Advances in Protograph-Based LDPC Codes and a Rate Allocation Problem
PhD Defense

Sudarsan Vasista Srinivasan Ranganathan
Advisors: Richard D. Wesel and Dariush Divsalar

Communication Systems Laboratory
University of California, Los Angeles
1. Incremental Redundancy (IR) and LDPC Codes
2. Quasi-Cyclic PBRL Design for Short Block-Lengths
3. A Property of PBRL Decoding
4. PBRL Codes for Universal Increment Ordering
5. A Rate Allocation Problem
1. **Incremental Redundancy (IR) and LDPC Codes**
   - Rate-Compatible Codes, Protograph LDPC Codes
   - Protograph QC-LDPC Codes

2. **Quasi-Cyclic PBRL Design for Short Block-Lengths**
   - Protograph-Based Raptor-Like LDPC (PBRL) Codes
   - Permanent-Bound-Based Design (PBD) Method
   - Simulation Results

3. **A Property of PBRL Decoding**

4. **PBRL Codes for Universal Increment Ordering**
   - UIO-PBRL Codes

5. **A Rate Allocation Problem**
Rate-Compatibility (RC)

- Useful in many communication applications

Highest-rate packet

Packet 1
Packet 2
... Packet \( \ell \)

Highest-rate code (HRC)
Incremental redundancy code (IRC)

Challenge today:
- Short block-lengths
- Ever-growing throughput requirements
- New applications
- Handling rate-compatibility
Rate-Compatibility (RC)

- Useful in many communication applications
- RC codes most recently proposed for the 5G wireless standard
Rate-Compatibility (RC)

- Useful in many communication applications
- RC codes most recently proposed for the 5G wireless standard
- Challenge today: short block-lengths, ever-growing throughput requirements, new applications, handling rate-compatibility
\[ H \text{ of size } 6 \times 9 \]

\[
H = \begin{bmatrix}
1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\
\end{bmatrix}
\]
Variable nodes

\[ H \text{ of size } 6 \times 9 \]

\[
H = \\
\begin{bmatrix}
1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0
\end{bmatrix}
\]
Linear Codes $\iff$ Tanner Graphs

Variable nodes

$H$ of size $6 \times 9$

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

Check nodes
Linear Codes $\iff$ Tanner Graphs

$H$ of size $6 \times 9$

$$H = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$
Linear Codes $\iff$ Tanner Graphs

Variable nodes

Check nodes

$H$ of size $6 \times 9$

$$H = \begin{bmatrix}
1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\
\end{bmatrix}$$
“Regular” bipartite Tanner graph
Variable node degree 2
Check node degree 3
Girth:
Length of smallest cycle in graph

R. Tanner, ”A recursive approach to low complexity codes,”
Low-Density Parity-Check Codes (LDPC Codes)

- Most important class of codes treated as graphs are **LDPC codes**

Low-Density Parity-Check Codes (LDPC Codes)

- Most important class of codes treated as graphs are **LDPC codes**
- LDPC: parity-check matrix is of “low density”

Initial channel belief
Iterative Decoding of (LDPC) Codes

Initial channel belief

V-to-C messages
Iterative Decoding of (LDPC) Codes

Initial channel belief
V-to-C messages
C-to-V messages
Iterative Decoding of (LDPC) Codes

Initial channel belief
V-to-C messages
C-to-V messages
Overall belief
Iterative Decoding Threshold

Outline

1. Incremental Redundancy (IR) and LDPC Codes
   - Rate-Compatible Codes, Protograph LDPC Codes
   - Protograph QC-LDPC Codes

2. Quasi-Cyclic PBRL Design for Short Block-Lengths
   - Protograph-Based Raptor-Like LDPC (PBRL) Codes
   - Permanent-Bound-Based Design (PBD) Method
   - Simulation Results

3. A Property of PBRL Decoding

4. PBRL Codes for Universal Increment Ordering
   - UIO-PBRL Codes

5. A Rate Allocation Problem
An $N \times N$ circulant permutation matrix (CPM), $x$-shifted $(0 \leq x \leq N - 1)$ is

- the identity matrix cyclically shifted by $x$ positions
Circulant Permutation Matrix (CPM)

Definition (Circulant permutation matrix (CPM))

An $N \times N$ circulant permutation matrix (CPM), $x$-shifted ($0 \leq x \leq N - 1$) is

- the identity matrix cyclically shifted by $x$ positions

Example, when $x = 2$, $N = 4$, right shift:

$$
\begin{bmatrix}
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0
\end{bmatrix}
$$
**Protograph**

A small Tanner graph

![Protograph Diagram]
Protograph and Protomatrix

**Protograph**
A small Tanner graph

**Corresponding Protomatrix**

\[
P = \begin{bmatrix}
1 & 1 & 1 \\
0 & 1 & 1 \\
\end{bmatrix}
\]

\[n_c = 2, \ n_v = 3\]

\[R = \frac{n_v - n_c}{n_v - n_p}\]

Protomatrix of complete protograph

\[
\begin{bmatrix}
1 & 1 & 1 \\
1 & 1 & 1 \\
\end{bmatrix}
\]
Protograph Quasi-Cyclic LDPC (Protograph QC-LDPC) Codes

