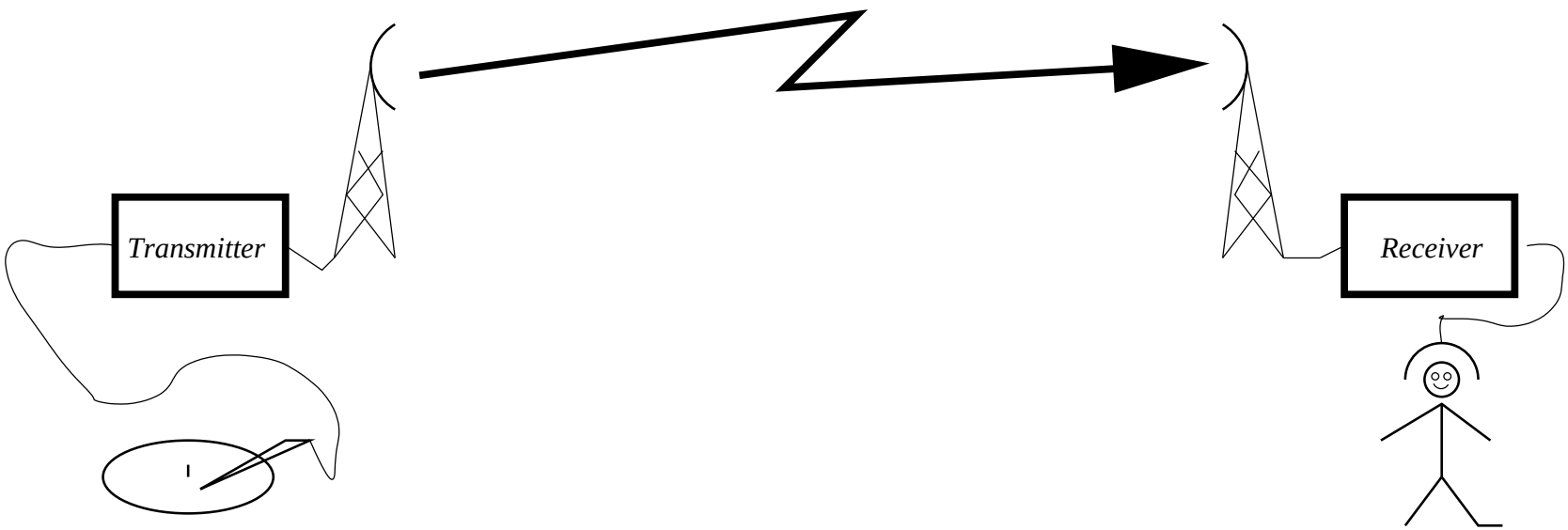
# Iterative Coding Techniques, Pseudocodewords, and their Relationship

Ralf Koetter Pascal O. Vontobel
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
email: {koetter,vontobel}@uiuc.edu

# The problem




Reliable transmission of information

# Decoding Algorithms


Assume that the codeword $\mathbf{x} \in \mathcal{C}$ was sent, the word $\mathbf{y} \in \mathscr{Y}^n$ was received, and based on $\mathbf{y}$ we would like to find the "most likely" transmitted codeword $\hat{\mathbf{x}}$.

- Symbol-wise MAP decoding gives

$$\hat{x}_i(\mathbf{y}) = \underset{x_i \in \mathscr{X}}{\operatorname{argmax}} P_{X_i|\mathbf{y}}(x_i|\mathbf{y}) \quad \text{(for each } i = 1, \ldots, n\text{)}.$$

- Block-wise MAP decoding gives

$$\hat{\mathbf{x}}(\mathbf{y}) = \underset{\mathbf{x} \in \mathcal{C}}{\operatorname{argmax}} P_{\mathbf{x}|\mathbf{y}}(\mathbf{x}|\mathbf{y}).$$

# Low-Density Parity-Check (LDPC) code

**Parity-check matrix:** $H$, $r \times n$ matrix

**Low-density:** $H$ is sparse

**Codes:**

$$\mathcal{C} = \left\{ \mathbf{c} \in \mathbb{F}_2 | H\mathbf{c}^T = \mathbf{0}^T \right\}$$

**Received:** $\mathbf{y} \in \mathscr{Y}$

**Problem:** Find the "most likely" transmitted word $\mathbf{c}$ .

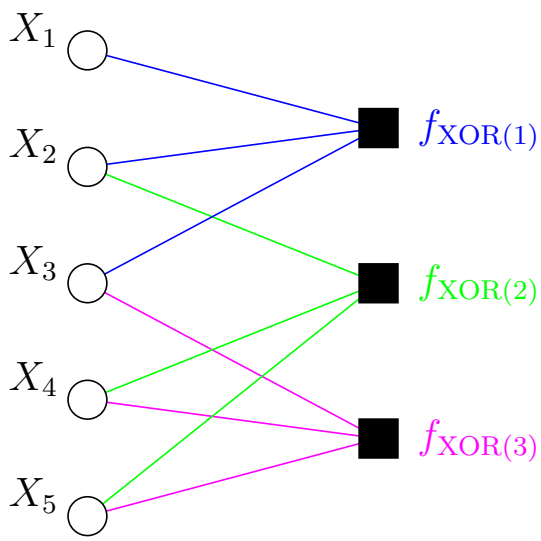
**Binary codes:** Log-likelihood ratios $\lambda_i = \ln \frac{\Pr(c_i=0|y_i)}{\Pr(c_i=1|y_i)}$

$$\boldsymbol{\lambda} = (\lambda_0, \lambda_1, \ldots, \lambda_{n-1})$$

# Tanner/Factor Graph of an LDPC Code

Example:

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

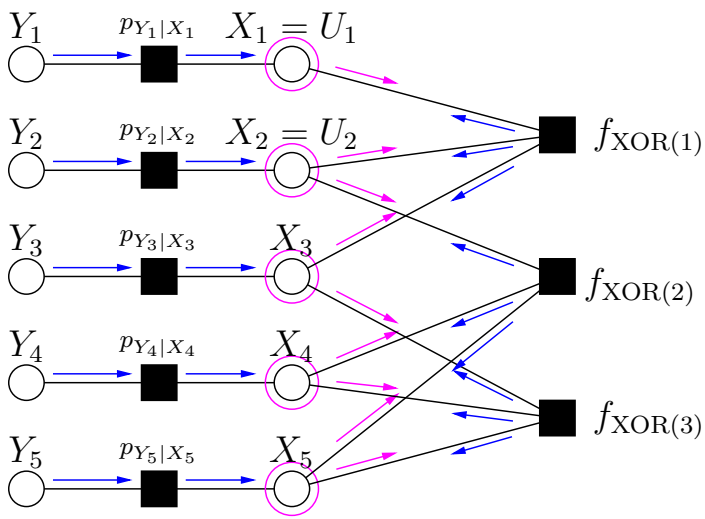



LDPC codes in general:

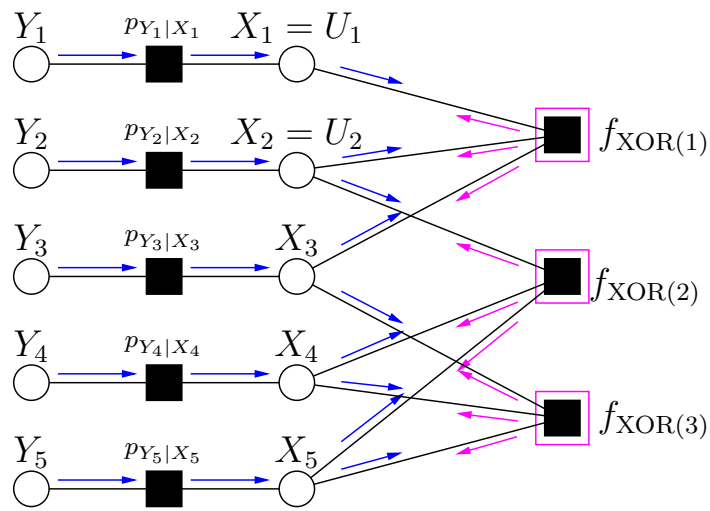
- An LDPC code has a matrix with very few ones.
- $(j, k)$-regular LDPC code: all bit nodes have degree $j$ and all check nodes have degree $k$.
- One can show that factor/Tanner graphs of good codes have cycles (under the assumption of bounded state-space sizes). (?)

# Message-Passing Decoding Algorithms

$i$-th iteration

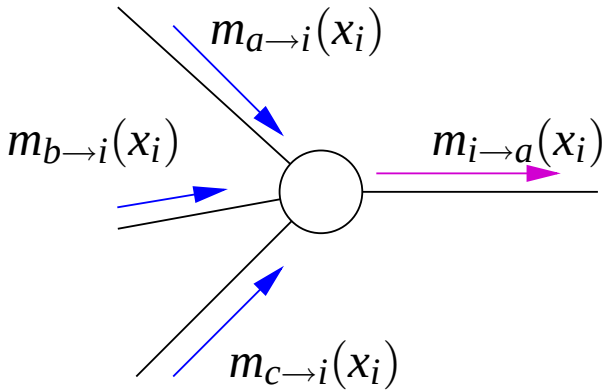


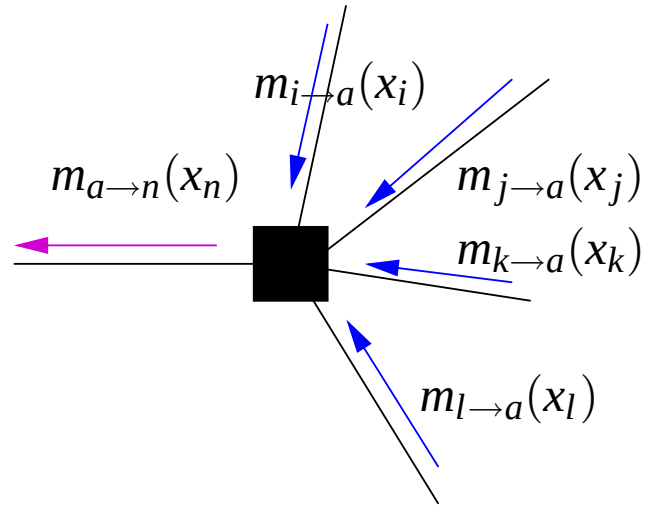

$i.5$-th iteration




A message-passing algorithm: all operations are performed locally!

