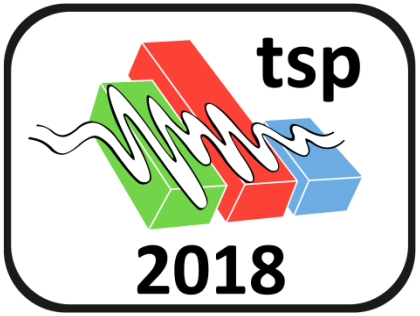
Simulated Annealing Method for Construction of High-Girth QC-LDPC Codes

Vasiliy Usatyuk,
South-West State University,
Kursk, L@Lcrypto.com

Ilya Vorobyev,
Institute for Information Transmission Problems,
Moscow, vorobyev.i.v@yandex.ru



Athens, Greece
July 5, 2018

Low Density Parity-Check Codes

$$H = \begin{array}{c} \begin{matrix} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \end{matrix} \\ \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \end{array} \begin{matrix} c_1 \\ c_2 \\ c_3 \end{matrix}$$

Low density parity-check codes (LDPC-codes), – block linear code of dimension *k* and code words x size *n*, defined by parity-check matrix *H* with size (*n-k*)·*n*, which contain low density-parity check.

Parity-check matrix of size (*n-k*)·*n contain about n checks.*

Every row of parity-check matrix *H define equation*:

$$Hx^T = 0$$

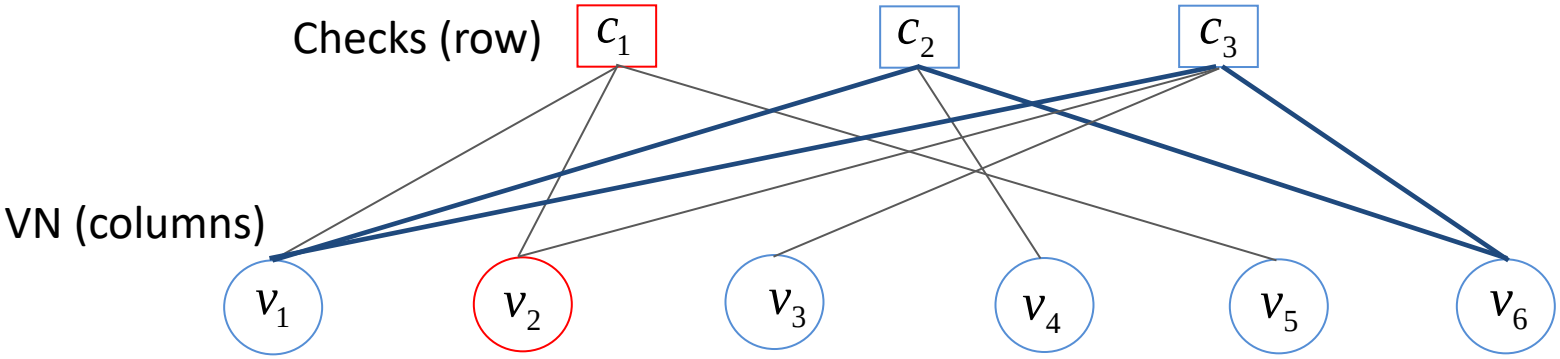
$$\begin{aligned} x_1 + x_2 + x_5 &= 0 \\ x_1 + x_4 + x_6 &= 0 \\ x_1 + x_2 + x_3 + x_6 &= 0 \end{aligned} \qquad GF(2)$$

Gallager R.G., “Low-density parity-check codes”, IRE Trans. Inform. Theory, vol. IT-8, pp. 21-28, Jan. 1962.

Tanner Graph. Cycles and Girth

$$H = \begin{array}{c} \begin{matrix} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \end{matrix} \\ \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \end{array} \begin{matrix} c_1 \\ c_2 \\ c_3 \end{matrix}$$

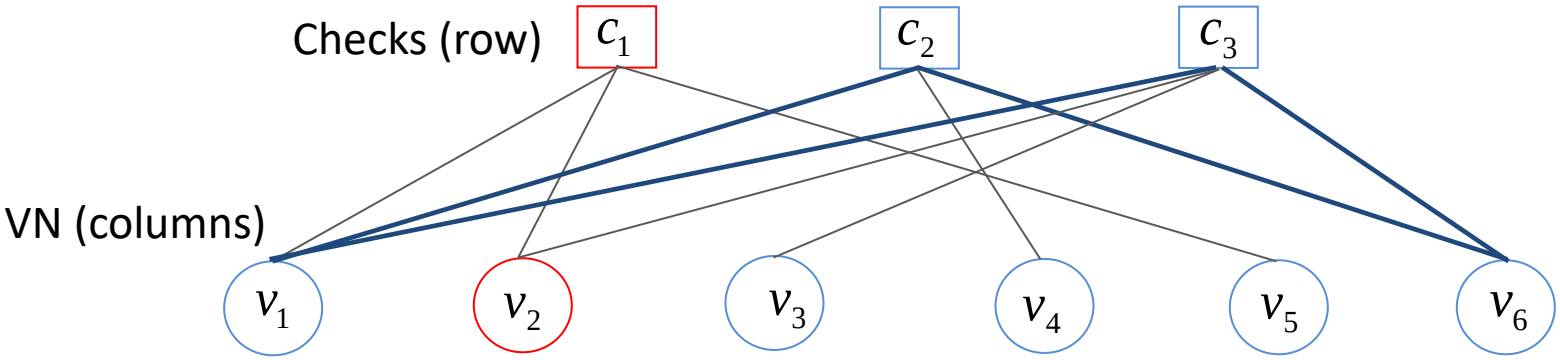
Tanner Graph– equivalent bipartite graph for the parity check matrix *H*



Tanner R.M., "A recursive approach to low complexity codes", IEEE Trans. Inform. Theory, IT-27, pp. 533-547, September 1981.

Tanner Graph. Cycles and Girth

Tanner Graph– equivalent bipartite graph for the parity check matrix *H*



Cycle - closed simple way in Tanner-graph.

Example of cycle 4: $c_2 \rightarrow v_1 \rightarrow c_3 \rightarrow v_6 \rightarrow c_2$

Girth – shortest cycles in Tanner-graph.

Tanner R.M., "A recursive approach to low complexity codes", IEEE Trans. Inform. Theory, IT-27, pp. 533-547, September 1981.

Quasi-Cyclic LDPC codes

$$H = \begin{array}{c} \begin{matrix} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \end{matrix} \\ \left[\begin{array}{cc|cc|cc} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ \hline 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{array}\right] \end{array} \begin{matrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{matrix}$$

Quasi-Cyclic LDPC(QC-LDPC codes)- LDPC-codes which parity-check matrix defined by structured block submatrix – Circulant Permutation matrix.

Applied QC-LDPC codes allow to simplify analysis of code and graph properties, increase throughput (parallelism inside circulant), decrease complexity of hardware implementation based on barrel shifter.

$$H_{QC} = \begin{bmatrix} I^0 & I^1 & I^1 \\ I^0 & I^{-1} & I^0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$Circulant\ \ Permutation\, Matrix\,(CPM)\ of\ \ size\ \ 2\times 2:\ \ I^0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, I^1 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, I^{-1} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

R. M. Tanner, D. Sridhara, T. Fuja, ”A class of group structured LDPC codes”, Proc. ICSTA 2001, Ambleside, England, 2001

Cycles at QC-LDPC Codes

QC-LDPC codes contain cycle of size $2i$ if (circulant permutation matrix) shifts satisfy equation

$$\sum_{k=0}^{i-1} \Delta_{j_k, j_{k+1}}(l_k) = 0 \bmod (Z)$$

For example for cycle 4, x *is a circulant shift in matrix*:

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

$$H_{QC} = \begin{bmatrix} I^0 & I^0 & I^1 \\ I^0 & I^0 & I^1 \end{bmatrix}, \textit{where } I \textit{ CPM of size } 2 \times 2$$

$$Auth(\mathrm{H}_{QC}) = size(I)$$

If you found cycles multiplied it to Authomorphism of Tanner graph, in our case Auth=2

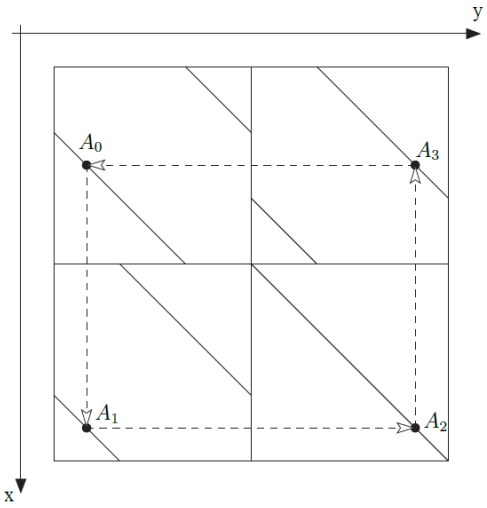
Fossorier M.P.C., “Quasi-cyclic low-density parity-check codes from circulant permutation matrices”, IEEE Trans. Inf. Theory, vol. 50, no. 8, pp. 1788–1793, 2004.