Protomatrix of complete protograph

\[
\begin{bmatrix}
1 & 1 & 1 \\
1 & 1 & 1 \\
\end{bmatrix}
\]

CPM blocks $N \times N$ in place of each 1

\[
\begin{bmatrix}
1 & 0 & 1 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 & 0 & 1 \\
1 & 0 & 1 & 0 & 0 & 1 \\
0 & 1 & 0 & 1 & 1 & 0 \\
\end{bmatrix}
\]
Protograph Quasi-Cyclic LDPC (Protograph QC-LDPC) Codes

Protomatrix of complete protograph

\[
\begin{bmatrix}
1 & 1 & 1 \\
1 & 1 & 1
\end{bmatrix}
\]

CPM blocks $N \times N$ in place of each 1

\[
\begin{bmatrix}
1 & 0 & 1 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 & 0 & 1 \\
1 & 0 & 1 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 & 1 & 0
\end{bmatrix}
\]

Parity-check matrix of complete protograph QC-LDPC code
Protograph Quasi-Cyclic LDPC (Protograph QC-LDPC) Codes

Protomatrix of complete protograph

\[
\begin{bmatrix}
1 & 1 & 1 \\
1 & 1 & 1 \\
\end{bmatrix}
\]

Lifting

CPM blocks $N \times N$ in place of each 1

Parity-check matrix of complete protograph QC-LDPC code

\[
\begin{bmatrix}
1 & 0 & 1 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 & 0 & 1 \\
1 & 0 & 1 & 0 & 0 & 1 \\
0 & 1 & 0 & 1 & 1 & 0 \\
\end{bmatrix}
\]
Protograph Quasi-Cyclic LDPC (Protograph QC-LDPC) Codes

Protomatrix of complete protograph

\[
\begin{bmatrix}
1 & 1 & 1 \\
1 & 1 & 1 \\
\end{bmatrix}
\]

CPM blocks \( N \times N \) in place of each 1

\[
\begin{bmatrix}
1 & 0 & 1 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 & 0 & 1 \\
1 & 0 & 1 & 0 & 0 & 1 \\
0 & 1 & 0 & 1 & 1 & 0 \\
\end{bmatrix}
\]

Parity-check matrix of complete protograph QC-LDPC code

Lifting factor \( N \)
Outline

1. Incremental Redundancy (IR) and LDPC Codes
   - Rate-Compatible Codes, Protograph LDPC Codes
   - Protograph QC-LDPC Codes

2. Quasi-Cyclic PBRL Design for Short Block-_lengths
   - Protograph-Based Raptor-Like LDPC (PBRL) Codes
   - Permanent-Bound-Based Design (PBD) Method
   - Simulation Results

3. A Property of PBRL Decoding

4. PBRL Codes for Universal Increment Ordering
   - UIO-PBRL Codes

5. A Rate Allocation Problem
Based on...


Protograph-Based Raptor-Like (PBRL) Codes

- Best performing RC-LDPC codes known so far

Protograph-Based Raptor-Like (PBRL) Codes

- Best performing RC-LDPC codes known so far
  - The code family in 5G for data communication

Protograph-Based Raptor-Like (PBRL) Codes

- Best performing RC-LDPC codes known so far
  - The code family in 5G for data communication
- General structure of a PBRL protomatrix

\[ P = \begin{bmatrix} P_{HRC} & 0 \\ P_{IRC} & I \end{bmatrix}_{n_c \times n_v} \]  

Protograph-Based Raptor-Like (PBRL) Codes

- Best performing RC-LDPC codes known so far
  - The code family in 5G for data communication
- General structure of a PBRL protomatrix

\[
P = \begin{bmatrix}
P_{\text{HRC}} & 0 \\
P_{\text{IRC}} & I
\end{bmatrix}_{n_c \times n_v}
\] (1)

- \(P_{\text{HRC}} = \text{HRC (highest-rate code) part}\)

Protograph-Based Raptor-Like (PBRL) Codes

- Best performing RC-LDPC codes known so far
  - The code family in 5G for data communication
- General structure of a PBRL protomatrix

\[
P = \begin{bmatrix}
P_{\text{HRC}} & 0 \\
P_{\text{IRC}} & I
\end{bmatrix}_{n_c \times n_v}
\] (1)

- \(P_{\text{HRC}} = \) HRC (highest-rate code) part
- \(P_{\text{IRC}} = \) IRC (incremental redundancy code) part

## PBRL Protomatrix

\[ P = \begin{bmatrix}
2 & 1 & 2 & 1 & 2 & 1 & 0 & 0 & 0 \\
1 & 2 & 1 & 2 & 1 & 2 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix} \]

- \( n_{CH} = 2, n_{vH} = 6 \)
- \( P_{HRC} \)

\[ P_{IRC} \quad n_c = 5, n_v = 9 \]
PBRL Protograph
PBRL Protograph

![PBRL Protograph Diagram]

