

# Coding, polytopes and low complexity algorithm

Ralf Koetter




Coordinated Science Laboratory

University of Illinois at Urbana-Champaign

email: {koetter}@uiuc.edu

Joint work with Pascal O. Vontobel, MIT

## Outline

1. Factor graphs, message passing — driving forces  
2. Linear programming decoding, belief propagation 
3. Properties of the structures appearing in LP decoding
4. Solving the linear program  $\perp$
5. Conclusions

## Low-Density Parity-Check (LDPC) code

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

Low-density:  $H$  is sparse

Codes:

$$\mathbb{C} = \{ \mathbf{c} \in \mathbb{F}_2^n \mid H\mathbf{c}^T = \mathbf{0}^T \}$$

Received:  $\mathbf{y} \in \mathcal{Y}$

Problem: Find the “most likely” transmitted word  $\mathbf{c}$ .

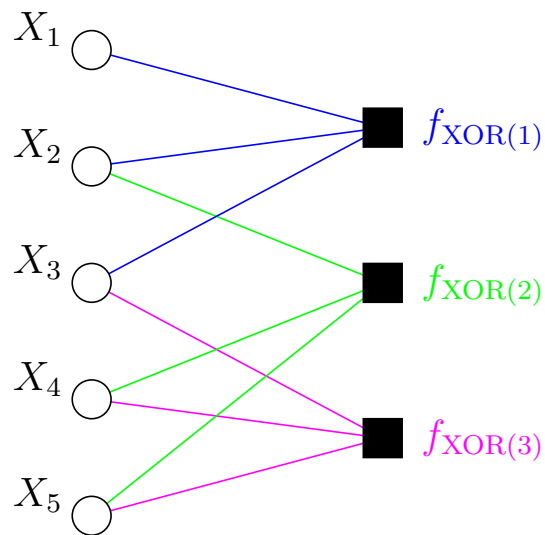
Binary codes: Log-likelihood ratios  $\lambda_i = \ln \frac{\Pr(c_i=0|y_i)}{\Pr(c_i=1|y_i)}$

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

## 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  $k$ .
- Each check defines a code  $\mathbb{C}_i = \{\mathbf{x} \in \mathbb{F}_2^n : \mathbf{h}_i \mathbf{x}^T = 0\}$

$$\mathbb{C} = \bigcap_i \mathbb{C}_i$$

*The main algorithmic problem: Decoding*

The decoding problem is to find the codeword  $\mathbf{x}$  that maximizes  $\log Pr(\mathbf{x}|\mathbf{y})$

$$\operatorname{argmax}_{\mathbf{x} \in \mathbb{C}} \log(Pr(\mathbf{x}|\mathbf{y})) = \operatorname{argmax}_{\mathbf{x} \in \mathbb{C}} \log\left(\frac{Pr(\mathbf{x}|\mathbf{y})}{Pr(\mathbf{0}|\mathbf{y})}\right) = \operatorname{argmax}_{\mathbf{x} \in \mathbb{C}} \log \prod_i \frac{Pr(x_i|y_i)}{Pr(0|y_i)} =$$

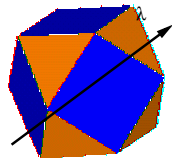
$$\operatorname{argmax}_{\mathbf{x} \in \mathbb{C}} \sum_i \log \left( \frac{Pr(x_i|y_i)}{Pr(0|y_i)} \right)^{x_i} = \operatorname{argmax}_{\mathbf{x} \in \mathbb{C}} \sum_i -x_i \lambda_i$$

ML Decoding problem:

$$\text{minimize } \langle \boldsymbol{\lambda}, \mathbf{x} \rangle, \mathbf{x} \in \mathbb{C}$$

## *The main algorithmic problem: Decoding*

Let a binary code  $\mathbb{C} \subseteq \{0, 1\}^n$  be given  $CH(\mathbb{C})$  denotes the convex hull of  $\mathbb{C}$  interpreted in  $\mathbb{R}^n$ .



Equivalent ML Decoding problem: minimize  $\langle \lambda, \mathbf{x} \rangle, \mathbf{x} \in CH(\mathbb{C})$

The central problem in practice: Construct codes for which this linear program (or approximations thereof) are solvable!

A relaxation of the LP

$CH(\mathbb{C})$  is hard to describe (typically the number of constraints is exponential in the code length).

$$CH(\mathbb{C}) = CH\left(\bigcap_{\text{all checks}} \mathbb{C}_i\right) \Rightarrow \mathbb{P} = \bigcap_{\text{all checks}} CH(\mathbb{C}_i)$$

$$\begin{array}{ll} \text{minimize} & \langle \lambda, \mathbf{x} \rangle \\ \text{subject to} & \mathbf{x} \in \mathbb{P} \end{array}$$

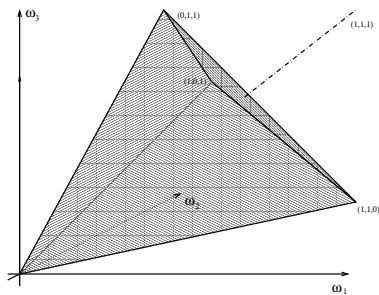
$CH(\mathbb{C}_i)$  has a small description  $\Rightarrow \mathbb{P} [\mathbb{P}(H)]$  has a small description. The key is to choose the effective length of  $\mathbb{C}_i$  at most as  $\log(N)$  in the overall length.

We obtain automatically a decoding algorithm with polynomial complexity....

- How to we really obtain "highly" efficient algorithms?
- How well is  $CH(\mathbb{C})$  approximated by  $\mathbb{P}$ ?
- What is a good polytope  $\mathbb{P}$  and how do we construct it?

## Describing $CH(\mathbb{C}_i)$

For a parity check code  $\mathbb{C}_i$  of length (degree) three  $CH(\mathbb{C}_i)$  is spanned by the corner points  $(0, 0, 0)$ ,  $(1, 1, 0)$ ,  $(1, 0, 1)$ ,  $(0, 1, 1)$ :



$$\begin{aligned} 0 &\leq \omega_1 \leq 1 \\ 0 &\leq \omega_2 \leq 1 \\ 0 &\leq \omega_3 \leq 1 \end{aligned}$$

and

$$\begin{aligned} -\omega_1 + \omega_2 + \omega_3 &\geq 0 \\ +\omega_1 - \omega_2 + \omega_3 &\geq 0 \\ +\omega_1 + \omega_2 - \omega_3 &\geq 0 \\ +\omega_1 + \omega_2 + \omega_3 &\leq 2 \end{aligned}$$

or, equivalently,

$$0 \leq \omega_i \leq 1 \quad \text{and} \quad \begin{aligned} \max\{\omega_1, \omega_2, \omega_3\} &\leq \frac{1}{2}(\omega_1 + \omega_2 + \omega_3) \\ \omega_1 + \omega_2 + \omega_3 &\leq 2 \end{aligned}$$

similar for higher degree parity check codes.

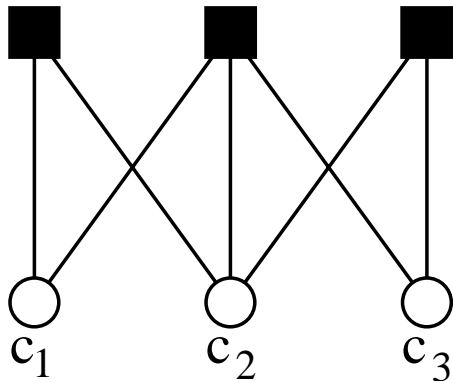
For higher degree checks the set of  $\omega_1, \omega_2, \dots, \omega_\delta$  is again given as the convex combination of even weight words. The hyperplanes describing this polytope are given as

$$0 \leq \omega_i \leq 1, \quad \omega_i \leq \frac{1}{2}(\omega_1 + \omega_2 + \dots + \omega_\delta)$$

and all hyperplanes obtained by replacing an even number of  $\omega_i$  with  $1 - \omega_i$ .

