LUND UNIVERSITY

Low-Density Parity-Check Codes and Spatial Coupling for Quantitative Group Testing

Michael Lentmaier[†]

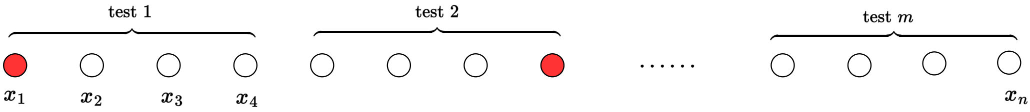
Joint work with Mgeni Makambi Mashauri[†] and Alexandre Graell i Amat[‡]

†Department of Electrical and Information Technology, Lund University, Sweden
‡Department of Electrical Engineering, Chalmers University of Technology, Sweden

Workshop on Application Driven Coding Theory
Simons Institute for the Theory of Computing
UC Berkeley, March 6, 2024

Background: Group Testing

- We have a large population of items
- Very few of them are "defective" (probability of being defective, γ is very small)



- **Goal:** Identify $\boldsymbol{x}$: defective $(x_i = 1)$, non-defective $(x_i = 0)$
- To reduce the number of tests: **test the items in groups** (pooling) [Dorfman1943]
- Rate, $\Omega = \frac{m}{n}$ (smaller is better)
- Adaptive vs non-adaptive test design
- We consider the asymptotic regime: $n \to \infty$

[Dorfman1943] Robert Dorfman, “The Detection of Defective Members of Large Populations,” *The Annals of Mathematical Statistics,*, vol. 14, no. 4, pp. 436–440, 1943.

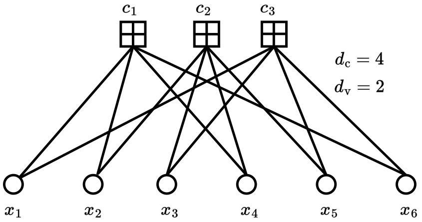
Background: Graphical Representation

- For non-adaptive group testing the pooling can be represented by a test matrix $\boldsymbol{A}$

$$\boldsymbol{A} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

$$\quad x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5 \quad x_6$$

- The matrix can be represented by a bipartite graph G



- We consider the scenario where the graph is sparse

Non-quantitative vs Quantitative

- **Non-quantitative:** test result, $s_i = 1$ if at least one item is defective otherwise $s_i = 0$ (logical OR)

●	●	○	○	○	●		Non-Quantitative	Quantitative
0	0	1	1	1	0	s_1	0	0
1	1	0	1	0	0	s_2	1	2
0	1	1	0	1	0	s_3	1	1
1	1	0	0	0	1	s_4	1	3

- For **quantitative** group testing, a test result shows the number of defective items

$$s_i = \sum_{j=1}^{n} x_j a_{ij} \quad \rightarrow \quad \boldsymbol{s} = A x$$

Group Testing with Sparse Graphs

- A popular non-quantitative scheme is SAFFRON [Lee2016]
- A variation of SAFFRON uses generalized LDPC (GLDPC) construction with SAFFRON as the signature matrix (component code) [Vem2017]

Signature matrix: $m_\text{b} \times d_\text{c}$

$$\boldsymbol{U} = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

Adjacency matrix: $m_\text{B} \times n$

$$\boldsymbol{B} = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

Test matrix: $m \times n$ $\quad \boldsymbol{A} = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$

$$m = m_\text{b} \times m_\text{B}$$

[Lee2016] K. Lee, R. Pedarsani, and K. Ramchandran, “SAFFRON: A fast, efficient, and robust framework for group testing based on sparse-graph codes,” in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Barcelona, Spain, July 2016.
[Vem2017] A. Vem, N. T. Janakiraman, and K. R. Narayanan, “Group testing using left-and-right-regular sparse-graph codes,” in *CoRR*, vol. abs/1701.07477, 2017.