S. V. S. Ranganathan
Protograph Codes, Rate Allocation
PhD Defense, 11/30/2018
Original PBRL Design Procedure

\[
P = \begin{bmatrix}
P_{HRC} \\
P_{IRC}
\end{bmatrix}
\begin{bmatrix}
2 & 1 & 2 & 1 & 2 & 1 & 2 & 1 \\
1 & 2 & 1 & 2 & 1 & 2 & 1 & 2
\end{bmatrix}
\]

Rate 6/7, Threshold = 3.07dB
$P = \begin{bmatrix}
2 & 1 & 2 & 1 & 2 & 1 & 2 & 1 & 0 \\
1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1
\end{bmatrix}$

$P_{HRC}$

Rate 6/7, Threshold = 3.07dB
Rate 6/8, Threshold = 2.11dB
Original PBRL Design Procedure

\[
P = \begin{bmatrix}
P_{\text{HRC}} \\
2 & 1 & 2 & 1 & 2 & 1 & 2 & 1 & 0 & 0 \\
1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 & 0 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\
\end{bmatrix}
\]

Rate 6/7, Threshold = 3.07dB
Rate 6/8, Threshold = 2.11dB
Rate 6/9, Threshold = 1.61dB
Original PBRL Design Procedure

\[
P = \begin{pmatrix}
2 & 1 & 2 & 1 & 2 & 1 & 2 & 1 & 0 & 0 & 0 \\
1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\
1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\
\end{pmatrix}

P_{\text{HRC}}

\text{Rate 6/7, Threshold = 3.07dB}
\text{Rate 6/8, Threshold = 2.11dB}
\text{Rate 6/9, Threshold = 1.61dB}
\text{6/10, Threshold = 1.17dB}

P_{\text{IRC}}
Original PBRL Design Procedure

\[
P = \begin{bmatrix}
P_{\text{HRC}} \\
2 & 1 & 2 & 1 & 2 & 1 & 2 & 1 \\
1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 \\
1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\
\cdot \\
\cdot \\
\cdot \\
\cdot \\
P_{\text{IRC}}
\end{bmatrix}
\]

Rate 6/7, Threshold = 3.07dB
Rate 6/8, Threshold = 2.11dB
Rate 6/9, Threshold = 1.61dB
6/10, Threshold = 1.17dB
1. Incremental Redundancy (IR) and LDPC Codes
   - Rate-Compatible Codes, Protograph LDPC Codes
   - Protograph QC-LDPC Codes

2. Quasi-Cyclic PBRL Design for Short Block-Lengths
   - Protograph-Based Raptor-Like LDPC (PBRL) Codes
   - Permanent-Bound-Based Design (PBD) Method
   - Simulation Results

3. A Property of PBRL Decoding

4. PBRL Codes for Universal Increment Ordering
   - UIO-PBRL Codes

5. A Rate Allocation Problem
Why a New Design Method?

- Threshold $\iff$ long-block-length property
Why a New Design Method?

- Threshold $\iff$ long-block-length property
  - Cannot be used for **low FER requirements at short block-lengths**
Why a New Design Method?

- Threshold $\leftrightarrow$ long-block-length property
  - Cannot be used for low FER requirements at short block-lengths
- No explicit design method known previously for short RC-LDPC codes
Why a New Design Method?

- Threshold $\Leftrightarrow$ long-block-length property
  - Cannot be used for low FER requirements at short block-lengths
- No explicit design method known previously for short RC-LDPC codes
  - Short convolutional codes are pretty much the only class of codes used so far for short block-length rate-compatibility
Why a New Design Method?

- Threshold \(\iff\) long-block-length property
  - Cannot be used for low FER requirements at short block-lengths
- No explicit design method known previously for short RC-LDPC codes
  - Short convolutional codes are pretty much the only class of codes used so far for short block-length rate-compatibility
- Because we can do better...
A Minimum Distance Upper Bound

The minimum distance of a protograph QC-LDPC code is upper bounded by a constant that does not depend upon $N$ and that depends only upon the protograph.


CPM blocks $N \times N$ in place of each 1

\[
\begin{bmatrix}
1 & 1 & 1 \\
1 & 1 & 1
\end{bmatrix}
\rightarrow
\begin{bmatrix}
\square & \square & \square \\
\square & \square & \square
\end{bmatrix}
\]
The minimum distance of a protograph QC-LDPC code is upper bounded by a constant that does not depend upon $N$ and that depends only upon the protograph.

Theorem (Complexity of Permanent Bound)

For a protomatrix of size $n_c \times n_v$, the complexity of computing this upper bound is $\Theta\left(\binom{n_v}{n_c+1}(n_c + 1) \cdot n_c 2^{n_c}\right)$.

Theorem (Complexity of Permanent Bound)

For a protomatrix of size $n_c \times n_v$, the complexity of computing this upper bound is $\Theta \left( \binom{n_v}{n_c+1} (n_c + 1) \cdot n_c 2^{n_c} \right)$.