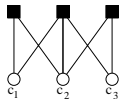
In the degree three case  $\omega_1 + \omega_2 + \omega_3 \leq 2$  is e.g. obtained from  $\omega_1 \leq \omega_2 + \omega_3$  as  $\omega_1 \leq (1 - \omega_2) + (1 - \omega_3)$

### A simple example

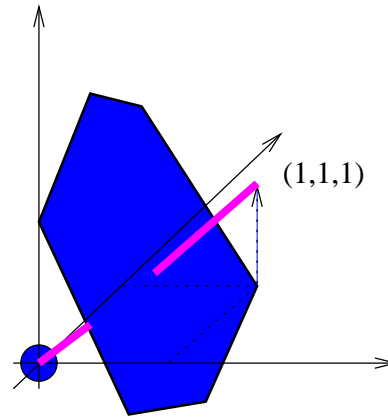
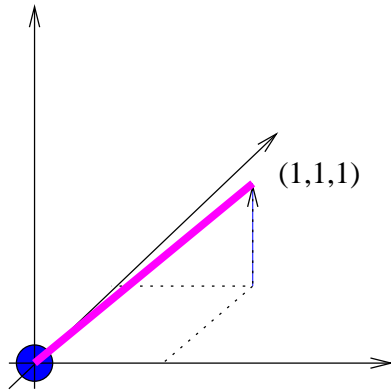


$$H = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$

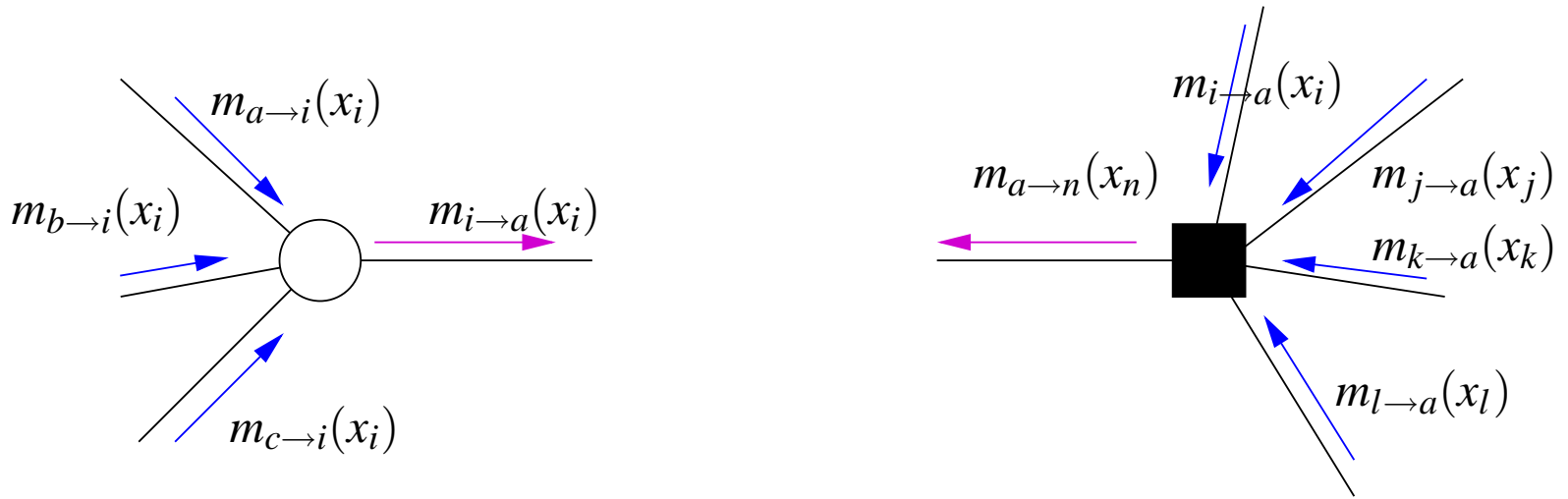
$\mathcal{C} = \{(0, 0, 0)\}$ ,  $d_H = \infty$ , ML decision is always  $(0, 0, 0)$



*continued...* Decision regions:



# "Highly" Efficient Message-Passing Decoding Algorithms



or

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

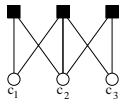
$$m_{a \to n}(x') = \min_{\substack{\mathbf{x}: x_n = x' \\ \mathbf{x} \text{ valid}}} \sum_{s \in \Gamma(a) \setminus n} m_{s \to a}(x_s)$$

Interpreting a result of Yedidia et al. we see that belief propagation obtains a zero gradient solution of the function

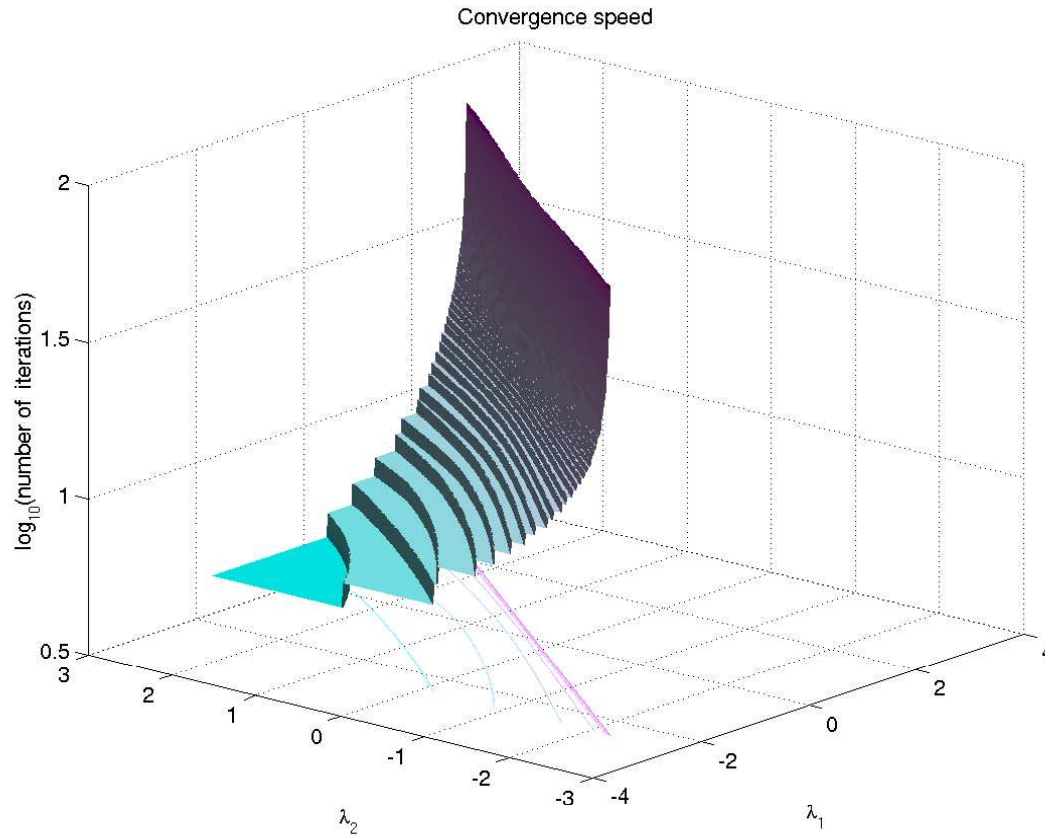
$$\langle \lambda, \mathbf{x} \rangle + \mathbb{I}[\mathbf{x} \in \mathbb{P}] + T(\text{Bethe entropy approximation})$$

$$\mathbb{I}[A] = \begin{cases} 0 & A \text{ is true} \\ \infty & \text{otherwise} \end{cases}$$

As  $T$  approaches zero minimizing the Bethe free energy is the same as solving the LP.

