



Designing Error Correction Codes for Iterative Decoding

Sarah Johnson

`sarah.johnson@nicta.com.au`

National ICT Australia, Sydney, Australia

Contents

- **Low-density parity-check (LDPC) codes**
 - Error correction codes
 - Iterative sum-product decoding
 - What makes a good LDPC code?
- **Algebraic LDPC codes**
 - Steiner 2-designs
 - Partial geometries
 - Overview of the codes
- **Work in progress**
 - Dual field codes
 - Burst error correction
 - Coding on Markov channels

Error correction codes

Parity check (error detection):

message $c_1 c_2 c_3 \rightarrow$ codeword $c_1 c_2 c_3 c_4$ where $c_1 \oplus c_2 \oplus c_3 \oplus c_4 = 0$

Parity checks (error correction):

message $c_1 c_2 c_3 \rightarrow$ codeword $\mathbf{c} = c_1 c_2 c_3 c_4 c_5 c_6$,

where

$$c_1 \oplus c_2 \oplus c_4 = 0$$

$$c_2 \oplus c_3 \oplus c_5 = 0$$

$$c_1 \oplus c_2 \oplus c_3 \oplus c_6 = 0$$

in matrix form:

$$\underbrace{\begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}}_H \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Parity-check matrix H

a codeword $\Leftrightarrow Hc^T = 0$

To generate the codeword for a given message, the code constraints can be rewritten:

$$\begin{array}{lcl}
 c_1 \oplus c_2 \oplus c_4 = 0 & & c_4 = c_1 \oplus c_2 \\
 c_2 \oplus c_3 \oplus c_5 = 0 & \iff & c_5 = c_2 \oplus c_3 \\
 c_1 \oplus c_2 \oplus c_3 \oplus c_6 = 0 & & c_6 = c_1 \oplus c_2 \oplus c_3
 \end{array}$$

In matrix form:

$$\left[\begin{array}{cccccc} c_1 & c_2 & c_3 & c_4 & c_5 & c_6 \end{array} \right] = \left[\begin{array}{ccc} c_1 & c_2 & c_3 \end{array} \right] \underbrace{\left[\begin{array}{cccccc} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{array} \right]}_G$$

e.g. $\left[\begin{array}{ccc} c_1 & c_2 & c_3 \end{array} \right] = 110$ gives

$$c_4 = 1 \oplus 1 = 0,$$

$$c_5 = 1 \oplus 0 = 1$$

$$c_6 = 1 \oplus 1 \oplus 0 = 0$$

hence the codeword 110010

ML decoding: choose the codeword closest to the received vector

Low-density parity-check codes

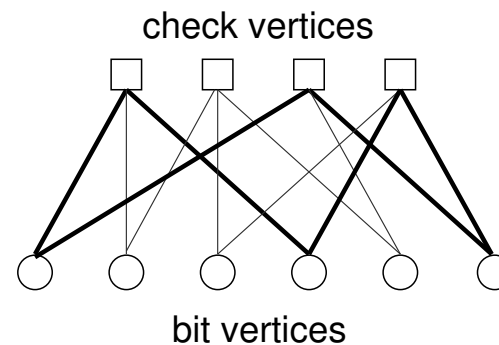
- introduced by Gallager (1962)
- rediscovered by MacKay and Neal (1996, 1999)
- linear block code defined as the null space of sparse parity-check matrix H

Tanner graph

bipartite graph representing linear code

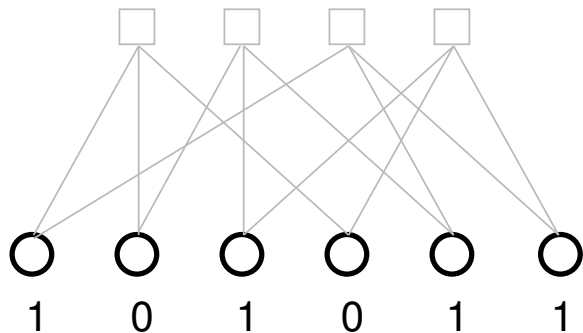
- edge in graph connects j -th bit vertex with i -th parity-check vertex iff $H_{i,j} = 1$
- *regularity*: all nodes of the same type have the same degree