Note the dependence on $n_c, n_v$ (the size of the whole protomatrix)

Permanent Bound Design (PBD) Method

\[ P = \begin{bmatrix} P_{HRC} \\ P_{IRC} \end{bmatrix} \]

Rate 6/7, Min. Dist. Upper Bound = 8
Rate 6/8, Min. Dist. Upper Bound = 12
Permanent Bound Design (PBD) Method

\[ P = \begin{bmatrix} P_{\text{HRC}} \\ 2 & 1 & 2 & 1 & 2 & 1 & 2 & 1 & 0 \\ 1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \]

Rate 6/7, Min. Dist. Upper Bound = 8
Rate 6/8, Min. Dist. Upper Bound = 12
Permanent Bound Design (PBD) Method

\[ P = \begin{bmatrix}
  P_{\text{HRC}} \\
  \begin{array}{cccccccc}
  2 & 1 & 2 & 1 & 2 & 1 & 2 & 1 \\
  1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 \\
  1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
  1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
  \end{array}
  \begin{array}{c}
  0 & 0 \\
  0 & 0 \\
  0 & 1 \\
  \end{array}
\end{bmatrix} \]

Rate 6/7, Min. Dist. Upper Bound = 8
Rate 6/8, Min. Dist. Upper Bound = 12
Rate 6/9, Min. Dist. Upper Bound = 14

\[ P_{\text{IRC}} \]
Permanent Bound Design (PBD) Method

\[
P = \begin{bmatrix}
P_{HRC} \\
2 & 1 & 2 & 1 & 2 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
\vdots & \vdots & & & & \vdots & \vdots \\
\end{bmatrix} \\
P_{IRC}
\]

Rate 6/7, Min. Dist. Upper Bound = 8
Rate 6/8, Min. Dist. Upper Bound = 12
Rate 6/9, Min. Dist. Upper Bound = 14
Complexity Reduction for PBRL Protomatrices

For a PBRL protomatrix of size $n_c \times n_v$, the complexity of computing the permanent upper bound is $\Theta \left( \frac{n_v}{n_c+1} (n_c + 1) \cdot \left( n_c + 1 \right) 2^{(n_c+1)} \right)$. 
Complexity Reduction for PBRL Protomatrices

Theorem (Permanent Bound Complexity for PBRL Protomatrices)

For a PBRL protomatrix of size $n_c \times n_v$, the complexity of computing the permanent upper bound is $\Theta\left(\binom{n_v}{n_{c_H}+1} (n_c + 1) \cdot (n_{c_H} + 1) 2^{(n_{c_H}+1)} \right)$.

Recall, it was previously $\Theta\left(\binom{n_v}{n_c+1} (n_c + 1) \cdot n_c 2^{n_c} \right)$.
Reduction in Complexity of Design Algorithm

\[ P = \begin{bmatrix}
    P_{\text{HRC}} \\
    2 & 1 & 2 & 1 & 2 & 1 & 2 & 1 \\
    1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 \\
\end{bmatrix} \]

\[ P_{\text{IRC}} \]

Rate 6/7, Min. Dist. Upper Bound = 8
Rate 6/8, Min. Dist. Upper Bound = 12
### Reduction in Complexity of Design Algorithm

Consider the matrix $P$ defined as follows:

$$P = \begin{bmatrix} P_{HRC} \\ \hline 2 & 1 & 2 & 1 & 2 & 1 & 2 & 1 & 0 \\ 1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

- $P_{HRC}$ represents a particular configuration with rate 6/7 and minimum distance upper bound equal to 8.
- $P_{IRC}$ represents another configuration with rate 6/8 and minimum distance upper bound equal to 12.
Reduction in Complexity of Design Algorithm

\[ P = \begin{bmatrix} P_{HRC} \\ \hline 2 & 1 & 2 & 1 & 2 & 1 & 2 & 1 & 0 & 0 \\ 1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \]

Rate 6/7, Min. Dist. Upper Bound = 8
Rate 6/8, Min. Dist. Upper Bound = 12
Rate 6/9, Min. Dist. Upper Bound = 14

\[ P_{IRC} \]
### Reduction in Complexity of Design Algorithm

Let's consider the matrices $P_{HRC}$ and $P_{IRC}$.

$$
P = \begin{bmatrix}
2 & 1 & 2 & 1 & 2 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\end{bmatrix}
$$

- Rate 6/7, Min. Dist. Upper Bound = 8
- Rate 6/8, Min. Dist. Upper Bound = 12
- Rate 6/9, Min. Dist. Upper Bound = 14
Reduction in Complexity of Design Algorithm

Turns out, we can do better!
Theorem (Reduced Complexity Design Algorithm)

The complexity of a "pre-compute" step is $\Theta(n^2 c^2 1^p n^c 1^q n^c 2^p n^c 2^q)$.

