# Optimisation of parity-check matrices of LDPC codes

**Yauhen Yakimenka**, Vitaly Skachek
Institute of Computer Science, University of Tartu

Ratnieki, Latvia October 5, 2014

# Outline

- Quick intro into coding theory

# Outline

- Quick intro into coding theory
- Problem description

# Outline

- Quick intro into coding theory
- Problem description
- Existing results

# Outline

- Quick intro into coding theory
- Problem description
- Existing results
- Our contribution

# Outline

- **Quick intro into coding theory**
- Problem description
- Existing results
- Our contribution

# Communication model

- Noisy channel model

$$(\Sigma_{in}, \Sigma_{out}, Prob)$$

$$Prob(a, b) = \mathsf{P}\left\{b \text{ received} \,|\, a \text{ transmitted}\right\}$$

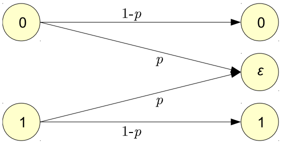
# Communication model

- Noisy channel model

$$(\Sigma_{in}, \Sigma_{out}, Prob)$$

$$Prob(a, b) = \mathsf{P}\left\{b \text{ received} \,|\, a \text{ transmitted}\right\}$$

- Binary erasure channel (BEC)



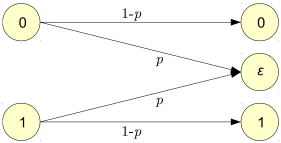
# Communication model

- Noisy channel model

$$(\Sigma_{in}, \Sigma_{out}, Prob)$$

$$Prob(a,b) = \mathsf{P}\left\{b \text{ received} \,|\, a \text{ transmitted}\right\}$$

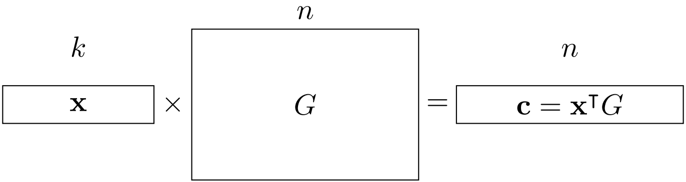
- Binary erasure channel (BEC)




- Example: $0110111 \to 01\varepsilon0\varepsilon11$

# Linear (binary) block codes

Generator matrix




Linear code $\mathcal{C}$ is a subspace of $\mathbb{F}_2^n$
$G$ is not unique (every basis of subspace will work)
$\mathcal{C}$ is denoted as $[n, k, d]$

# Dual code

Dual code $\mathcal{C}^\perp$ is orthogonal compliment of $\mathcal{C}$

$$\mathcal{C}^\perp = \{\mathbf{c}' \in \mathbb{F}_2^n : \mathbf{c}' G^\intercal = 0\}$$

$\mathcal{C}^\perp$ is also a binary linear code: $[n, n-k, d^\perp]$

# Parity-check matrix

- Another way to define $\mathcal{C}$

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

# Parity-check matrix

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

- Another way to define $\mathcal{C}$
- $\mathcal{C}$ is a space of solutions $\mathbf{c}$ of

$$H\mathbf{c}^\intercal = 0$$

# Parity-check matrix

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

- Another way to define $\mathcal{C}$
- $\mathcal{C}$ is a space of solutions $\mathbf{c}$ of

$$H\mathbf{c}^\intercal = 0$$

- $\operatorname{rank} H = n - k$ (but $H$ can have more rows, which are redundant)

# Parity-check matrix

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

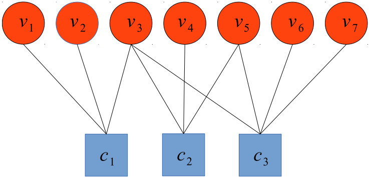
- Another way to define $\mathcal{C}$
- $\mathcal{C}$ is a space of solutions $\mathbf{c}$ of

$$H\mathbf{c}^{\mathsf{T}} = 0$$

- $\operatorname{rank} H = n - k$ (but $H$ can have more rows, which are redundant)
- $H$ is any matrix of rank $n - k$ whose rows are codewords in $\mathcal{C}^{\perp}$

# Tanner graph

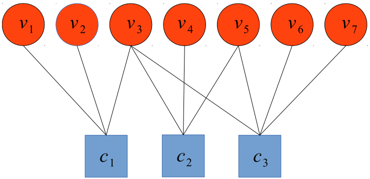
$$H = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix}$$