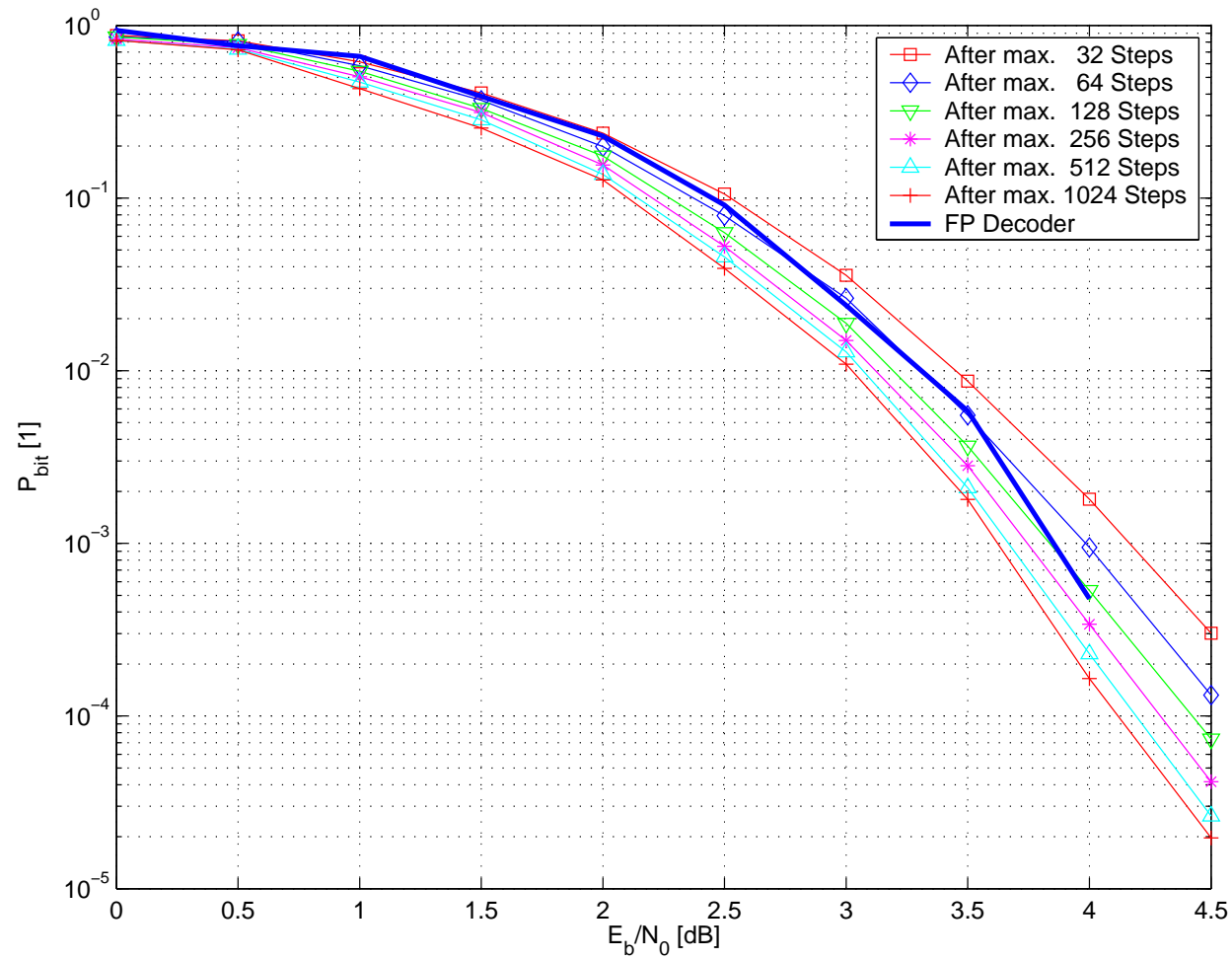


*continued... Convergence speed:*



The convergence speed towards the plane predicted by  $\mathbb{P}$

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



## A quick interim-summary

- High performance coding schemes are driven by decoding algorithms in contrast to code construction considerations
- The main contributors to these codes are combinatorial and convex optimization
- The object at the center of virtually all high performance decoding algorithms is the polytope  $\mathbb{P}(H)$  — a property of the code representation via  $H$  rather than a property of the code  $\mathbb{C}$ .
- The central charge is to understand  $\mathbb{P}(H)$  and to make LP decoding as efficient as message passing!

## Observations around the polytope $P$

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

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

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 $P$

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

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

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  $\frac{g-2}{2}$ .

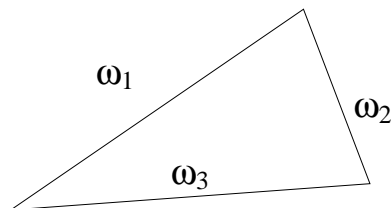
## Observations around the polytope $P$

The metric polytope of a binary code (matroid) is defined as:

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

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

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



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

Translating a theorem for binary matroids we get:

Theorem[Seymour, 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  $F_7^*$ ,  $M(K_5)$ ,  $R_{10}$

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

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

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

*....so far, so good...*

What about the interplay of  $P(H)$  and channels?

*The distance properties of  $\mathbb{P}(H)$*

$\mathbb{P}(H)$  is just a function of  $H$ , i.e. independent of the channel.

How can we assess if a parity check matrix is “good” for a given channel?

Traditionally we would consider the minimum Hamming distance of a code, reflecting the pairwise error probability.

For  $\mathbb{P}(H)$  the pairwise error probability between the word  $\mathbf{0}$  and  $\omega \in \mathbb{P}(H)$  is determined by the condition  $\langle \omega, \lambda \rangle \stackrel{?}{>} 0$

The effect of  $\langle \omega, \lambda \rangle \stackrel{?}{>} 0$  in different channels

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

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

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

$$r > \text{supp}(\omega) \Rightarrow d_p^{BEC}(\mathbf{0}, \omega) = \text{supp}(\omega)$$

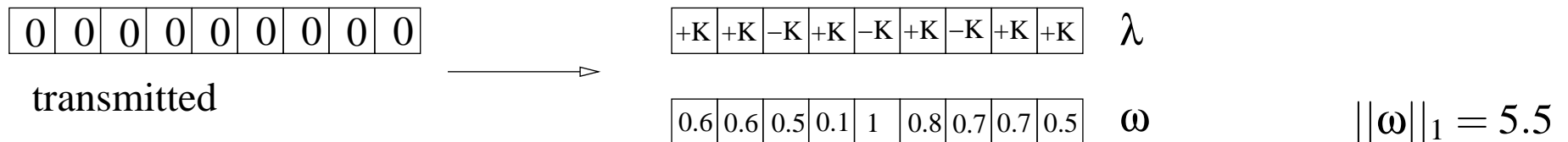
(Stopping sets)

## Binary symmetric channel

Vector in  $\mathbb{P}(H)$ :  $\omega = (\omega_1, \omega_2, \dots, \omega_n)$ ,

Error vector:  $e = (e_1, e_2, \dots, e_n)$

Log-likelihood ratios:  $\lambda_i \in \{+K, -K\}$



The condition  $\langle \omega, \lambda \rangle \stackrel{?}{>} 0$  may be satisfied with  $t$  errors if there exists a set  $J = \{i_1, i_2, \dots, i_t\} \subset \{1 \dots n\}$  such that:

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

## Effect on BSC

We have

$$d_p^{BSC}(\mathbf{0}, \boldsymbol{\omega}) = 2 \min_J \{ |J| : \sum_{i \in J} \omega_i > \sum_{i \notin J} \omega_i \} - 1$$

Note that:  $d_p^{BSC}(\mathbf{0}, \boldsymbol{\omega}) \geq \frac{\|\boldsymbol{\omega}\|_1}{\max_i \{\omega_i\}} \geq \|\boldsymbol{\omega}\|_1$