Cycles at QC-LDPC Codes

QC-LDPC codes contain cycle of size $2i$ if (circulant permutation matrix) shifts satisfy equation

$$\sum_{k=0}^{i-1} \Delta_{j_k, j_{k+1}}(l_k) = 0 \bmod (Z)$$

For example for cycle 4:



$$A_0 := \begin{pmatrix} x \\ (x + \delta_{A_0}) \bmod z \end{pmatrix} \quad A_3 := \begin{pmatrix} (y_{A_3} - \delta_{A_3}) \bmod z \\ y_{A_3} \end{pmatrix}$$

$$A_1 := \begin{pmatrix} (y_{A_0} - \delta_{A_1}) \bmod z \\ y_{A_0} \end{pmatrix} \quad A_2 := \begin{pmatrix} x_{A_1} \\ (x_{A_1} + \delta_{A_2}) \bmod z \end{pmatrix}$$

$$(\delta_{A_0} - \delta_{A_1} + \delta_{A_2} - \delta_{A_3}) \bmod z = 0$$

Fossorier M.P.C., "Quasi-cyclic low-density parity-check codes from circulant permutation matrices", IEEE Trans. Inf. Theory, vol. 50, no. 8, pp. 1788–1793, 2004.

Construction QC-LDPC Codes

We have protograph's base matrix and circulant size .

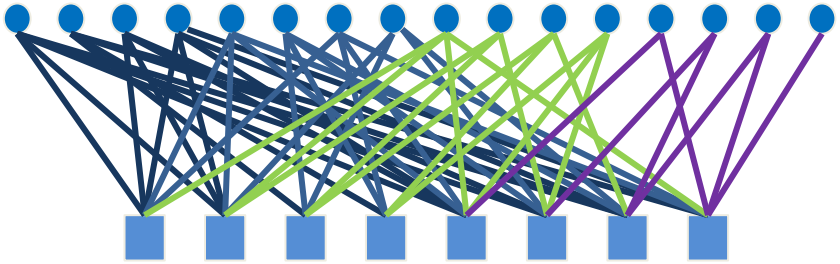
Construct QC-LDPC code:

"Simple" target. with maximal girth

"Industrial" target. special structures of cycles

Protograph

16 variable nodes



8 parity-check nodes

8x16 protograph base matrix

$$H_{proto} = \begin{pmatrix} 1&0&1&0&1&0&1&0&1&0&0&0&0&0&0&0 \\ 1&0&1&0&1&0&0&1&1&1&0&0&0&0&0&0 \\ 0&1&0&1&0&1&0&1&0&1&1&0&0&0&0&0 \\ 0&1&0&1&0&1&1&0&0&0&1&1&0&0&0&0 \\ 1&0&1&0&1&0&1&0&1&0&0&1&1&0&0&0 \\ 1&0&1&0&0&1&0&1&0&1&0&1&0&1&0&0 \\ 1&1&0&1&0&1&0&1&0&0&1&0&0&1&1&0 \\ 0&1&0&1&1&0&1&0&1&0&0&0&1&0&1&1 \end{pmatrix}$$

42 circulant size

9 circulant

$$H = \begin{pmatrix} 33&0&15&0&8&0&28&0&26&0&0&0&0&0&0&0 \\ 14&0&25&0&11&0&0&9&18&33&0&0&0&0&0&0 \\ 0&8&0&11&0&29&0&30&0&20&18&0&0&0&0&0 \\ 0&31&0&39&0&22&31&0&0&0&37&11&0&0&0&0 \\ 33&0&9&0&5&0&24&0&25&0&0&28&23&0&0&0 \\ 20&0&30&0&0&20&0&12&0&30&0&22&0&12&0&0 \\ 2&20&0&11&0&31&0&7&0&0&36&0&0&17&6&0 \\ 0&7&0&32&24&0&39&0&30&0&0&0&26&0&38&28 \end{pmatrix}$$

Code length N=16*42=672

$$H = \begin{pmatrix} 2&-1&6&-1&7&-1&7&-1&8&-1&-1&-1&-1&-1&-1&-1 \\ 7&-1&6&-1&1&-1&-1&3&6&3&-1&-1&-1&-1&-1&-1 \\ -1&2&-1&8&-1&2&-1&8&-1&1&8&-1&-1&-1&-1&-1 \\ -1&5&-1&6&-1&2&4&-1&-1&-1&0&1&-1&-1&-1&-1 \\ 4&-1&6&-1&3&-1&1&-1&5&-1&-1&5&4&-1&-1&-1 \\ 7&-1&4&-1&-1&2&-1&0&-1&6&-1&5&-1&4&-1&-1 \\ 8&1&-1&4&-1&4&-1&5&-1&-1&8&-1&-1&2&7&-1 \\ -1&1&-1&3&4&-1&5&-1&3&-1&-1&-1&4&-1&4&3 \end{pmatrix}$$

Code length N=16*9=144

Fossorier's "Guess-and-Test" method for construction QC-LDPC codes

1.Randomly choice of circulant shift in parity-check matrix

2. Check according equation:

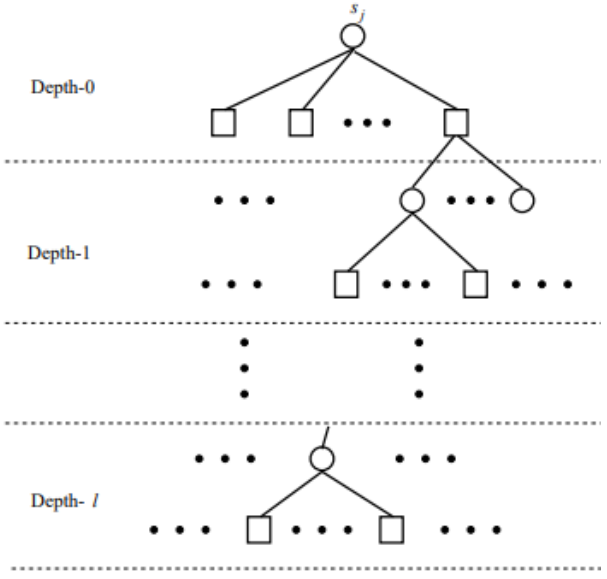
$$\sum_{k=0}^{i-1} \Delta_{j_k, j_{k+1}}(l_k) = 0 \bmod (Z)$$

3.If not satisfied girth(ACE/EMD) requirement, go to step 1.

Fossorier M.P.C., "Quasi-cyclic low-density parity-check codes from circulant permutation matrices", IEEE Trans. Inf. Theory, vol. 50, no. 8, pp. 1788–1793, 2004.

Greedy QC-LDPC code construction methods - PEG methods

Greedy choice(which maximize require cycle sizes) of shift values for every variable nodes. Dioud-Declercq-Fossorie improve greedy equation (to choice of shift) to avoid undetected cycles.



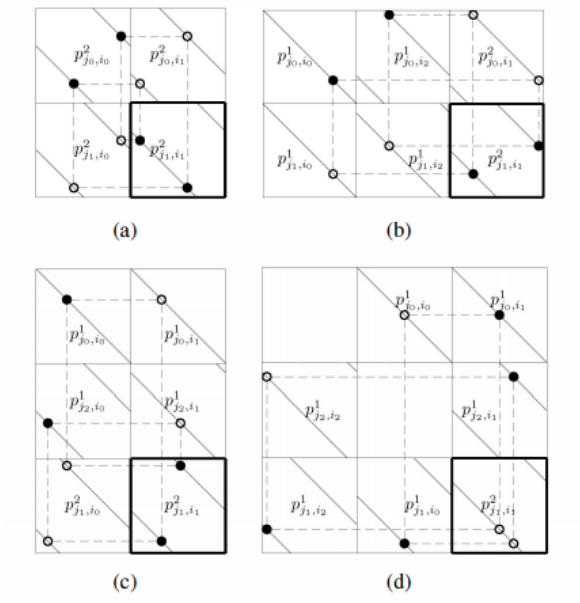


Fig. 2. Different configurations of PEG-undetectable 8-cycles

$$
\begin{array}{ccc}
I_0 & * & * \\
I_6 & * & * \\
I_c & * & *
\end{array}
\Rightarrow
\begin{array}{ccc}
I_0 & I_D & * \\
I_6 & * & * \\
I_3 & * & *
\end{array}
\Rightarrow
\begin{array}{ccc}
I_0 & I_0 & * \\
I_6 & I_E & * \\
I_3 & * & *
\end{array}
\Rightarrow
\begin{array}{ccc}
I_0 & I_0 & I_0 \\
I_6 & I_7 & I_3 \\
I_3 & I_1 & I_8
\end{array}
$$

X.-Y. Hu, E. Eleftheriou, and D.-M. Arnold, Regular and irregular progressive edge-growth tanner graphs, IEEE Trans. Inf. Theory, vol. 51, no. 1, pp. 386-398, Jan. 2005.

M. Diouf, D. Declercq, M. Fossorier, S. Ouya and B. Vasic, ”Improved PEG construction of large girth QC-LDPC codes,” 2016 9th International Symposium on Turbo Codes and Iterative Information Processing (ISTC), Brest, 2016, pp. 146-150.

Hill Climbing method for construction QC-LDPC codes

Our algorithm iteratively updates a sequence of edge labels. At each iteration, it changes the single edge label to the value that effects the greatest reduction in a cost function. The cost function we use depends on the number of cycles in the current code that have length less than the desired girth. Shorter cycles are weighted to be more costly than longer cycles. The algorithm terminates when either (a) the current values of all coefficients give zero cost (and thus the code has the desired girth), or (b) when we can no longer change any single coefficient to a value that further reduces the cost (and thus the number of undesired cycles). When the tree topology of the code implies the existence of inevitable cycles (b) will always be the stopping criterion. Updates are performed subject to the sibling constraint on edge labels. This preserves the tree topology of the code and thus, e.g., the protograph structure of the code is an invariant under the updates. We note that a change in a single edge label will, in general, have a trickle-down effect on a number of code coefficients (equal to the number of leaves in the tree that are a descendent of that edge).

