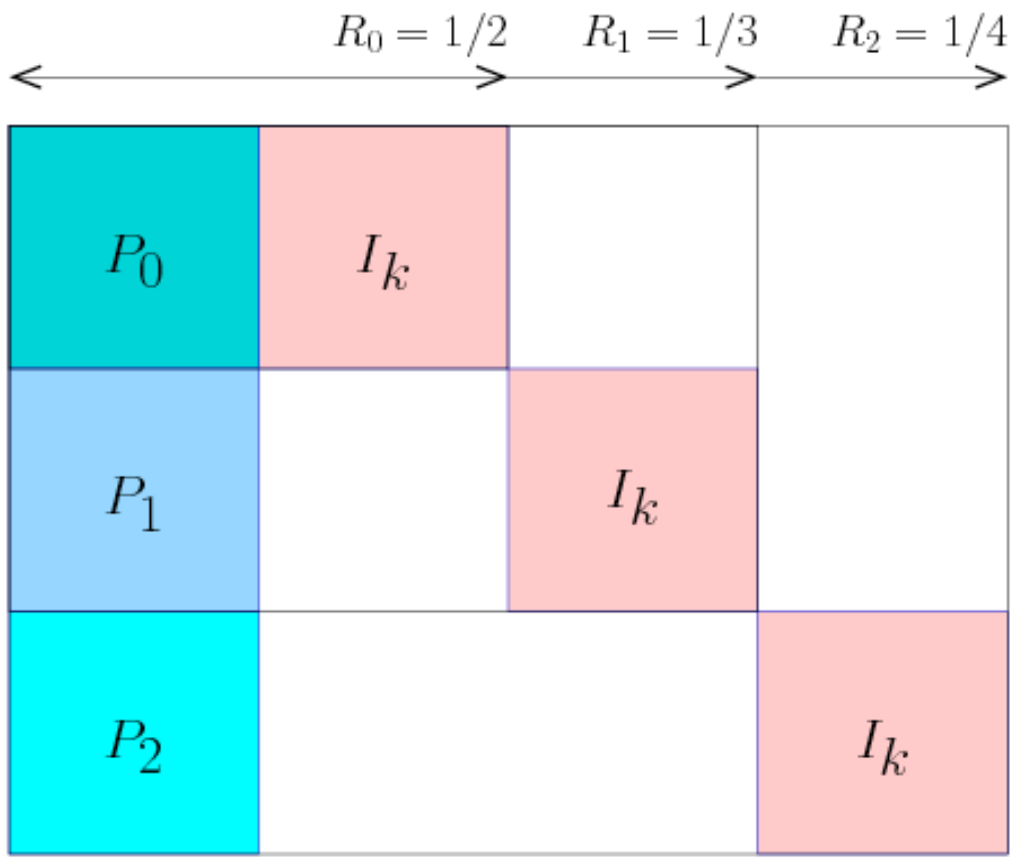
and B_k denotes the EXIT function of a degree k check node.

Rate-compatible irregular LDPC design



- Start with high-rate daughter code H_0 and extend to lower rates
- Obtain n^{th} code by optimizing the parity extension submatrix H_n, where H_{n-1}, H_{n-2},... appear as constraints
- The degree profile of each code is optimal at its rate

Systematic representation



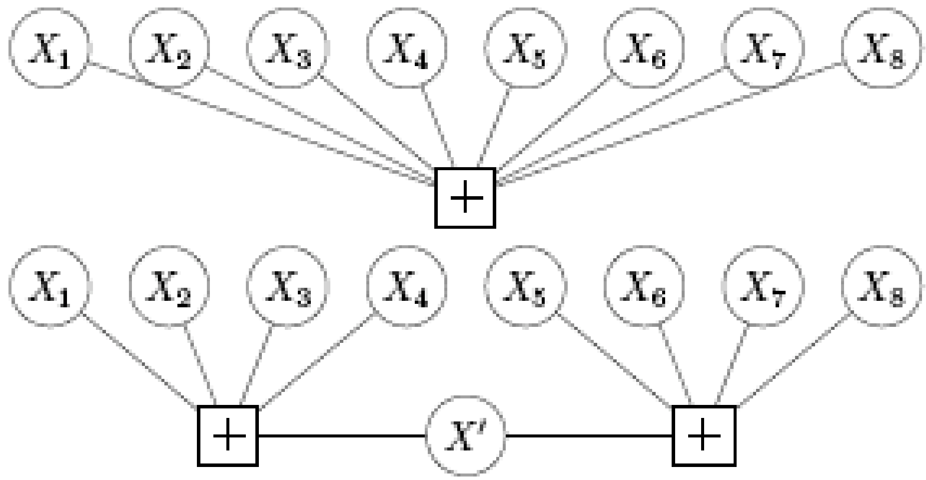
Generator matrix for lth code:

$$G_l = [I_k \;\; -P_0^T \;\; \cdots \;\; -P_l^T],\; l = 0, 1, \ldots$$

Girth conditioning

- At moderate block lengths, cycles of parity matrix significantly affect code performance
- PEG algorithm greedily assigns edges in graph in a column-by-column fashion such that local girth is maximized
- Constrained PEG algorithm accepts base matrix as parameter and attempts to maximize girth of extension parity rows

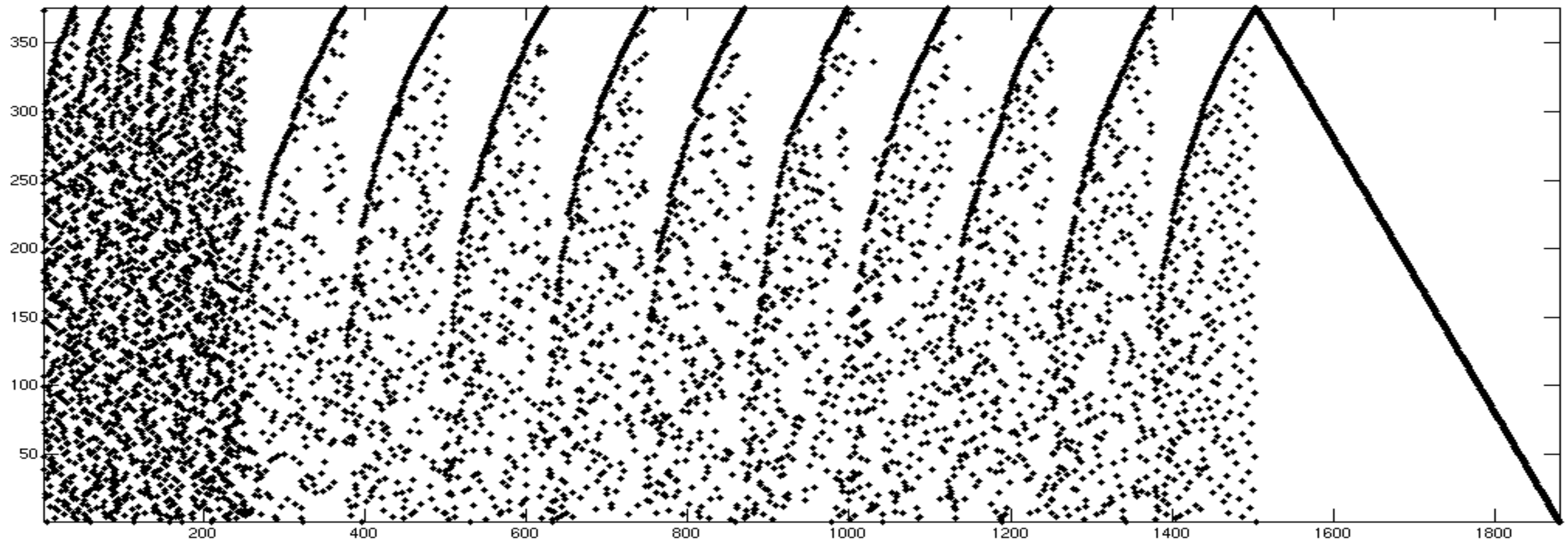
Parity splitting



- Check-regular constructions yield graphs of large girth
- Parity splitting enables a level of control over parity degree concentration
- New degree-two redundancy symbols are created by splitting an existing parity equation