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

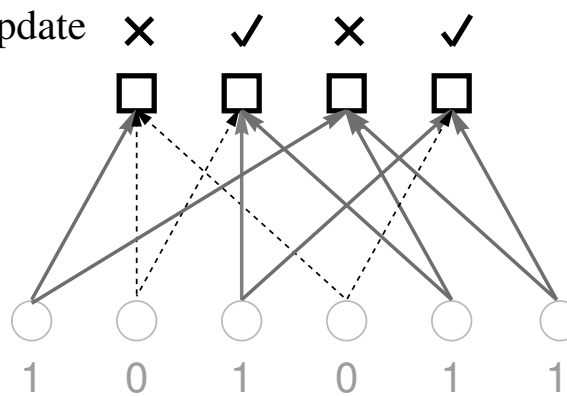


Iterative decoding

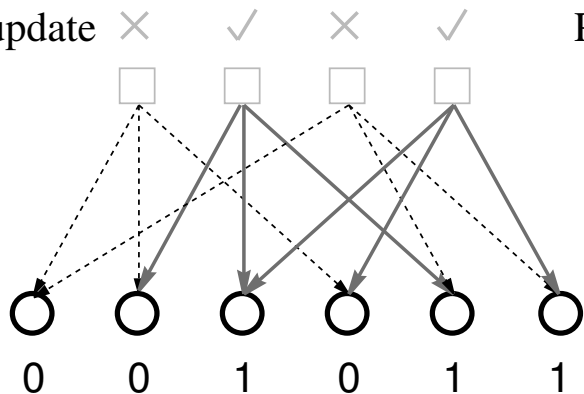
Initialization



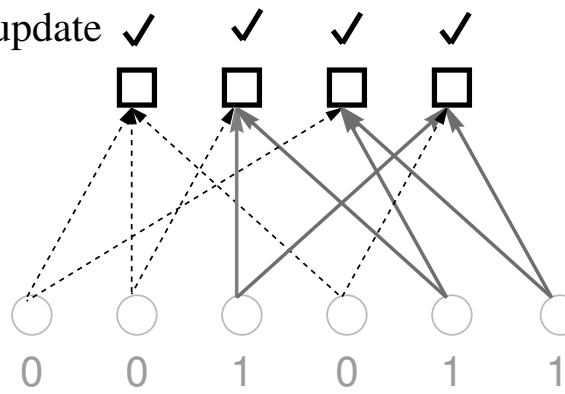
Parity update



Bit update

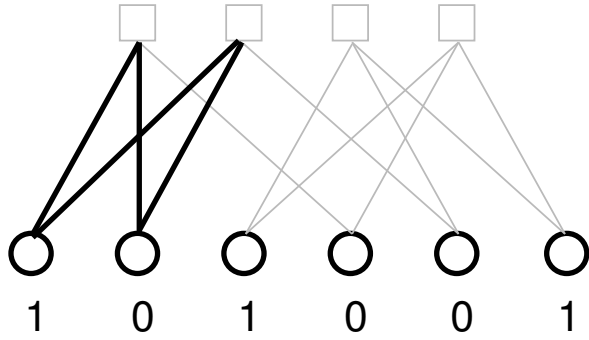


Parity update

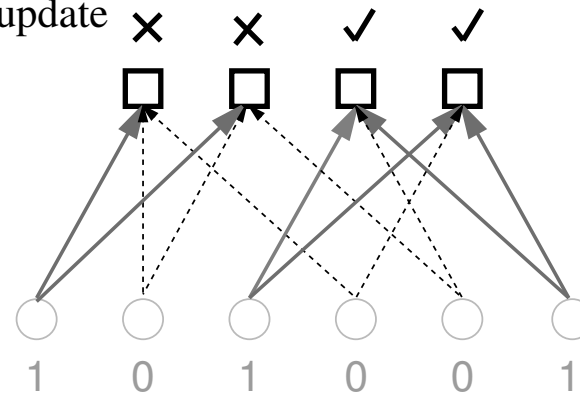


Bit-flipping decoding: sent word 001001, received word 101001

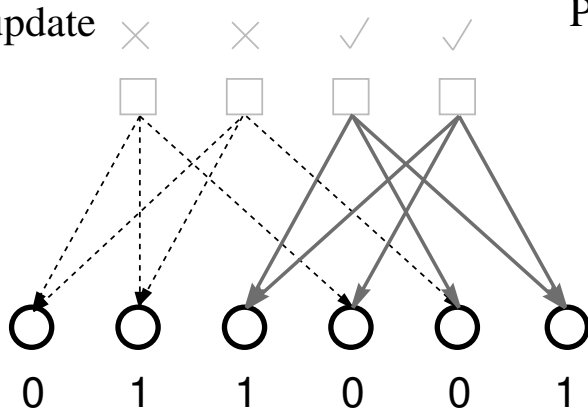
Initialization



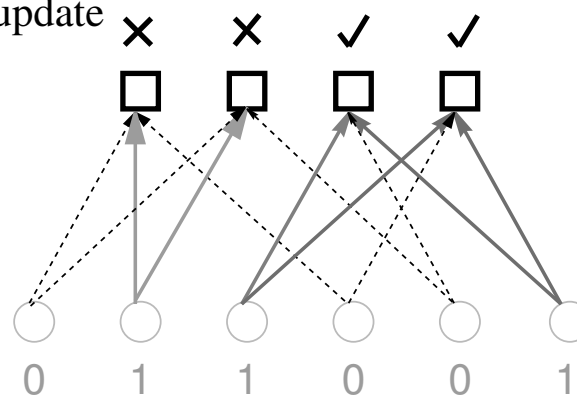
Parity update



Bit update



Parity update



Bit-flipping decoding: sent word 001001, received word 101001

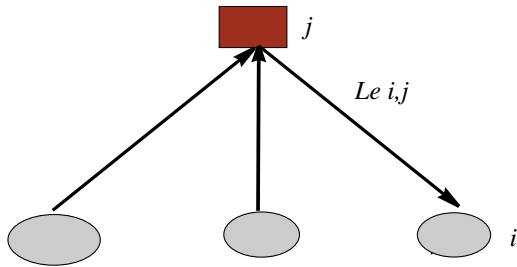
Sum-product decoding

Step 1 (Initialisation)