For the design rows, the complexity is $O(n^v c^1 n^c 1^p n^c 1^q 2^p n^c 2^q)$.
Theorem (Reduced Complexity Design Algorithm)

- The complexity of a “pre-compute” step is

\[\Theta \left( \binom{n_v}{n_c+1} (n_c + 1) \cdot n_c 2^{n_c} + \binom{n_v}{n_c+1} \cdot (n_c + 1) 2^{(n_c+1)} \right)\].

\[\Theta \left( \binom{n_v}{n_c+1} (n_c + 1) \cdot n_c 2^{n_c} + \binom{n_v}{n_c+1} \cdot (n_c + 1) 2^{(n_c+1)} \right)\]
Reduction in Complexity of Design Algorithm

Theorem (Reduced Complexity Design Algorithm)

- The complexity of a “pre-compute” step is
  \[ \Theta \left( \binom{n_{vH}}{n_{cH}+1} (n_{cH} + 1) \cdot n_{cH} 2^{n_{cH}} + \binom{n_{vH}}{n_{cH}+1} \cdot (n_{cH} + 1) 2^{(n_{cH}+1)} \right). \]

- For the design rows, the complexity is
  \[ O \left( \binom{n_{vH}}{n_{cH}+1} n_{vH} \right). \]
Theorem (Design of a Row $\iff$ ILP)

The design of one row of the IRC part according to the PBD method is an integer linear program with $\text{dom} = \{\text{candidate rows}\}$. 
Theorem (ILP $\neq$ LP relaxation)

The relaxation of the ILP is not exact.
The LP Relaxation is Still Useful

Theorem (New Upper Bound for PBRL Protomatries)

\[
\begin{array}{c|c}
\mathcal{H}_C & \mathcal{H}_V \\
\hline
\mathcal{N}_C \times \mathcal{N}_V \\
\end{array}
\]
Theorem (New Upper Bound for PBRL Protomatrices)

- For a given HRC part, the LP relaxation provides a new set of upper bounds at all lower design rates.
The LP Relaxation is Still Useful

\[ n_{CH} \times n_{vH} \]

\[ n_C \times n_V \]

Theorem (New Upper Bound for PBRL Protomatrices)

- **For a given HRC part, the LP relaxation provides a new set of upper bounds at all lower design rates.**
- **These upper bounds can be obtained without even having to go through the actual design procedure.**
Outline

1. Incremental Redundancy (IR) and LDPC Codes
   - Rate-Compatible Codes, Protograph LDPC Codes
   - Protograph QC-LDPC Codes

2. Quasi-Cyclic PBRL Design for Short Block-Lengths
   - Protograph-Based Raptor-Like LDPC (PBRL) Codes
   - Permanent-Bound-Based Design (PBD) Method
   - Simulation Results

3. A Property of PBRL Decoding

4. PBRL Codes for Universal Increment Ordering
   - UIO-PBRL Codes

5. A Rate Allocation Problem
New PBRL Protograph for 5G at 192 Information Bits

\[
\begin{bmatrix}
1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\]
New PBRL Protograph for 5G at 192 Information Bits

\[
\begin{bmatrix}
1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
\end{bmatrix}
\]
New Codes for 5G at 192 Information Bits

Frame error rate (FER)

Original 5G PBRL LDPC, $k = 192$

PBD method 5G PBRL LDPC, $k = 192$

$E_b/N_0$ (dB)

10^{-8}
10^{-6}
10^{-4}
10^{-2}
10^{0}

Rate 6/12
Rate 6/10
Rate 6/15
New Codes for 5G at 192 Information Bits

Frame error rate (FER) vs. $E_b/N_0$ (dB) for different rates and methods.

- Original 5G PBRL LDPC, $k = 192$
- PBD method 5G PBRL LDPC, $k = 192$

Rates compared:
- Rate 6/11
- Rate 6/13
- Rate 6/14
Outline

1. Incremental Redundancy (IR) and LDPC Codes
   - Rate-Compatible Codes, Protograph LDPC Codes
   - Protograph QC-LDPC Codes

2. Quasi-Cyclic PBRL Design for Short Block-Lengths
   - Protograph-Based Raptor-Like LDPC (PBRL) Codes
   - Permanent-Bound-Based Design (PBD) Method
   - Simulation Results

3. A Property of PBRL Decoding

4. PBRL Codes for Universal Increment Ordering
   - UIO-PBRL Codes

5. A Rate Allocation Problem
Based on...

Recall LDPC Decoding... 

- Initial channel belief
- V-to-C messages
- C-to-V messages
- Overall belief
Theorem

It is sufficient to check whether the HRC variable nodes have converged to a codeword.
Checking Only HRC Parities – No Penalty

Frame error rate (FER)

Criterion C1 - check all parities
Criterion C2 - check HRC parities

$E_b/N_0$ (dB)

- $10^{-8}$
- $10^{-6}$
- $10^{-4}$
- $10^{-2}$
- $10^{-1}$

$10^0$
Checking Only HRC Parities – Faster Convergence

Total number of convergences over 100 iterations: 811832
Outline

1. Incremental Redundancy (IR) and LDPC Codes
   - Rate-Compatible Codes, Protograph LDPC Codes
   - Protograph QC-LDPC Codes

2. Quasi-Cyclic PBRL Design for Short Block-Lengths
   - Protograph-Based Raptor-Like LDPC (PBRL) Codes
   - Permanent-Bound-Based Design (PBD) Method
   - Simulation Results

3. A Property of PBRL Decoding

4. PBRL Codes for Universal Increment Ordering
   - UIO-PBRL Codes

5. A Rate Allocation Problem
Increment Arrival Order

- Highest-rate packet
- Highest-rate code (HRC)

Design and arrival order
- Packet 1
- Packet 2
- \( \cdots \)
- Packet \( \ell \)