Quantitative Group Testing with Sparse Graphs: Prior work

- The test results show the number of defectives
- Best known scheme with sparse graph uses GLDPC [KAR2019]

$$\boldsymbol{U} = \begin{pmatrix} \overbrace{\begin{matrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{matrix}}^{d_\mathsf{c}} \\ \begin{matrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{matrix} \end{pmatrix} \quad t = 1, m_\mathsf{u} = 3$$

- $d_\mathsf{c} = 2^{m_\mathsf{u}} - 1 \rightarrow m_\mathsf{u} = \log_2(d_\mathsf{c} + 1)$
- Tests per subcode $= t\log_2(d_\mathsf{c}+1)+1$
- Rate, $\Omega = \frac{m}{n} = \frac{d_\mathsf{v}}{d_\mathsf{c}}\left(t\left\lceil \log_2(d_\mathsf{c}+1)\right\rceil + 1\right)$

- A t-error-correcting BCH code is used as a component code
- An additional row of ones to identify # of defective items

[KAR2019] E. Karimi, F. Kazemi, A. Heidarzadeh, K. R. Narayanan, and A. Sprintson, “Sparse graph codes for non-adaptive quantitative group testing,” in *Proc. IEEE Inf. Theory Workshop (ITW)*, 2019.

Prior Work

- **Density Evolution**

  For each iteration ℓ

  $q^{(\ell)}$: probability a test sends resolved to item
  $p^{(\ell)}$: probability a defective item is unresolved

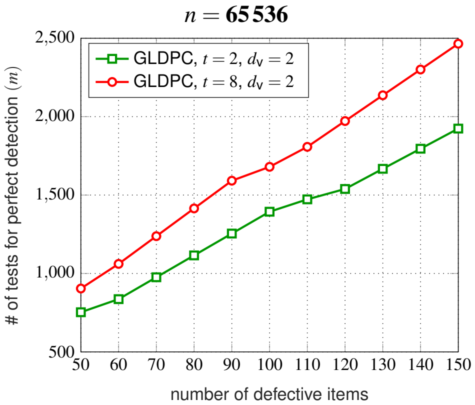
  Test to item:

  $$q^{(\ell)} = \sum_{i=0}^{t-1} \binom{d_c - 1}{i} \left(p^{(\ell-1)}\right)^i \left(1 - p^{(\ell-1)}\right)^{d_c - 1 - i}$$

  Item to test:

  $$p^{(\ell)} = \gamma (1 - q^{(\ell-1)})^{d_v - 1}$$

- Small number of tests for a large population size
- Increasing t improves error correction
- Penalized by increasing number of tests
  $m = n \frac{d_v}{d_c} \left(t \lceil \log_2(d_c + 1) \rceil + 1 \right)$



■ What about using $t = 0$?

Proposed scheme: Group Testing with LDPC

- With $t = 0$ we loose local error correcting capability
- We can observe and utilize two events
  - Syndrome equal zero: $s_i^{(\ell)} = 0$
    Infer all items as 0 (non-defective)
  - Syndrome equals test degree: $s_i^{(\ell)} = d_\mathrm{c}^{(\ell)}$
    Infer all items as 1 (defective)

?	?	?	?	?	?	$\boldsymbol{s}^{(1)}$	$\boldsymbol{d}_\mathrm{c}^{(1)}$
0	0	1	1	1	0	3	3
1	1	0	1	0	0	2	3
0	1	1	0	1	0	2	3
1	0	1	0	0	1	2	3

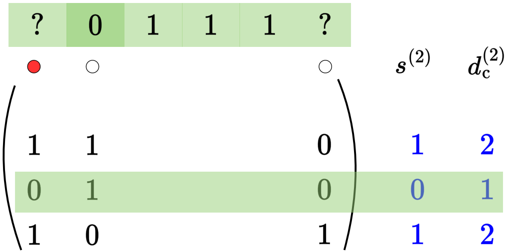
Proposed scheme: Group Testing with LDPC

- With $t = 0$ we loose local error correcting capability
- We can observe and utilize two events
  - Syndrome equal zero: $s_i^{(\ell)} = 0$
    Infer all items as 0 (non-defective)
  - Syndrome equals test degree: $s_i^{(\ell)} = d_{\mathrm{c}}^{(\ell)}$
    Infer all items as 1 (defective)
- We then peel off *resolved* items (reducing the syndrome accordingly)

?	?	1	1	1	?		
●	○	●	●	●	○	$s^{(1)}$	$d_{\mathrm{c}}^{(1)}$
0	0	1	1	1	0	3	3
1	1	0	1	0	0	2	3
0	1	1	0	1	0	2	3
1	0	1	0	0	1	2	3