For each bit i : $L_i = \log_e \left(\frac{1-P_i}{P_i} \right)$

Set $L_{i,j} = L_i \forall j$

Step 2 - Extrinsic probability metrics



For each check j calculate $Le_{i,j}$ the extrinsic probability of bit i from check j :

$$Le_{i,j} = \log \left(\frac{1 + \prod_{k|k \in B_j, k \neq i} \tanh(L_{k,j}/2)}{1 - \prod_{k|k \in B_j, k \neq i} \tanh(L_{k,j}/2)} \right)$$

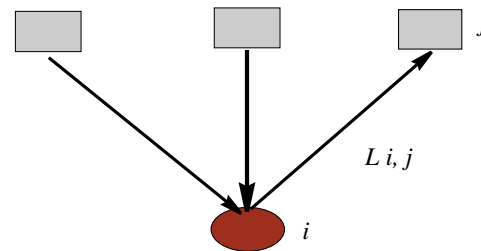
Step 3 - Check for a valid codeword

$$z_i = L_i + \sum_{j|i \in B_j} Le_{i,j}$$

$$\hat{z}_i = \begin{cases} 1, & z_i \leq 0 \\ 0, & z_i > 0 \end{cases}$$

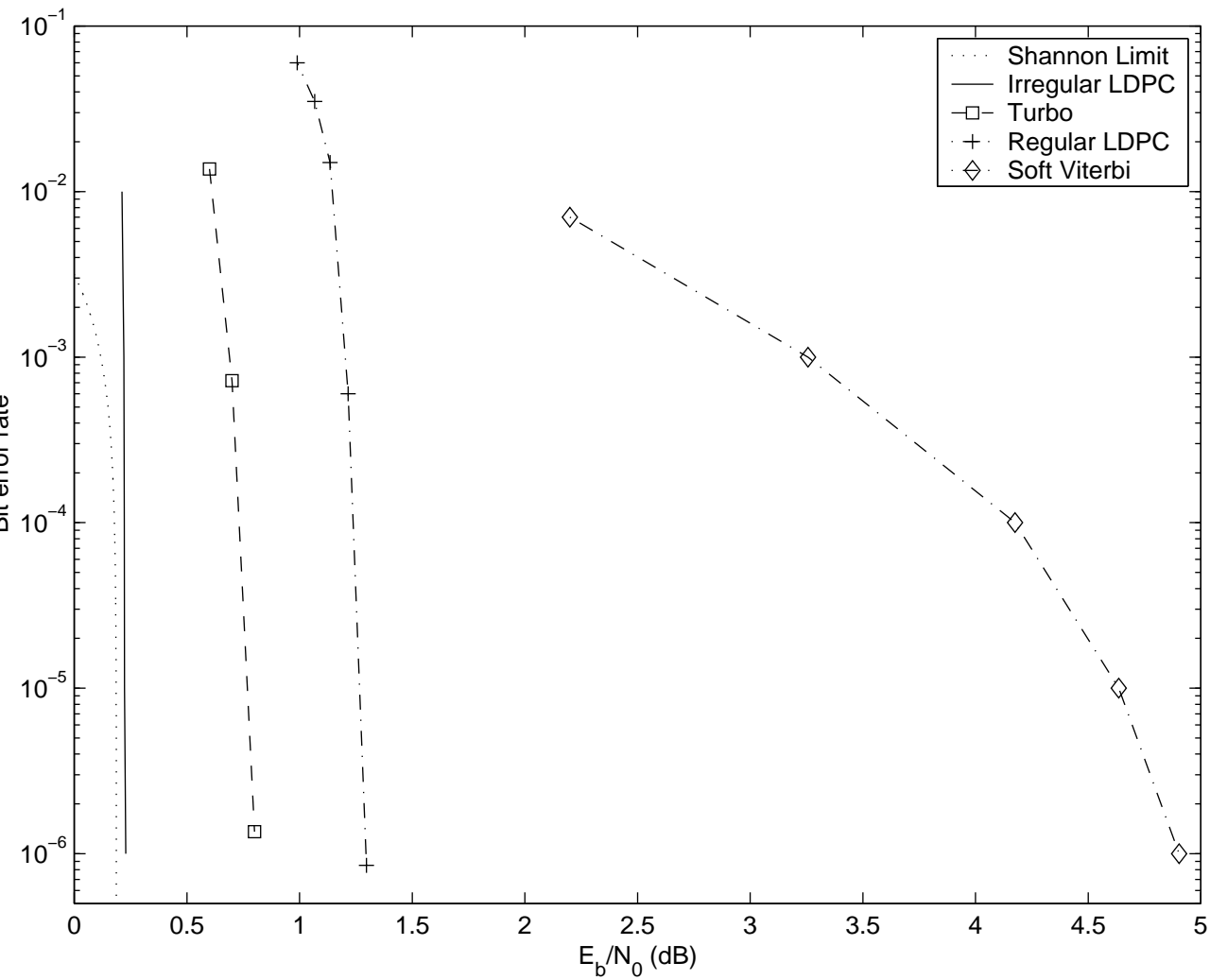
If $\hat{z}H^T = 0$ or max iterations reached finish here,
otherwise return to Step 1.