# Message-Passing Decoding Algorithms




$$m_{i\to a}(x_i) = \prod_{s\in\Gamma(i)\setminus a} m_{s\to i}(x_i)$$

$$m_{a\to n}(x') = \sum_{\mathbf{x}:x_n=x'} f_a(\mathbf{x}) \prod_{s\in\Gamma(a)\setminus n} m_{s\to a}(x_s)$$

## Analysis of message passing algorithms in finite length graphs

Wiberg 1996, (pseudo-codewords, computation tree, pseudodistance, deviation set)

Horn 1999, pseudo-codewords, cycle codes

Forney et al., 2001 (extensions to other channels, tail-biting-trellis)

Frey et al., 2001 (signal-space interpretation of iterative decoding)

Di et al., 2002 (stopping sets, erasure channel)

MacKay et al. 2002 (near codewords)

Feldman 2003, (linear progamming decoding)

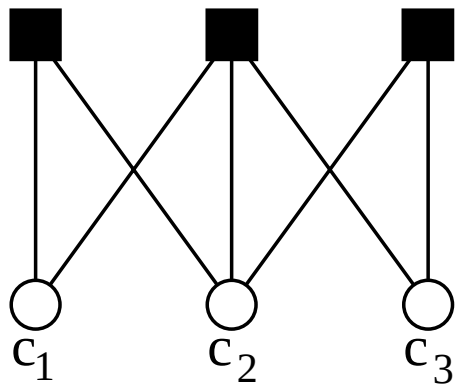
Richardson 2003, (trapping sets)

Wainwright, Jordan 2003, (belief propagation)

Yedidia, Freeman, Weiss 2004 (Bethe free energy)

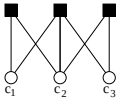
and enormous anecdotal evidence.....

## A simple example

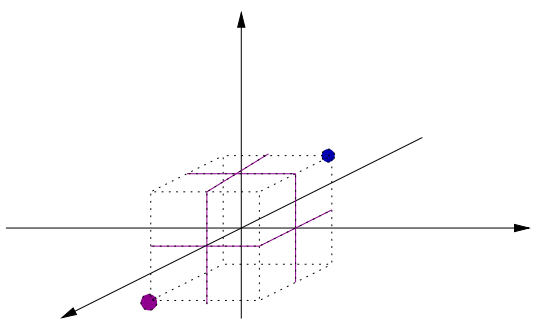


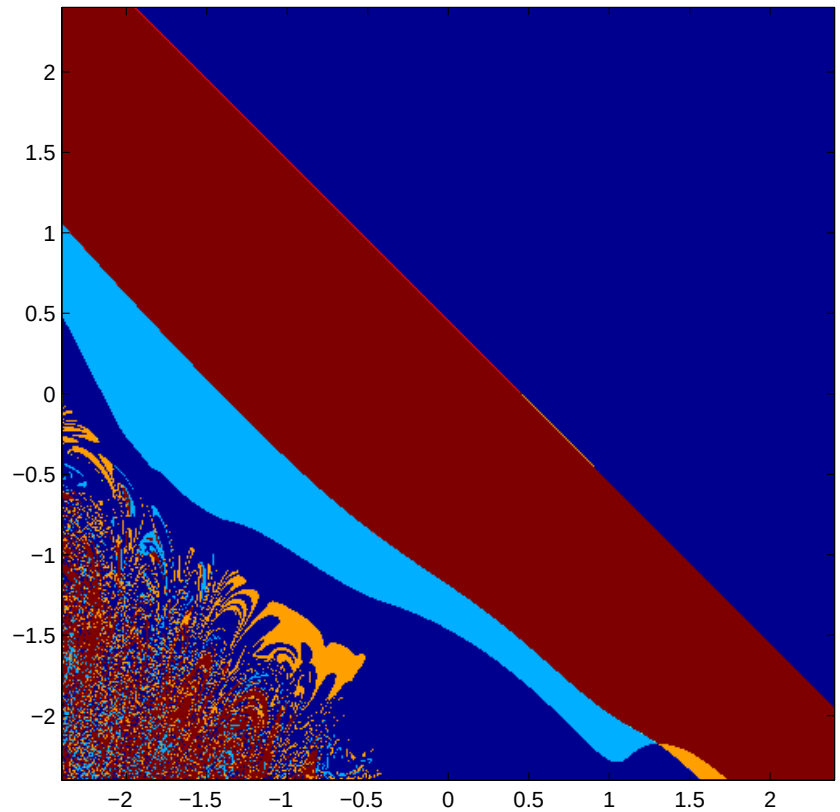

$$H = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$

$\mathcal{C} = \{(0,0,0)\}$, $d_H = \infty$, ML decision is always $(0,0,0)$



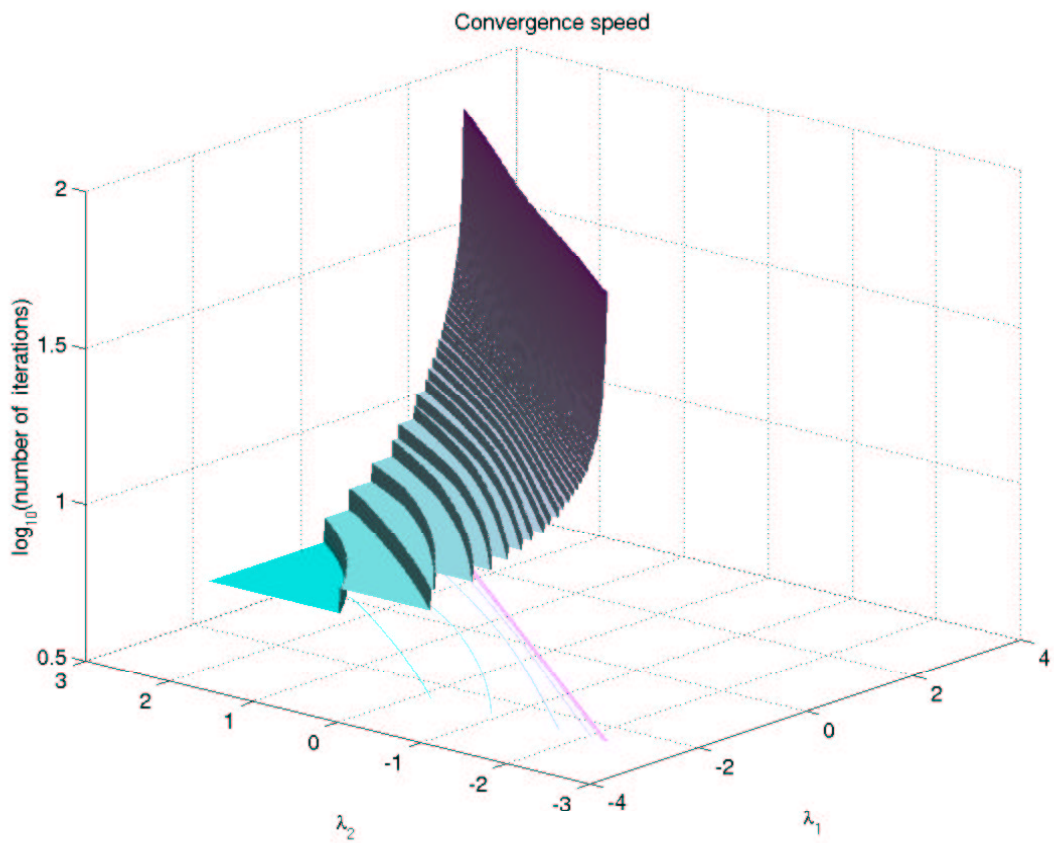
continued... Decision regions ($\lambda_3 = -0.45$):





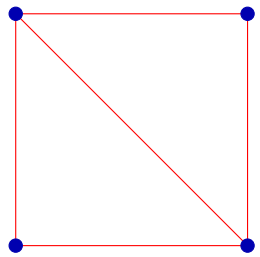

Decision region is described by $\lambda_1 + \lambda_2 + \lambda_3 > 0$

continued... Convergence speed:

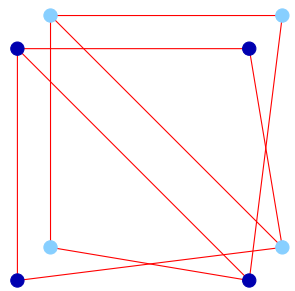



The convergence speed towards the plane $\lambda_1 + \lambda_2 + \lambda_3 = 0$

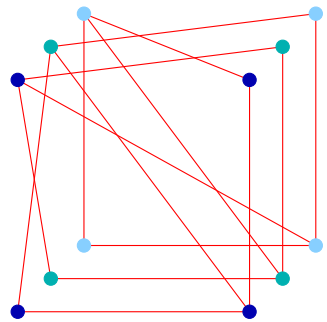
# Graph Covers



original graph



(a possible)
double cover of
the original graph

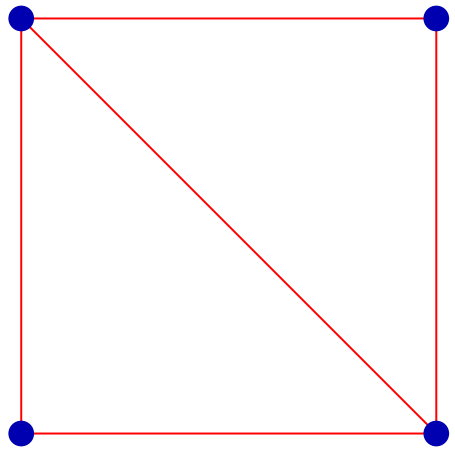


(a possible)
triple cover of
the original graph

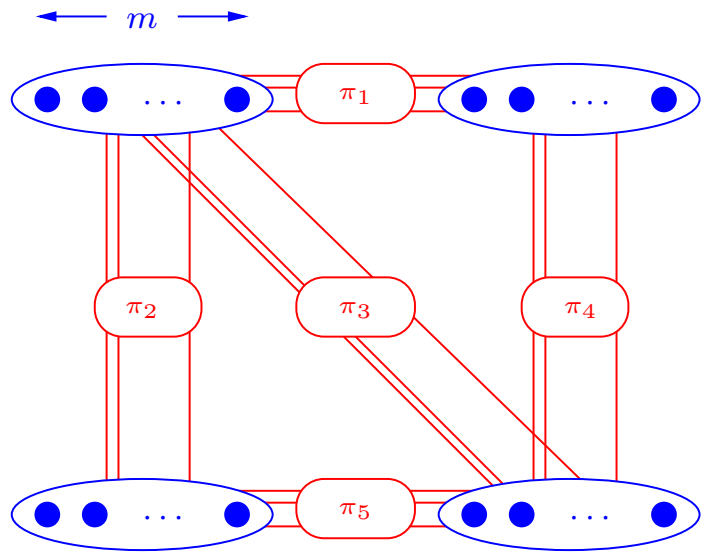
...

Besides double covers, a graph also has many triple covers, quadruple covers, quintuple covers, etc.