Alcatel·Lucent

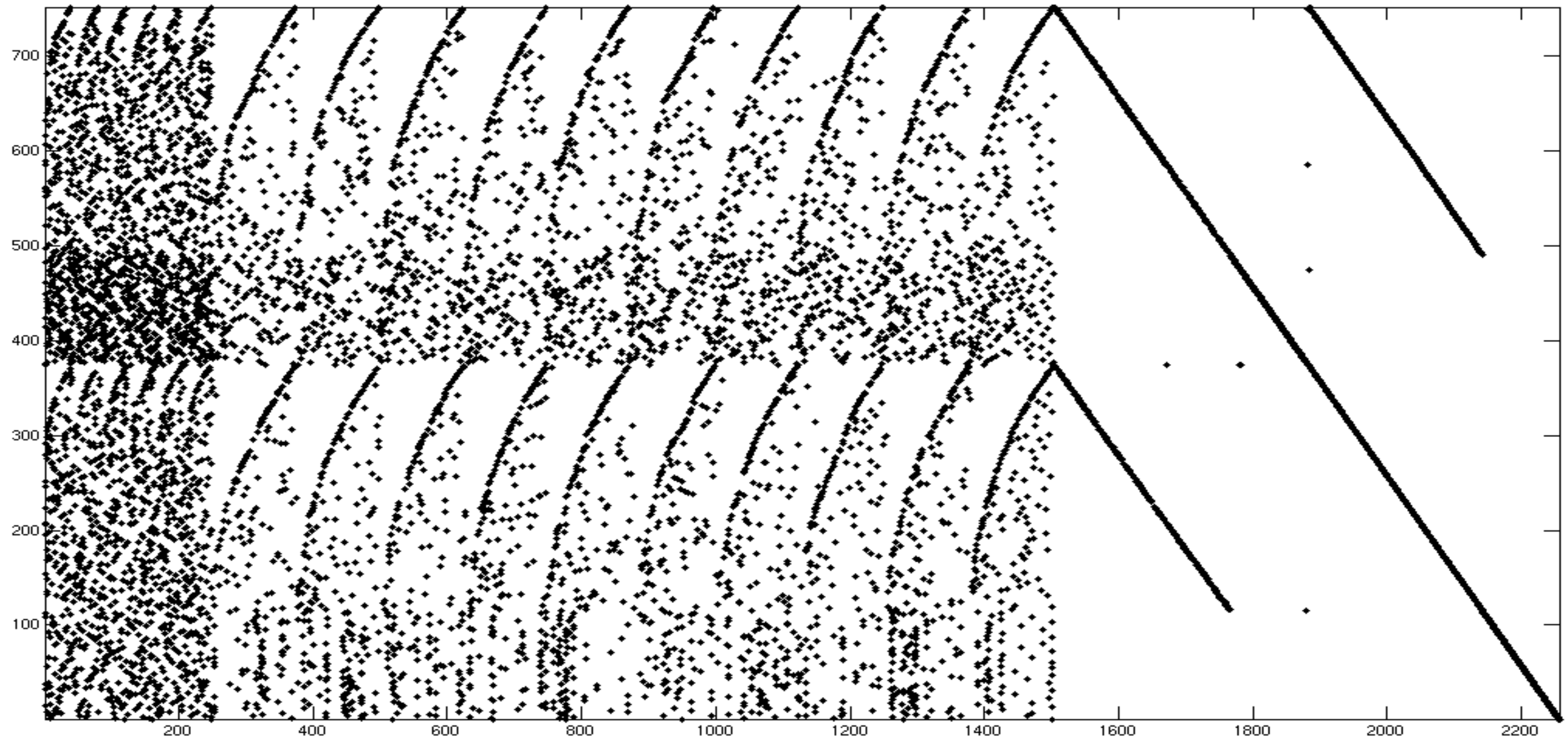
Example: rate-1/2 daughter code



- Scatter plot representation of parity check matrix
- Dot represents edge in graph
- Daughter code created with PEG algorithm