- Bipartite graph: columns and rows of $H$

# Tanner graph

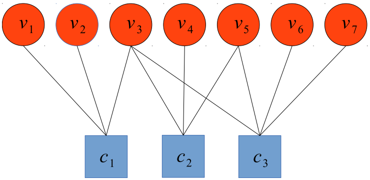
$$H = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix}$$




- Bipartite graph: columns and rows of $H$
- Edge present if element in $H$ is 1

# Tanner graph

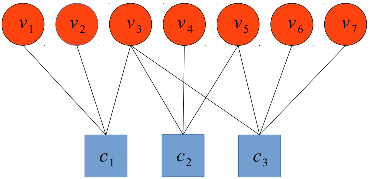
$$H = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix}$$




- Bipartite graph: columns and rows of $H$
- Edge present if element in $H$ is 1
- Variable nodes $(v_1, v_2, \ldots)$ represent bits of codeword

# Tanner graph

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




- Bipartite graph: columns and rows of $H$
- Edge present if element in $H$ is 1
- Variable nodes $(v_1, v_2, \dots)$ represent bits of codeword
- Check nodes $(c_1, c_2, \dots)$ represent parity requirements

# Outline

- Quick intro into coding theory
- **Problem description**
- Existing results
- Our contribution

# Iterative decoding on BEC

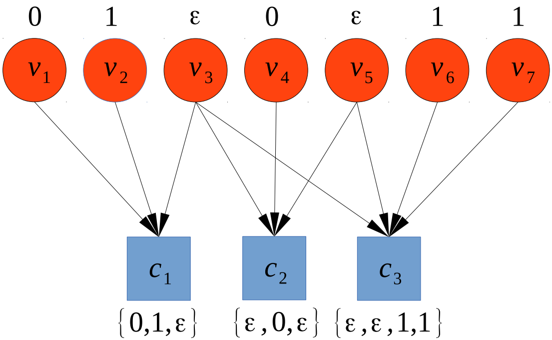



Figure : Step 1 of 4

# Iterative decoding on BEC

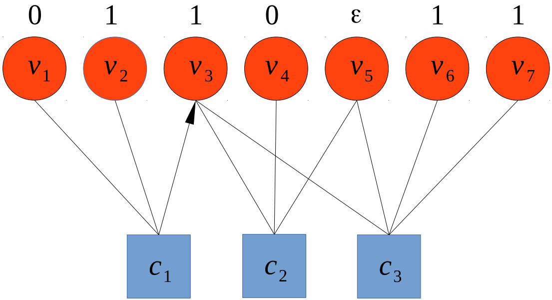



Figure : Step 2 of 4

# Iterative decoding on BEC

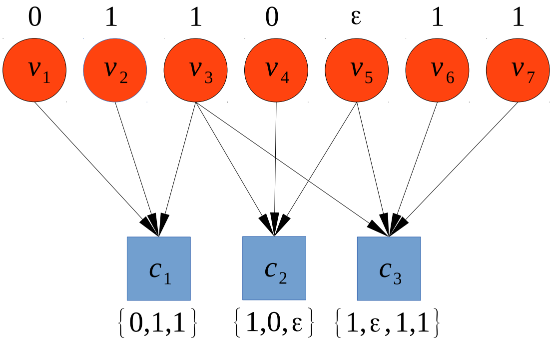



Figure : Step 3 of 4

# Iterative decoding on BEC

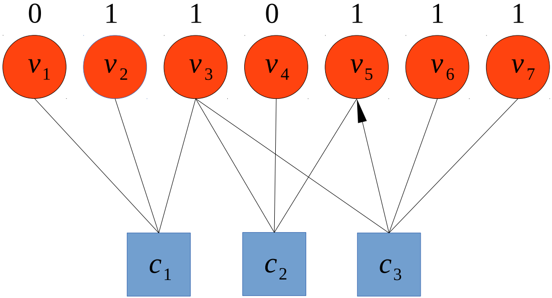
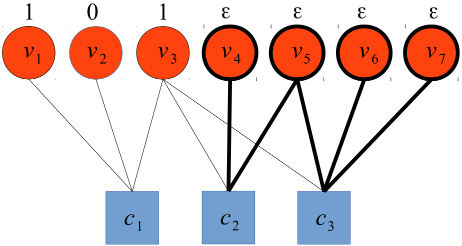