Step 1 - Update bit probability metrics



$$L_{i,j} = \sum_{l|i \in B_l, l \neq j} Le_{i,j} + L_i$$

Why all the excitement?



Rate-1/2 codes on AWGN

- Soft Viterbi decoding of a constraint length 7 convolutional code
- Regular Gallager code length 65389;
- Turbo code with 2+32 states, 16384 bit interleaver, and 18 iterations
- Length 10^7 optimized irregular code
- Shannon limit at rate 1/2

Constructing LDPC codes

- Gallager gave a (semi-) random construction of H :
 - ◇ column weight γ
 - ◇ row weight ρ
 - ◇ $\gamma, \rho \ll n$
 - ◇ $H : J \times n$, not necessarily full rank
- For very long codes \Rightarrow good codes are easily constructed randomly
 - average girth grows with code length
 - convergence to an ensemble average in codeword limit

Q how do we implement codes with these huge random parity-check matrices?

.. storage ... encoding ... flexibility

- ◇ Can we construct H without 4-cycles systematically?
 - ▷ Yes - projective and Euclidean geometries, balanced incomplete block designs (BIBD) + others

Combinatorial designs

The problem of selecting subsets ('blocks') from a finite set ('points') in such a way that specified conditions on subset intersection are met.

Steiner 2-design

- each block has γ points and each point in r blocks
- two points occur together in exactly one block together

eg $\gamma = 2, r = 3$

points $\{1, 2, 3, 4\}$
blocks $[1,2], [1,3], [2,3],$
 $[2,4], [1,4], [3,4]$

incidence matrix

$$N_{i,j} = \begin{cases} 1 & \text{if point } i \in \text{Block } j, \\ 0 & \text{otherwise} \end{cases}$$

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

Combinatorial designs for LDPCs

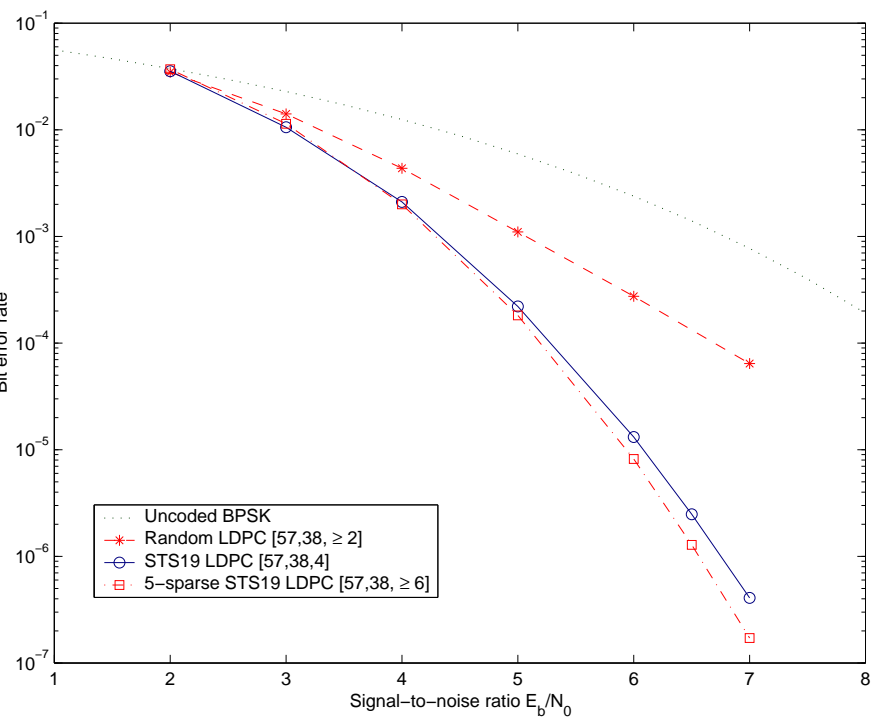
Idea is to use incidence matrix of a Steiner 2-design as the parity-check matrix of an LDPC code.