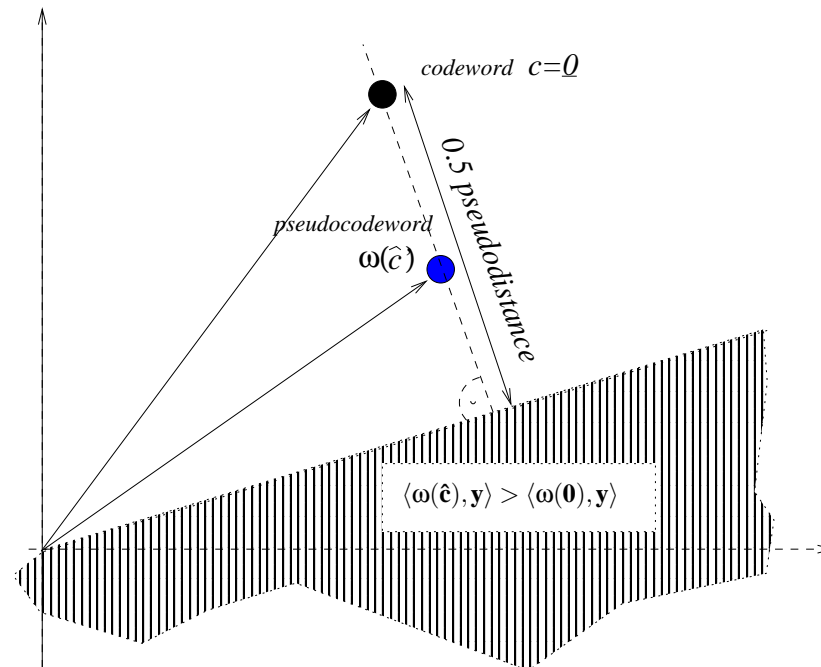
$\frac{\|\boldsymbol{\omega}\|_1}{\max_i \{\omega_i\}}$  is called the max-fractional distance and  $\|\boldsymbol{\omega}\|_1$  is called the fractional distance in this context.

While  $\|\boldsymbol{\omega}\|_1$  is easy to compute it is only a lower bound on the error correction capability.

# Additive White Gaussian Noise

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

$$\langle \frac{1}{2}(1 - \tilde{\omega}(\hat{c})), y \rangle \stackrel{?}{>} 0$$



On an AWGN channel the "distance" to the word  $\omega$  equals

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

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)	$10.8 < w_{p,\min}^{\text{AWGNC}} < 16.4$
Minimum pseudo-weight (BSC)	$w_{p,\min}^{\text{BSC}} < 10$

## The general case!

Assume a memoryless channel with binary input  $x \in \{0, 1\}$  output alphabet  $\mathcal{Y}$  and channel law  $p(y|x)$ .

$$P(\langle \omega, \lambda \rangle > 0) \leq E[\exp(-s(\langle \omega, \lambda \rangle))], \quad s > 0$$

$$E[\exp(-s(\sum_i \omega_i \log \frac{p(y|0)}{p(y|1)}))] = E[\prod_{i=1}^N (\frac{p(y|1)}{p(y|0)})^{s\omega_i}]$$

$$\prod_{i=1}^N E[(\frac{p(y|1)}{p(y|0)})^{s\omega_i}] = \exp(\sum_{i=1}^N \log(E[(\frac{p(y|1)}{p(y|0)})^{s\omega_i}]))$$

$$P(\langle \boldsymbol{\omega}, \boldsymbol{\lambda} \rangle > 0) \leq \exp\left(\sum_{i=1}^N \log(E[(\frac{p(y|1)}{p(y|0)})^{s\omega_i}])\right)$$

Consider the elements of  $\boldsymbol{\omega}$  as realizations of a r.v. drawn from the empirical distribution given by  $\boldsymbol{\omega}$ .

$$\sum_{i=1}^N \log(E[(\frac{p(y|1)}{p(y|0)})^{s\omega_i}]) = N E_{\boldsymbol{\omega}} \left[ \log(E_y[(\frac{p(y|1)}{p(y|0)})^{s\omega}]) \right]$$

The generalization of Bhattacharya distance to pseudodistance then becomes:

$$d_p = - \min_{s>0} E_{\boldsymbol{\omega}} \left[ \log(E_y[(\frac{p(y|1)}{p(y|0)})^{s\omega}]) \right]$$

$$d_p = - \min_{s>0} E_{\omega} \left[ \log \left( E_y \left[ \left( \frac{p(y|1)}{p(y|0)} \right)^{s\omega} \right] \right) \right]$$

and we get

$$P(\langle \omega, \lambda \rangle > 0) \leq \exp(-Nd_p)$$

(Bhattacharya distance:  $\omega_i \in \{0, 1\}$ ,  $s = 0.5$ ,  $d = -\log(E[\sqrt{p(y|1)p(y|0)}])$ )

## What's next?

We know how to evaluate the fundamental polytope on different channels based on the notion of pseudodistance.

This can be applied to arbitrary channels by the Bhattacharya pseudodistance

The task is now to construct matrices  $H$  such that  $d_p$  is large for given length and rate.

We need general bounding techniques for pseudodistance!

$$d_p^{BEC} \geq \left( \begin{array}{c} d_p^{BSC} \\ d_p^{AWGN} \end{array} \right) \geq \frac{\|\omega\|_1}{\max_i \{\omega_i\}} \geq \|\omega\|_1$$

## *Bounds on the Minimum Pseudo-Weight for the AWGN channel (?)*

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

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

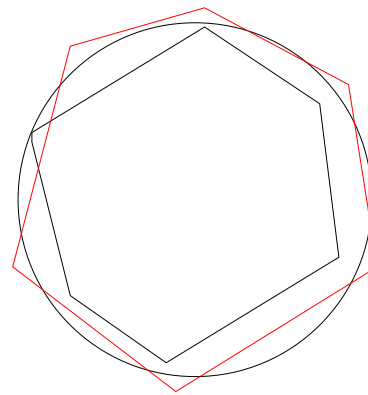
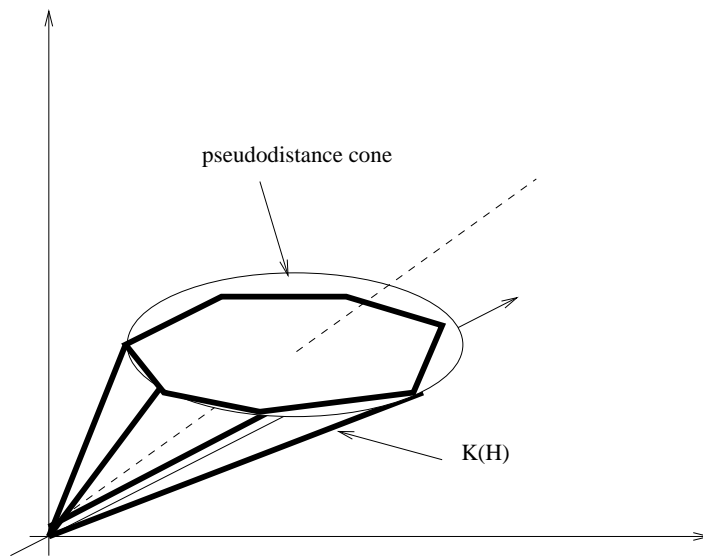
Techniques for obtaining **upper bounds** on the **min. pseudo-weight**:

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

## Some geometry

The pseudoweight on an AWGN is given as

$$\frac{\|\omega\|_1^2}{\|\omega\|_2^2} = \frac{\langle \omega, \mathbf{1} \rangle^2}{\|\omega\|_2^2} = \frac{1}{N} \cos^2(\angle(\omega, \mathbf{1})).$$