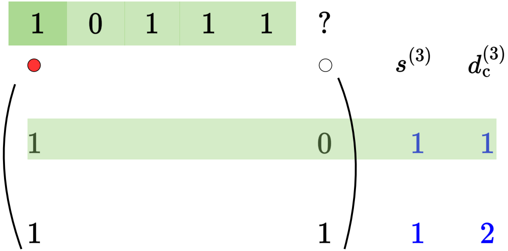
Proposed scheme: Group Testing with LDPC

- With $t = 0$ we loose local error correcting capability
- We can observe and utilize two events
  - Syndrome equal zero: $s_i^{(\ell)} = 0$
    Infer all items as 0 (non-defective)
  - Syndrome equals test degree: $s_i^{(\ell)} = d_c^{(\ell)}$
    Infer all items as 1 (defective)
- We then peel off *resolved* items (reducing the syndrome accordingly)
- This is repeated until no item to peel



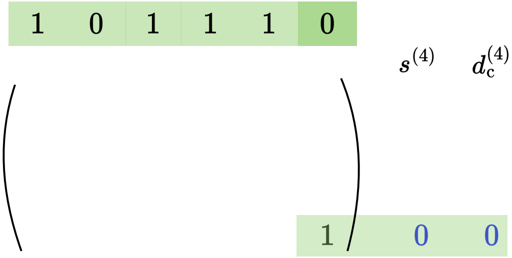
Proposed scheme: Group Testing with LDPC

- With $t = 0$ we loose local error correcting capability
- We can observe and utilize two events
  - Syndrome equal zero: $s_i^{(\ell)} = 0$
    Infer all items as 0 (non-defective)
  - Syndrome equals test degree: $s_i^{(\ell)} = d_c^{(\ell)}$
    Infer all items as 1 (defective)
- We then peel off *resolved* items (reducing the syndrome accordingly)
- This is repeated until no item to peel



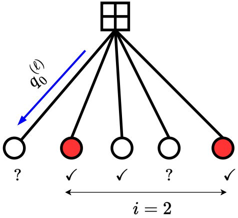
Proposed scheme: Group Testing with LDPC

- With $t = 0$ we loose local error correcting capability
- We can observe and utilize two events
  - Syndrome equal zero: $s_i^{(\ell)} = 0$
    Infer all items as 0 (non-defective)
  - Syndrome equals test degree: $s_i^{(\ell)} = d_{\text{c}}^{(\ell)}$
    Infer all items as 1 (defective)
- We then peel off *resolved* items (reducing the syndrome accordingly)
- This is repeated until no item to peel



Density Evolution

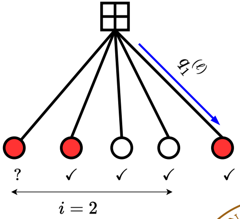
- $p_1^{(\ell)}$: probability that a message from a defective is *unresolved*
- $q_0^{(\ell)}$: **probability that a message to a non-defective is *resolved***
- $p_0^{(\ell)}$: probability a message from non-defective is *unresolved*
- $q_1^{(\ell)}$: **probability that a message to a defective is *resolved***



From test to item

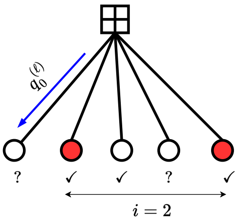
$$q_0^{(\ell)} = \sum_{i=0}^{d_c-1} \binom{d_c - 1}{i} \gamma^i (1-\gamma)^{d_c-1-i} \left(1 - p_1^{(\ell-1)}\right)^i$$

$$q_1^{(\ell)} = \sum_{i=0}^{d_c-1} \binom{d_c - 1}{i} \gamma^i (1-\gamma)^{d_c-1-i} \left(1 - p_0^{(\ell-1)}\right)^{d_c-1-i}$$



Density Evolution

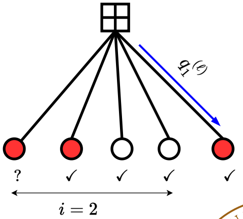
- $p_1^{(\ell)}$: **probability that a message from a defective is *unresolved***
- $q_0^{(\ell)}$: **probability that a message to a non-defective is *resolved***
- $p_0^{(\ell)}$: **probability a message from non-defective is *unresolved***
- $q_1^{(\ell)}$: **probability that a message to a defective is *resolved***