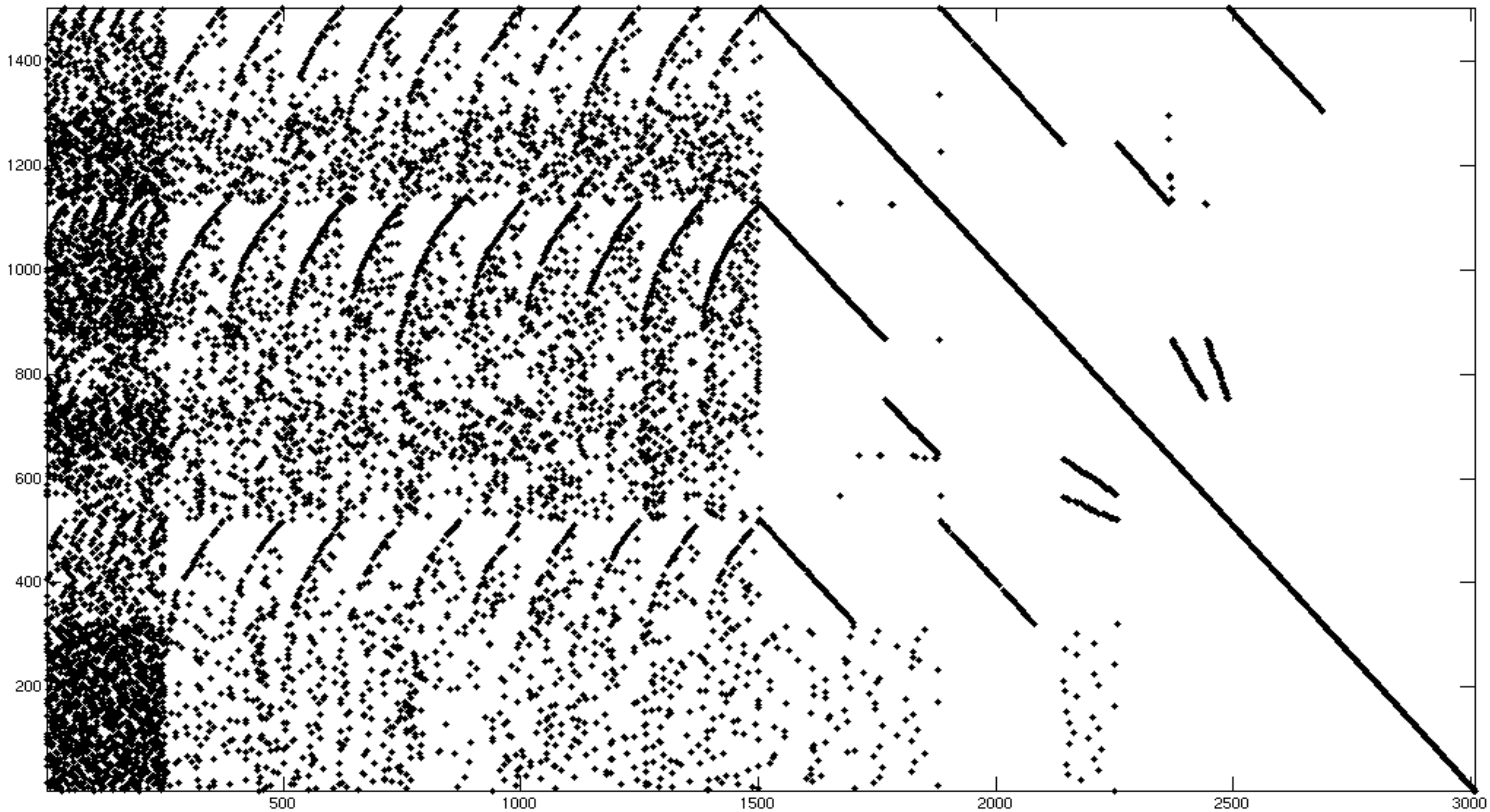
Alcatel·Lucent

1st extension code: rate-2/3



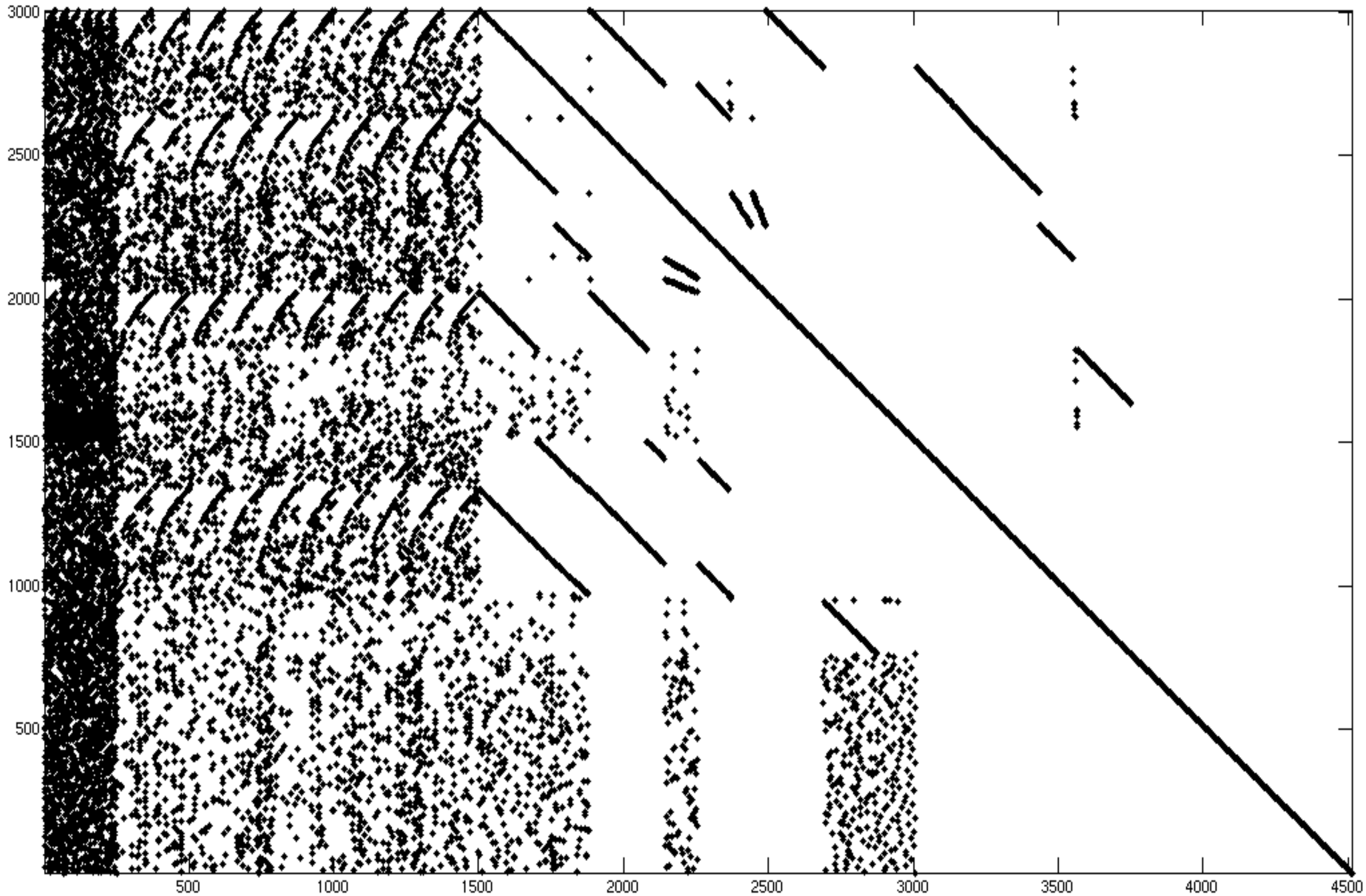
- Extension matrix is developed via a combination of parity splitting and modified PEG algorithm
- Flexibility of supported rates and block lengths