# Graph Covers



original graph

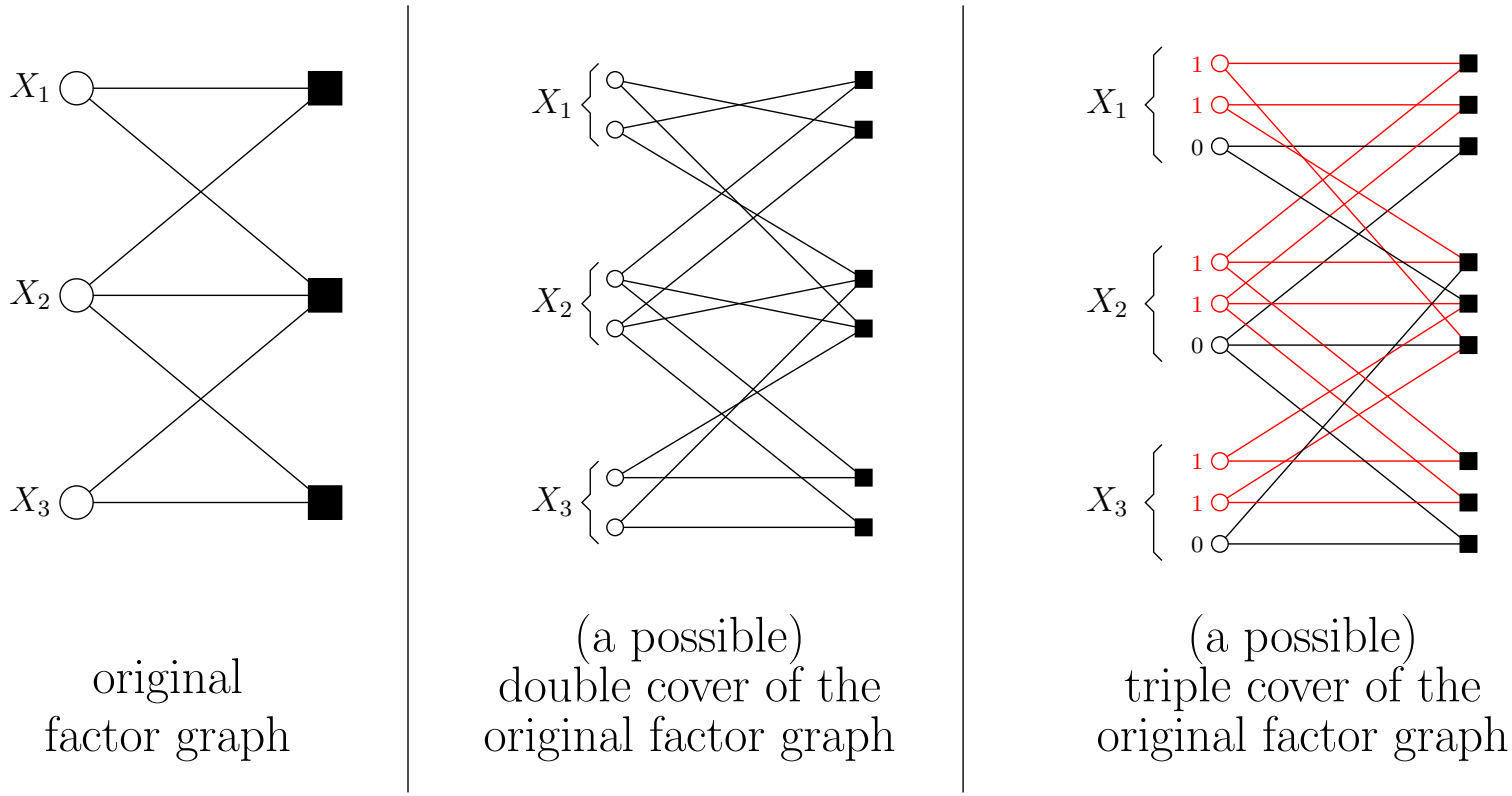



(possible)
$m$-fold cover of
original graph

An $m$-fold cover is also called a cover of degree $m$.

Note: there are many possible $m$-fold covers of a graph.

# A Simple Example



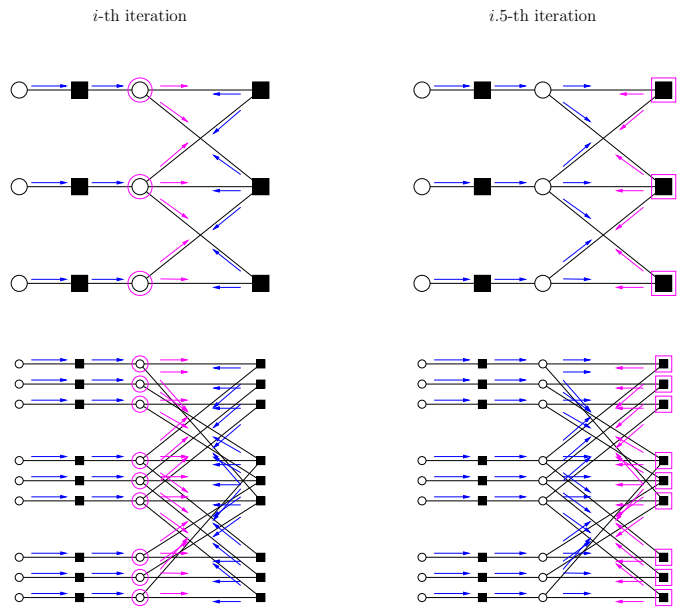

original factor graph

(a possible) double cover of the original factor graph

(a possible) triple cover of the original factor graph

$\lambda_1 + \lambda_2 + \lambda_3 < 0 \Rightarrow$ The indicated (valid) configuration in the triple cover has a larger likelihood than the the all-zeros configuration.

# A Simple Example




A locally operating decoding algorithm cannot distinguish if it is decoding on the original factor graph or on any of its covers.

The messages in the triple cover factor graph correspond to three identical copies of the messages in the original factor graph.

The original code $\mathcal{C}$ has length $n$

Code of length $mn$ in a degree $m$ graph cover: $\widehat{\mathcal{C}}$ with codewords $\widehat{\mathbf{c}}$

Any bit $c_i$ is lifted to a set of bits $\widehat{c}_{i,l}, l = 1, 2, \ldots, m$

The received values $y_i$ are lifted to $\widehat{y}_{i,l} = y_i$

Consider the two codewords case $\widehat{\mathbf{0}}, \widehat{\mathbf{c}} \neq \mathbf{0}$.

$$\log \frac{Pr(\widehat{\mathbf{0}}|\widehat{\mathbf{y}})}{Pr(\widehat{\mathbf{c}}|\widehat{\mathbf{y}})} = \sum_{i=1}^{n} \sum_{l=1}^{m} \log \frac{Pr(\widehat{0}_{i,l}|\widehat{y}_i)}{Pr(\widehat{c}_{i,l}|\widehat{y}_i)} = \langle \left| \left\{ \ell \mid \widehat{c}_{i,\ell} = 1 \right\} \right|, \boldsymbol{\lambda} \rangle$$

All we need to know is how often do the bits in a cover assume the value 1 or 0?

$$\omega_i(\widehat{\mathbf{c}}) \stackrel{\text{def}}{=} \frac{|\{\ell : \widehat{c}_{i,\ell} = 1\}|}{m},$$

$$\log \frac{Pr(\mathbf{0}|\widehat{\mathbf{y}})}{Pr(\widehat{\mathbf{c}}|\widehat{\mathbf{y}})} > 0 \Longleftrightarrow \langle \boldsymbol{\omega}(\widehat{\mathbf{c}}), \boldsymbol{\lambda} \rangle > 0$$

Pseudo-Codewords: $\boldsymbol{\omega}(\widehat{\mathbf{c}}) = (\omega_1(\widehat{\mathbf{c}}), \omega_2(\widehat{\mathbf{c}}), \ldots, \omega_{n-1}(\widehat{\mathbf{c}}))$

In any <u>locally operating</u> message passing algorithm the set of pseudo-codewords competes with codewords for being the "best" solution !

## The effect of $\langle \boldsymbol{\omega}(\hat{\mathbf{c}}), \boldsymbol{\lambda} \rangle \overset{?}{>} 0$ in different channels

Erasure Channel: $\lambda_i \in \{0, \infty\}$

Unless the component-wise product $\boldsymbol{\omega} \circ \boldsymbol{\lambda} = \mathbf{0}$ holds
$\langle \boldsymbol{\omega}(\hat{\mathbf{c}}), \boldsymbol{\lambda} \rangle > 0$ is satisfied

The minimal number of erasures $r$ equals so that an error may occur is

$$r > \mathrm{supp}(\boldsymbol{\omega})$$

(Stopping sets)

## Pseudodistance:BSC

Pseudocodeword: $\boldsymbol{\omega} = (\omega_1, \omega_2, \ldots, \omega_n)$,
Error vector: $\mathbf{e} = (e_1, e_2, \ldots, e_n)$
Log-likelihood ratios: $\lambda_i \in \{+K, -K\}$

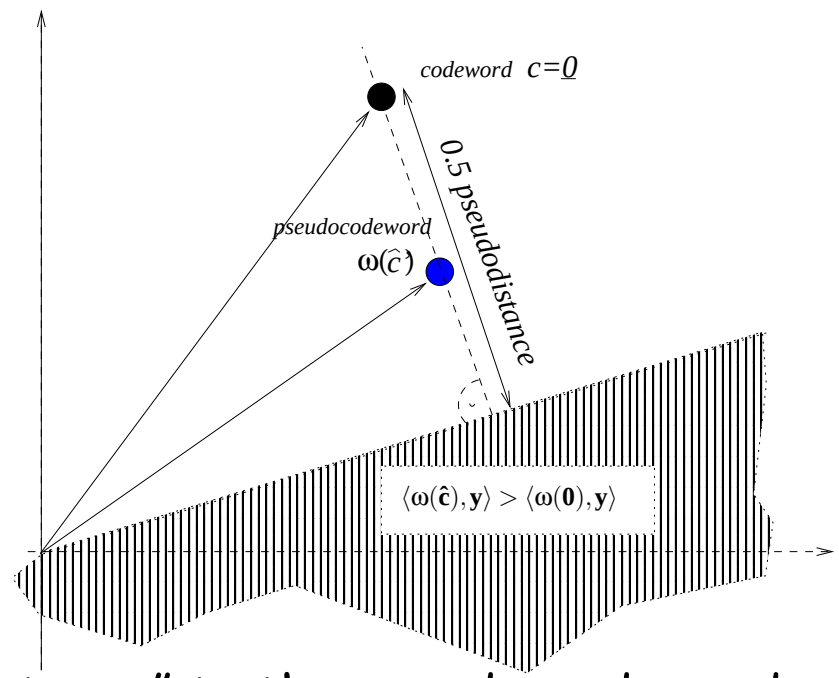
The condition $\langle \boldsymbol{\omega}(\hat{\mathbf{c}}), \boldsymbol{\lambda} \rangle \overset{?}{>} 0$ may be satisfied with $t$ errors if there exists a set $J = \{i_1, i_2, \ldots, i_t\} \subset \{1 \ldots n\}$ such that:

$$\sum_{i \in J} \omega_i > \sum_{i \notin J} \omega_i$$

# Pseudodistance:AWGN

Antipodal signaling: $c_i \to \tilde{c}_i = 1 - 2c_i \quad c_i = \frac{1}{2}(1 - \tilde{c}_i), \quad \boldsymbol{\lambda} \propto \mathbf{y}$

$$\langle \tfrac{1}{2}(1 - \tilde{\boldsymbol{\omega}}(\hat{\mathbf{c}})), \mathbf{y} \rangle \overset{?}{>} 0$$