Wang Y., Draper S.C. and Yedidia J.S., “Hierarchical and high-girth QC LDPC codes”, IEEE Trans. Inf. Theory, vol. 59, no. 7, pp. 4553-4583, July 2013.

Hill Climbing method for construction QC-LDPC codes

Get initial QC-LDPC codes

$$\begin{matrix} I_0 & * & * & I_0 & 0 & I_1 \\ I_6 & * & * & I_0 & I_0 & 0 \\ I_c & * & * & 0 & I_0 & I_0 \end{matrix} \quad (c-6+1-0) \bmod 8 \neq 0,\ C \neq \{7,6\}$$

$$\begin{matrix} I_0 & I_D & * & I_0 & 0 & I_1 \\ I_6 & * & * & I_0 & I_0 & 0 \\ I_3 & * & * & 0 & I_0 & I_0 \end{matrix} \quad D \neq \{0\}$$

$$\begin{matrix} I_0 & I_0 & * & I_0 & 0 & I_1 \\ I_6 & I_E & * & I_0 & I_0 & 0 \\ I_3 & * & * & 0 & I_0 & I_0 \end{matrix} \quad E \neq \{0,6\}$$

$$\begin{matrix} I_0 & I_0 & I_0 & I_0 & 0 & I_1 \\ I_6 & I_7 & I_3 & I_0 & I_0 & 0 \\ I_3 & I_1 & I_8 & 0 & I_0 & I_0 \end{matrix} \quad J \neq \{3,0,1,5,7\}$$

$$\begin{matrix} I_0 & I_0 & I_0 & I_0 & 0 & I_1 \\ I_6 & I_7 & I_3 & I_0 & I_0 & 0 \\ I_3 & I_1 & I_8 & 0 & I_0 & I_0 \end{matrix} \quad J \neq \{3,0,1,5,7\}$$

1. Choice **any** shift value (not row by row or column by column) with maximal number of cycles;

2. Choice another shift value from allowed shifts list to decrease cycles number.

Wang Y., Draper S.C. and Yedidia J.S., "Hierarchical and high-girth QC LDPC codes", IEEE Trans. Inf. Theory, vol. 59, no. 7, pp. 4553-4583, July 2013.

Hill Climbing method for construction QC-LDPC codes

Get initial QC-LDPC codes

Disadvantage:
Iterative decreasing number of cycles usually provide suboptimal solution like Greedy Search.
Solution:
1.Give some freedom of choice at the begin of method and decrease with Iteration. Apply Simulated Annealing method.
2. Make choice of circulant shifts more soft by involving pdf function of cycles.

Wang Y., Draper S.C. and Yedidia J.S., “Hierarchical and high-girth QC LDPC codes”, IEEE Trans. Inf. Theory, vol. 59, no. 7, pp. 4553-4583, July 2013.

Proposed Simulated Annealing method for construction QC-LDPC codes

Algorithm 1 Simulated Annealing method for construction of QC-LDPC codes

Require: $M(\mathbf{H})$−mother matrix, L− circulant size, g-girth of lifted matrix, EMD-minimal EMD value, $Iter$− maximal number of iterations, $seed$- a seed to be used in a pseudo-random number generator, $Temp$− initial value of temperature

1: $Nstep = 0$
2: $i, j = rnd(seed)$
3: **for** $it = 0; it <= Iter; it = it + 1$ **do**
4: **while** $M_{ij}(\mathbf{H}) = 0$ **do**
5: $i, j = rnd(seed)$
6: **end while**
7: **for** $k = 0; k <= L - 1; k = k + 1$ **do**
8: $\Theta_k = enumcirccycles(i, j, k, g, EMD)$

More Soft

$$P(k) = w(k) / \sum_{m=0}^{L-1} w(m),$$

$$w(k) = e^{\frac{-\Theta_k}{Temp}},$$

where Θ_k-number of cycles through $E_{ij}(\mathbf{H})$-CPM with shift value k, $P(k)$− probability of k-shift CPM value choice, $w(k)$−probability weight function;

9: **end for**
10:

$$\Phi = enumcycles(E(\mathbf{H}), g, EMD),$$

where Φ-total number of cycles in exponent matrix $E(\mathbf{H})$;

11:

$$E_{ij}(\mathbf{H}) = rndshift(P, Temp)$$

12: Nstep=Nstep+1
13:

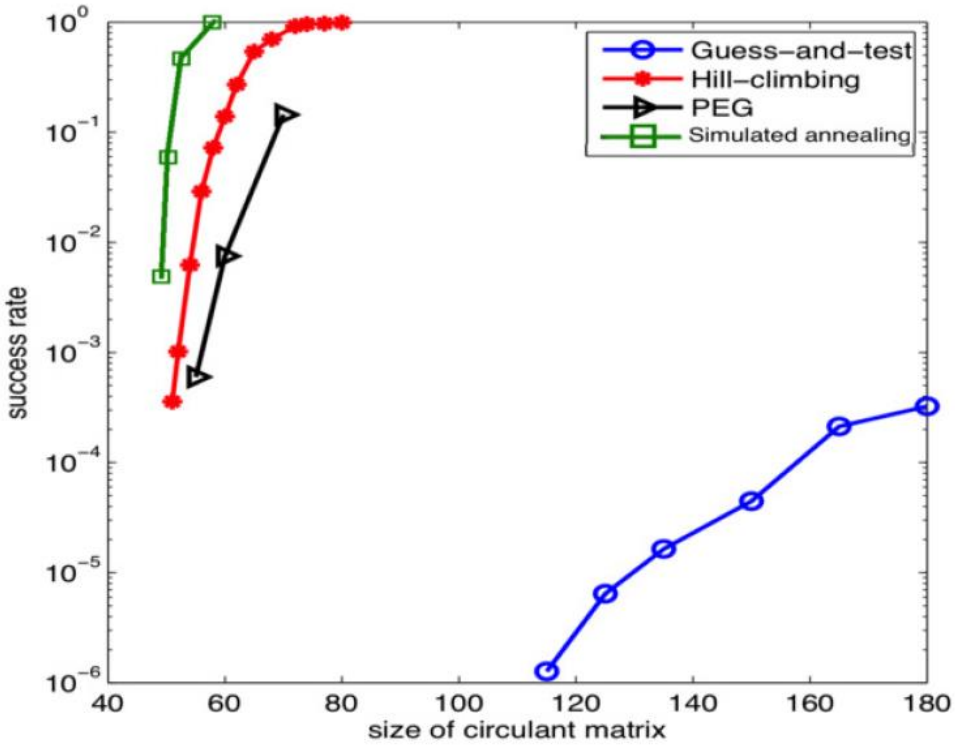
Less Greedy

$$Temp = \eta \frac{\Phi}{Nstep^2},$$

where η-some constant value;

14: **end for**
return $E(\mathbf{H})$

$$H^{(3\times 12)} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$



Comparison of the success rate of guess-and-test, hill climbing, simulated annealing and PEG methods in finding a girth-8 regular QCLDPC code when mother matrix have 3 rows and 12 columns.

Minimal value of circulant *L* for regular base matrix with row number *m*=3 and column number *n* with girth 10

Column number, *L*	Our approach	Hill Climbing*	Improved QC-PEG**
4	37	39	37
5	61	63	61
6	91	103	91
7	155	160	155
8	215	233	227
9	304	329	323
10	412	439	429
11	545	577	571
12	709	758	-

Minimal value of circulant *L* for regular base matrix with row number *m*=3 and column number *n* with girth 12

Column number, *L*	Our approach	Improved QC-PEG**	TableV***
4	73	73	97
5	160	163	239
6	320	369	479
7	614	679	881
8	1060	1291	1493
9	1745	1963	2087
10	2734	-	-
11	4083	-	-
12	5964	-	-

*Y. Wang, S. C. Draper and J. S. Yedidia, "Hierarchical and High-Girth QC LDPC Codes," in *IEEE Transactions on Information Theory*, vol. 59, no. 7, pp. 4553-4583, July 2013.
**M. Diouf, D. Declercq, M. Fossorier, S. Ouya and B. Vasić, "Improved PEG construction of large girth QC-LDPC codes," 2016 9th International Symposium on Turbo Codes and Iterative Information Processing (ISTC), Brest, 2016, pp. 146-150.
***M.E. O'Sullivan. "Algebraic construction of sparse matrices with large girth". IEEE Trans. Inf. Theory, vo1.S2, no.2, pp.718-727, Feb. 2006.