From test to item

$$q_0^{(\ell)} = \sum_{i=0}^{d_c-1} \binom{d_c-1}{i} \gamma^i (1-\gamma)^{d_c-1-i} \left(1-p_1^{(\ell-1)}\right)^i$$

$$q_1^{(\ell)} = \sum_{i=0}^{d_c-1} \binom{d_c-1}{i} \gamma^i (1-\gamma)^{d_c-1-i} \left(1-p_0^{(\ell-1)}\right)^{d_c-1-i}$$



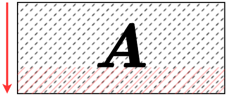
From item to test

$$p_0^{(\ell)} = \left(1-q_0^{(\ell-1)}\right)^{d_v-1}$$

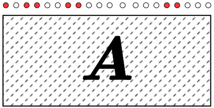
$$p_1^{(\ell)} = \left(1-q_1^{(\ell-1)}\right)^{d_v-1} .$$

Performance Comparison

- We consider two scenarios
  - **Fixing the proportion of defective items γ and changing the rate $\Omega = \frac{m}{n}$**

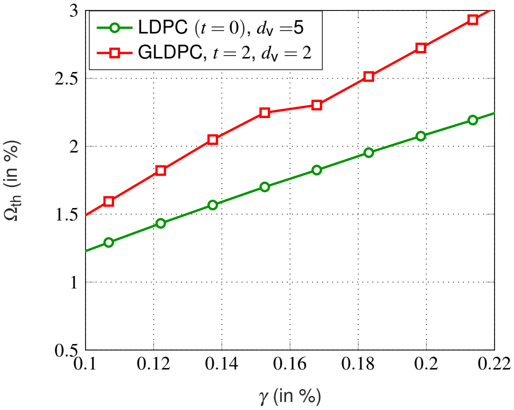


  - Same as in previous work [KAR2019]
  - **Fixing the rate Ω and changing γ**



  - A new perspective considering $\boldsymbol{A}$ (code) as fixed

Minimum rate required for a fixed γ



[KAR2019] E. Karimi, F. Kazemi, A. Heidarzadeh, K. R. Narayanan, and A. Sprintson, "Sparse graph codes for non-adaptive quantitative group testing," in *Proc. IEEE Inf. Theory Workshop (ITW)*, 2019.

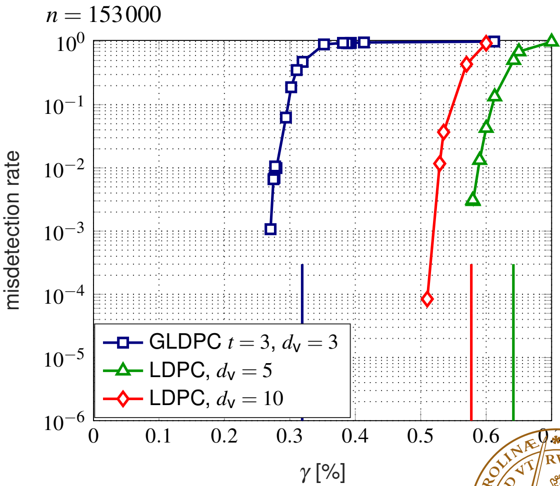
Performance Comparison: Fixed Rate, $\Omega = 5\%$

Table: GLDPC Based

t	d_v	γ_{th}
	2	0.2487
1	3	0.3708
	4	0.3510
	2	0.3983
2	3	0.3372
	4	0.2884
	2	0.3784
3	3	0.3189
	4	0.2441
	2	0.3418
5	3	0.2686
	4	0.2014

Table: LDPC Based

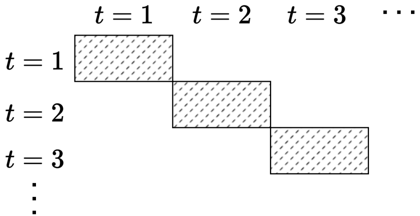
d_v	γ_{th}
3	0.4555
4	0.5982
5	0.6416
6	0.6464
7	0.6353
10	0.5773

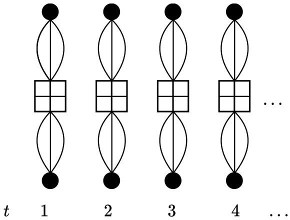


- Can we get consistently better with increasing d_v? What about spatial coupling?