2nd extension code: rate-1/2

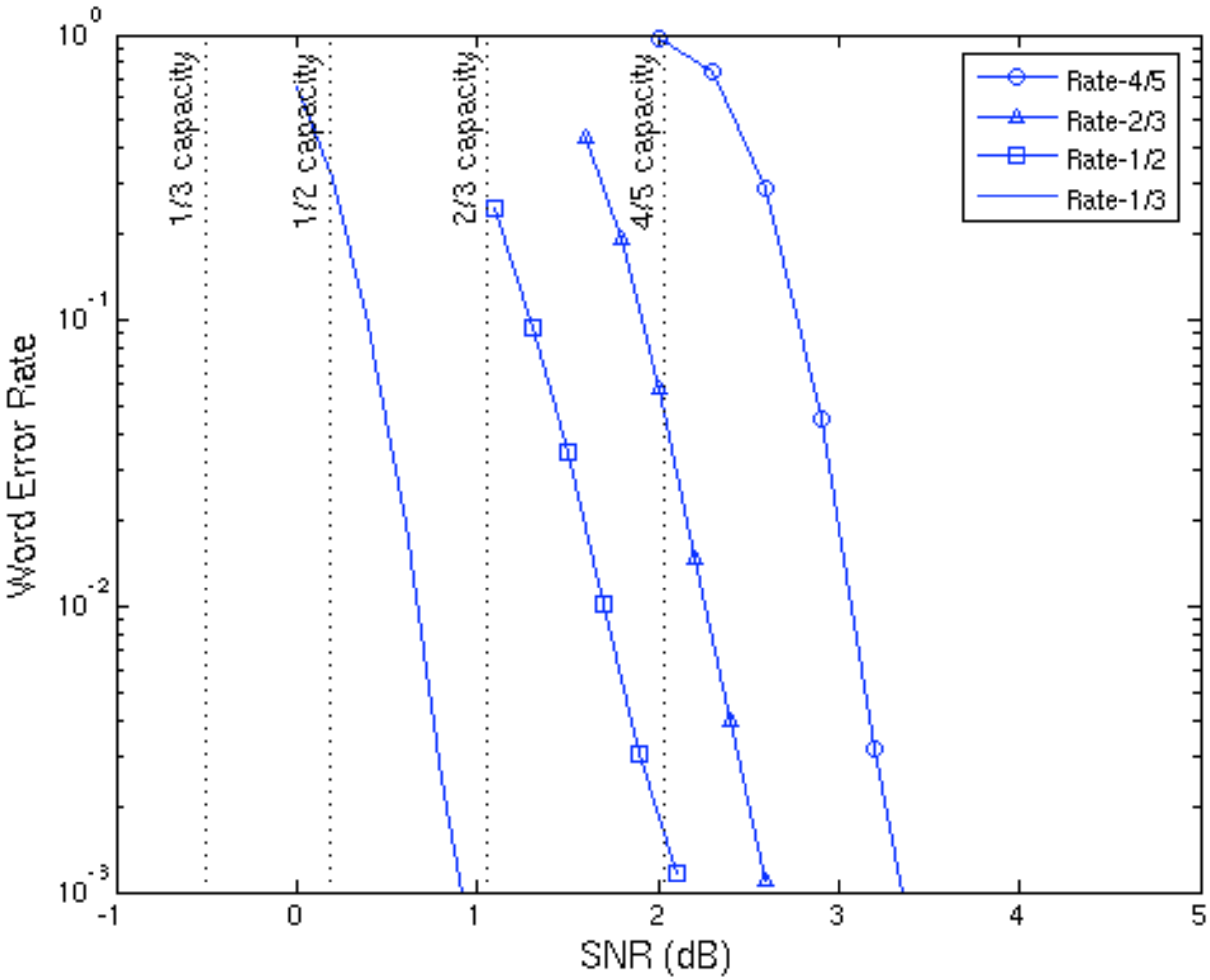


Alcatel·Lucent

3rd extension code: rate-1/3



Performance of rate-compatible LDPC



Alcatel·Lucent

Requirements for the adoption of LDPC codes in Rev-C

- In Revision-C there is a finite number of spectral efficiencies
- The standard will not specify the exact set of transport blocks other than potentially the maximum transport size beyond which segmentation must be done
- LDPC designs must address this requirement that seems incompatible with handcrafted approaches tailored to a specific transport block size
- LDPC code construction must be optimized for a wide variety of unspecified transport blocks and rates