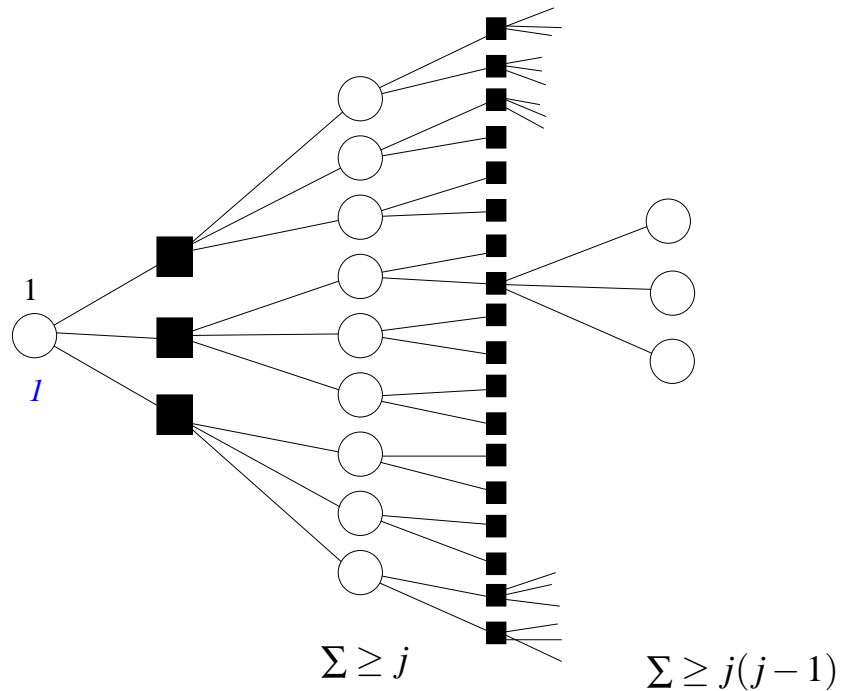


The simplest bound:  $\|\omega\|_1 \leq \frac{\|\omega\|_1}{\max_i \omega_i} \leq \frac{\|\omega\|_1^2}{\|\omega\|_2^2}$

$$\frac{\|\omega\|_1}{\max_i \omega_i} = \frac{\|\omega\|_1^2}{\max_i \omega_i \|\omega\|_1} = \frac{\|\omega\|_1^2}{\sum_j \omega_j \max_i \omega_i} \leq \frac{\|\omega\|_1^2}{\|\omega\|_2^2}$$

The normalized  $L_1$  norm is a simple lower bound on the pseudodistance of both the BSC and the AWGN channel and can be efficiently computed by a linear program. (Since we have a polynomial number of faces in  $P(H)$  we can find the vertex of minimum  $L_1$  norm in each face. The smallest nonzero  $L_1$  norm is the sought after quantity)

Assume the underlying graph has girth  $g$ .



$$\frac{\|\omega\|_1}{\max_i \omega_i} \geq j \sum_{i=1}^{\lceil \frac{g}{4} - 1 \rceil} (j-1)^i \approx (j-1)^{\lceil \frac{g}{4} \rceil}$$

## 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  $\mathbf{L} \triangleq \mathbf{H}^T \mathbf{H}$ .
- 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

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

Assume  $\omega \in K(H) \Rightarrow \forall j = 1 \dots m \|\omega_{\Gamma(j)}\|_1^2 \geq 2\|\omega_{\Gamma(j)}\|_2^2$

Consider a vector  $y = \omega H^T$ :

$$\|y\|_2^2 = \sum_{j=1}^m \|\omega_{\Gamma(j)}\|_1^2 \geq \sum_{j=1}^m 2\|\omega_{\Gamma(j)}\|_2^2 = 2j\|\omega\|_2^2$$

Let  $\omega$  be represented in the eigenspaces of  $L$  as  $\omega = \sum_i \alpha_i x_i$  with  $x_1$  being the normalized all one vector with eigenvalue  $\mu_1$  and  $\alpha_1 = 1/N\|\omega\|_1$ .

We have

$$\|y\|_2^2 = \omega L \omega^T = \sum \alpha_i^2 \mu_i \leq \alpha_1^2 \mu_1 + \mu_2 (\sum_{i=2}^N \alpha_i^2) = \alpha_1^2 \mu_1 + \mu_2 (\|\omega\|_2^2 - \alpha_1^2) = 1/N\|\omega\|_1^2 (\mu_1 - \mu_2) + \mu_2 \|\omega\|_2^2$$

We get:

$$1/N \|\omega\|_1^2 (\mu_1 - \mu_2) + \mu_2 \|\omega\|_2^2 \geq 2j \|\omega\|_2^2$$

or

$$\frac{\|\omega\|_1^2}{\|\omega\|_2^2} \geq N \frac{2j - \mu_2}{\mu_1 - \mu_2}$$

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

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

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

has the following properties:

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

Associate with each entry in  $M$  a variable  $M_{i,j}$  The linear program reads:

Maximize  $\sum_i M_{i,i}$

subject to  $M_{:,j} \in K(H); M_{i,:} \in K(H); \sum_{i,j} M_{i,j} = 1; M_{i,j} \geq 0; M_{i,j} = M_{j,i}$

The linear program can be improved by creating  $N$  LPs using constraints of type:  $M_{i,j} \geq M_{i,k}, \forall k \neq i$  etc.

## Upper bounds

We exhibit pseudocodewords of low weight. Assume we have an idea that a vector  $\mathbf{x}$  is similar to a bad pseudocodeword.

For a given parity check matrix  $H$  we can consider the program

Minimize:  $\langle \mathbf{x}, \boldsymbol{\omega} \rangle$

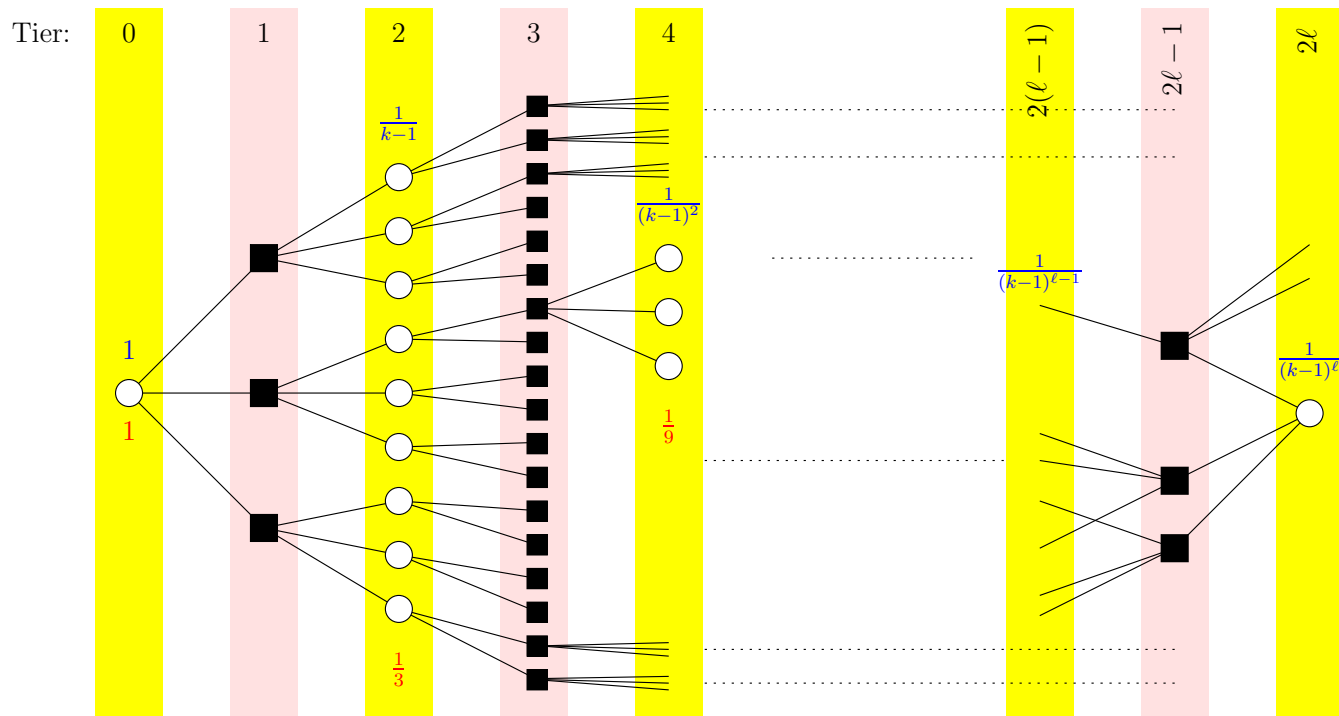
subject to:  $\boldsymbol{\omega} \in K(H); \|\boldsymbol{\omega}\|_1 = 1$

Repeating the procedure replacing  $\mathbf{x}$  with the found  $\boldsymbol{\omega}$  reveals a  $\boldsymbol{\omega}$  with low pseudoweight close to  $\mathbf{x}$ .