- Incremental redundancy code (IRC)
Increment Arrival Order

Design and arrival order

Highest-rate packet

Packet 1

Packet 2

\ldots

Packet \ell

Highest-rate code (HRC)

Incremental redundancy code (IRC)

Different arrival order

Packet \ell

Packet 1

\ldots

Packet 2

Highest-rate packet
**Objective**

- **Metric 1:** Require every order of arrival have same performance

  Inspired by requirement of inter-frame coding

Objective

- **Metric 1**: Require every order of arrival have same performance
  - Inspired by requirement of inter-frame coding

Objective

- **Metric 1**: Require every order of arrival have same performance
  - Inspired by requirement of inter-frame coding
- **Metric 2**: Require best throughput as you decode in a feedback system

Original PBRL Code – Different Arrival Orderings

PBRL, Rate 8/11, k=16384

4 of 6 possible increment combinations at rate 8/11

FER

$E_b/N_0$

BI-AWGNC Capacity

S. V. S. Ranganathan
Protograph Codes, Rate Allocation
PhD Defense, 11/30/2018
UIO-PBRL Code Designed for Metric 1

UIO-PBRL, Rate 8/11, k=16384

FER

10^0

10^{-1}

10^{-2}

10^{-3}

10^{-4}

EB/N0

1.4 1.6 1.8 2 2.2 2.4 2.6 2.8

BI-AWGNC Capacity

Iterative threshold

6 codes
UIO-PBRL Code Designed for Metric 2

Randomly Permutated Order of Incremental Packets

Throughput vs. Channel SNR for Original PBRL, k=16384, UIO-PBRL Metric 1, UIO-PBRL Metric 2, and Original PBRL, in-order.
1. Incremental Redundancy (IR) and LDPC Codes
   - Rate-Compatible Codes, Protograph LDPC Codes
   - Protograph QC-LDPC Codes

2. Quasi-Cyclic PBRL Design for Short Block-Lengths
   - Protograph-Based Raptor-Like LDPC (PBRL) Codes
   - Permanent-Bound-Based Design (PBD) Method
   - Simulation Results

3. A Property of PBRL Decoding

4. PBRL Codes for Universal Increment Ordering
   - UIO-PBRL Codes

5. A Rate Allocation Problem
Problem Setup

packet 1  packet 2  \ldots  packet m \quad k \text{ nats per packet}
Problem Setup

Erasure coding

rate $R_E \leq 1$

$k$ nats per packet
Problem Setup

Erasure coding

packet 1

packet 2

\ldots

packet m

\text{transmit each packet with channel code, rate } R_C

\text{rate } R_E \leq 1

\text{k nats per packet}

\text{packet m}/R_E
Problem Setup

Erasure coding
rate $R_E \leq 1$

Transmit each packet with channel code, rate $R_C$

Overall $T$ channel time units to transmit
Problem Setup

Packet 1  Packet 2  ...  Packet m
          |          |      |  \[k \text{ nats per packet}\]
          |          |      |  \[R_E \leq 1\]
Packet 1  Packet 2  ...  Packet m  Packet m/R_E

Erasure coding

Transmit each packet with channel code, rate \( R_C \)

Overall \( T \) channel time units to transmit \[ \frac{k}{R_C} \] channel symbols per packet
Problem Setup

Erasure coding

Rate $R_E \leq 1$

Transmit each packet with channel code, rate $R_C$

Overall $T$ channel time units to transmit

$$\frac{k}{R_C} \cdot \frac{m}{R_E} = T$$
Problem Setup

Erasure coding

Transmit each packet with channel code, rate $R_C$

Overall $T$ channel time units to transmit

\[ \frac{k}{R_C} \cdot \frac{m}{R_E} = T \quad \Rightarrow \quad \frac{mk}{T} = R_E R_C \]
Questions

- Fixing a target probability of failure $\lambda$ and $T$, what is the minimum operating SNR at PHY
Questions

- Fixing a target probability of failure $\lambda$ and $T$, what is the minimum operating SNR at PHY?
- For the fixed $T$, what should be $R_C^*, R_E^*$?
Questions

- Fixing a target probability of failure $\lambda$ and $T$, what is the minimum operating SNR at PHY?
- For the fixed $T$, what should be $R_C^*$, $R_E^*$?
- How do $R_C^*$, $R_E^*$ behave as a function of $T$, as $T \to \infty$?
Rayleigh fading channel

Average SNR = \( P \)
Fading Model

Rayleigh fading channel
Average SNR = $P$

$X \in \mathcal{X}$

$Y = HX + Z$

$H$

$Z$ AWGN

Courtade and Wesel
Block fading

Channel codeword for
two values of $R_C$
Fading Model

Rayleigh fading channel
Average SNR = \( P \)

\[ X \in \mathcal{X} \rightarrow H \rightarrow Y = HX + Z \]

Courtade and Wesel
Block fading

Proportional-diversity
block fading
(PD block fading)

