# On the Decoding of Tail-Biting UM-LDPC Codes

Igor Zhilin     Pavel Rybin     Fedor Ivanov     Victor Zyablov

Institute for Information Transmission Problems
Russian Academy of Sciences

XIV International Workshop on Algebraic and Combinatorial Coding Theory

# Outline

- Construction of (P)UM-LDPC codes
- Decoding algorithms
  - Belief Propagation
  - Single direction decoding
  - Double direction decoding
- Simulation results
- Conclusions

# Construction of (P)UM-LDPC codes

UM-LDPC code is defined by its semi-infinite parity-check matrix $\mathbf{H}'$:

$$\mathbf{H}' = \begin{pmatrix} \ddots & \ddots & & & \\ & \mathbf{H}_{t,1} & \mathbf{H}_{t,0} & & \\ & & \mathbf{H}_{1,1} & \mathbf{H}_{1,0} & \\ & & & \ddots & \ddots \end{pmatrix}$$

where $\mathbf{H}_{i,0}$, $\mathbf{H}_{i,1}$ — $r \times n$ parity-check matrices of component Gallagers LDPC block codes, $r = n - k$, $i = \overline{1..t}$, $t$ is a period.
$\mathbf{H}_{i,0}$ must have full rank.
$\mathbf{H}_{i,1}$ may have lesser rank if the code is PUM: $rank\left(\mathbf{H}_{i,0}\right) = r$, $rank\left(\mathbf{H}_{i,1}\right) = r_1 \leq r$.
The code rate $R'$ of constructed UM-LDPC code is equal to code rate of code with parity-check matrix $\mathbf{H}_{i,0}$.

# Tail-biting (P)UM-LDPC codes

Tail-biting UM-LDPC code with length $N = nt$ have the following parity-check matrix $\mathbf{H}$ (with tail-biting on the period $t$):

$$\mathbf{H} = \begin{pmatrix} \mathbf{H}_{1,1} & \mathbf{H}_{1,0} & & & \\ & \mathbf{H}_{2,1} & \mathbf{H}_{2,0} & & \\ & & \ddots & \ddots & \\ \mathbf{H}_{t,0} & & & & \mathbf{H}_{t,1} \end{pmatrix}$$

its size is $tr \times tn$.

So, the code rate $R$ of tail-biting UM-LDPC is given by:

$$R \geq 1 - \frac{tr}{tn} = 1 - \frac{r}{n} = R_{i,\{0,1\}}.$$

*Constructed tail-biting UM-LDPC code is itself also an LDPC code with special construction of parity-check matrix.*

# Decoding algorithms

- $\mathcal{A}(i_{max})$ — Belief Propagation, where $i_{max}$ — maximum number of iterations
- $\mathcal{B}(i_{max}, j_{max})$ — single direction decoding, $i_{max}$ is the number of inner iterations and $j_{max}$ is the number of outer iterations
- $\mathcal{C}(i_{max}, j_{max})$ — decoding in both directions

The received sequence $\mathbf{r}$ can be represented as
$\mathbf{r} = \begin{pmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \dots & \mathbf{r}_t \end{pmatrix}$,
where $\mathbf{r}_i$ is a vector with length $n$, that corresponds to the parity-check matrices $\mathbf{H}_{i,1}$ and $\mathbf{H}_{((i-1) \bmod t)+1,0}$.

# Algorithm $\mathcal{B}$

$\mathbf{r}_k^{(0)} \longleftarrow \mathbf{r}_k, \ \forall k : 1 \leq k \leq t$

$\mathbf{r}_1^{(1)} \longleftarrow \mathbf{r}_1$

$\boldsymbol{\Delta}_k^{(0)} \longleftarrow 0, \ \forall k : 1 \leq k \leq t$

**for** $j = 1$ **to** $j_{max}$ **do**

  **for** $k = 1$ **to** $t$ **do**

    $k_1 \longleftarrow k, \ k_2 \longleftarrow (k \bmod t) + 1$

    $\mathbf{x}_{k_1} \longleftarrow \mathbf{r}_{k_1}^{(j)}$

    $\mathbf{x}_{k_2} \longleftarrow \mathbf{r}_{k_2}^{(j-1)} - \boldsymbol{\Delta}_{k_2}^{(j-1)}$

    $\left(\mathbf{y}_{k_1} \quad \mathbf{y}_{k_2}\right) \longleftarrow D_k^{(i_{max})}\left(\left(\mathbf{x}_{k_1} \quad \mathbf{x}_{k_2}\right)\right)$

    $\boldsymbol{\Delta}_{k_2}^{(j)} \longleftarrow \mathbf{y}_{k_2} - \mathbf{x}_{k_2}$