We can use this LP on all binary starting vectors of weight up to  $w$  which typically yields pseudocodewords of low weight very quickly.

(If none of the values of the program is below 0.5 then we can guarantee that the pseudoweight on a BSC is at least  $2w + 1$ )

# 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)

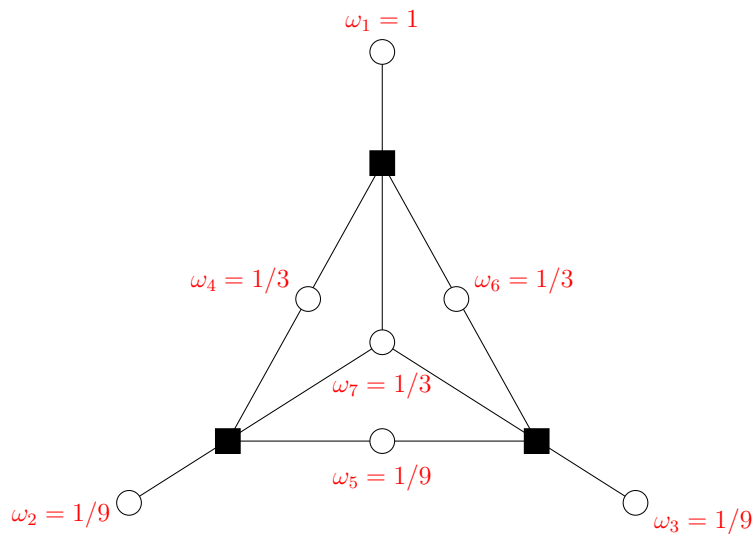
Example: [7, 4, 3] binary Hamming code.

The canonical completion **starting at  $\omega_1$**  is

$$\omega = \left( 1 \quad \frac{1}{9} \quad \frac{1}{9} \quad \frac{1}{3} \quad \frac{1}{9} \quad \frac{1}{3} \quad \frac{1}{3} \right).$$

The pseudo-weight of  $\omega$  is

$$\begin{aligned} w_p^{\text{AWGNC}}(\omega) &= \frac{\|\omega\|_1^2}{\|\omega\|_2^2} \\ &= \frac{\left( 1 + \frac{1}{9} + \frac{1}{9} + \frac{1}{3} + \frac{1}{9} + \frac{1}{3} + \frac{1}{3} \right)^2}{1 + \frac{1}{81} + \frac{1}{81} + \frac{1}{9} + \frac{1}{81} + \frac{1}{9} + \frac{1}{9}} \\ &= 3.973. \end{aligned}$$



Assume we have a canonically completed pseudocodeword  $\omega$ . Let bit  $i$  be at distance  $2t$  from the root.  $\omega_i = \frac{1}{(k-1)^t}$

$$\|\omega\|_1 = \sum_{t=0} N_{2t} \frac{1}{(k-1)^t}$$

$$\|\omega\|_2^2 = \sum_{t=0} N_{2t} \frac{1}{(k-1)^{2t}} \geq 1$$

$$N_{2t} \leq \frac{j}{j-2} (j-1)^{t-1} (k-1)^t \text{ and we get } \|\omega\|_1 \leq 1 + \sum_{t=1}^T \frac{j}{j-2} (j-1)^{t-1} (k-1)^t \leq \frac{j}{j-2} (j-1)^T$$

where  $T$  is chosen minimal so that  $\sum_{t=1}^T j(j-1)^{t-1}(k-1)^t \geq N$

Hence,  $\sum_{t=1}^{T-1} j(j-1)^{t-1}(k-1)^t < N$  and  $N > ((j-1)(k-1))^{T-1}$ .

It now follows  $T \leq 1 + \frac{\log N}{\log((j-1)(k-1))}$  and  $\|\omega\|_1 \leq \frac{j}{j-2}(j-1)^{1 + \frac{\log N}{\log((j-1)(k-1))}}$ .

$$\frac{\|\omega\|_1^2}{\|\omega\|_2^2} \leq \left(\frac{j(j-1)}{j-2}\right)^2 (j-1)^{2 \frac{\log N}{\log((j-1)(k-1))}}$$

or.....

## 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

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

where

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

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

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

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

(right hand side : Wiberg, Lentmaier et al.)

## One more surprising observation

For an AWGN channel we have:

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

where

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

For a BSC channel Felman et al. proved that a **constant fraction of errors** is decodable for some good enough expanders.

But then:

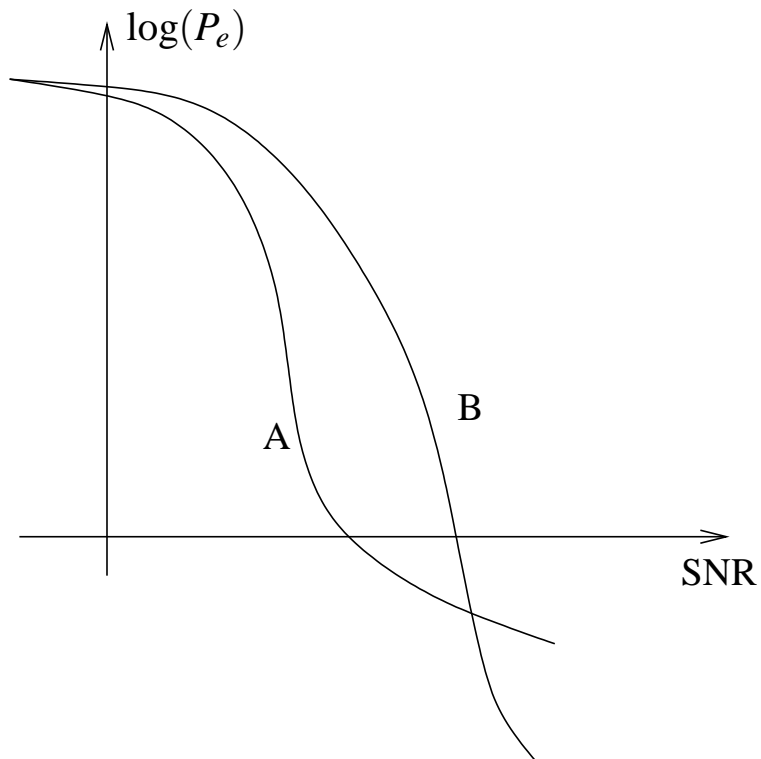
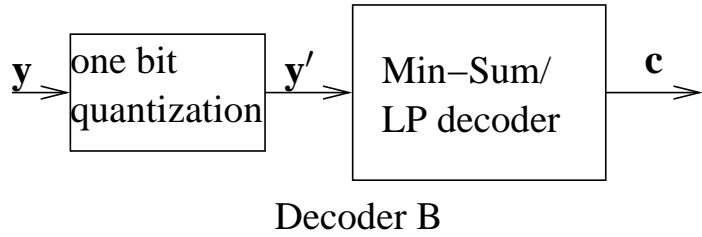
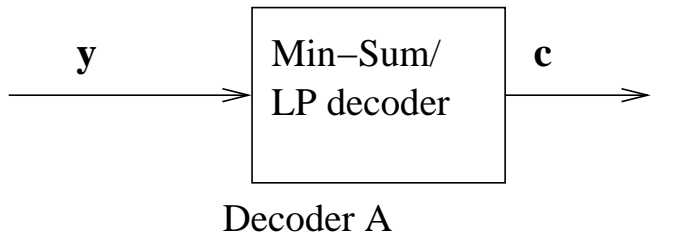
On AWGN:  $P_e > Ke^{-\alpha n^\beta}$

On BSC:  $P_e < K'ne^{-\alpha'n}$

$$P_e^{AWGN} > \alpha e^{\beta'_{j,k} \cdot n^{\beta_{j,k}}} > \alpha' e^{\beta'' n} > P_e^{BSC}$$

for large enough  $n$

Quantizing the received symbols will actually give a better performance !



## *How to solve the LP?*

The grand plan: Look at the dual program and use a suitably modified version of the Min-Sum or Sum-Product algorithm as computational engine.