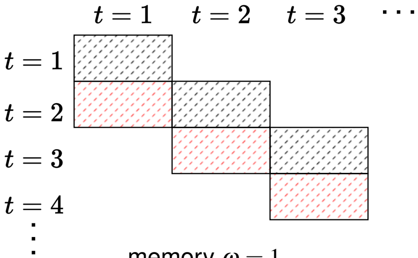
Group Testing with Spatial Coupling

- **Classical approach**: test each block of items separately

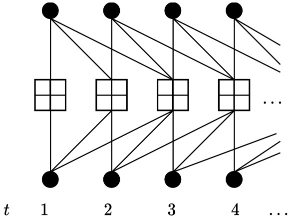




- **Spatial Coupling**: Interconnect blocks (motivated by results in coding theory)



memory $\omega = 1$



Group Testing with Spatial Coupling

- The chain is terminated after length L
- Density evolution

$$q_{0,\tau}^{(\ell)} = \frac{1}{\omega+1}\sum_{j=0}^{\omega}\sum_{i=0}^{d_c-1} \text{Bino}(d_c-1,i,\gamma)\left(1-p_{1,\tau-j}^{(\ell-1)}\right)^{i}$$

$$q_{1,\tau}^{(\ell)} = \frac{1}{\omega+1}\sum_{j=0}^{\omega}\sum_{i=0}^{d_c-1} \text{Bino}(d_c-1,i,\gamma)\left(1-p_{0,\tau-j}^{(\ell-1)}\right)^{d_c-1-i}$$

$$p_{0,\tau}^{(\ell)} = \frac{1}{\omega+1}\sum_{j=0}^{\omega}\left(1-q_{0,\tau+j}^{(\ell-1)}\right)^{d_v-1}$$

$$p_{1,\tau}^{(\ell)} = \frac{1}{\omega+1}\sum_{j=0}^{\omega}\left(1-q_{1,\tau+j}^{(\ell-1)}\right)^{d_v-1} .$$

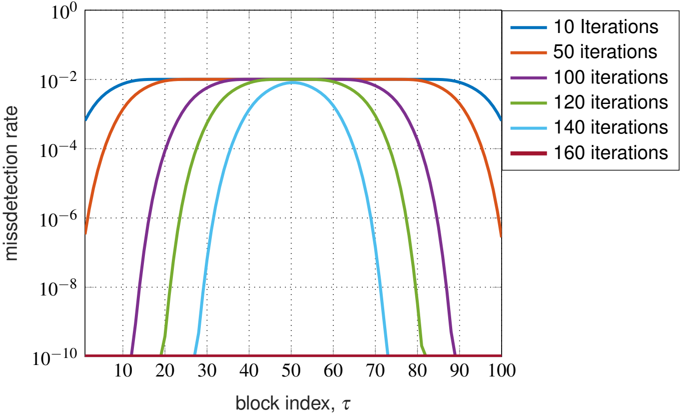
- $p_{0,\tau}^{(\ell)} = p_{0,\tau}^{(\ell)} = 0$ for $\tau < 0$ and $\tau > L$
- Rate becomes

$$\Omega_{SC} = \Omega\left(1+\frac{\omega}{L}\right)$$

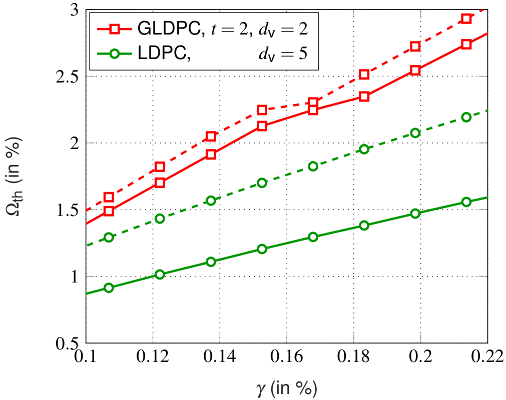
- The rate increase vanishes as L increases