On an AWGN channel the "distance" to the pseudo-codeword equals

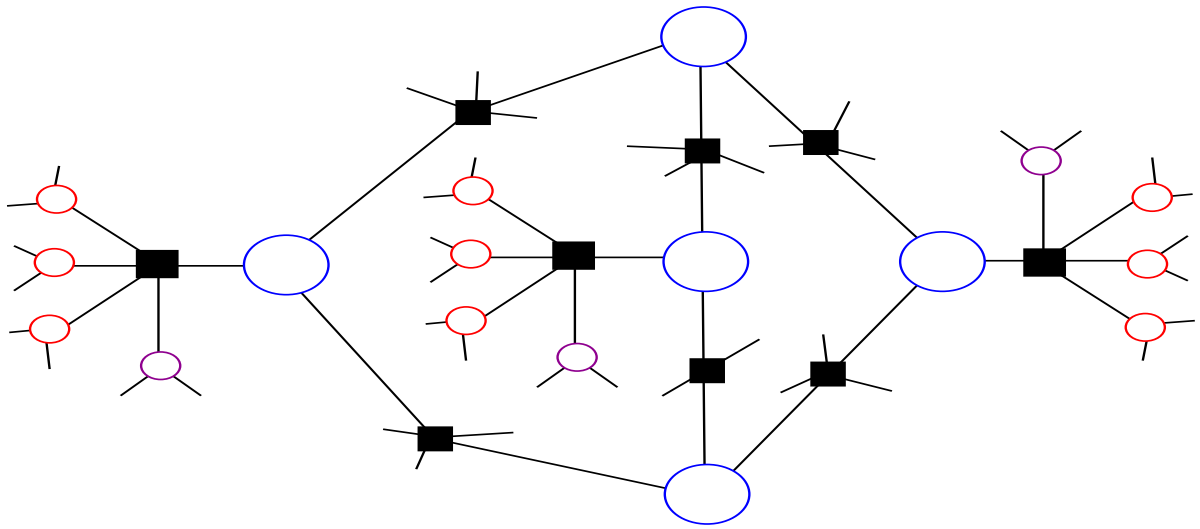
$$\boxed{w_P(\boldsymbol{\omega}) = \left(\frac{||\boldsymbol{\omega}||_1}{||\boldsymbol{\omega}||_2}\right)^2.}$$

## A $[155, 64, 20]$ Code by Tanner (Part 1)

A $(3, 5)$-regular LDPC code constructed by Tanner.

| | |
|---|---|
| Codelength | 155 |
| Rate | $64/155 = 0.4129$ |
| Girth of the factor graph | 8 (optimal) |
| Diameter of the factor graph | 6 (optimal) |
| Minimum Hamming weight | 20 |
| Minimum pseudo-weight (AWGN) | $10.8 < w_{\text{p,min}}^{\text{AWGNC}} < 16.4$ |
| Minimum pseudo-weight (BSC) | $w_{p,min}^{BSC} < 10$ |

# A $[155, 64, 20]$ Code by Tanner (Part 2)



after sorting:

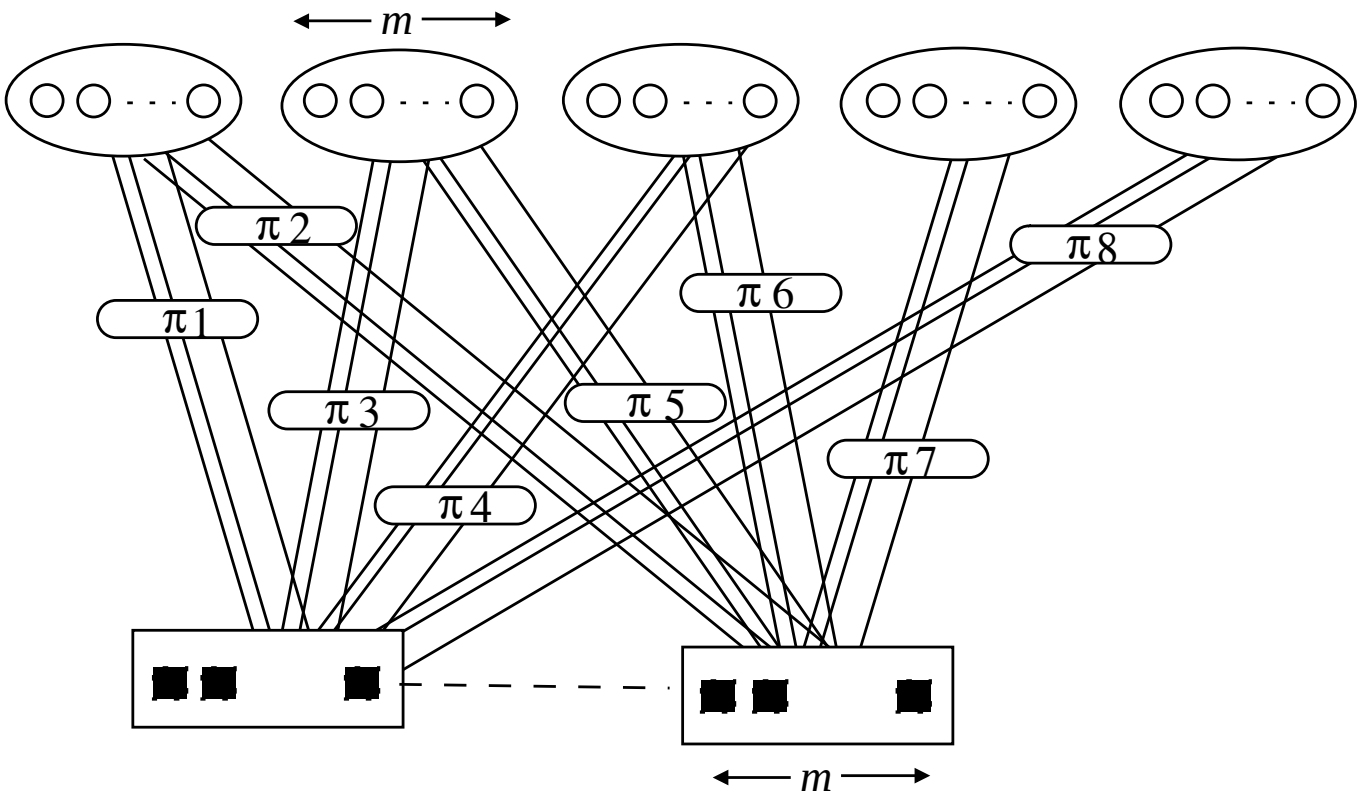
$\omega = (1, 1, 1, 1, 1, \frac{3}{10}, \frac{3}{10}, \frac{3}{10}, \frac{3}{10}, \frac{3}{10}, \frac{3}{10}, \frac{3}{10}, \frac{3}{10}, \frac{3}{10}, \frac{3}{10}, \frac{3}{20} \frac{3}{20} \frac{3}{20}, \frac{1}{10} \ldots)$

and

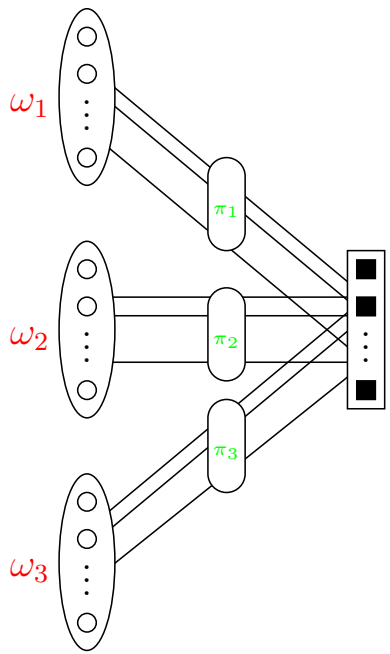
$\omega = (\frac{9}{10}, \frac{9}{10}, \frac{9}{10}, \frac{9}{10}, \frac{9}{10}, \ldots)$

## So far so good....

How to characterize the set of pseudo-codewords $\omega$ from the union of all degree $m$ covers for $m = 1, 2, 3, \ldots$



For a typical check we have:




We can find permutations $\pi_1$, $\pi_2$, $\pi_3$ for the tuple $\omega_1, \omega_2, \omega_3$ if and only if

$$\begin{aligned} 0 &\le \omega_1 \le 1 \\ 0 &\le \omega_2 \le 1 \\ 0 &\le \omega_3 \le 1 \end{aligned} \quad \text{and} \quad \begin{aligned} -\omega_1 + \omega_2 + \omega_3 &\ge 0 \\ +\omega_1 - \omega_2 + \omega_3 &\ge 0 \\ +\omega_1 + \omega_2 - \omega_3 &\ge 0 \\ +\omega_1 + \omega_2 + \omega_3 &\le 2 \end{aligned}$$

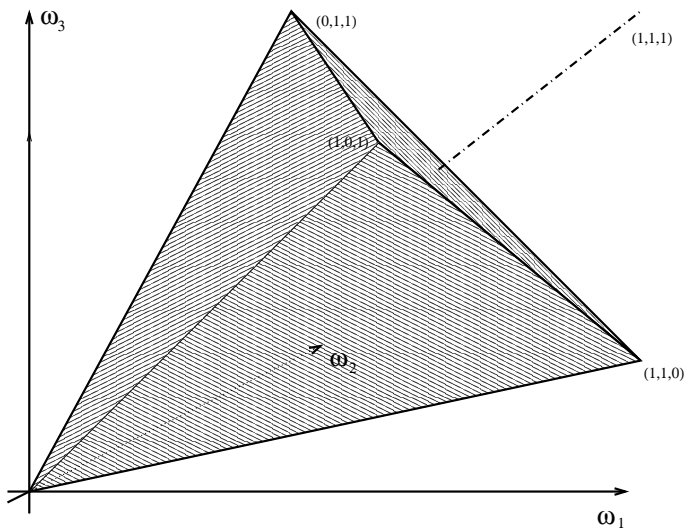
or, equivalently,

$$0 \le \omega_i \le 1 \quad \text{and} \quad \begin{aligned} \max\left\{\omega_1, \omega_2, \omega_3\right\} &\le \tfrac{1}{2}\left(\omega_1 + \omega_2 + \omega_3\right) \\ \omega_1 + \omega_2 + \omega_3 &\le 2 \end{aligned}$$

continued...

or

$$\omega_1 \leq \omega_2 + \omega_3$$
$$\omega_2 \leq \omega_1 + \omega_3$$
$$\omega_3 \leq \omega_1 + \omega_2$$