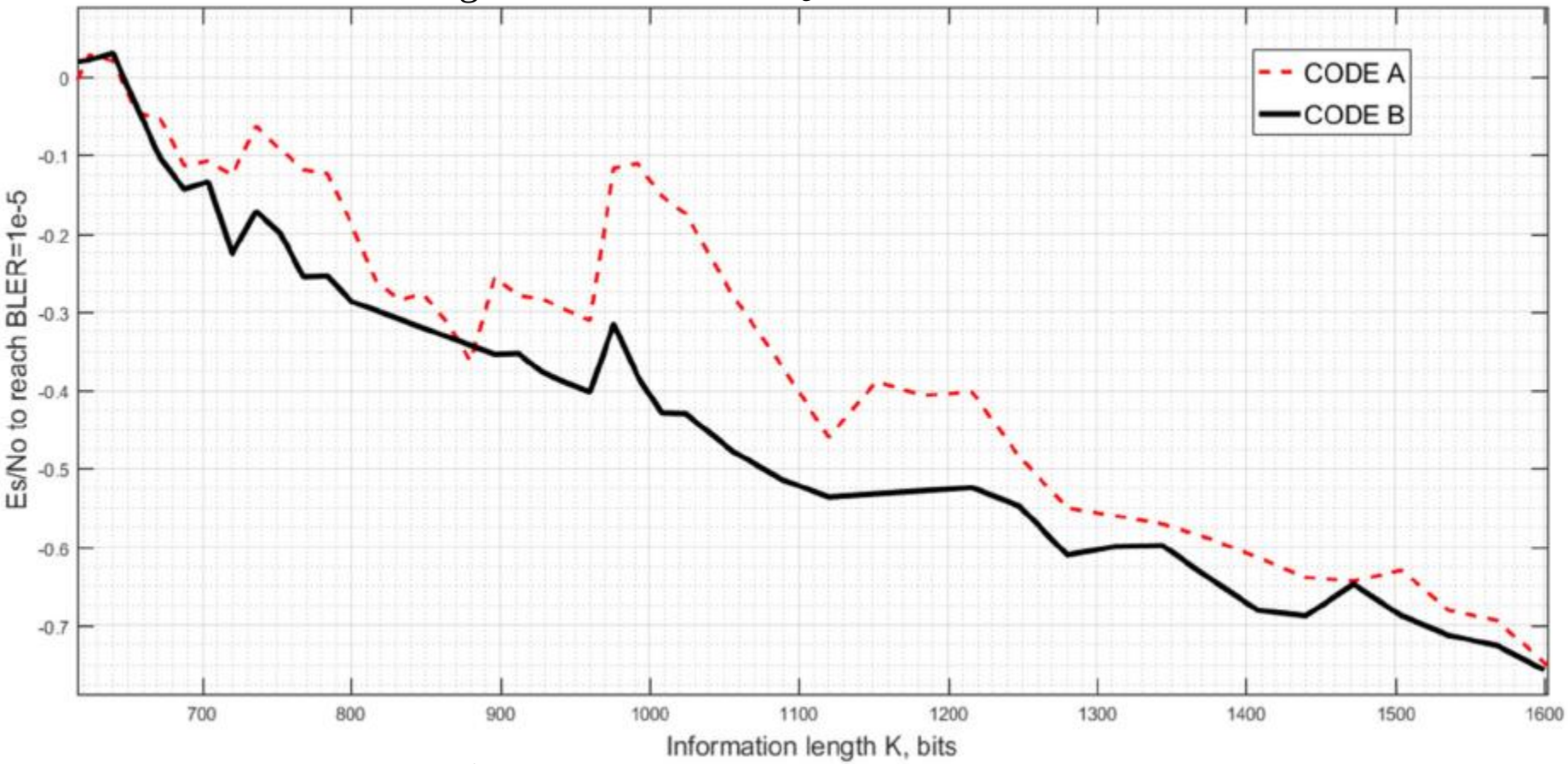
Apply Simulated Annealing method under 1/3 rate 5G Base graph 2 (get 8 set)

1	1	1	1	0	0	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	1	1	1	1	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	1	1	0	0	0	1	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	0	1	1	1	1	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	1	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	1	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	1	0	1	0	0	0	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	1	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

R1-1711982 Nokia WF on LDPC parity check matrices 3GPP RAN1- NR2 Qingdao, China, 27th 30th June 2017

Compare proposed method with QC-LDPC codes constructed by industrial leader

We apply our method to construction QC-LDPC codes from 5G standard, and for base graph 2 Rate 1/3 we construct QC-LDPC codes with better performance at error-floor level than joint solution of Nokia, Samsung, ZTE, MediaTek, Qualcomm, LG, Ericsson, Intel, CATT .



SNR required to achieve 10^{-5} Frame Error Rate for information lengths , Code A-5G;Code B-SA

Summary

1. We propose Simulated Annealing method for QC-LDPC codes Construction.
2. Our method exceed previous described methods according girth(EMD) properties: Bit-Filling, Guess-And-Test, Improved Progressive Edge Grown, Hill-Climbing.
3. Application of our methods exceed in error-floor region joint solution of Nokia, Samsung, Huawei, ZTE, MediaTek, LG, Qualcomm, Ericsson, Intel, CATT for 5G BG2 QC-LDPC Codes.
4. To support reproducible and further research of this method we provide source code.

Source codes:
https://github.com/Lcrypto/Simulated-annealing-lifting-QC-LDPC

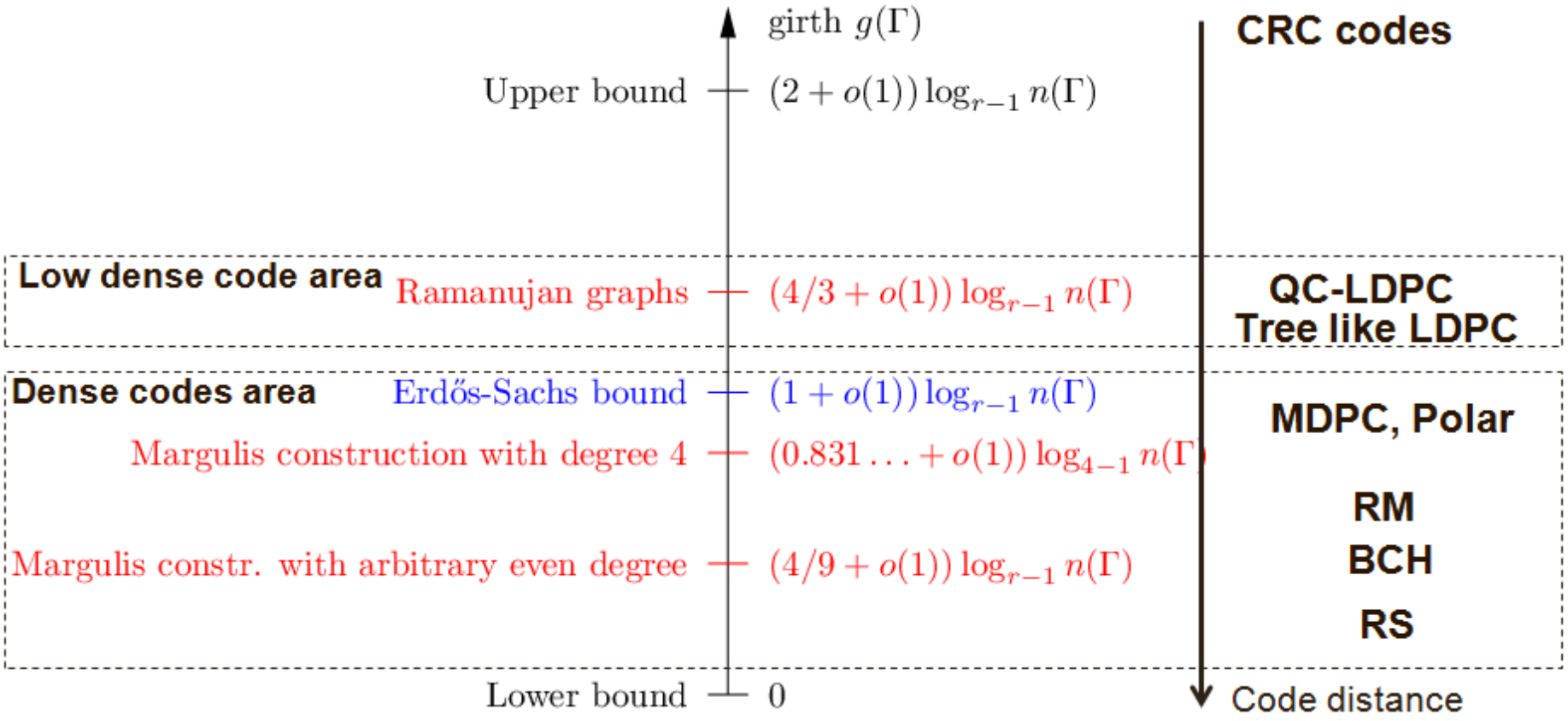
Thank You!

Source codes:
https://github.com/Lcrypto/Simulated-annealing-lifting-QC-LDPC

5G Base graph 2 Example of code constructed using our approach for family a=2

0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	24	-1	-1	-1	-1	-1	104	-1	-1	-1	-1	-1	57	-1
-1	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	92	-1	-1	-1	-1	-1	-1	-1	17	-1	-1	54
-1	-1	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	33	-1	-1	-1	-1	-1	-1	-1	-1	77	29
-1	-1	-1	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	45	-1	4	-1	-1	127	-1
-1	-1	-1	-1	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	120	6	-1	44	-1	-1	-1	-1	-1	-1	-1	-1	74
-1	-1	-1	-1	-1	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	122	87	-1	-1	18	-1	-1	-1	-1	-1	-1	-1	66
-1	-1	-1	-1	-1	-1	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	14	124	-1	-1	-1	-1	-1	-1	-1	-1	6	-1
-1	-1	-1	-1	-1	-1	-1	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	40	-1	15	-1	-1	-1	-1	84	-1	-1	-1	-1	-1	92
-1	-1	-1	-1	-1	-1	-1	-1	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	75	-1	-1	-1	-1	-1	114	-1	-1	-1	-1	-1	37	66
-1	-1	-1	-1	-1	-1	-1	-1	-1	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	27	-1	-1	-1	-1	-1	114	-1	-1	-1	-1	92
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	-1	-1	-1	-1	-1	-1	-1	109	-1	-1	-1	-1	16	-1	-1	-1	52	-1	-1	104	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	77	-1	113	-1	-1	80	90
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	-1	-1	-1	-1	-1	-1	-1	73	34	-1	-1	24	-1	-1	-1	-1	-1	-1	103
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	-1	-1	-1	-1	-1	11	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	4	48
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	-1	-1	-1	41	-1	79	-1	89	-1	-1	-1	-1	95	-1	-1	-1	70
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	-1	-1	-1	-1	125	-1	105	22	-1	-1	-1	68	-1	-1	68	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	-1	-1	-1	126	-1	103	-1	-1	-1	-1	79	-1	-1	124	23
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	-1	-1	108	-1	-1	-1	-1	-1	-1	-1	-1	-1	117	44
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	-1	-1	0	67	57	105	11	-1	4	122	12	-1	34
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	-1	83	-1	-1	114	-1	35	-1	84	-1	96	46
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	-1	21	125	88	104	16	66	123	-1	84	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	-1	72	-1	76	37	-1	-1	118	51	22