- **low density** \Rightarrow fraction of ones: γ/v
- **minimum distance** $\Rightarrow d_{\min} \geq \gamma + 1$
- **no 4-cycles** \Rightarrow girth ≥ 6
 - 6-cycles: $N_6 = \frac{b}{3} \binom{\gamma}{2} (r - 1)(\gamma - 1)$
- **regular** – column weight γ , row weight r

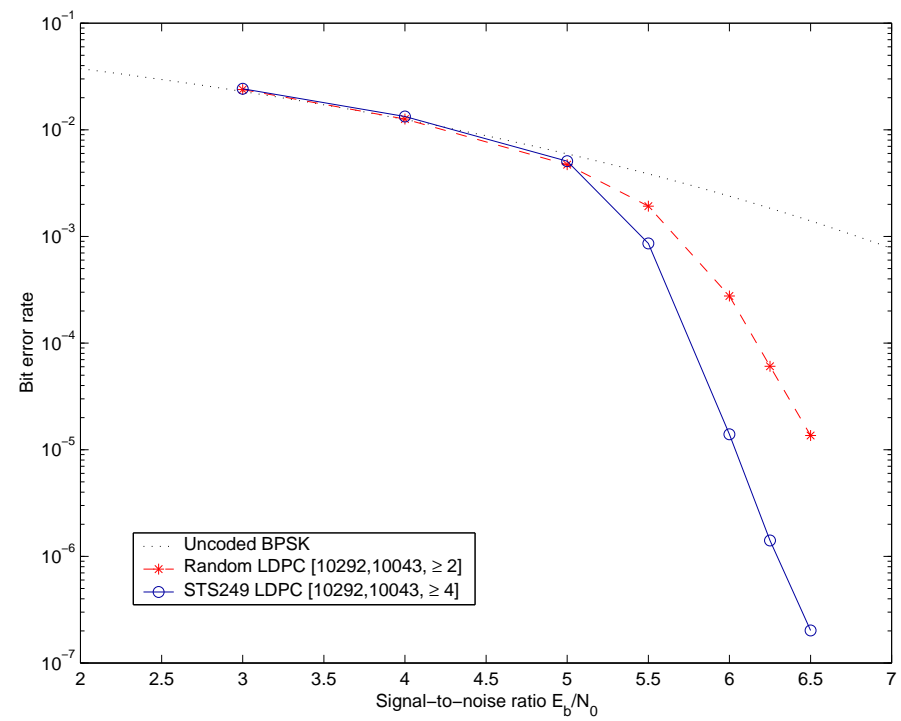
Classes

- Steiner triple systems (STS), constant $\gamma = 3$
- Steiner 2-designs with fixed $\gamma \geq 4$.
- oval designs, specified by $v = (\gamma - 1)(\gamma - 2)/2$ for all integers γ ,
- unital designs, specified by $v = \gamma^3 - 3\gamma^2 + 3\gamma$ for all integers γ ,