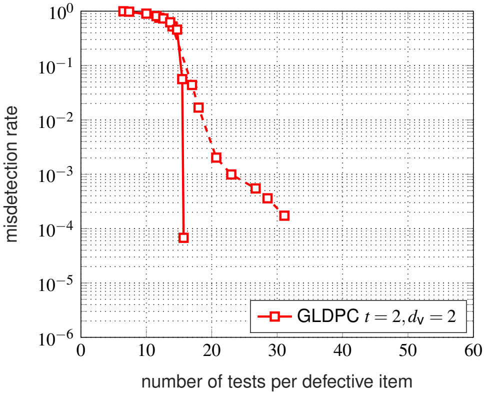
Spatial Coupling: Wave Effect

- Tests at boundary have lower degree
  - The nodes at the boundary can be resolved with higher probability
  - The effect spreads within the chain as a wave



Spatial coupling: Performance Changing Rate





- coupled (solid) uncoupled (dashed)
- memory $\omega = 5$
- $n = 153\,000$
- Low error floor with coupling

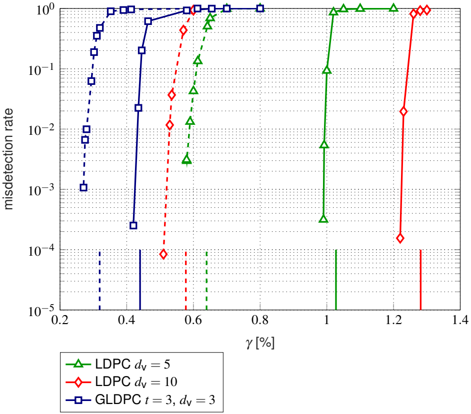
Spatial coupling: Performance for Fixed Rate

Table: γ_{th} for $\Omega = 5\%$ with GLDPC Code-Based

t	d_{v}	$\omega = 0$	$\omega = 1$	$\omega = 5$	$\omega = 10$
1	3	0.3708	0.4166	0.4166	0.4166
	4	0.3510	0.4395	0.4425	0.4425
3	3	0.3189	0.4257	0.4379	0.4395
	4	0.2441	0.3662	0.4028	0.4028
5	3	0.2686	0.3784	0.4089	0.4089
	4	0.2014	0.3159	0.3769	0.3769

Table: γ_{th} for $\Omega = 5\%$ with LDPC Code-Based

d_{v}	$\omega = 0$	$\omega = 1$	$\omega = 5$	$\omega = 10$
4	0.5982	0.8423	0.8540	0.8540
5	0.6416	0.9682	1.0274	1.0250
6	0.6464	1.0044	1.1325	1.1327
10	0.5773	0.9188	1.2814	1.2816



- $n = 153\,000$, $L = 200$, $\omega = 5$
- solid(coupled) dashed(uncoupled)

Can We Prove Threshold Saturation?

- Finding a density evolution recursion that follows the conditions of a vector admissible system.

$$q_0^{(\ell)} = \sum_{i=0}^{d_c-1} \binom{d_c-1}{i} \gamma^i (1-\gamma)^{d_c-1-i} \left(1-p_1^{(\ell-1)}\right)^i$$

$$q_1^{(\ell)} = \sum_{i=0}^{d_c-1} \binom{d_c-1}{i} \gamma^i (1-\gamma)^{d_c-1-i} \left(1-p_0^{(\ell-1)}\right)^{d_c-1-i}$$

$$p_0^{(\ell)} = \left(1-q_0^{(\ell-1)}\right)^{d_v-1}$$

$$p_1^{(\ell)} = \left(1-q_1^{(\ell-1)}\right)^{d_v-1} .$$

With $x_0^{(\ell)} = 1-q_0^{(\ell)}$, $x_1^{(\ell)} = 1-q_1^{(\ell)}$, $y_0^{(\ell)} = (1-\gamma)q_0^{(\ell)}$, $y_1^{(\ell)} = \gamma p_1^{(\ell)}$
we can write:

$$x_0^{(\ell)} = 1-\left(1-y_1^{(\ell-1)}\right)^{d_c-1}$$

$$x_1^{(\ell)} = 1-\left(1-y_0^{(\ell-1)}\right)^{d_c-1}$$

$$y_0^{(\ell)} = (1-\gamma)\left(x_0^{(\ell-1)}\right)^{d_v-1}$$

$$y_1^{(\ell)} = \gamma\left(x_1^{(\ell-1)}\right)^{d_v-1} .$$

With spatial coupling:

$$x_{0,\tau}^{(\ell)} = 1-\frac{1}{\omega+1}\sum_{j=0}^{\omega}\left(1-y_{1,\tau-j}^{(\ell-1)}\right)^{d_c-1}$$