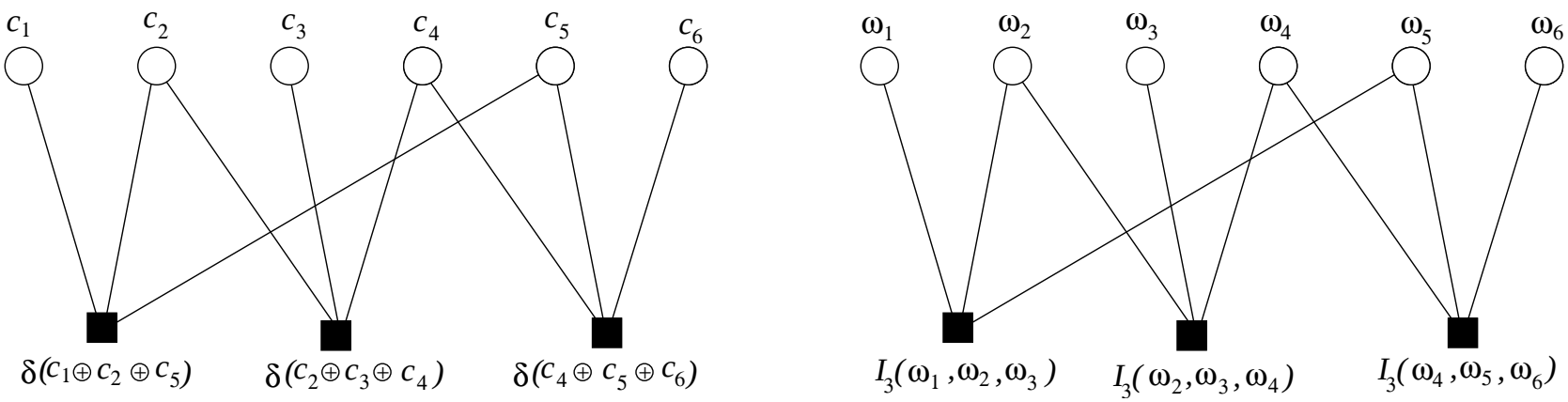


In general we have that a check of degree $\delta$ constrains the set of allowable $\omega_1, \omega_2, \ldots, \omega_\delta$ to values such that $\max\{\omega_1, \omega_2, \ldots, \omega_\delta\} < \frac{1}{2}\sum_{i=1}^{\delta} \omega_i$, and $0 \leq \omega_i \leq 1$

We define an indicator function

$$I_\delta(\omega_1, \omega_2, \ldots, \omega_\delta) = \begin{cases} 1 & \max\{\omega_1, \omega_2, \ldots, \omega_\delta\} < \frac{1}{2}\sum_{i=1}^{\delta} \omega_i \\ 0 & \text{otherwise.} \end{cases}$$

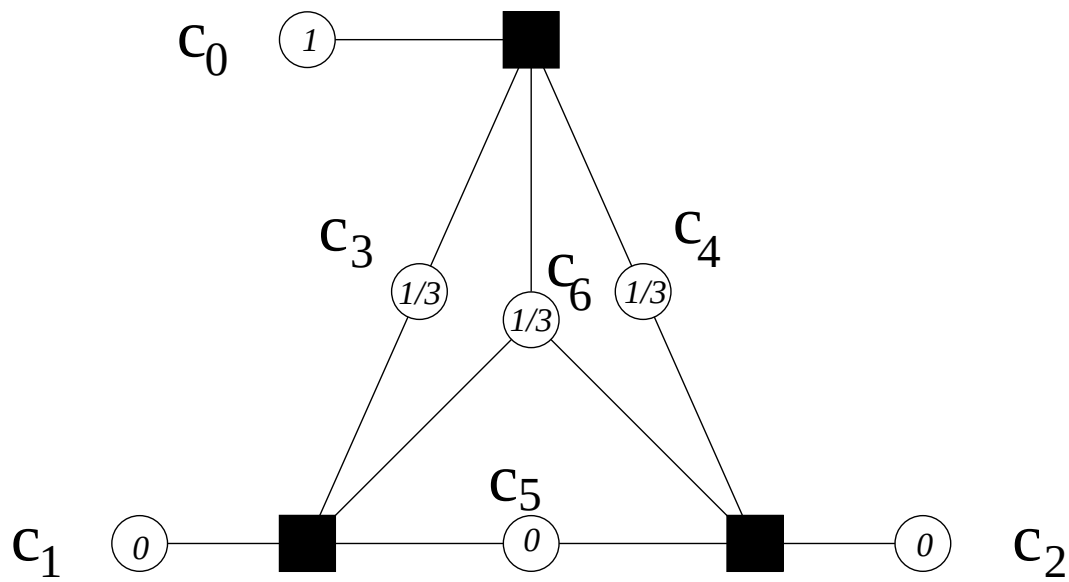
The indicator funtions $I_\delta(\omega_1, \omega_2, \ldots, \omega_\delta)$ allow us to write a factor graph for the pseudo-codeword indicator function:




The set of all possible pseudo-codewords is elegantly described in a factor graph. In particular, the set of pseudo-codewords is dense in the polytope in $\mathbb{R}^n$ that is cut out by the individual indicator functions.

Given an LDPC code graph we want to find the minimum pseudodistance rather than the minimum Hamming distance!

Some examples: $[7, 4, 3]$ Hamming code



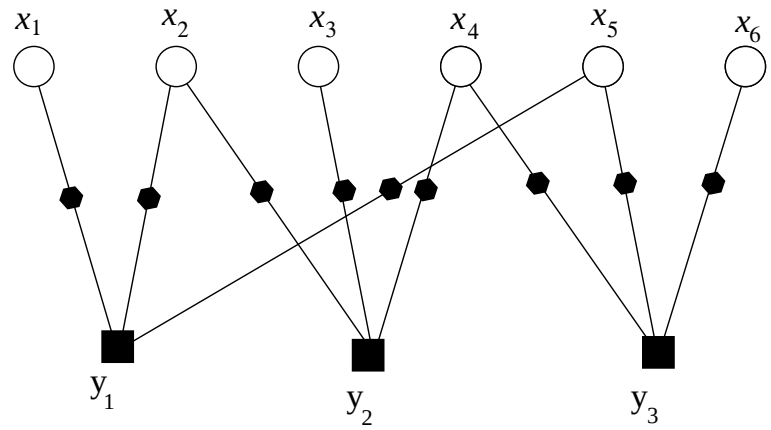

The pseudo-codeword is $\omega' = (1, 0, 0, \frac{1}{3}, \frac{1}{3}, 0, \frac{1}{3})$. The pseudo-codeword is at distance 3. (AWGN)

## More ways to get to the polytope...

1) The polytope described by all pseudocodewords is the same polytope as the one found by Feldman by relaxing an integer program.

2) Bethe free energy: Belief propagation attempts to minimize the Bethe free energy [Yedidia, Weiss, Freeman 04]

Fixed points of Belief Propagation are zero gradient points of the Bethe free energy.




$y_i$ *is in {(0,0,0),(1,1,0),(1,0,1),(0,1,1)}*

*corresponds to a function* $f(x_i, y_j)$

*to each variable we associate a function* $f(x_i)$

$$\beta \cdot F_{\text{Bethe}} = \sum_{\substack{i,j \\ i>j}} \sum_{x_i, y_j} b_{ij}(x_i, y_j) \log \frac{b_{ij}(x_i, y_j)}{f(x_i, y_j)} - \sum_i (n_i - 1) \sum_{v_i} b_i(v_i) \log \frac{b_i(v_i)}{f(v_i)},$$

or, with

$$\beta U_{\text{Bethe}} = -\sum_{\substack{i,j \\ i>j}} \sum_{x_i, y_j} b_{ij}(x_i, y_j) \log f(x_i, y_j) + \sum_i (n_i - 1) \sum_{v_i} b_i(v_i) \log f_i(v_i),$$

$$-H_{\text{Bethe}} = \sum_{\substack{i,j \\ i>j}} \sum_{x_i, y_j} b_{ij}(x_i, y_j) \log b_{ij}(x_i, y_j) - \sum_i (n_i - 1) \sum_{v_i} b_i(v_i) \log b_i(v_i),$$

$$F_{\text{Bethe}} = U_{\text{Bethe}} - T H_{\text{Bethe}}$$

## The optimization problem:

Minimize $F_{\text{Bethe}} = U_{\text{Bethe}} - TH_{\text{Bethe}}$ over all choices of "beliefs" $b_{ij}(x_i, y_j)$ and $b_i(v_i)$ such that

$$
\begin{aligned}
0 \leq \quad b_{ij}(x_i, y_j) \quad &\leq 1 \\
0 \leq \quad b_i(v_i) \quad &\leq 1 \\
&\sum_{y_j} b_{ij}(x_i, y_j) = b_i(x_i) \\
&\sum_{x_j} b_{ij}(x_i, y_j) = b_j(y_j)
\end{aligned}
$$

Moreover the minimization is only feasible if $b_{ij}(x_i, y_j)$ is compatible with $x_i$ and $y_j$.

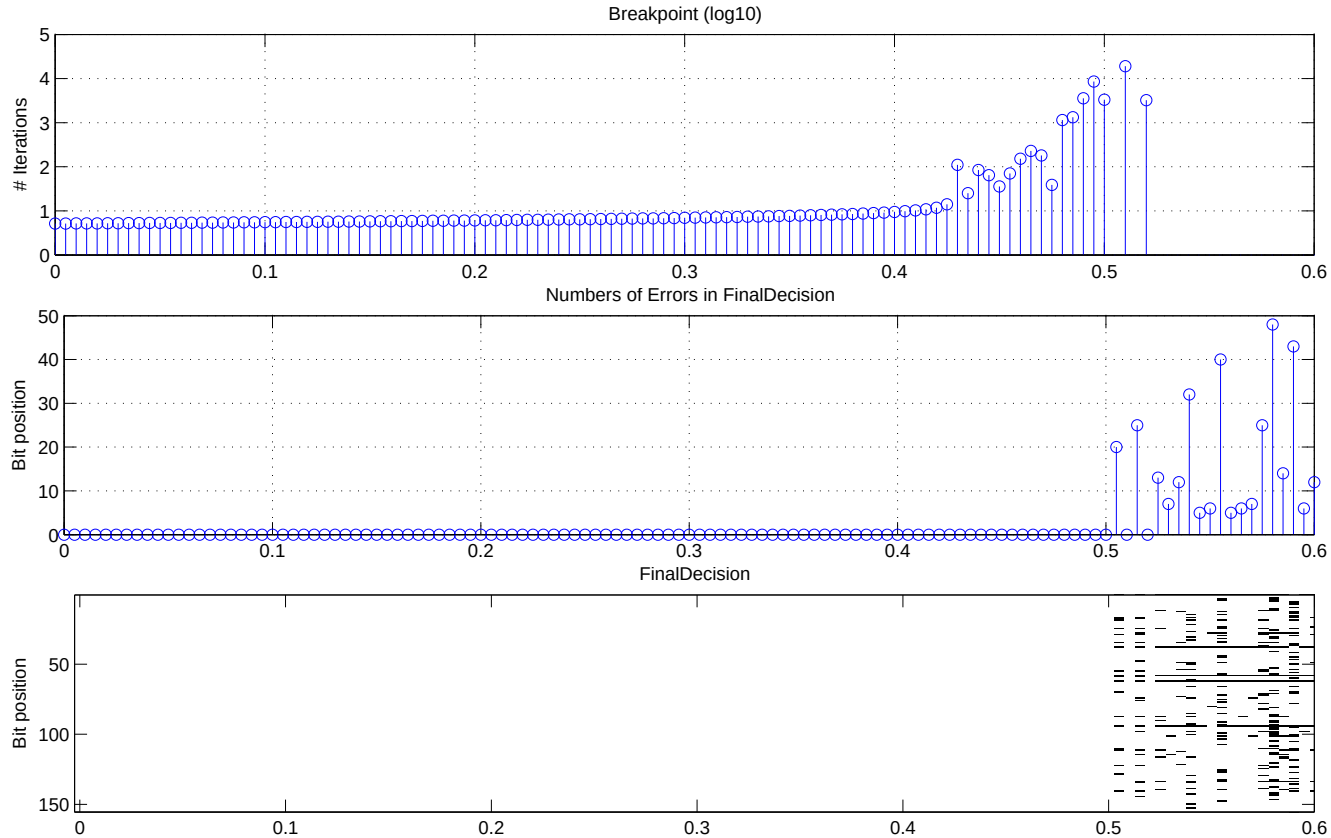
This is just the polytope found before!