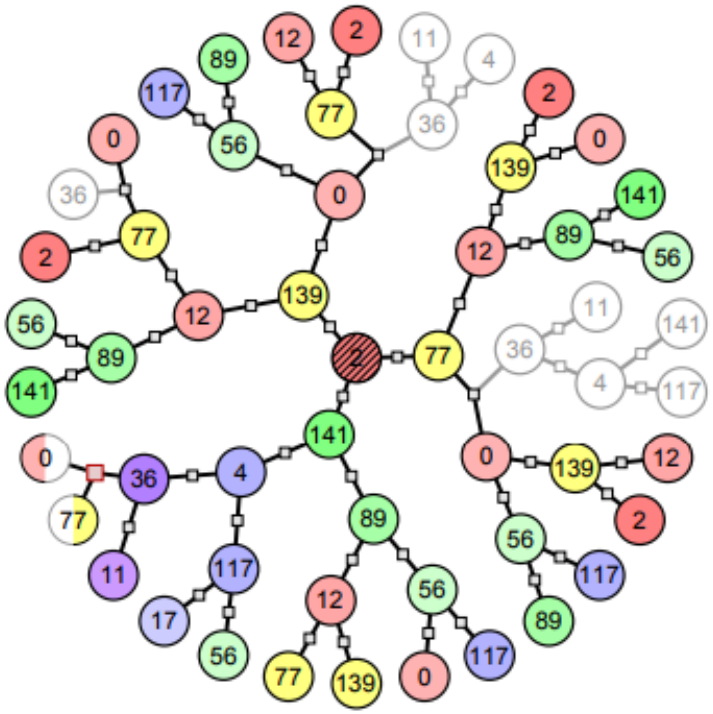
Asymptotical properties of graph girth(shortest cycles) and hamming distance grown from graph cardinally



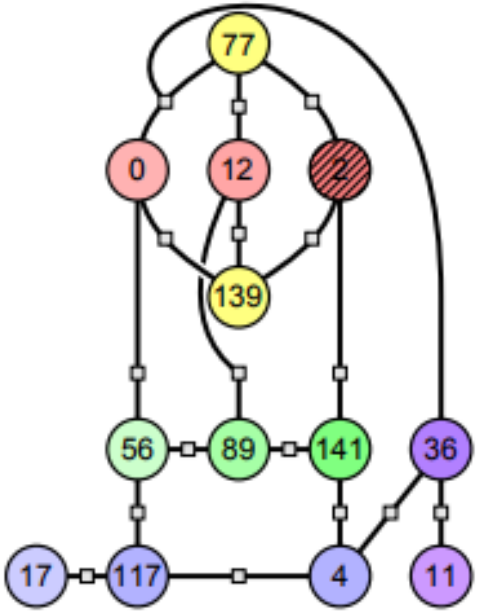
To have enough error correcting capability (code distance) necessary to have a lot of short cycles in Tanner graph

After m decoding iterations Belief Propagation algorithm under the tanner graph with girth g produce wrong decoding result due cycles:

$$m < \frac{g}{4} \leq m+1.$$



Tanner QC-LDPC code [155,64,20] computation(Wiberg) tree after 4 iterations(4 generations) of BP decoder



Example of Subgraph for which "$BP \neq MAP$"

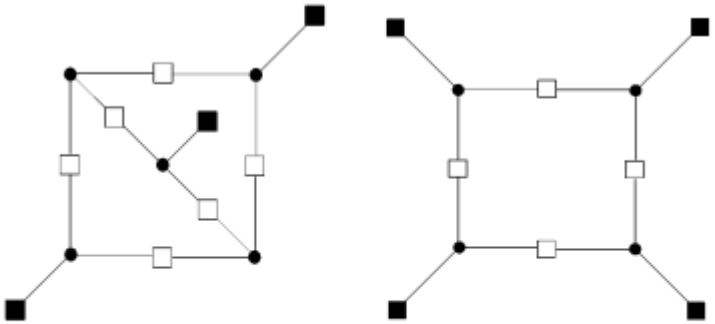
Circle is variable nodes, black square – check nodes, gray variable nodes show “weak nodes” which produce uncorrected error under BP which can corrected according code distance

Trapping sets - subgraphs formed by cycles or it's union:

After m decoding iterations Belief Propagation algorithm under the tanner graph with girth g produce wrong decoding result due Trapping sets:

$$m < \frac{g}{4} \leq m+1.$$

Trapping set (a,b) is a sub-graph with a variable nodes and b odd degree checks.



(a) (b)

- represents a variable node
- represents an even degree check node
- represents an odd degree check node

Trapping sets: (a) $(5,3)$ and (b) $(4,4)$

For example, TS(5,3) produced by three 8-cycles;
TS(a,0) is most harmfulness pseudocodeword of weight a formed by cycles 2*a.

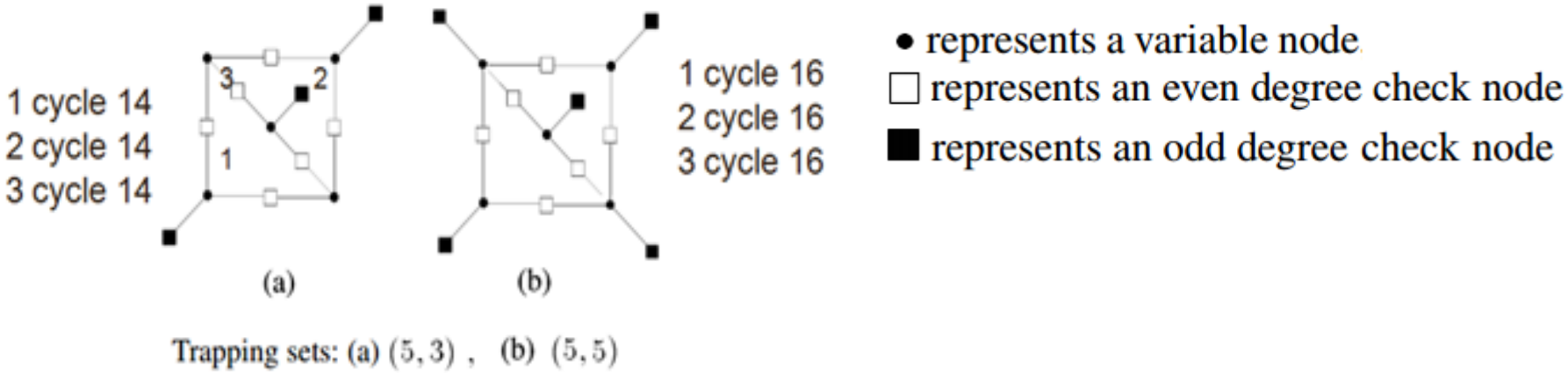
Simple way to broke harmful TS improve EMD/ACE:

Approximate cycle extrinsic message degree metric for Codes graph:

$$ACE(C) = \sum_{v \in C} \left(d(v) - 2 \right)$$

$d(v)$ – degree of node in cycles

Trapping set (a,b) **is a sub-graph with** a **variable nodes and** b **odd degree checks.**



Trapping sets: (a) $(5,3)$, (b) $(5,5)$

For example, TS(5,3) produced by 3 cycle of ACE 14 and TS(5,5) by 3 cycles of ACE 16.
Improve of this metrics improve TS structures.
Decoder properties like quantization, normalize and offset values change harmfulness of TS.
This is why solution of TS elimination under LDPC code construction and choice properties of decoder need doing simultaneously.

But what TS sets broke mean from performance point of view, especially under AWGN with L=7 (iterations)?

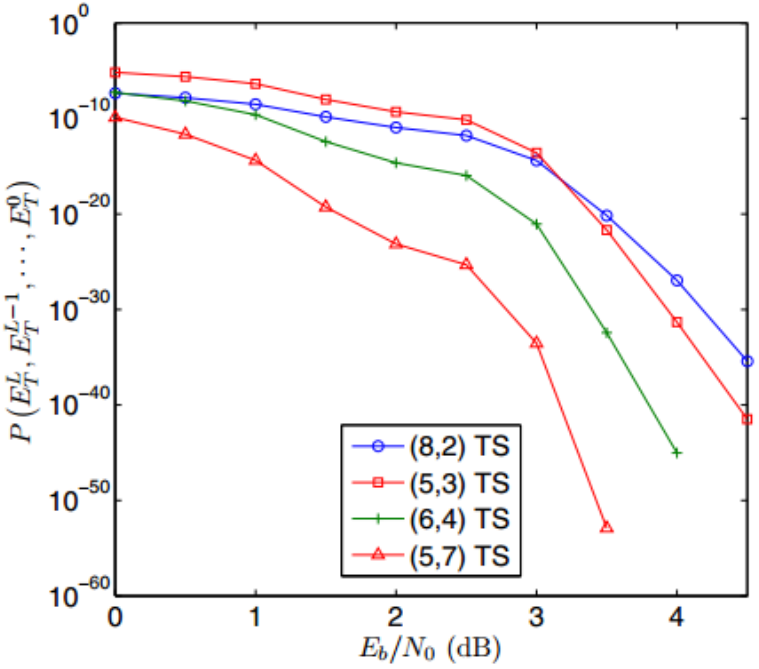


Fig. 5. Comparison of the (5,3), (5,7), (6,4) and (8,2) TSs of the Tanner (155,64,20) code