Figure : Step 4 of 4

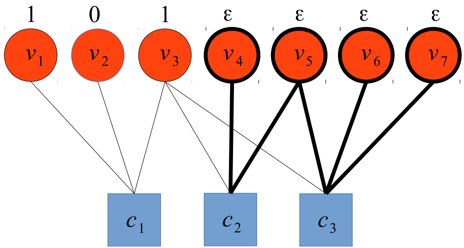
# Stopping sets

- In Tanner graph



# Stopping sets

- In Tanner graph




- In parity-check matrix

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

# Stopping sets

- Decoder fails $\Leftrightarrow$ stopping set is erased

# Stopping sets

- Decoder fails $\Leftrightarrow$ stopping set is erased
- Stopping sets undesirable

# Stopping sets

- Decoder fails $\Leftrightarrow$ stopping set is erased
- Stopping sets undesirable
- Erasure of stopping sets of small size is more probable

# Stopping sets

- Decoder fails $\Leftrightarrow$ stopping set is erased
- Stopping sets undesirable
- Erasure of stopping sets of small size is more probable
- (!) Stopping sets are defined for $H$, not for $\mathcal{C}$

# Stopping sets

- Decoder fails $\Leftrightarrow$ stopping set is erased
- Stopping sets undesirable
- Erasure of stopping sets of small size is more probable
- (!) Stopping sets are defined for $H$, not for $\mathcal{C}$
- Additional (redundant) rows could eliminate stopping sets

# Stopping sets

- Decoder fails $\Leftrightarrow$ stopping set is erased
- Stopping sets undesirable
- Erasure of stopping sets of small size is more probable
- (!) Stopping sets are defined for $H$, not for $\mathcal{C}$
- Additional (redundant) rows could eliminate stopping sets
- Every codeword of $\mathcal{C}$ induces stopping set $\Rightarrow$ those not possible to eliminate

## Stopping sets

- Decoder fails $\Leftrightarrow$ stopping set is erased
- Stopping sets undesirable
- Erasure of stopping sets of small size is more probable
- (!) Stopping sets are defined for $H$, not for $\mathcal{C}$
- Additional (redundant) rows could eliminate stopping sets
- Every codeword of $\mathcal{C}$ induces stopping set $\Rightarrow$ those not possible to eliminate
- Idea: use redundant $H$ which eliminates all stopping sets of size $< d$

# Stopping sets

- Decoder fails $\Leftrightarrow$ stopping set is erased
- Stopping sets undesirable
- Erasure of stopping sets of small size is more probable
- (!) Stopping sets are defined for $H$, not for $\mathcal{C}$
- Additional (redundant) rows could eliminate stopping sets
- Every codeword of $\mathcal{C}$ induces stopping set $\Rightarrow$ those not possible to eliminate
- Idea: use redundant $H$ which eliminates all stopping sets of size $< d$
- **Stopping redundancy** $\rho(\mathcal{C})$ is the minimum number of rows in $H$ s.t. there are no stopping of size $< d$

# Stopping sets

- Decoder fails $\Leftrightarrow$ stopping set is erased
- Stopping sets undesirable
- Erasure of stopping sets of small size is more probable
- (!) Stopping sets are defined for $H$, not for $\mathcal{C}$
- Additional (redundant) rows could eliminate stopping sets
- Every codeword of $\mathcal{C}$ induces stopping set $\Rightarrow$ those not possible to eliminate
- Idea: use redundant $H$ which eliminates all stopping sets of size $< d$
- **Stopping redundancy** $\rho(\mathcal{C})$ is the minimum number of rows in $H$ s.t. there are no stopping of size $< d$
- Always achievable: $\rho(\mathcal{C}) \leq 2^{n-k} - 1$

# Example

$$H = \begin{array}{c} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \end{array} \begin{pmatrix} \mathbf{1} & 1 & \mathbf{0} & 1 & 1 & 0 & 1 & 1 & 1 & \mathbf{1} \\ \mathbf{1} & 1 & \mathbf{0} & 1 & 1 & 0 & 0 & 0 & 1 & \mathbf{1} \\ \mathbf{1} & 0 & \mathbf{1} & 0 & 0 & 1 & 0 & 1 & 0 & \mathbf{1} \\ \mathbf{0} & 0 & \mathbf{1} & 1 & 1 & 0 & 1 & 1 & 0 & \mathbf{1} \\ \mathbf{0} & 0 & \mathbf{1} & 0 & 1 & 0 & 0 & 1 & 1 & \mathbf{1} \\ \mathbf{0} & 0 & \mathbf{0} & 1 & 1 & 0 & 0 & 1 & 0 & \mathbf{0} \\ \mathbf{1} & 0 & \mathbf{0} & 1 & 0 & 0 & 1 & 1 & 0 & \mathbf{1} \end{pmatrix}$$