## The equivalent optimization problem:

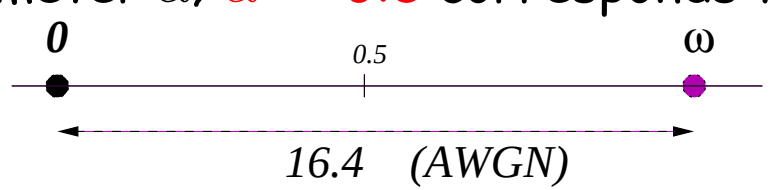
Minimize $F_{\text{Bethe}} = U_{\text{Bethe}} - TH_{\text{Bethe}}$ over all choices of "beliefs" which lie within the previously found polytope.

As $T$ approaches zero the LP decoder of Feldman is recovered ( $U_{\text{Bethe}}$ is a linear function of the beliefs.)

The pseudocodeword polytope is actually identical to the "belief" polytope of the Bethe approximation of the Gibbs free energy.




The horizontal axis shows the parameter $\alpha$; $\alpha = 0.5$ corresponds to the hypthetical decision boundary.



## Observations around the polytope $P$

Let $C$ be a liner code in $\mathbb{F}_2^n$ and let $C^\perp$ be its dual.

$CH(C) \triangleq$ convex hull of $C$ in $\mathbb{R}^n$

Given a vector $\mathbf{h}$ let $P_h$ be defined as:

$$P_h = CH(\mathbf{h}^\perp)$$

$$P = \bigcap_{h \in \{rows\ in\ H\}} P_h$$

## Observations around the polytope $P$

$$H = \begin{pmatrix} \vdots & & & & & & \vdots \\ \dots & 1 & 1 & 0 & 1 & 0 & \dots \\ \dots & 0 & 1 & 1 & 1 & 1 & \dots \\ \vdots & & & & & & \vdots \end{pmatrix} \begin{array}{l} \\ \boldsymbol{\omega} \in P_{h_1} \\ \boldsymbol{\omega} \in P_{h_2} \\ \\ \end{array}$$

$$H = \begin{pmatrix} \vdots & & & & & & \vdots \\ \dots & 1 & 1 & 0 & 1 & 0 & \dots \\ \dots & 0 & 1 & 1 & 1 & 1 & \dots \\ \dots & 1 & 0 & 1 & 0 & 1 & \dots \\ \vdots & & & & & & \vdots \end{pmatrix} \begin{array}{l} \\ \boldsymbol{\omega} \in P_{h_1} \\ \boldsymbol{\omega} \in P_{h_2} \\ \boldsymbol{\omega} \in P_{h_3},\ h_3 = h_1 + h_2 \\ \\ \end{array}$$

If $h_1$ and $h_2$ coincide in at least two positions $h_3$ is facet defining:

$$CH(h_1^{\perp}) \cap CH(h_2^{\perp}) \supset CH(h_1^{\perp}) \cap CH(h_2^{\perp}) \cap C(h_3^{\perp})$$

## Observations around the polytope $P$

$$H = \begin{pmatrix} \vdots & & & & & & \vdots \\ \dots & 1 & 1 & 0 & 1 & 0 & \dots \\ \dots & 0 & 1 & 1 & 1 & 1 & \dots \\ \vdots & & & & & & \vdots \end{pmatrix} \begin{array}{l} \\ \boldsymbol{\omega} \in P_{h_1} \\ \boldsymbol{\omega} \in P_{h_2} \\ \\ \end{array}$$

$$H = \begin{pmatrix} \vdots & & & & & & \vdots \\ \dots & 1 & 1 & 0 & 1 & 0 & \dots \\ \dots & 0 & 0 & 1 & 1 & 1 & \dots \\ \dots & 1 & 1 & 1 & 0 & 1 & \dots \\ \vdots & & & & & & \vdots \end{pmatrix} \begin{array}{l} \\ \boldsymbol{\omega} \in P_{h_1} \\ \boldsymbol{\omega} \in P_{h_2} \\ \boldsymbol{\omega} \in P_{h_3},\ h_3 = h_1 + h_2 \\ \\ \end{array}$$

If $h_1$ and $h_2$ coincide in at most one positions $h_3$ is not facet defining:

$$CH(h_1^{\perp}) \cap CH(h_2^{\perp}) = CH(h_1^{\perp}) \cap CH(h_2^{\perp}) \cap C(h_3^{\perp})$$

## Observations around the polytope $P$

If there are no four cycles in the graph we get all inequalities involving the sum of two rows for free.

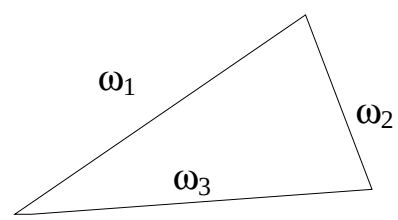
**Proposition** Let $H$ describe a graph with girth $g$. The polytope $P$ is not further restricted by linear combinations of rows of $H$ if the weight of the linear combination does not exceed $\frac{g-2}{2}$.

## Observations around the polytope $P$

$$MET(C^{\perp}) \triangleq \bigcap_{c \in C^{\perp}} CH(c^{\perp})$$

Example: $i$th word $c_i$ in $C^{\perp}$ equals $(1, 1, 1, 0, \ldots, 0)$

$$CH(C_i) = \{\boldsymbol{\omega} \in \mathbb{R}^n : \left| \begin{array}{c} \omega_1 \leq \omega_2 + \omega_3 \\ \omega_2 \leq \omega_3 + \omega_1 \\ \omega_3 \leq \omega_2 + \omega_1 \\ \omega_1 + \omega_2 + \omega_3 \leq 2 \end{array} \right|, 0 \leq \omega_i \leq 1\}$$



$$MET(C^{\perp}) \stackrel{?}{=} CH(C)$$

Translating a theorem for binary matroids we get:

Theorem[Seymor, 81] $MET(C^{\perp}) = CH(C)$ if and only if there is no way to shorten and puncture $C$ such that we get the codes $F_7^*, M(K_5), R_{10}$

$F_7^* : [7, 3, 4]$

$M(K_5) : [10, 6, 3]$

$R_{10} : [10, 5, 4]$

## Bounds on the Minimum Pseudo-Weight (?)

Techniques for obtaining upper bounds on the min. pseudo-weight:

- The pseudo-weight of any valid pseudo-codeword gives an upper bound.
- Canonical completion.

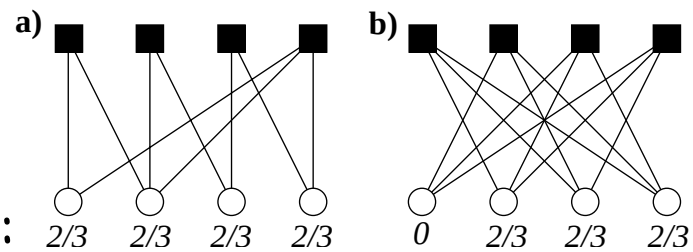
Techniques for obtaining lower bounds on the min. pseudo-weight:

- Bounds based on largest and second largest eigenvalue of $\mathbf{H}^{\mathsf{T}} \cdot \mathbf{H}$.
- Linear programming bounds.

Thank you!

## The trouble makers...

Stopping sets:



## The trouble makers...

Near codewords, trapping sets: $(m, l)$ near codewords are words of weight $m$ that do not satisfy $l$ check equations. The significance of $(m, l)$ near codewords is that they often give rise to low weight pseudocodewords. A canonical completion of the $(m, l)$ near codeword yields a word of pseudoweight at most

$$w_{\text{p,min}}^{\text{AWGNC}}(\mathcal{C}) \leq \beta'_{j,k} \cdot n^{\beta_{j,k}},$$

where

$$\beta'_{j,k} = \left(\frac{l(j-1)}{m(j-2)}\right)^2, \quad \beta_{j,k} = \frac{\log\left((j-1)^2\right)}{\log\left((j-1)(k-1)\right)} < 1.$$

## The trouble makers...

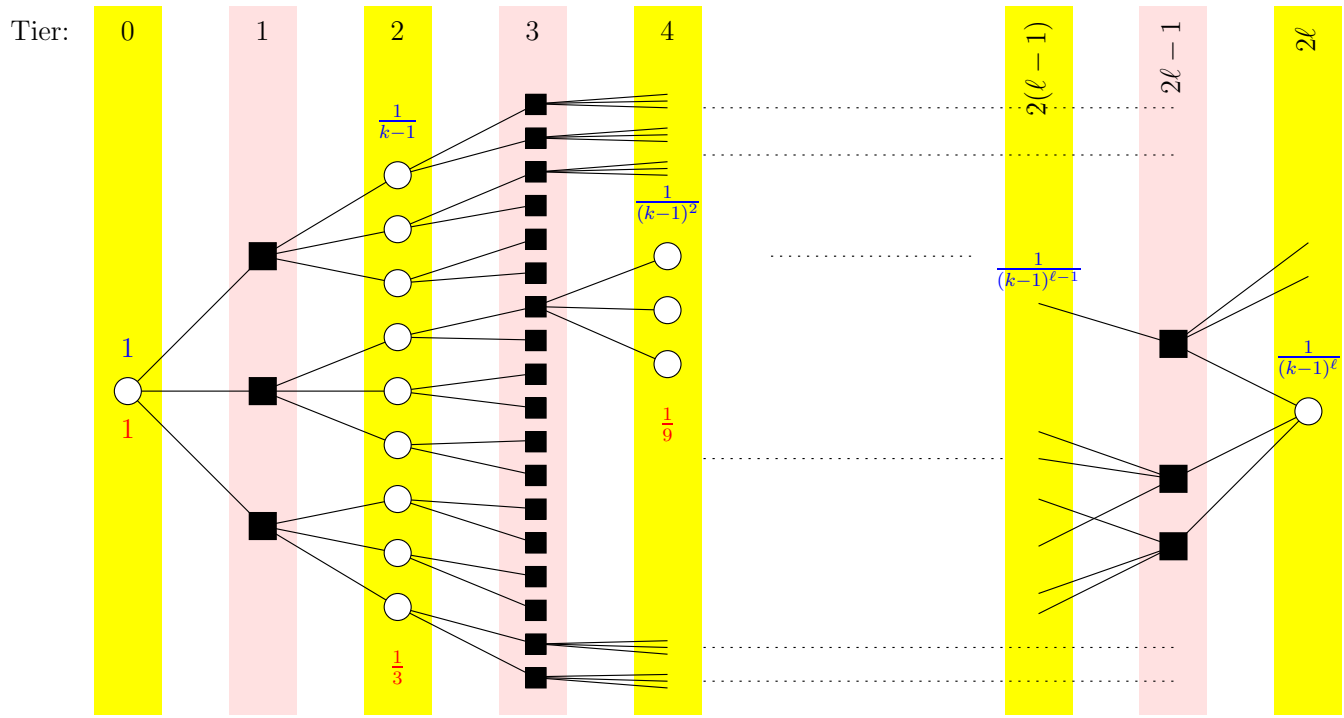
The best estimate based on a suspected near codeword with indicator vector $v$ is given by the LP:

$$minimize \;\; \langle 1 - v, w \rangle$$

$$||w||_1 = 1, w \in P$$

Near codwords and stopping sets are excellent candidates to find low weight pseudocodewords. The LP allows us to quantify if a stopping set or a near codeword is indeed a problem.