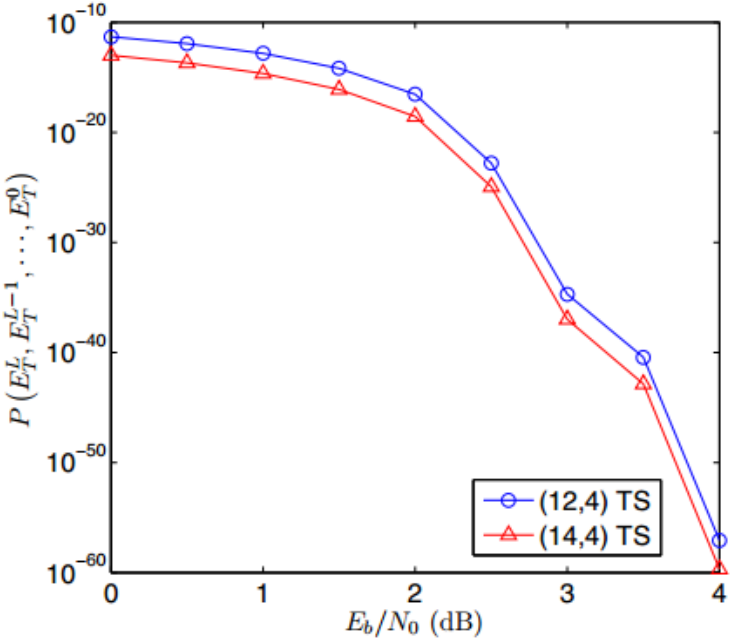


Fig. 3. Comparison of the (12,4) and the (14,4) TSs of the Margulis (2640,1320) code

Joint probability of all VNs in TS in error as a measure of harmfulness of TS

Deka K., Rajesh A., Bora P.K. Comparison of the Detrimental Effects of Trapping Sets in LDPC Codes

For parity-check matrix:

$$H=\begin{pmatrix}0 & 1 & 0 & 1\\ 0 & 0 & 1 & 1\\ 1 & 1 & 1 & 1\end{pmatrix} \qquad \begin{aligned} x_2 \oplus x_4 &= 0\\ x_3 \oplus x_4 &= 0\\ x_1 \oplus x_2 \oplus x_3 \oplus x_4 &= 0\end{aligned}$$

We construct constrain systems using inequality which is equivalent

$$\sum_{i\in V} x_i' - \sum_{i\in h_i/V} x_i' \le |V| - 1, \; where \quad |V| \quad odd \quad 0 \le x_i' \le 1$$

Using it on the first row of parity-check matrix:

$$\begin{pmatrix}+1 & -1\\ -1 & +1\end{pmatrix}\begin{pmatrix}x_2'\\ x_4'\end{pmatrix} \le \begin{matrix}1-1 & V=\{x_2'\}\\ 1-1 & V=\{x_4'\}\end{matrix}, \quad \begin{pmatrix}0 & +1 & 0 & -1\\ 0 & -1 & 0 & +1\end{pmatrix}\begin{pmatrix}x_1'\\ x_2'\\ x_3'\\ x_4'\end{pmatrix} \le \begin{matrix}1-1 & V=\{x_2'\}\\ 1-1 & V=\{x_4'\}\end{matrix}$$

For second row:

$$\begin{pmatrix}+1 & -1\\ -1 & +1\end{pmatrix}\begin{pmatrix}x_3'\\ x_4'\end{pmatrix} \le \begin{matrix}1-1 & V=\{x_3'\}\\ 1-1 & V=\{x_4'\}\end{matrix} \quad \begin{pmatrix}0 & 0 & +1 & -1\\ 0 & 0 & -1 & +1\end{pmatrix}\begin{pmatrix}x_1'\\ x_2'\\ x_3'\\ x_4'\end{pmatrix} \le \begin{matrix}1-1 & V=\{x_3'\}\\ 1-1 & V=\{x_4'\}\end{matrix}$$

For third row:

$$\begin{pmatrix}+1 & -1 & -1 & -1\\ -1 & +1 & -1 & -1\\ -1 & -1 & +1 & -1\\ -1 & -1 & -1 & +1\\ +1 & +1 & +1 & -1\\ +1 & +1 & -1 & +1\\ +1 & -1 & +1 & +1\\ -1 & +1 & +1 & +1\end{pmatrix}\begin{pmatrix}x_1'\\ x_2'\\ x_3'\\ x_4'\end{pmatrix} \le \begin{matrix}1-1 & V=\{x_1'\}\\ 1-1 & V=\{x_2'\}\\ 1-1 & V=\{x_3'\}\\ 1-1 & V=\{x_4'\}\\ 3-1 & V=\{x_1', x_2', x_3'\}\\ 3-1 & V=\{x_1', x_2', x_4'\}\\ 3-1 & V=\{x_1', x_3', x_4'\}\\ 3-1 & V=\{x_2', x_3', x_4'\},\end{matrix}$$

We can weigh of cycle and it union $\omega(p)$ under some defined statistical model.

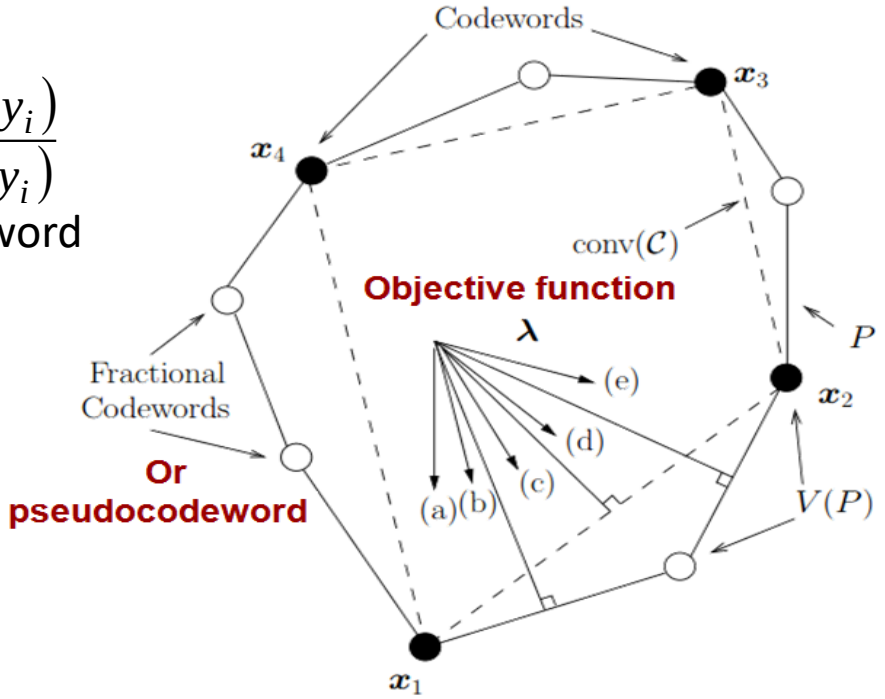
For example, binary-input additive white Gaussian noise:

$$\lambda = (\lambda_0, \lambda_1, \ldots, \lambda_{n-1}), \qquad where\ \lambda_i = \ln \frac{\Pr(c_i = 0 \mid y_i)}{\Pr(c_i = 1 \mid y_i)}$$