Stopping sets of size $< d$:
$\{1, 3, 10\}$, $\{1, 5, 8\}$, $\{4, 8, 10\}$, $\{5, 8, 10\}$.

# Example

$$
H' = \begin{array}{l} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \\ c_1 + c_2 + c_3 \\ c_2 + c_3 \end{array}
\begin{pmatrix}
\mathbf{1} & 1 & \mathbf{0} & 1 & 1 & 0 & 1 & 1 & 1 & \mathbf{1} \\
\mathbf{1} & 1 & \mathbf{0} & 1 & 1 & 0 & 0 & 0 & 1 & \mathbf{1} \\
\mathbf{1} & 0 & \mathbf{1} & 0 & 0 & 1 & 0 & 1 & 0 & \mathbf{1} \\
\mathbf{0} & 0 & \mathbf{1} & 1 & 1 & 0 & 1 & 1 & 0 & \mathbf{1} \\
\mathbf{0} & 0 & \mathbf{1} & 0 & 1 & 0 & 0 & 1 & 1 & \mathbf{1} \\
\mathbf{0} & 0 & \mathbf{0} & 1 & 1 & 0 & 0 & 1 & 0 & \mathbf{0} \\
\mathbf{1} & 0 & \mathbf{0} & 1 & 0 & 0 & 1 & 1 & 0 & \mathbf{1} \\
\mathbf{1} & 0 & \mathbf{1} & 0 & 0 & 1 & 1 & 0 & 0 & \mathbf{1} \\
\mathbf{0} & 1 & \mathbf{1} & 1 & 1 & 1 & 0 & 1 & 1 & \mathbf{0}
\end{pmatrix}
$$

No small stopping sets $\Rightarrow \rho(\mathcal{C}) \leq 9$

# Outline

- Quick intro into coding theory
- Problem description
- **Existing results**
- Our contribution

## Han-Siegel-Vardy'08

Probabilistic approach

# Han-Siegel-Vardy'08

Probabilistic approach

$t$ codewords

Step 1

- Choose $t$ random codewords of $\mathcal{C}^{\perp}$ (no repetition)

# Han-Siegel-Vardy'08

Probabilistic approach

$t$ codewords

Step 1

- Choose $t$ random codewords of $\mathcal{C}^{\perp}$ (no repetition)
- Find expectations of

# Han-Siegel-Vardy'08

Probabilistic approach

$t$ codewords

Step 1

- Choose $t$ random codewords of $\mathcal{C}^{\perp}$ (no repetition)
- Find expectations of
  - number of stopping sets left

# Han-Siegel-Vardy'08

Probabilistic approach

| |
|---|
| $t$ codewords |

Step 1

- Choose $t$ random codewords of $\mathcal{C}^{\perp}$ (no repetition)
- Find expectations of
  - number of stopping sets left
  - rank deficiency

# Han-Siegel-Vardy'08

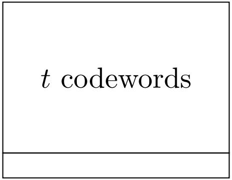
Probabilistic approach

$t$ codewords

Step 1

- Choose $t$ random codewords of $\mathcal{C}^{\perp}$ (no repetition)
- Find expectations of
  - number of stopping sets left
  - rank deficiency
- Guaranteed existence

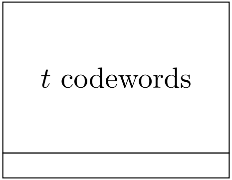
# Han-Siegel-Vardy'08




Step 2

- Add one more random codeword of $\mathcal{C}^{\perp}$ (no repetition)

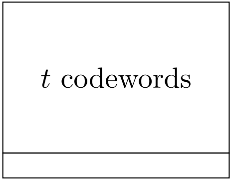
# Han-Siegel-Vardy'08




Step 2

- Add one more random codeword of $\mathcal{C}^\perp$ (no repetition)
- Find expectations of