Example: triple systems

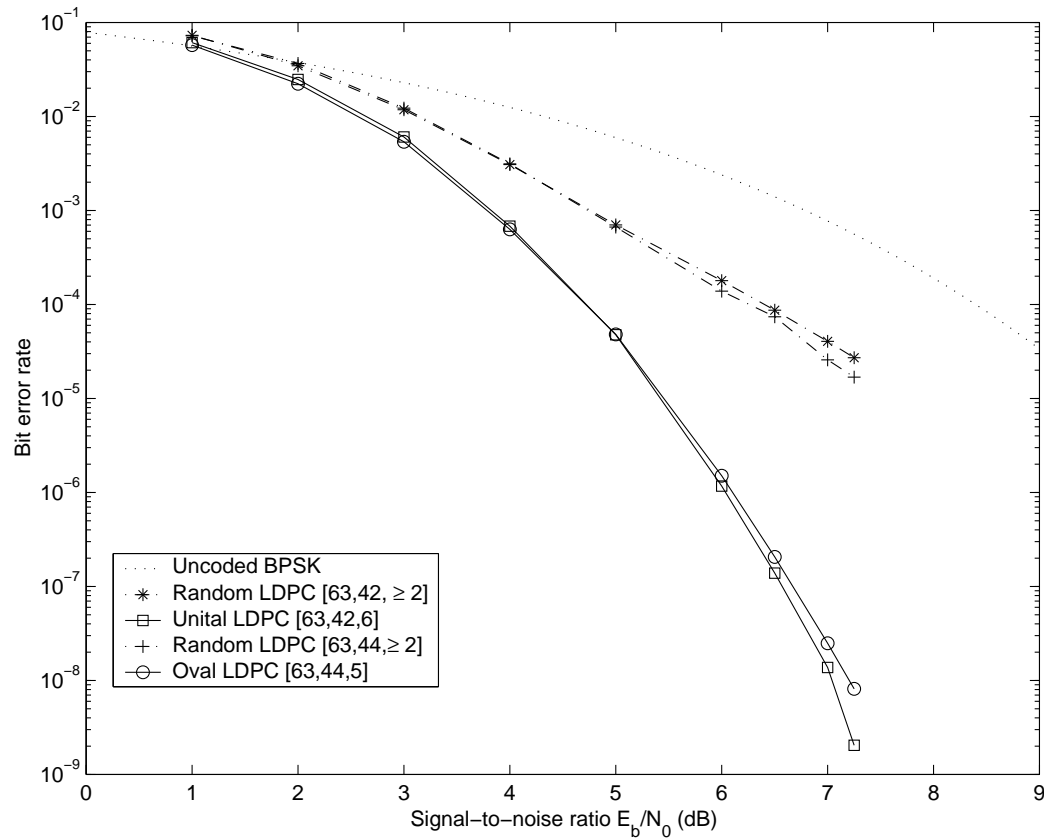


(c) length 57 column weight 3 LDPC codes



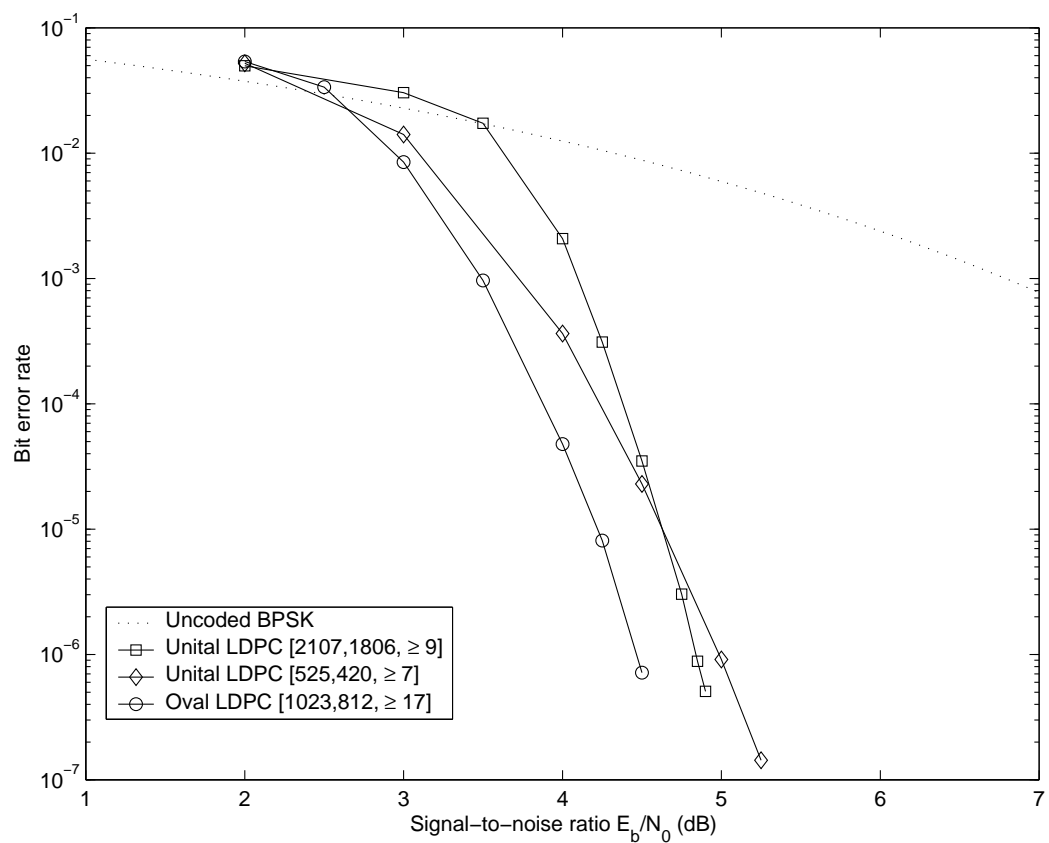
(d) length 10292 column weight 3 LDPC codes

Example: unital and oval codes



- sum-product decoding with max. 10 iterations
- for the random codes used MacKay–Neal heuristics to remove 4-cycles while attempting to keep row weights constant

Example: unital and oval codes

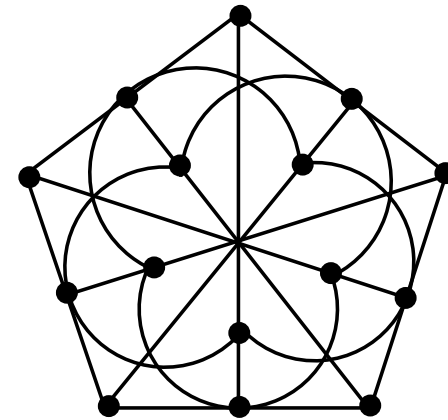


- sum-product decoding with max. 10 iterations
- larger col weight \Rightarrow better performance in low noise channels

Partial geometries

A set of points and lines $\text{pg}(s, t, \alpha)$:

1. each point on $t + 1$ lines, each line through $s + 1$ points
2. two points on **at most** one line together
3. point p **not** on line $L \Rightarrow \alpha$ lines through p intersect L