The goal of LP is to find maximum likelihood codeword

$$\min \lambda^T c$$

$$c \in \left\{ \bar{c} \in F_2^n \mid H\bar{c}^T = \bar{0}^T \bmod 2 \right\}$$



$$\omega(p) = \frac{e^T p}{\|p\|_2}$$

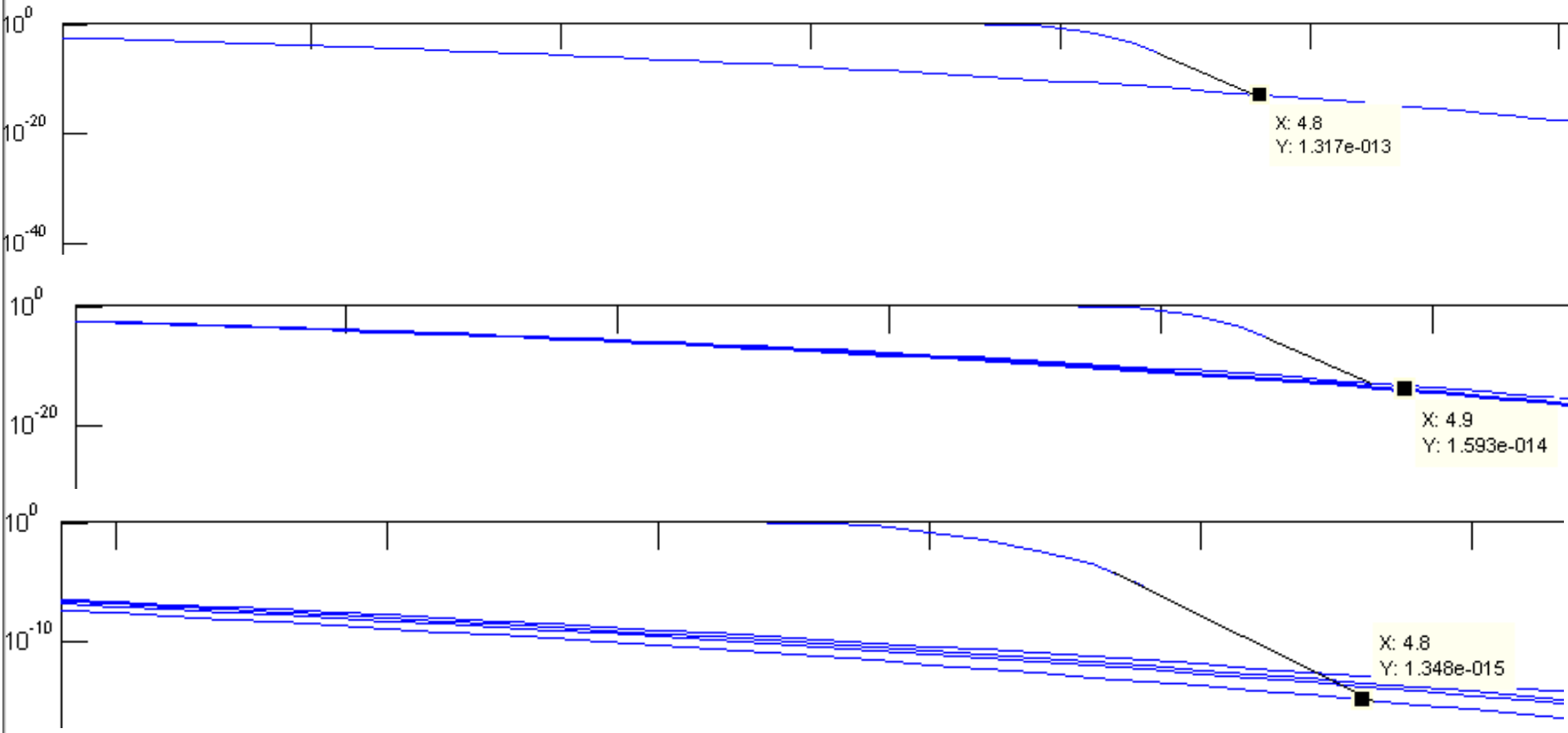
$$\mathrm{P}(\bar{c}_0 \to cycles) = Q\left(\sqrt{\sigma^2 \cdot \omega(p)}\right)$$

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{\frac{-t^2}{2}}\, dt$$

e – all-one vector in Euclidean space of LP, p – pseucodeword (weight of cycles and it union).

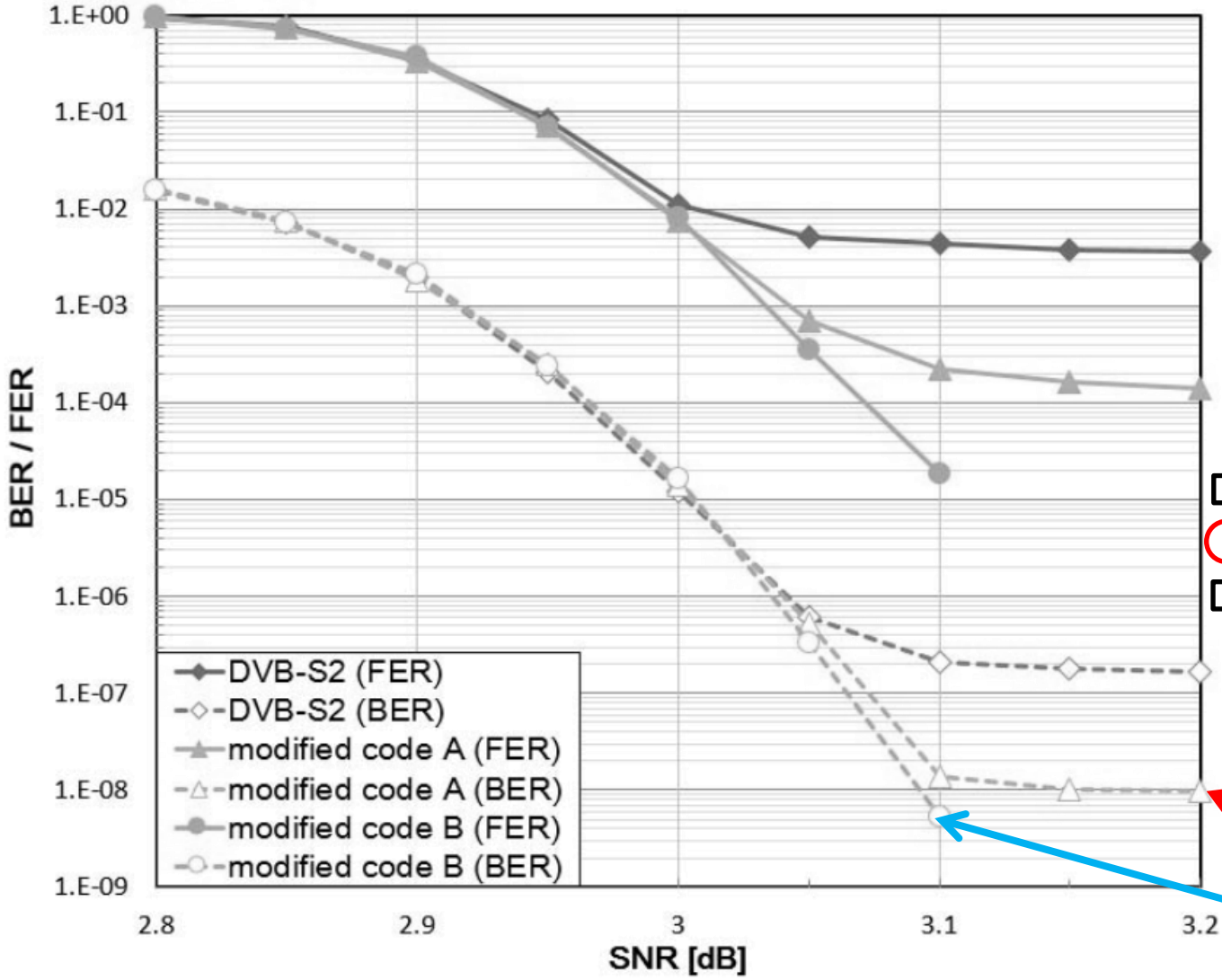
TS(a,b)	pseudo weight of TS, $\omega(p)$	Volume nodes
(7, 6)	[11.953960]	164 1846 2741 2751 3410 4429 8627
(6, 6)	[12.928841]	1001 2213 6379 6646 8176 9495
(6, 6)	[13.009602]	251 1640 5006 5745 8085 8347
(7, 6)	[13.077987]	1289 1616 1856 2501 2761 4439 8637
(7, 6)	[13.120068]	3362 3751 5558 6785 7224 7495 8370
(8, 8)	[13.164738]	1 1545 3839 5179 6719 6873 7144 7970
(6, 6)	[13.207715]	1846 2741 2751 3410 4429 8627
(9, 6)	[13.267264]	419 769 1275 1352 1990 2896 4222 5501 9739
(6, 6)	[13.295882]	209 1501 2968 3233 7738 9995
(6, 6)	[13.335162]	1 831 1933 2838 3001 8714
(6, 6)	[13.585278]	501 2414 3774 4263 6038 9212
(7, 6)	[13.737830]	210 2251 4348 6890 7552 7775 8597
(6, 6)	[13.871107]	3294 3901 4042 4303 5001 7477
(6, 6)	[13.891160]	3501 5086 5610 5999 6953 7297
(7, 6)	[14.179450]	1001 1294 1773 2038 2899 7499 9897
(8, 8)	[14.199011]	1351 1678 1918 2563 2823 3482 4251 8699
(7, 6)	[14.259830]	454 624 2763 3163 3751 7642 8218
(6, 6)	[14.295253]	3251 4260 4671 5208 5344 5402
(6, 6)	[14.363166]	1412 2094 3001 5719 7706 8978
…	…	…
(12, 12)	[32.320236]	155 2231 2781 5001 5953 7174 7554 8921 9004 9374 9414 9909

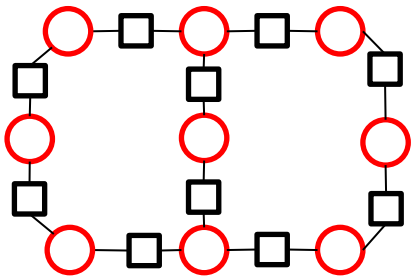
TS analysis (importance Sampling for predict Error-floor):



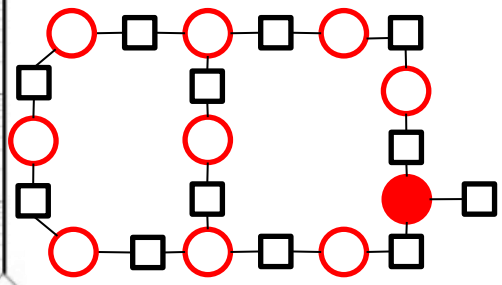
FER error-floor of 4x40x250.txt under 6 iteration.

Error-floor performance Fig. Second line without TS(7, 6) [11.953960]. Third line without first 5 TS: (7, 6) [11.953960], (6, 6) [12.928841], (6, 6) [13.009602], (7, 6) [13.077987], (7, 6) [13.120068]. The fifth line without 15 first TS.