    $\mathbf{r}_{k_2}^{(j)} \longleftarrow \mathbf{r}_{k_2}^{(j-1)} + \boldsymbol{\Delta}_{k_2}^{(j)}$

  **end for**

**end for**

**return** $\mathbf{r}^{(j_{max})} = \left(\mathbf{r}_1^{(j_{max})} \mathbf{r}_2^{(j_{max})} \ldots \mathbf{r}_t^{(j_{max})}\right)$

# Algorithm $\mathcal{C}$

$\mathbf{r}_k^{(0)} \longleftarrow \mathbf{r}_k,\ \forall k : 1 \leq k \leq t$

$\mathbf{\Delta}_{k,\{0,1\}}^{(0)} \longleftarrow 0,\ \forall k : 1 \leq k \leq t$

**for** $j = 1$ **to** $j_{max}$ **do**

  **for** $k = 1$ **to** $t$ **do**

    $k_1 \longleftarrow k,\ k_2 \longleftarrow (k \bmod t) + 1$

    $\mathbf{x}_{k_1} \longleftarrow \mathbf{r}_{k_1}^{(j-1)} - \mathbf{\Delta}_{k_1,1}^{(j-1)}$

    $\mathbf{x}_{k_2} \longleftarrow \mathbf{r}_{k_2}^{(j-1)} - \mathbf{\Delta}_{k_2,0}^{(j-1)}$

    $\left(\mathbf{y}_{k_1} \quad \mathbf{y}_{k_2}\right) \longleftarrow D_k^{(i_{max})}\left(\left(\mathbf{x}_{k_1} \quad \mathbf{x}_{k_2}\right)\right)$

    $\mathbf{\Delta}_{k_1,1}^{(j)} \longleftarrow \mathbf{y}_{k_1} - \mathbf{x}_{k_1}$

    $\mathbf{\Delta}_{k_2,0}^{(j)} \longleftarrow \mathbf{y}_{k_2} - \mathbf{x}_{k_2}$

  **end for**

  **for** $k = 1$ **to** $t$ **do**

    $\mathbf{r}_k^{(j)} \longleftarrow \mathbf{r}_k^{(j-1)} + \mathbf{\Delta}_{k,0}^{(j)} + \mathbf{\Delta}_{k,1}$

  **end for**

**end for**

**return** $\mathbf{r}^{(j_{max})} = \left(\mathbf{r}_1^{(j_{max})}\mathbf{r}_2^{(j_{max})} \ldots \mathbf{r}_t^{(j_{max})}\right)$

# Computer Simulation

UM-LDPC code, period $t = 4$, based on (2,4) LDPC codes with parity-check matrices $\mathbf{H}_{i,\{0,1\}}$, s.t. parity-check matrices $\mathbf{H}_i = \begin{pmatrix} \mathbf{H}_{i,1} & \mathbf{H}_{i,0} \end{pmatrix}$ have girth $\geq 4$.

$R = 1 - \frac{2}{4} = 0.5$,
$N = 2032$.

BPSK, AWGN, soft decision.

# Results of the Simulation

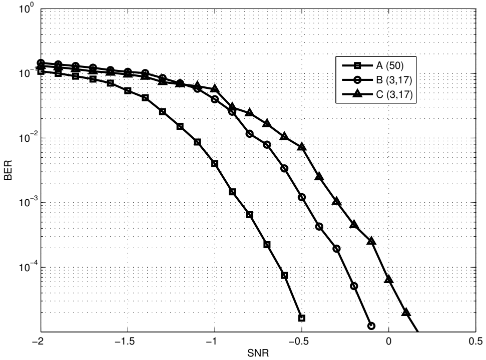


Figure: Simulation results for UM-LDPC code with code rate $R{=}0.5$, based on the LDPC codes (2,4), under decoding algorithms $\mathcal{A}\,(50)$, $\mathcal{B}\,(3,17)$ and $\mathcal{C}\,(3,17)$

# Conclusion

- $\mathcal{A}\,(50)$ has the best decoding performance. $\mathcal{B}\,(3,17)$ performs worse than $\mathcal{A}\,(50)$ by almost 0.4 dB at $\mathrm{BER} = 10^{-5}$. Performance of $\mathcal{C}\,(3,17)$ is worse than performance of $\mathcal{B}\,(3,17)$ by almost 0.2 dB at the same BER.
- Complexities of algorithms $\mathcal{B}$ and $\mathcal{C}$ is asymptotically the same as complexity of $\mathcal{A}$ up to a constant factor which is close to 1. Since $\mathcal{A}$ performs better than $\mathcal{B}$ or $\mathcal{C}$, it is unreasonable to use algorithms $\mathcal{B}$ and $\mathcal{C}$.

Thank you for your attention.