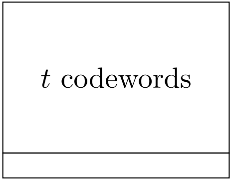
# Han-Siegel-Vardy'08




Step 2

- Add one more random codeword of $\mathcal{C}^\perp$ (no repetition)
- Find expectations of
  - *decrease* of number of stopping sets left

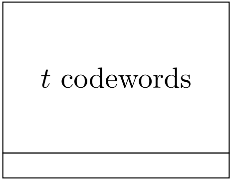
# Han-Siegel-Vardy'08




Step 2

- Add one more random codeword of $\mathcal{C}^{\perp}$ (no repetition)
- Find expectations of
  - *decrease* of number of stopping sets left
  - *increase* of rank

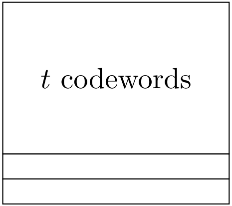
# Han-Siegel-Vardy'08




Step 2

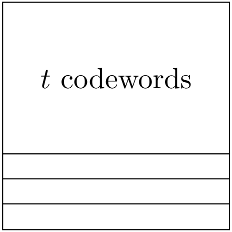
- Add one more random codeword of $\mathcal{C}^{\perp}$ (no repetition)
- Find expectations of
  - *decrease* of number of stopping sets left
  - *increase* of rank
- Guaranteed existence

# Han-Siegel-Vardy’08



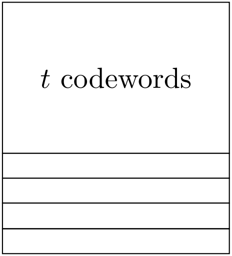

- Iterate

# Han-Siegel-Vardy'08



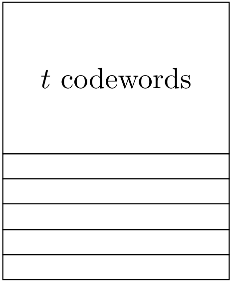

- Iterate
- Iterate

# Han-Siegel-Vardy'08




- Iterate
- Iterate
- Iterate

# Han-Siegel-Vardy'08




- Iterate
- Iterate
- Iterate
- Stop when there are no small stopping sets left and rank is $n - k$

# Outline

- Quick intro into coding theory
- Problem description
- Existing results
- **Our contribution**

# Main trick

$\tau$, non-random

- Choose some first row(s) non-randomly and carefully

# Main trick

$\tau$, non-random

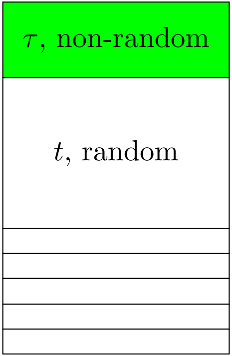
- Choose some first row(s) non-randomly and carefully
- ...so that we know how many stopping sets left

# Main trick

$\tau$, non-random

- Choose some first row(s) non-randomly and carefully
- ...so that we know how many stopping sets left
- ...or can bound their number

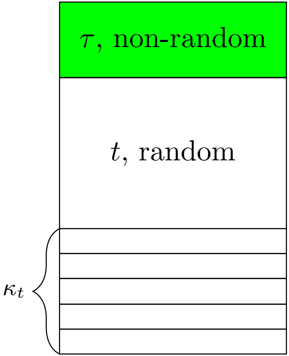
# Main trick




- Choose some first row(s) non-randomly and carefully
- ...so that we know how many stopping sets left
- ...or can bound their number
- And then apply technique of Han-Siegel-Vardy

# Main theorem

**Theorem:** $\rho(\mathcal{C}) \leq \tau + \min_{t \geq r} \{t + \kappa_t\}$
where:




$$
\begin{aligned}
\kappa_t &= \min \left\{ k \in \mathbb{N} : Q_k(\lfloor \mathcal{D}_t \rfloor) = 0 \right\} \\
Q_k(x) &= P_k(P_{k-1}(\dots P_1(x)\dots)) \\
P_j(x) &= \left\lfloor x \left( 1 - \frac{(d-1)2^{r-d+1}}{2^r - (t + \tau + j)} \right) \right\rfloor \\
\mathcal{D}_t &= \sum_{i=1}^{d-1} u_i \prod_{j=\tau+1}^{t+\tau} \left( 1 - \frac{i 2^{r-i}}{2^r - j} \right) \\
&\quad + \frac{1}{2^{t-r}} \left( 1 + \frac{2/3}{2^{t-r+1} - 1} \right)
\end{aligned}
$$

# Candidates for non-random rows

Just take some codewords of $\mathcal{C}^{\perp}$, calculate straightforward.
E.g. some conventional $H$ or some rows of it.