## Primal and dual LPs

Primal program: minimize  $\langle \mathbf{x}, \mathbf{a} \rangle$  subject to  $A\mathbf{x}^T + \mathbf{b}^T \leq \mathbf{0}^T$ .

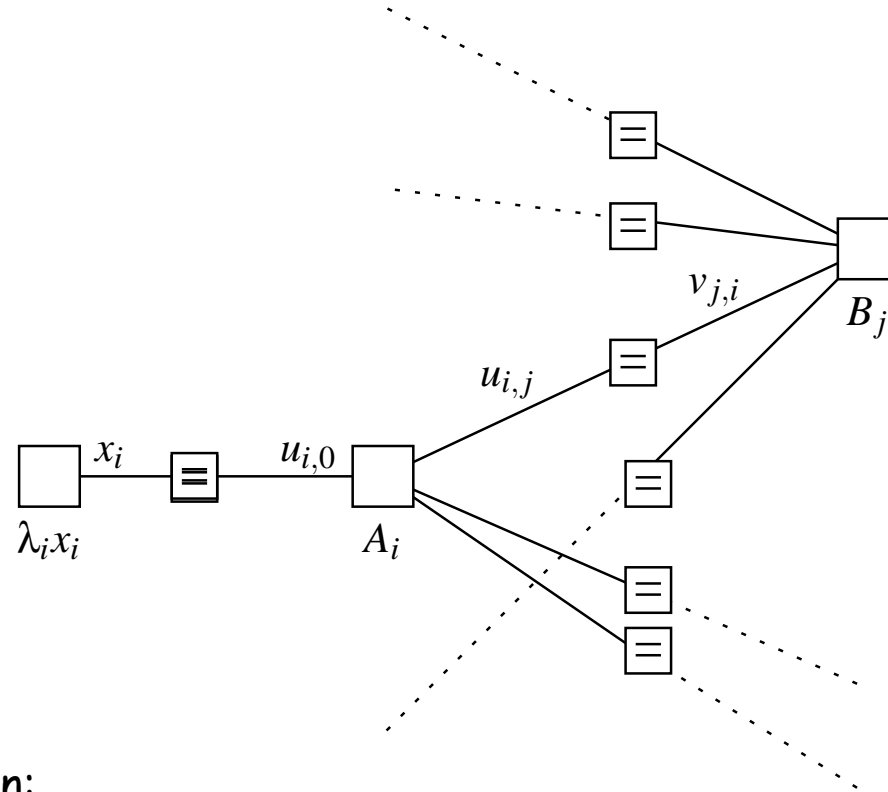
Dual program: maximize  $\langle \boldsymbol{\lambda}, \mathbf{b} \rangle$  subject to  $\boldsymbol{\lambda}A + \mathbf{a} = \mathbf{0}$ ,  $\boldsymbol{\lambda} \succeq \mathbf{0}$ .

$$\begin{aligned} J(\boldsymbol{\lambda}) &= \min_{\mathbf{x}} \{ \langle \mathbf{x}, \mathbf{a} \rangle + \boldsymbol{\lambda}(A\mathbf{x}^T + \mathbf{b}^T) \} \\ &= \min_{\mathbf{x}} \{ \langle \boldsymbol{\lambda}, \mathbf{b} \rangle + (\boldsymbol{\lambda}A + \mathbf{a})\mathbf{x}^T \} = \langle \boldsymbol{\lambda}, \mathbf{b} \rangle - \llbracket \boldsymbol{\lambda}A + \mathbf{a} = \mathbf{0} \rrbracket \end{aligned}$$

Assume  $\mathbf{x}^*$  is the solution to the primal problem.

$$\langle \mathbf{x}^*, \mathbf{a} \rangle \stackrel{\boldsymbol{\lambda} \succeq \mathbf{0}}{\geq} \langle \mathbf{x}^*, \mathbf{a} \rangle + \boldsymbol{\lambda}(A\mathbf{x}^{*T} + \mathbf{b}^T) \geq J(\boldsymbol{\lambda})$$

# A graphical representation of the primal LP



Objective function:

$$\sum_i \lambda_i x_i + \sum_i \llbracket x_i = u_{i,0} \rrbracket + \sum_i \llbracket \underline{u}_i \in CH(A_i) \rrbracket + \sum_{i,j} \llbracket u_{i,j} = v_{i,j} \rrbracket + \sum_j \llbracket \underline{v} \in CH(B_j) \rrbracket$$

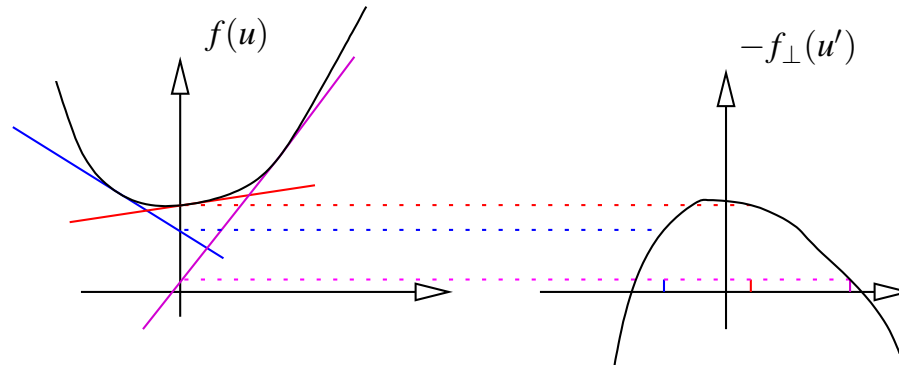


## Dualizing a graphical convex optimization problems

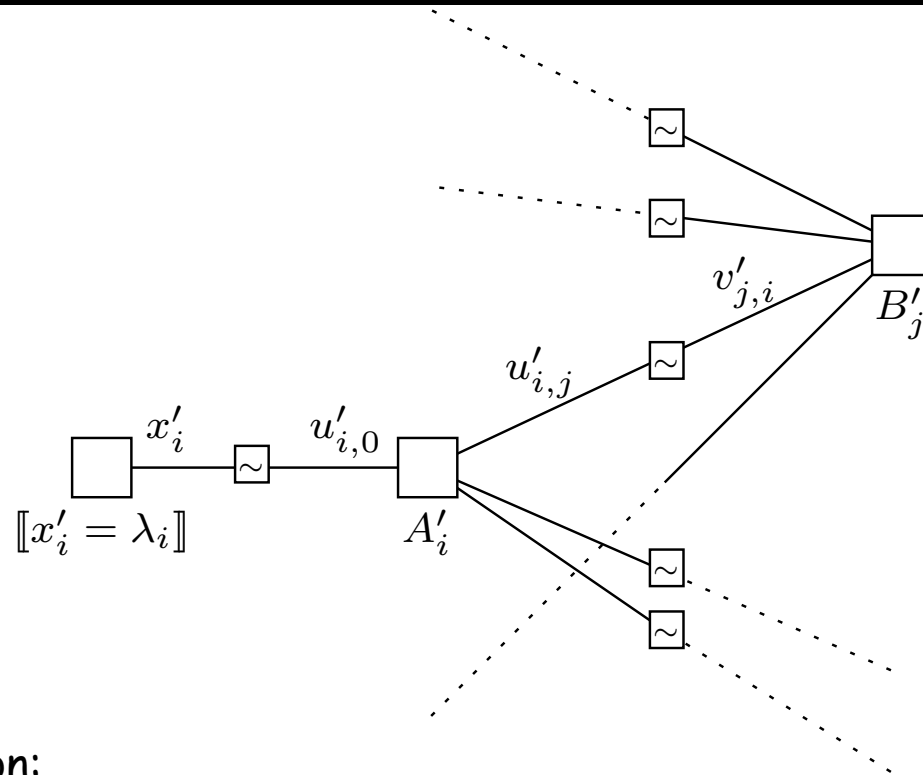
The dual of a graphical convex optimization problem is obtained by dualizing the individual cost contributions:

Fenchel duality:  $f_{\perp}(\underline{u}') \leftrightarrow f(\underline{u})$

$$f_{\perp}(\underline{u}') = \sup_{\underline{u}} (\langle \underline{u}, \underline{u}' \rangle - f(\underline{u}))$$