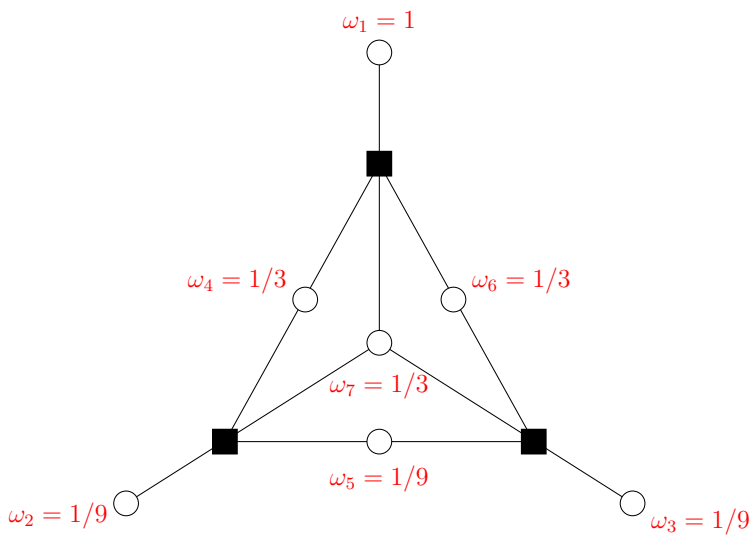
# An Upper Bound based on the Canonical Completion (Part 1)




The canonical completion for a $(3, 4)$-regular LDPC code. On check-regular graphs the canonical completion always gives a (valid) pseudo-codeword.

## An Upper Bound based on the Canonical Completion (Part 2)

Example: $[7, 4, 3]$ binary Hamming code.




The canonical completion starting at $\omega_1$ is

$$\boldsymbol{\omega} = \begin{pmatrix} 1 & \frac{1}{9} & \frac{1}{9} & \frac{1}{3} & \frac{1}{9} & \frac{1}{3} & \frac{1}{3} \end{pmatrix}.$$

The pseudo-weight of $\boldsymbol{\omega}$ is

$$w_{\mathrm{p}}^{\mathrm{AWGNC}}(\boldsymbol{\omega}) = \frac{||\boldsymbol{\omega}||_1^2}{||\boldsymbol{\omega}||_2^2}$$

$$= \frac{\left(1 + \frac{1}{9} + \frac{1}{9} + \frac{1}{3} + \frac{1}{9} + \frac{1}{3} + \frac{1}{3}\right)^2}{1 + \frac{1}{81} + \frac{1}{81} + \frac{1}{9} + \frac{1}{81} + \frac{1}{9} + \frac{1}{9}}$$

$$= 3.973.$$

## An Upper Bound based on the Canonical Completion

Theorem: Let $\mathcal{C}$ be a $(j,k)$-regular LDPC code with $3 \leq j < k$. Then the minimum pseudo-weight is upper bounded by

$$w_{\text{p,min}}^{\text{AWGNC}}(\mathcal{C}) \leq \beta'_{j,k} \cdot n^{\beta_{j,k}},$$

where

$$\beta'_{j,k} = \left(\frac{j(j-1)}{j-2}\right)^2, \quad \beta_{j,k} = \frac{\log\left((j-1)^2\right)}{\log\left((j-1)(k-1)\right)} < 1.$$

Corollary: The minimum relative pseudo-weight for any sequence $\{\mathcal{C}_i\}$ of $(j,k)$-regular LDPC codes of increasing length satisfies

$$\lim_{n\to\infty}\left(\frac{w_{\text{p,min}}^{\text{AWGNC}}(\mathcal{C}_i)}{n}\right) = 0.$$

($exp(-a' n^{\beta_{j,k}}) < P_B < n\ exp(-a n^{\beta_{j,k}/4})$ )

(right hand side: Lentmaier et al.)

## A Lower Bounds on the Minimum Pseudo-Weight based on Eigenvalues

Let $\mathcal{C}$ be a $(j,k)$-regular code of length $n$.

- Let $\mathbf{H}$ be the parity-check matrix. component.
- Let $\mathbf{L} \triangleq \mathbf{H}^{\mathsf{T}}\mathbf{H}$.
- Let $\mu_1$ and $\mu_2$ be the largest and second largest eigenvalue, respectively, of $\mathbf{L}$.

Then the minimum Hamming weight and the minimum AWGNC pseudo-weight of $\mathcal{C}$ are lower bounded by

$$w_{\mathsf{H}}^{\min}(\mathcal{C}) \geq w_{\mathsf{p}}^{\min}(\mathsf{C}) \geq n \cdot \frac{2j - \mu_2}{\mu_1 - \mu_2}.$$

## A Lower Bound on The Minimum Pseudo-Weight based on Linear Programming (?)

Let $\boldsymbol{\omega}$ be any pseudo-codeword with $||\boldsymbol{\omega}||_1 = 1$. Then the (rank-1) matrix

$$\mathbf{M} \triangleq \boldsymbol{\omega}^{\mathsf{T}} \cdot \boldsymbol{\omega} = \begin{pmatrix} \omega_1^2 & \omega_1\omega_2 & \omega_1\omega_3 & \cdots & \omega_1\omega_n \\ \omega_2\omega_1 & \omega_2^2 & \omega_2\omega_3 & \cdots & \omega_2\omega_n \\ \omega_3\omega_1 & \omega_3\omega_2 & \omega_3^2 & \cdots & \omega_3\omega_n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \omega_n\omega_1 & \omega_n\omega_2 & \omega_n\omega_3 & \cdots & \omega_n^2 \end{pmatrix}$$

has the following properties:

- entries are non-negative,
- $\sum_{i,j}[\mathbf{M}]_{i,j} = 1$,
- $\mathrm{Trace}(\mathbf{M}) = ||\boldsymbol{\omega}||_2^2$,
- row $i$ of $\mathbf{M}$ equals $\omega_i \cdot \boldsymbol{\omega}$,
- column $j$ of $\mathbf{M}$ equals $\omega_j \cdot \boldsymbol{\omega}^{\mathsf{T}}$.

## A Lower Bound on The Minimum Pseudo-Weight based on Linear Programming

Maximizing

$$\mathrm{Trace}(N)$$

over all $n \times n$-matrices $\mathbf{N}$ (not necessarily of rank $1$) that fulfill

- entries are non-negative,
- $\sum_{i,j} [\mathbf{N}]_{i,j} = 1$,
- the rows are in the fundamental cone,
- the columns are in the fundamental cone,

we obtain $1/\mathrm{Trace}(N)$ as a lower bound on the minimum pseudo-weight. (Note: the above optimization problem is a linear program.)

## Pseudo-Weights for other Channels (?)

Let $\boldsymbol{\omega}$ be a (valid) pseudo-codeword. For each channel we can define a pseudo-weight, see [Wiberg:96], [FKKR:01].

- The AWGN channel pseudo-weight $w_{\mathrm{p}}(\boldsymbol{\omega})$ of $\boldsymbol{\omega}$ is given by

$$w_{\mathrm{p}}^{\mathrm{AWGNC}}(\boldsymbol{\omega}) = \frac{||\boldsymbol{\omega}||_1^2}{||\boldsymbol{\omega}||_2^2} = \frac{(|\omega_1| + \cdots + |\omega_n|)^2}{|\omega_1|^2 + \cdots + |\omega_n|^2} = \frac{(\omega_1 + \cdots + \omega_n)^2}{\omega_1^2 + \cdots + \omega_n^2}.$$

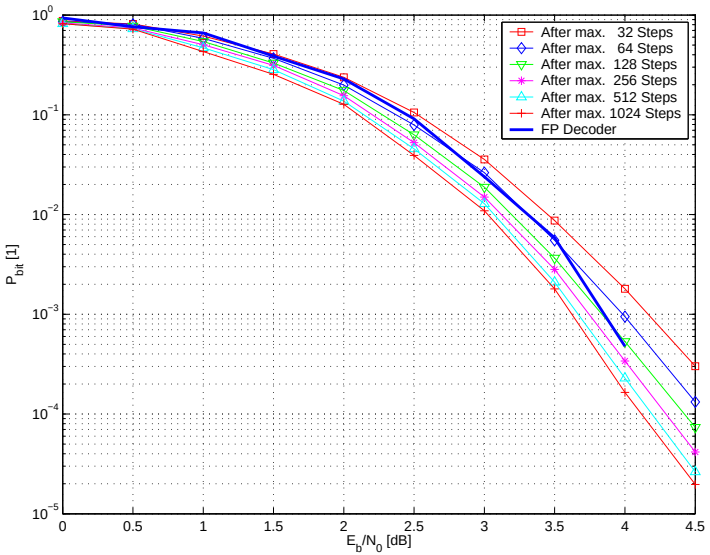
- The BSC channel pseudo-weight $w_{\mathrm{p}}^{\mathrm{BSC}}(\boldsymbol{\omega})$ is twice the median of the descendingly sorted vector $\boldsymbol{\omega}$.