Channel codeword for
two values of \( R_C \)
Proportional-Diversity Block Fading

Overall rate $mk/T$

Optimal $R_E$

$m = \hat{m} = 64$, $\epsilon = 0.1$, $\lambda = 10^{-6}$

- $k/l_f = 1$
- $k/l_f = 2$
- $k/l_f = 5$
- $k/l_f = 10$

S. V. S. Ranganathan
Protograph Codes, Rate Allocation
PhD Defense, 11/30/2018
Theoretical Result

Theorem

Let the coding scheme use an arbitrary erasure code capable of producing any number of packets and a “very good” channel code. Let us assume a Rayleigh proportional-diversity block-fading channel.
Theoretical Result

Theorem

Let the coding scheme use an arbitrary erasure code capable of producing any number of packets and a “very good” channel code. Let us assume a Rayleigh proportional-diversity block-fading channel. Then, for any sufficiently large $T$ the optimal value of $R_E$ is equal to its highest possible value.
Conclusions

- We designed new PBRL codes for short block-lengths
Conclusions

- We designed new PBRL codes for short block-lengths
  - Demonstrated the effectiveness using a design for 5G
Conclusions

- We designed new PBRL codes for short block-lengths
  - Demonstrated the effectiveness using a design for 5G
- We studied a decoding property of PBRL codes
Conclusions

• We designed new PBRL codes for short block-lengths
  › Demonstrated the effectiveness using a design for 5G
• We studied a decoding property of PBRL codes
• We looked at an application of PBRL codes for universal increment ordering
Conclusions

- We designed new PBRL codes for short block-lengths
  - Demonstrated the effectiveness using a design for 5G
- We studied a decoding property of PBRL codes
- We looked at an application of PBRL codes for universal increment ordering
- We studied a cross-layer rate allocation problem
Conclusions

- We designed new PBRL codes for short block-lengths
  - Demonstrated the effectiveness using a design for 5G
- We studied a decoding property of PBRL codes
- We looked at an application of PBRL codes for universal increment ordering
- We studied a cross-layer rate allocation problem
  - We showed that erasure coding is unnecessary if there is enough diversity at the PHY layer
I would like to thank...

- Rick & Dariush
I would like to thank...

- Rick & Dariush
- Kasra Vakilinia & Haobo Wang
Acknowledgement

I would like to thank . . .

• Rick & Dariush
• Kasra Vakilinia & Haobo Wang
• Prof. Dolecek

• BZ Shen
• National Science Foundation (NSF), Broadcom Foundation, S. A. Photonics – for their funding support
• Engineering Graduate Students Association
I would like to thank…

- Rick & Dariush
- Kasra Vakilinia & Haobo Wang
- Prof. Dolecek
- My committee members – Profs. Fragouli, Diggavi, and Lu
Acknowledgement

I would like to thank...

- Rick & Dariush
- Kasra Vakilinia & Haobo Wang
- Prof. Dolecek
- My committee members – Profs. Fragouli, Diggavi, and Lu
- BZ Shen
I would like to thank...

- Rick & Dariush
- Kasra Vakilinia & Haobo Wang
- Prof. Dolecek
- My committee members – Profs. Fragouli, Diggavi, and Lu
- BZ Shen
- National Science Foundation (NSF), Broadcom Foundation, S. A. Photonics – for their funding support
I would like to thank…

- Rick & Dariush
- Kasra Vakilinia & Haobo Wang
- Prof. Dolecek
- My committee members – Profs. Fragouli, Diggavi, and Lu
- BZ Shen
- National Science Foundation (NSF), Broadcom Foundation, S. A. Photonics – for their funding support
- Engineering Graduate Students Association
Appendix – Permanent Upper Bound

**Theorem (Smarandache and Vontobel)**

Let a protomatrix $P$ with a positive design rate and no punctured variable nodes be of size $n_c \times n_v$. If $S \subseteq [n_v]$, denote by $P_S$ the sub-matrix of $P$ formed by the columns indexed by elements of $S$. Then, any QC-LDPC code $C$ obtained from the protomatrix $P$ has a minimum distance $d_{\text{min}}(C)$ that is upper bounded as

$$d_{\text{min}}(C) \leq \min^* \left\{ \sum_{i \in S} \text{perm} \left( P_{S \setminus i} \right) \right\},$$

(2)

where $|\cdot|$ refers to the cardinality of a set, $S \setminus i$ is shorthand for $S \setminus \{i\}$, and $\min^*$ returns the smallest non-zero value in a set of non-negative values with at least one positive value or $+\infty$ if the set is $\{0\}$. 
Appendix – Permanent Bound Example

\[ P = \begin{bmatrix}
2 & 1 & 2 & 1 & 2 & 1 & 0 & 0 & 0 \\
1 & 2 & 1 & 2 & 1 & 2 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix} \]

\[ S = \{1, 3, 4, 6, 7, 9\} \]
\[ P = \begin{bmatrix} 2 & 1 & 2 & 1 & 2 & 1 & 0 & 0 & 0 \\ 1 & 2 & 1 & 2 & 1 & 2 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \]

\[ S = \{1, 3, 4, 6, 7, 9\} \]

\[ \sum_{i \in S} \text{perm} (P_{S \setminus i}) \]
Appendix – Permanent Bound Example

\[ P = \begin{bmatrix}
2 & 1 & 2 & 1 & 2 & 1 & 0 & 0 & 0 \\
1 & 2 & 1 & 2 & 1 & 2 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix} \]