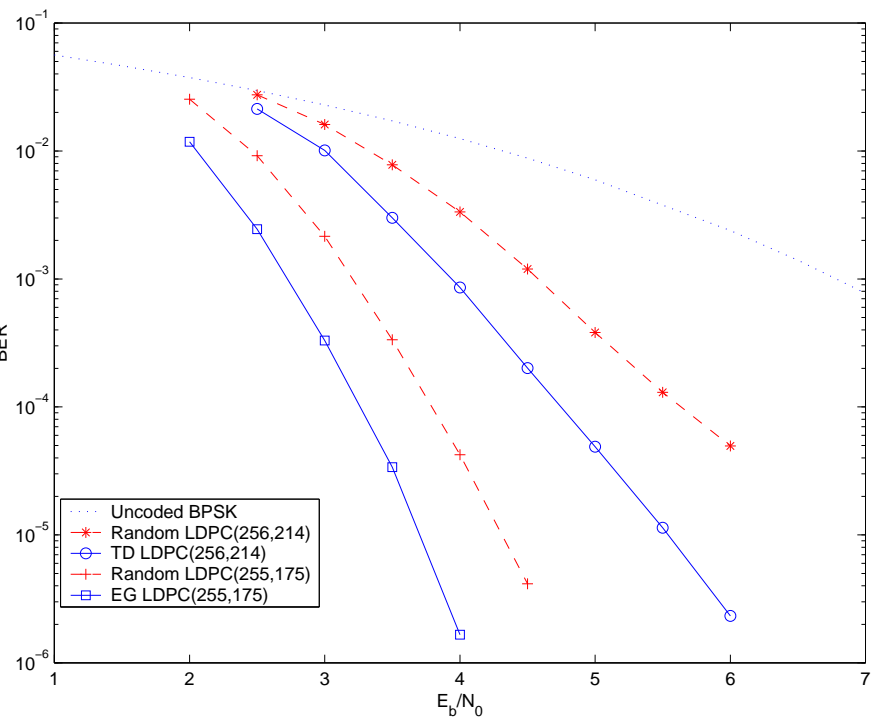


$\text{pg}(2,2,1)$

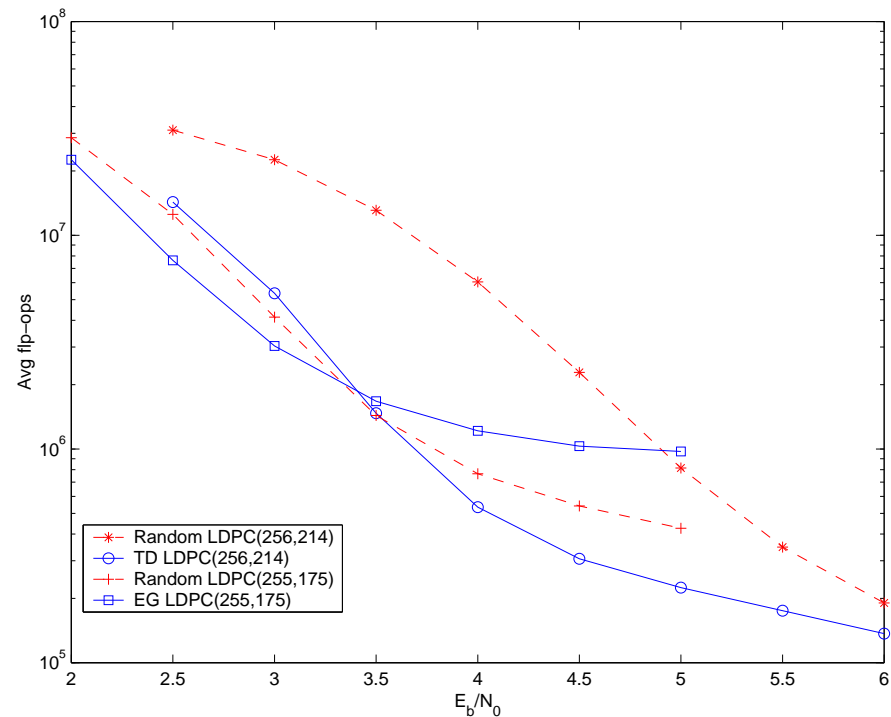
LDPC codes from partial geometries

- **low density** \Rightarrow fraction of ones: $\alpha/(st + \alpha)$
- **minimum distance** $\Rightarrow d_{\min} \geq \max\{(t + 1)(s + 1 - t + \alpha)/\alpha, 2(s + \alpha)/\alpha\}$
- **no 4-cycles** \Rightarrow girth ≥ 6
- **dependent rows in \mathbf{H}** \Rightarrow at least $\frac{s(s+1-\alpha)(st+\alpha)}{\alpha(s+t+1-\alpha)}$ dependent rows
- **regularity** $\Rightarrow (s + 1, t + 1)$ -regular

Example: Transversal designs



(e) Error correction performance (BER vs. E_b/N_0)