- The BEC channel pseudo-weight $w_{\mathrm{p}}^{\mathrm{BEC}}(\boldsymbol{\omega})$ is the support of $\boldsymbol{\omega}$, i.e.

$$w_{\mathrm{p}}^{\mathrm{BEC}}(\boldsymbol{\omega}) = \mathrm{supp}(\boldsymbol{\omega}) = \left|\left\{i \in \{1, \ldots, n\} \mid \omega_i \neq 0\right\}\right|.$$

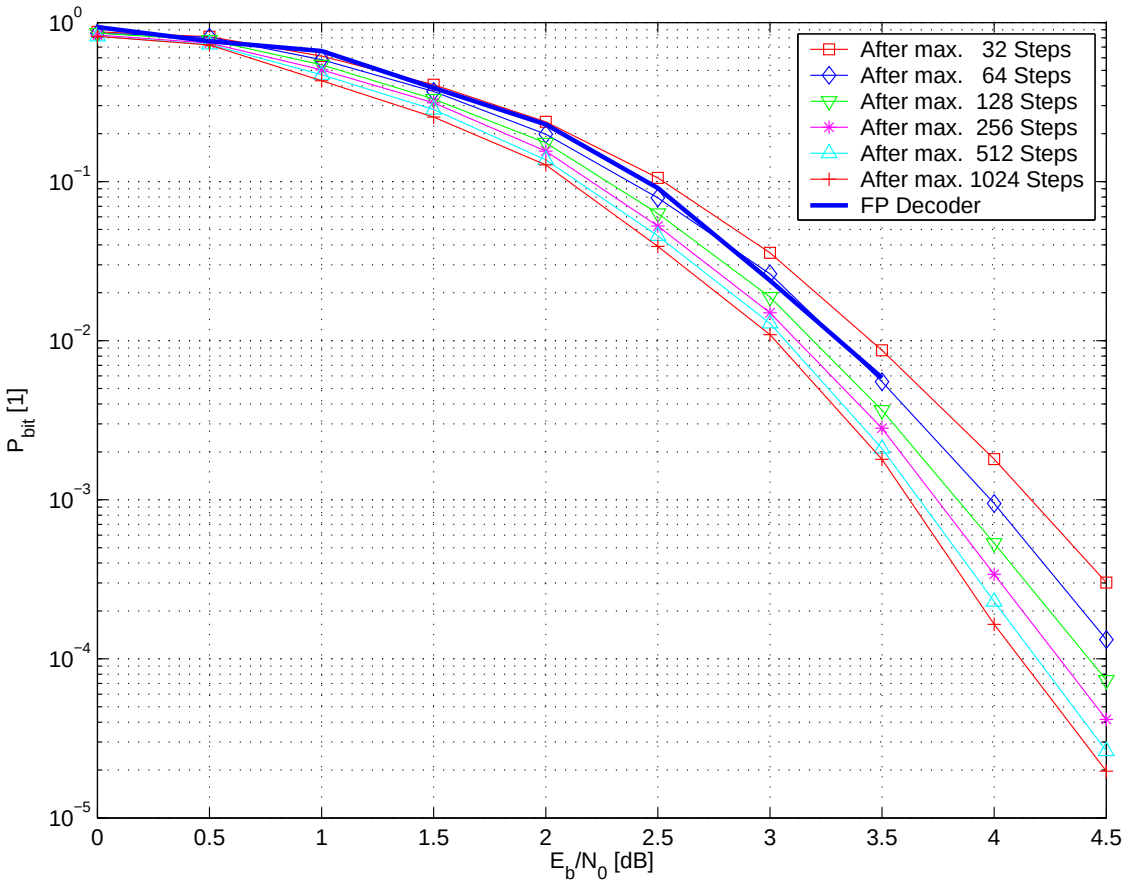
Note: the pseudo-weight definition depends on the channel but the fundamental polytope/cone is independent of the channel!

# Connections to the Linear Programming Decoder

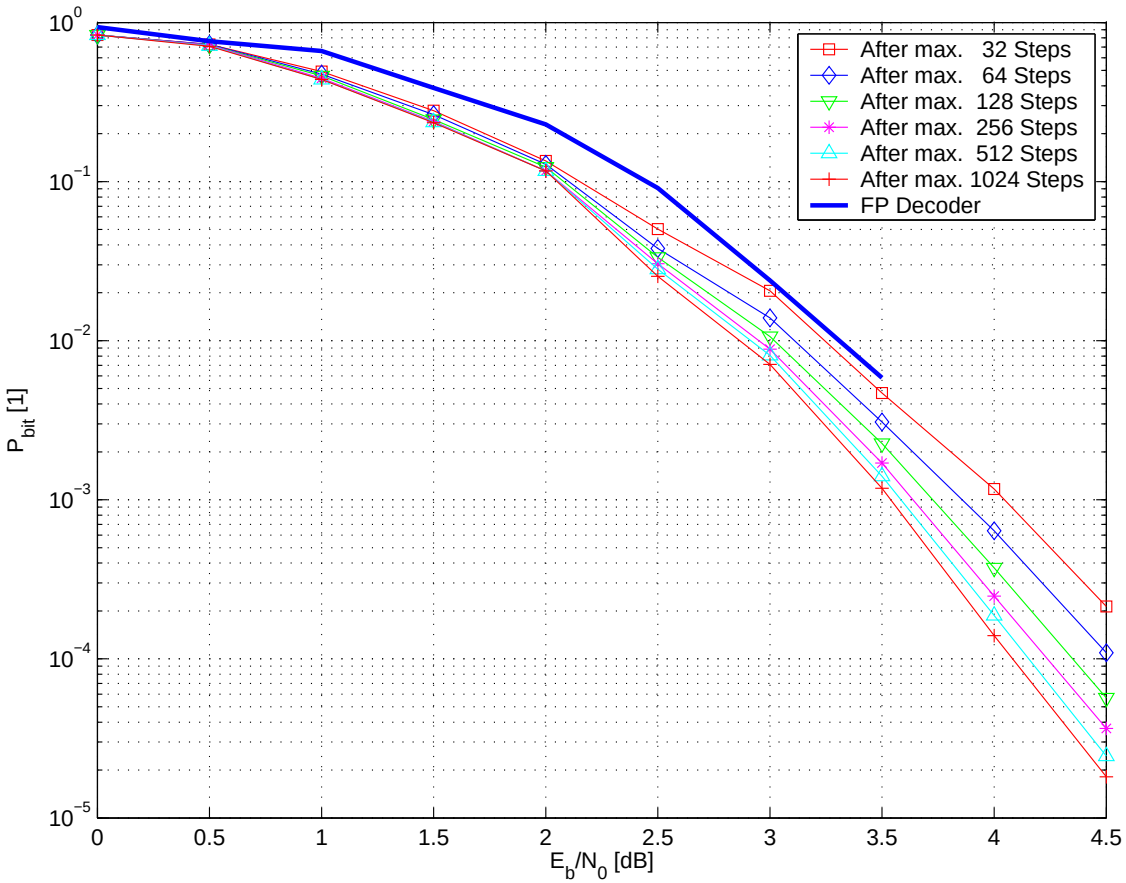



- The linear programming decoder was recently introduced by Feldman and Karger.
- They formulate the decoding of a code as a relaxed integer programming problem in order to obtain a linear program.
- The most canonical relaxation yields exactly the polytope that we called the fundamental polytope.

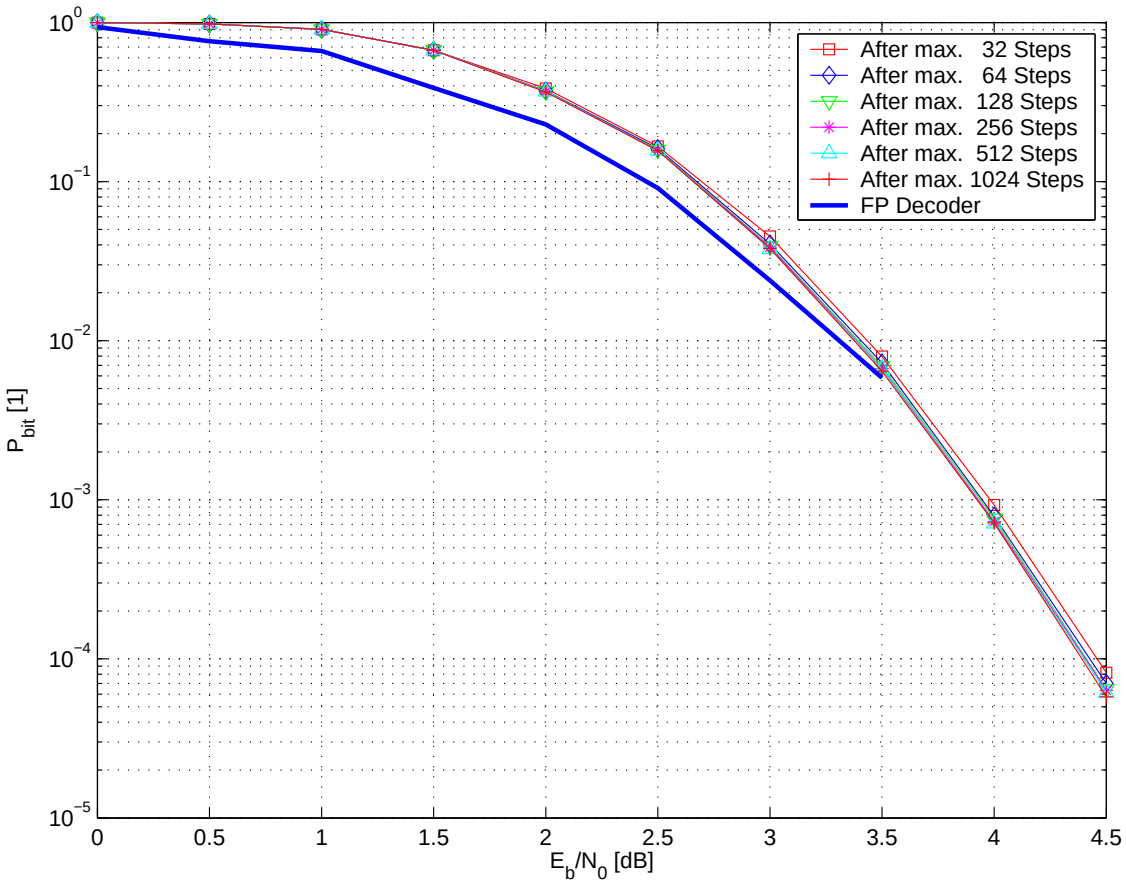
A comparison for the linear programming decoder and iterative decoding: ([155, 64, 20], Max/Product Algorithm):



A comparison for the linear programming decoder and iterative decoding ([155, 64, 20],Sum/Product Algorithm):



A comparison for the linear programming decoder and iterative decoding: ([155, 64, 20], Max/Product Algorithm, scaling = $2^{-0.35}$):



## Conclusions:

- Pseudo-codewords compete with codewords for the solution of the optimization problem that is solved with iterative decoding
- Codewords in graph covers are the systemic price one has to pay for using any locally operating message passing algorithm.
- The error floor performance in the large SNR limit is typically determined by pseudo-codewords!
- For large length the minimum relative pseudodistance vanishes for the AWGN channel!
- We have shown several techniques to lower/upper bound the minimum pseudo-weight of a factor graph realization of a code.
- Future work: LDPC code construction based on the avoidance of "bad patterns".
- An intriguing question: How to design codes with good pseudodistance!

Thanks you!