TS(9,0)



TS(10,1)

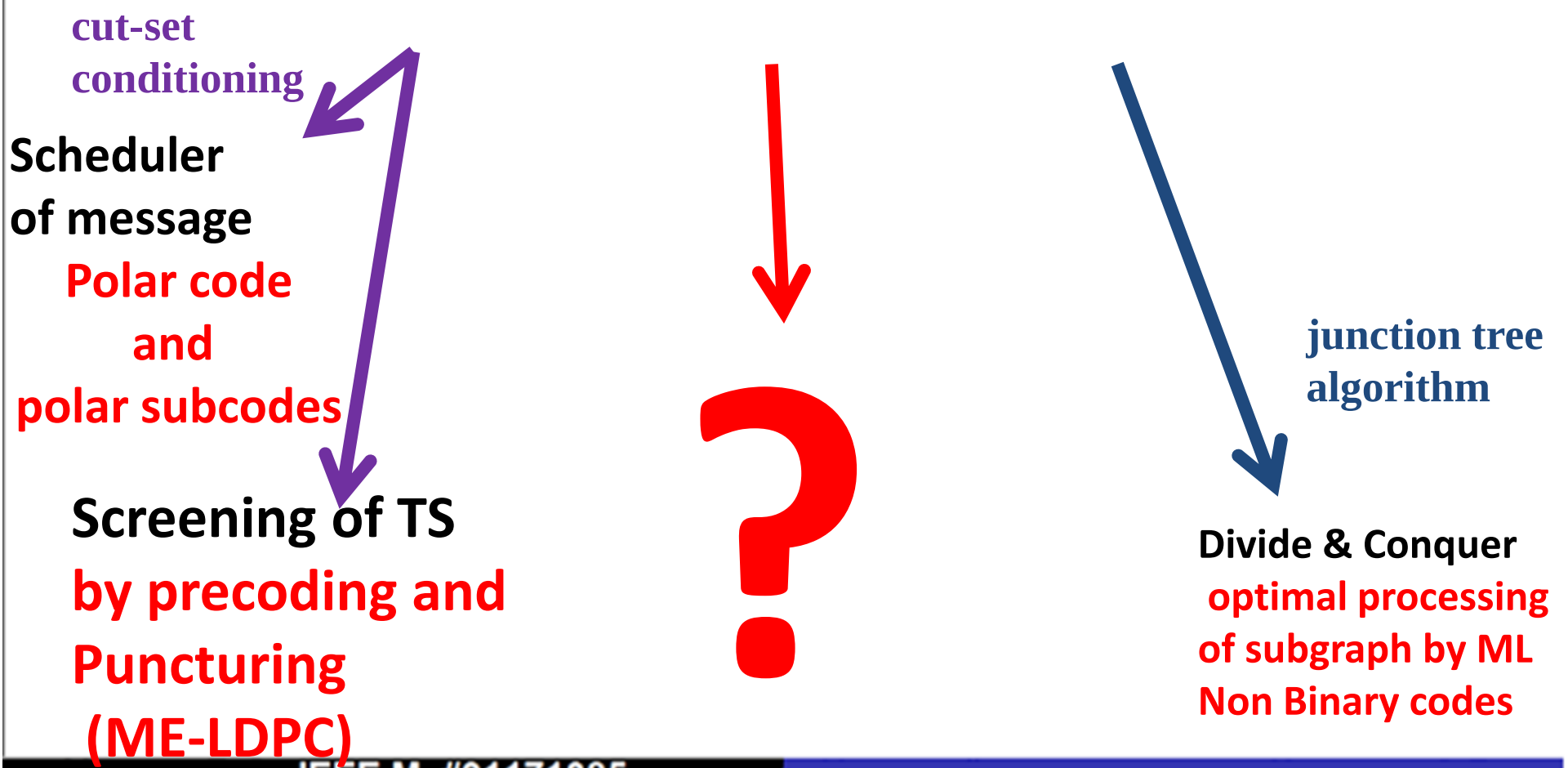
Without TS(9,0).

Without TS(9,0) And TS(10,1).

BER and FER performance of the original and modified DVB-S2 QC-LDPC codes of information length K=43200 and rate 2/3.

Sublinear size TS from number of nodes ('small cycles') make error-floor problem

Can there exist a new method to reach theoretical threshold-> increase «dense» of graph (make more correlations between nodes, improve weight spectrum properties) and solve trouble of TS with linear/'very' subexponentialy complexity?

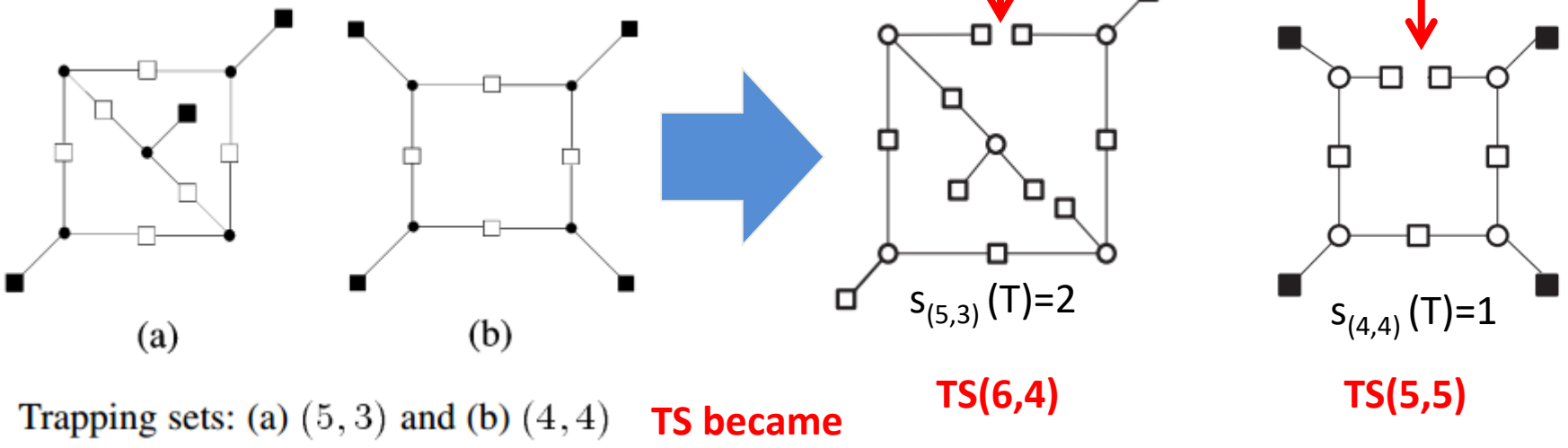


Consider of junction tree approach to trapping sets elimination

Definition. Let T(a, b) be an elementary trapping set. Let $C_k = \{c_1, c_2, \ldots, c_k\}$ be a set of check nodes of degree 2 in T. A set $S \subseteq C_k$. Is called critical

if by converting the single parity checks in S to the super checks (super factor), the trapping set is not harmful anymore.

Definition. Let T(a, b) be an elementary trapping set. The minimum size of a critical set in T is denoted by $s_{(a,b)}(T)$.

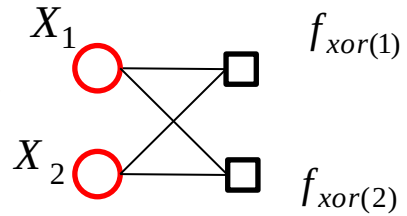


Trapping sets: (a) $(5, 3)$ and (b) $(4, 4)$

Problem related to Trapping sets bypass using BP message scheduler:

We can bypass Trapping sets using sequentially BP decoder scheduler under general graph:

$$H = \begin{array}{c} \begin{matrix} X_1 & X_2 \end{matrix} \\ \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \end{array} \begin{matrix} f_{xor(1)} \\ f_{xor(2)} \end{matrix}$$

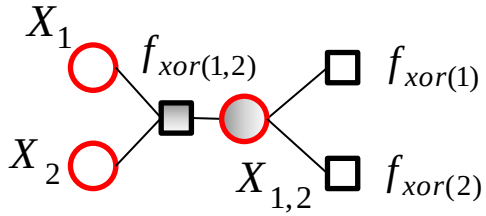


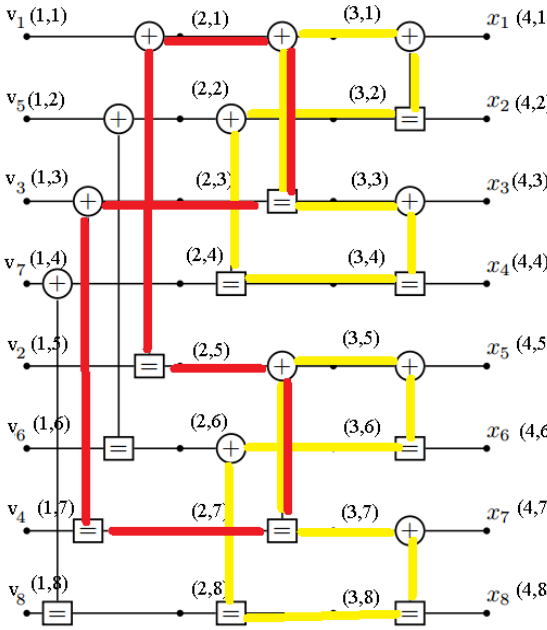
and construct graph in such way that some *variable more reliable (have some a prior information about graph structures).*

Base on this knowledge we can sequentially decode graph by BP:

Make new unobservable variable node $X_{1,2} = X_1 + X_2$ **and corresponding check node** $f_{xor(1,2)} = X_1 + X_2 + X_{1,2} = 0.$

$$H = \begin{array}{c} \begin{matrix} X_1 & X_2 & X_{1,2} \end{matrix} \\ \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix} \end{array} \begin{matrix} f_{xor(1)} \\ f_{xor(2)} \\ f_{xor(1,2)} \end{matrix}$$





Girth in Polar code graph for N=8

Cycle 4 bypass by cost of sequential decoding step.

In Polar code using this method sequential BP decoder (Successive cancellation) bypass short cycles. As result, harmfulness TS weight equal to code distance.