# A graphical representation of the dual LP



Objective function:

$$\begin{aligned}
 & - \sum_i [\lambda_i = x'_i] - \sum_i [x'_i = -u'_{i,0}] - \sum_i \min_{\mathbf{a} \in CH(A_i)} \langle \underline{u}'_i, \mathbf{a} \rangle - \\
 & \sum_{i,j} [u_{i,j} = -v_{i,j}] - \sum_j \min_{\mathbf{b} \in CH(B_j)} \langle \underline{v}'_j, \mathbf{b} \rangle
 \end{aligned}$$

## Solving the dual LP

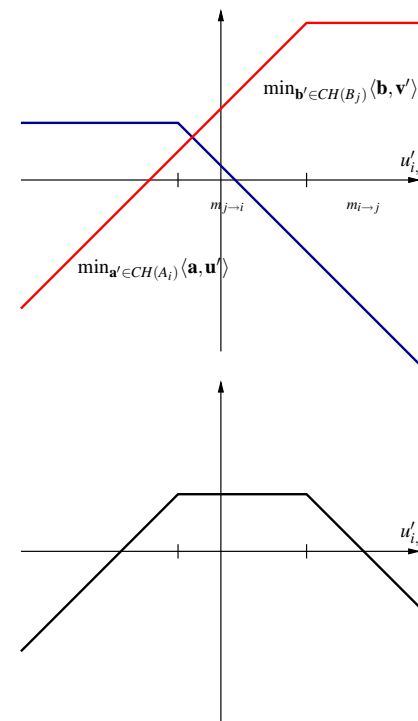
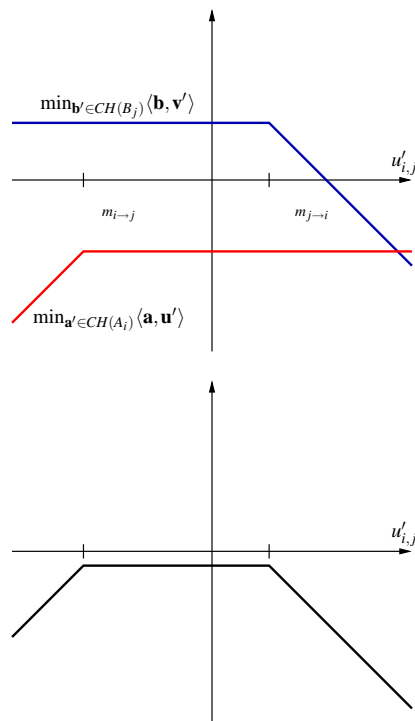
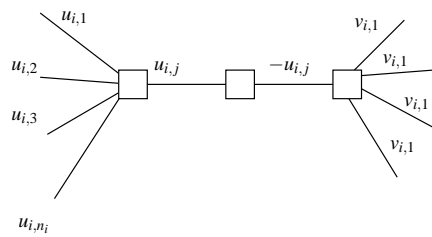
Find an assignment of variables that maximizes

$$\sum_i \min_{\mathbf{a} \in CH(A_i)} \langle \underline{u}'_i, \mathbf{a} \rangle + \sum_j \min_{\mathbf{b} \in CH(B_j)} \langle \underline{v}'_j, \mathbf{b} \rangle$$

subject to  $u'_{i,j} = -v'_{i,j}$  and  $u'_{i,0} = -\lambda_i$

A variety of algorithms is available for this task....

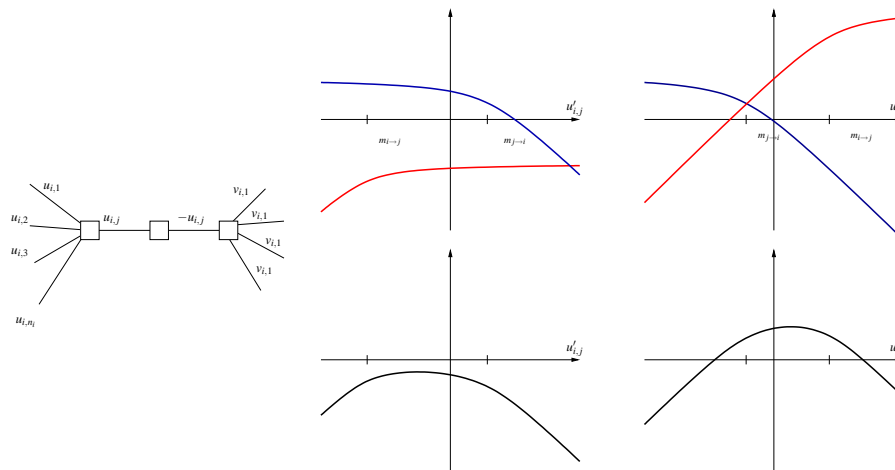
Consider a single edge  $(i, j)$  in the graph. Assume all values of dual variables are fixed except  $u'_{i,j}$  and  $v'_{i,j} = -u'_{i,j}$ .



## A modification

We replace the minimization with "soft" minima

$$\min_{\mathbf{a} \in CH(C)} \langle \underline{u}, \mathbf{a} \rangle \rightarrow \kappa \log \sum_{\mathbf{a} \in CH(C)} e^{-\frac{1}{\kappa} \langle \underline{u}, \mathbf{a} \rangle}$$



This leads to Sum-Product vs. Max-Product

The "soft min" version solves

$$\langle \lambda, \mathbf{x} \rangle + \mathbb{I}[\mathbf{x} \in \mathbb{P}] - \kappa(H(\alpha) + H(\beta))$$

$\alpha, \beta$  are the probability mass function for the minimizing point in  $CH(A_i)$  and  $CH(B_j)$ .

## An Algorithm

- 1) Run an iteration of the Sum-Product / Min-Sum algorithm
  - 2) Project the found values  $u_{i,j}$  and  $v_{i,j}$ , i.e. the messages  $m_{i \rightarrow j}$  and  $m_{j \rightarrow i}$  onto the line  $u_{i,j} + v_{i,j} = 0$ .
- \* Convergence of the algorithm is guaranteed(!) if one considers one edge at a time.
  - \* For the converged algorithm exactness of the LP solution can be guaranteed ( $\kappa \rightarrow 0$ )
  - \* Correctness of the hard decisions can be guaranteed for the decidable positions.

*Conclusions: What we know and what we don't know: LP — BP*

- Precise characterization of the algorithm (✓, ?)
- Equivalent minimum distance characterization (✓, ?)
- Cycle codes are well understood, (✓, ✓)
- Ensemble behavior (?, ✓?)
- Analytic error bounds (?, ✓)
- Analytic proof of thresholds (✓, ✓)

## Conclusions

- LP decoding performance is close to iterative decoding performance (typically, between Sum-Product and Min-Sum decoding)
- The polytope  $\mathbb{P}$  characterizes the LP decoder completely.
- On different channels the effects of  $\mathbb{P}$  are different - and fairly well understood
- The main problem becomes the construction of polytopes  $\mathbb{P}$  with large pseudodistance — the new game is  $(n, k, d_p(H))$  instead of  $(n, k, d_H)$
- General purpose LP solvers run in polynomial time but are fairly inefficient.
- Belief-propagation algorithms can be (slightly) modified to provide highly efficient LP solvers (typically for integer, low-density check structures)

- LP decoding becomes an attractive opportunity, especially if theoretical guarantees are required.

## References

Feldman, Karger, Wainwright, "LP Decoding", Allerton 2003

Kolmogorov and Wainwright, "On the optimality of tree-reweighted max-product messagepassing", UAI,2005

Wainwright, Jaakkola, Willsky, "Map estimation via agreement on (hyper)trees: Message passing and linear programming approaches", Allerton 2002

Yedidia, Freeman, and Weiss, "Constructing free energy approximations and generalized belief propagation algorithms", IEEE-IT 2005

Weiss, Yanover, Meltzer, "MAP Estimation, Linear Programming and Belief Propagation with Convex Energies" preprint 2006.

[www.pseudocodewords.info](http://www.pseudocodewords.info)