$$x_{1,\tau}^{(\ell)} = 1-\frac{1}{\omega+1}\sum_{j=0}^{\omega}\left(1-y_{0,\tau-j}^{(\ell-1)}\right)^{d_c-1}$$

$$y_{0,\tau}^{(\ell)} = (1-\gamma)\frac{1}{\omega+1}\sum_{j=0}^{\omega}\left(x_{0,\tau+j}^{(\ell-1)}\right)^{d_v-1}$$

$$y_{1,\tau}^{(\ell)} = \gamma\frac{1}{\omega+1}\sum_{j=0}^{\omega}\left(x_{1,\tau+j}^{(\ell-1)}\right)^{d_v-1} .$$

First approach: finding maximum γ for a fixed Ω

- **Vector admissible system:** [YED2012] a recursion $(\mathbf{f}, \boldsymbol{g})$ with

$$\boldsymbol{x}^{(\ell)} = \mathbf{f}\left(\boldsymbol{g}(\boldsymbol{x}^{(\ell-1)}); \varepsilon\right) \;, \quad \boldsymbol{x}^{(0)} = \mathbf{1} \;, \; \varepsilon \in [0,1]$$

  where $\mathbf{f}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), \cdots, f_d(\boldsymbol{x})]$ and $\boldsymbol{g}(\boldsymbol{x}) = [g_1(\boldsymbol{x}), \cdots, g_d(\boldsymbol{x})]$ are twice continuously differentiable and strictly increasing in all arguments.

- With $\varepsilon = \gamma$ we get from density evolution equations:

$$\mathbf{f}_\gamma(x_0, x_1; \gamma) = \left[(1-\gamma)\cdot x_0^{d_\mathsf{v}-1}, \quad \gamma \cdot x_1^{d_\mathsf{v}-1}\right]$$
$$\mathbf{g}_\gamma(y_0, y_1) = \left[1-(1-y_1)^{d_\mathsf{c}-1}, \quad 1-(1-y_0)^{d_\mathsf{c}-1}\right].$$

- **Problem:** $(1-\gamma)$ and γ cannot both increase $\Rightarrow$ conditions not fulfilled

[YED2012] A. Yedla, Y.-Y. Jian, P. S. Nguyen, and H. D. Pfister, "A simple proof of threshold saturation for coupled vector recursions," in *Proc. IEEE Inf. Theory Workshop (ITW)*, 2012.

Second approach: finding minimum Ω for a fixed γ

- **Vector admissible system:** [YED2012] a recursion $(\mathbf{f}, \boldsymbol{g})$ with

$$\boldsymbol{x}^{(\ell)} = \mathbf{f}\left(\boldsymbol{g}(\boldsymbol{x}^{(\ell-1)}); \varepsilon\right) \;, \quad \boldsymbol{x}^{(0)} = \mathbf{1} \;,\; \varepsilon \in [0,1]$$

where $\mathbf{f}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), \cdots, f_d(\boldsymbol{x})]$ and $\boldsymbol{g}(\boldsymbol{x}) = [g_1(\boldsymbol{x}), \cdots, g_d(\boldsymbol{x})]$ are twice continuously differentiable and strictly increasing in all arguments.

- Setting $\varepsilon = 1 - \frac{1}{d_\mathsf{c}}$ we get from density evolution equations:

$$\mathbf{f}(y_0, y_1; \varepsilon) = \left[1 - (1-y_1)^{\frac{\varepsilon}{1-\varepsilon}}, \quad 1 - (1-y_0)^{\frac{\varepsilon}{1-\varepsilon}}\right]$$

$$\boldsymbol{g}(x_0, x_1) = \left[(1-\gamma) \cdot x_0^{d_\mathsf{v}-1}, \quad \gamma \cdot x_1^{d_\mathsf{v}-1}\right]$$