(f) Decoding complexity (Average number of floating point operations to decode a codeword)

Quasi-cyclic codes

Motivation \Rightarrow encoding can be done in $O(n)$ time

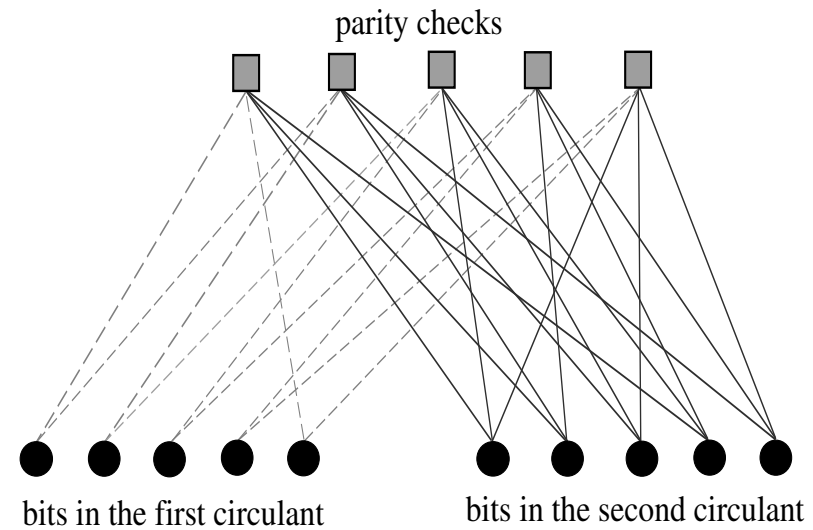
A rate 1/2 quasi-cyclic code from circulants

$$H = \left[\begin{array}{cccc|cccc} 1 & 1 & & & 1 & 1 & & 1 \\ & 1 & 1 & & 1 & 1 & & 1 \\ & & 1 & 1 & & 1 & 1 & 1 \\ & & & 1 & 1 & & 1 & 1 \\ 1 & & & & 1 & 1 & & 1 \end{array} \right]$$

Parity check matrix with two circulants $[A_1|A_2]$

$$G = \left[\begin{array}{cccc|cccc} 1 & & & & 1 & & & 1 \\ & 1 & & & & 1 & & 1 \\ & & 1 & & & & 1 & 1 \\ & & & 1 & & & & 1 \\ & & & & 1 & & & 1 \end{array} \right]$$

Generator matrix $[I|A_1^{-1}A_2]$



Take circulant matrices from difference families \Rightarrow regular, 4-cycle free, quasi-cyclic codes

Summary: LDPC codes

codes	checks	col, row wgt	length	lin. dep. rows	rate \approx	d_{\min}
STS(v)	$v \equiv 1, 3 \pmod{6}$	3 , $\frac{v-1}{2}$	$\frac{v(v-1)}{6}$	$\leq \log_2(v+1)$	$(\mathbf{v-7})/(\mathbf{v-1})$	≥ 4 (6)
DF(v)	$v \equiv 1 \pmod{\gamma(\gamma-1)}$	$\gamma = \mathbf{2 \cdots 7}$, $\frac{v-1}{\gamma-1}$	$\frac{v(v-1)}{\gamma(\gamma-1)}$		$\frac{(\mathbf{v-1})-\gamma(\gamma-1)}{\mathbf{v-1}}$	$\geq \gamma + 1$
KTS(v)	$v \equiv 3 \pmod{6}$	3 , any \mathbf{r}	$r * v/3$	0 or 1	$(\mathbf{r-3})/\mathbf{r}$	≥ 4

codes	checks	col wgt	row wgt	length	lin. dep. rows	d_{\min}
oval(m)	$2^{m-1}(2^m - 1)$	2^{m-1}	$2^m + 1$	$2^{2m} - 1$	$\frac{2^{2m}-2^m}{2} - 3^m + 2^m$	$\geq \mathbf{2^{m-1} + 1}$
unital(m)	$m^3 + 1$	$m + 1$	m^2	$m^2 \frac{m^3+1}{m+1}$	m^2 (if $2 m+1$)	$\geq \mathbf{m + 2}$

codes	checks	col, row wgt	length	lin. dep. rows	d_{\min}
PPG(s, t, α)	$\frac{(s+1)(st+\alpha)}{\alpha}$	$s + 1, t + 1$	$\frac{(t+1)(st+\alpha)}{\alpha}$	$\geq \frac{s(s+1-\alpha)(st+\alpha)}{\alpha(s+t+1-\alpha)}$	(previous)
TD(s, t)	$(s+1)(t+1)$	$s + 1, t + 1$	$(t+1)^2$	s	$\geq \mathbf{s + 2}$

codes	checks	col, row wgt	length	lin. dep. rows	d_{\min}
quasi-cyclic	v	$\gamma \leq 7$, any $r \in [\gamma, 2\gamma, \dots, \frac{v-1}{\gamma-1}]$	rv/γ	0	$\gamma + 1$