# Candidates for non-random rows

Just take some codewords of $\mathcal{C}^{\perp}$, calculate straightforward.
E.g. some conventional $H$ or some rows of it.
$\Rightarrow$ too slow! (and not always good results for our method)

# Candidates for non-random rows

Matrix of rows of weight $d^{\perp}$.
One such row

| | |
|---|---|
| $d^{\perp}$ | $n - d^{\perp}$ |

# Candidates for non-random rows

Matrix of rows of weight $d^\perp$.
One such row

| | |
|---|---|
| $d^\perp$ | $n - d^\perp$ |

covers(=eliminates) so many SS's of size $i$ $(i = 1, 2, \ldots, d-1)$:

$$d^\perp \binom{n - d^\perp}{i - 1}$$

# Candidates for non-random rows

Matrix of rows of weight $d^{\perp}$.
One such row

| $d^{\perp}$ | $n - d^{\perp}$ |
|---|---|

covers(=eliminates) so many SS's of size $i$ $(i = 1, 2, \ldots, d-1)$:

$$d^{\perp} \binom{n - d^{\perp}}{i - 1}$$

We could generalise to $\tau$ different rows of weight $d^{\perp}$ using *principle of inclusion-exclusion (PIE)*.

# Numerical results

Table : Upper bounds on the stopping redundancy

| | [24, 12, 8] Golay | [48, 24, 12] QR |
|---|---|---|
| Schwartz-Vardy'06, Th4 | 2509 | 4 540 385 |
| Han-Siegel-Vardy'08, Th1 | 198 | 3655 |
| Han-Siegel-Vardy'08, Th3 | 194 | 3655 |
| Han-Siegel-Vardy'08, Th4 | 187 | 3577 |
| Han-Siegel-Vardy'08, Th7 | 182 | 3564 |

# Numerical results

Table : Upper bounds on the stopping redundancy

| | [24, 12, 8] Golay | [48, 24, 12] QR |
|---|---|---|
| Schwartz-Vardy'06, Th4 | 2509 | 4 540 385 |
| Han-Siegel-Vardy,Th1 | 198 | 3655 |
| Han-Siegel-Vardy,Th3 | 194 | 3655 |
| Han-Siegel-Vardy,Th4 | 187 | 3577 |
| Han-Siegel-Vardy,Th7 | 182 | 3564 |
| $\tau = 1$ | 180 | 3538 |

# Numerical results

Table : Upper bounds on the stopping redundancy

| | [24, 12, 8] Golay | [48, 24, 12] QR |
|---|---|---|
| Schwartz-Vardy'06, Th4 | 2509 | 4 540 385 |
| Han-Siegel-Vardy,Th1 | 198 | 3655 |
| Han-Siegel-Vardy,Th3 | 194 | 3655 |
| Han-Siegel-Vardy,Th4 | 187 | 3577 |
| Han-Siegel-Vardy,Th7 | 182 | 3564 |
| $\tau = 1$ | 180 | 3538 |
| $\tau = 2$ | 176 | 3509 |

# Numerical results

Table : Upper bounds on the stopping redundancy

| | [24, 12, 8] Golay | [48, 24, 12] QR |
|---|---|---|
| Schwartz-Vardy'06, Th4 | 2509 | 4 540 385 |
| Han-Siegel-Vardy,Th1 | 198 | 3655 |
| Han-Siegel-Vardy,Th3 | 194 | 3655 |
| Han-Siegel-Vardy,Th4 | 187 | 3577 |
| Han-Siegel-Vardy,Th7 | 182 | 3564 |
| $\tau = 1$ | 180 | 3538 |
| $\tau = 2$ | 176 | 3509 |
| $\tau = 3$ | 172 | 3477 |
| ... | | |

# Acknoledgements

- Norwegian-Estonian Research Cooperation Programme (grant EMP133)
- Estonian Research Council (grant IUT2-1)

# Dziakuj