\[ S = \{1, 3, 4, 6, 7, 9\} \]

\[ \sum_{i \in S} \text{perm}(P_{S \setminus i}) = 10 \]
Appendix – Permanent Bound Example

\[ P = \begin{bmatrix}
2 & 1 & 2 & 1 & 2 & 1 & 0 & 0 & 0 \\
1 & 2 & 1 & 2 & 1 & 2 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix} \]

\[ S = \{1, 3, 4, 6, 7, 9\} \]

\[ \sum_{i \in S} \text{perm} \left( P_{S \setminus i} \right) = 10 + 10 \]
Appendix – Permanent Bound Example

\[ P = \begin{bmatrix}
2 & 1 & 2 & 1 & 2 & 1 & 0 & 0 & 0 \\
1 & 2 & 1 & 2 & 1 & 2 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
\end{bmatrix} \]

\[ S = \{1, 3, 4, 6, 7, 9\} \]

\[ \sum_{i \in S} \text{perm} \left( P_{S \setminus i} \right) = 10 + 10 + 4 \]
Appendix – Permanent Bound Example

\[ P = \begin{bmatrix}
2 & 1 & 2 & 1 & 2 & 1 & 0 & 0 & 0 \\
1 & 2 & 1 & 2 & 1 & 2 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix} \]

\[ S = \{1, 3, 4, 6, 7, 9\} \]

\[ \sum_{i \in S} \text{perm} \left( P_{S \setminus i}\right) = 10 + 10 + 4 + 4 \]
Appendix – Permanent Bound Example

\[ P = \begin{bmatrix}
2 & 1 & 2 & 1 & 2 & 1 & 0 & 0 & 0 \\
1 & 2 & 1 & 2 & 1 & 2 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix} \]

\[ S = \{1, 3, 4, 6, 7, 9\} \]

\[ \sum_{i \in S} \text{perm} \left( P_{S \setminus i} \right) = 10 + 10 + 4 + 4 + 20 \]
Appendix – Permanent Bound Example

\[ P = \begin{bmatrix}
2 & 1 & 2 & 1 & 2 & 1 & 0 & 0 & 0 \\
1 & 2 & 1 & 2 & 1 & 2 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix} \]

\[ S = \{1, 3, 4, 6, 7, 9\} \]

\[ \sum_{i \in S} \text{perm} \left( P_{S \setminus i} \right) = 68 \]

\[ \min^* \left( \{68, \ldots \} \right) \]
$P = \begin{bmatrix}
2 & 1 & 2 & 1 & 2 & 1 & 0 & 0 & 0 \\
1 & 2 & 1 & 2 & 1 & 2 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
\end{bmatrix}$

$S = \{1, 2, 3, 4, 5, 6\}$

$\sum_{i\in S} \text{perm} (P_{S\setminus i}) = 164$

$\min^* (\{68, 164, \ldots\})$
\[ P = \begin{bmatrix}
2 & 1 & 2 & 1 & 2 & 1 & 0 & 0 & 0 \\
1 & 2 & 1 & 2 & 1 & 2 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
\end{bmatrix} \]

\[ \binom{n_v}{n_c+1} \text{ sets of size } n_c + 1 \]

For each set \( n_c + 1 \) permanents of size \( n_c \times n_c \)

\[ \binom{n_v}{n_c+1}(n_c + 1) \text{ permanents of size } n_c \times n_c \]
Appendix – Permanent

- A a square matrix, \( \text{perm}(A) = \sum_{\sigma} \prod_j A(\sigma(j), j) \)
Appendix – Permanent

\[ A = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 4 \\ 2 & 1 & 2 \end{bmatrix} \in \mathbb{Z}^{3 \times 3} \]

\[ \text{perm}(A) = 8+ \]
Appendix – Permanent

$$A = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 4 \\ 2 & 1 & 2 \end{bmatrix} \in \mathbb{Z}^{3 \times 3}$$

$$\text{perm}(A) = 8 + 8$$
Appendix – Permanent

\[
A = \begin{bmatrix}
2 & 1 & 0 \\
1 & 2 & 4 \\
2 & 1 & 2 \\
\end{bmatrix} \in \mathbb{Z}^{3 \times 3}
\]

\[
\text{perm}(A) = 8 + 8 + 2
\]
Appendix – Permanent

\[ A = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 4 \\ 2 & 1 & 2 \end{bmatrix} \in \mathbb{Z}^{3 \times 3} \]

\[ \text{perm}(A) = 8 + 8 + 2 + 0 \]
Appendix – Permanent

\[ A = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 4 \\ 2 & 1 & 2 \end{bmatrix} \in \mathbb{Z}^{3 \times 3} \]

\[ \text{perm}(A) = 8 + 8 + 2 + 0 + 8 \]
Appendix – Permanent

\[ A = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 4 \\ 2 & 1 & 2 \end{bmatrix} \in \mathbb{Z}^{3 \times 3} \]

\[ \text{perm}(A) = 26 \]
Best algorithm (Ryser) is of complexity $\Theta(\ell \cdot 2^\ell)$ for matrix of size $\ell \times \ell$