- Threshold saturation occurs

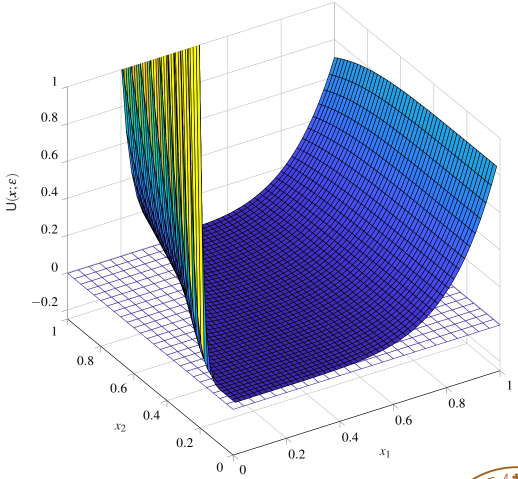
- The potential function is then given as

$$\mathsf{U}(\boldsymbol{x}; \varepsilon) = \int_0^1 \left((z(\lambda) - \mathbf{f}(\boldsymbol{g}(z(\lambda)); \varepsilon)) \boldsymbol{D}\boldsymbol{g}'(z(\lambda)) \right) \cdot z'(\lambda) \mathrm{d}\lambda$$

Potential function

$$\mathsf{U}(\boldsymbol{x};\varepsilon) = (1-p)x_1^{d_\mathsf{v}-1}\left((1-\varepsilon)\frac{1-\left(1-px_2^{d_\mathsf{v}-1}\right)^{\frac{1}{1-\varepsilon}}}{px_2^{d_\mathsf{v}-1}} + \frac{(d_\mathsf{v}-1)}{d_\mathsf{v}}x_1 - 1\right)$$

$$+px_2^{d_\mathsf{v}-1}\left((1-\varepsilon)\frac{1-\left(1-(1-p)x_2^{d_\mathsf{v}-1}\right)^{\frac{1}{1-\varepsilon}}}{(1-p)x_1^{d_\mathsf{v}-1}} + \frac{(d_\mathsf{v}-1)}{d_\mathsf{v}}x_2 - 1\right)$$

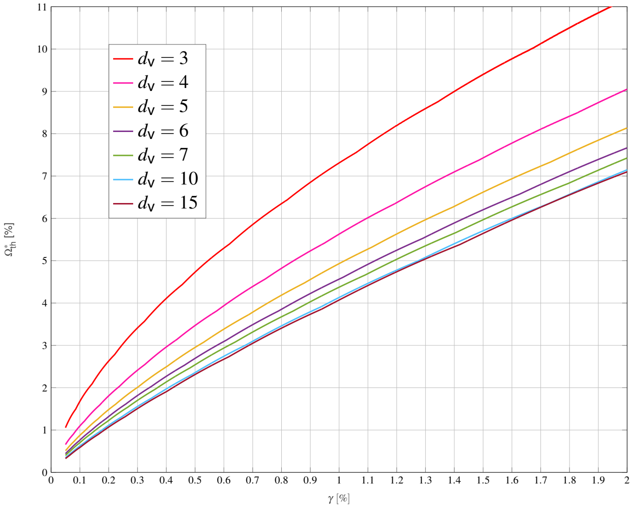


Potential threshold:

$$\varepsilon^* = \sup\{\varepsilon \in [0,1] \mid \min_{\boldsymbol{x}} \mathsf{U}(\boldsymbol{x};\varepsilon) \geq 0\}\,.$$

$d_\mathsf{v} = 6$, $\gamma = 1\%$ with $\varepsilon^* = 0.9924$. $\mathsf{U}(\boldsymbol{x};\varepsilon)$ is above the $z = 0$ plane since $\varepsilon = 0.9667 < \varepsilon^*$.

Potential thresholds



$$\Omega^*_{\mathsf{th}} = \frac{d_{\mathsf{v}}}{d_{\mathsf{c}}} = d_{\mathsf{v}}(1-\varepsilon^*)\,.$$

$$\varepsilon^* = \sup\{\varepsilon \in [0,1] \,|\, \min_{\boldsymbol{x}} \mathsf{U}(\boldsymbol{x};\varepsilon) \geq 0\}\,.$$

The minimum rate Ω^*_{th} for a fixed γ computed from the potential threshold ε^*.

Conclusions and Outlook

Conclusions

- Using a simple LDPC code significantly outperforms a GLDPC construction with t-error-correcting component code
- With spatial coupling we can improve the performance of both schemes
- We can measure the performance by two different approaches
  - Fixing the proportion γ and determining minimum rate Ω
  - Fixing the rate, Ω and determining the maximum γ
- Threshold saturation: with coupling the BP decoder achieves the potential threshold

Outlook

- Bundling of tests: non-binary messages can improve performance
- Looking at soft message passing