Density Evolution for Deterministic Generalized Product Codes with Higher-Order Modulation

Christian Häger[1] Alexandre Graell i Amat[2]
Henry D. Pfister[1] Fredrik Brännström[2]

[1] Department of Electrical and Computer Engineering, Duke University, Durham

[2] Department of Signals and Systems, Chalmers University of Technology, Gothenburg

International Symposium on Turbo Codes & Iterative Information Processing
Brest, France, September 6, 2016

FORCE
FIBER-OPTIC COMMUNICATIONS
RESEARCH CENTER

CHALMERS

Motivation and Outline

Motivation and Outline

- Error-correcting codes for high-speed fiber-optical communications: Generalized product codes with iterative bounded-distance decoding are very appealing (low complexity)

Motivation and Outline

- Error-correcting codes for high-speed fiber-optical communications: Generalized product codes with iterative bounded-distance decoding are very appealing (low complexity)
- Code proposals are often very structured (i.e., deterministic):
  - Conventional product codes [Justesen et al., 2010],
  - Spatially-coupled (or convolutional-like) versions such as staircase codes [Smith et al., 2012] and braided codes [Jian, 2013]

Motivation and Outline

- Error-correcting codes for high-speed fiber-optical communications: Generalized product codes with iterative bounded-distance decoding are very appealing (low complexity)
- Code proposals are often very structured (i.e., deterministic):
    - Conventional product codes [Justesen et al., 2010],
    - Spatially-coupled (or convolutional-like) versions such as staircase codes [Smith et al., 2012] and braided codes [Jian, 2013]
- Asymptotic density evolution analysis possible for the binary erasure channel (BEC) without ensemble argument [Häger et al., 2015]

Motivation and Outline

- Error-correcting codes for high-speed fiber-optical communications: Generalized product codes with iterative bounded-distance decoding are very appealing (low complexity)
- Code proposals are often very structured (i.e., deterministic):
  - Conventional product codes [Justesen et al., 2010],
  - Spatially-coupled (or convolutional-like) versions such as staircase codes [Smith et al., 2012] and braided codes [Jian, 2013]
- Asymptotic density evolution analysis possible for the binary erasure channel (BEC) without ensemble argument [Häger et al., 2015]
- Recent trend towards spectrally-efficient fiber-optical systems

Motivation and Outline

- Error-correcting codes for high-speed fiber-optical communications: Generalized product codes with iterative bounded-distance decoding are very appealing (low complexity)
- Code proposals are often very structured (i.e., deterministic):
  - Conventional product codes [Justesen et al., 2010],
  - Spatially-coupled (or convolutional-like) versions such as staircase codes [Smith et al., 2012] and braided codes [Jian, 2013]
- Asymptotic density evolution analysis possible for the binary erasure channel (BEC) without ensemble argument [Häger et al., 2015]
- Recent trend towards spectrally-efficient fiber-optical systems

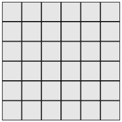
In This Talk . . .

- Deterministic code construction that recovers product codes, staircase codes, and block-wise braided codes as special cases
- Rigorous density evolution analysis over parallel BECs
- Application: Bit mapper (interleaver) optimization for coded modulation

Introduction: Product Codes and Staircase Codes

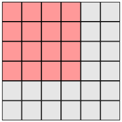
Introduction: Product Codes and Staircase Codes

rectangular array [Elias, 1954]



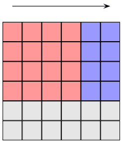
Introduction: Product Codes and Staircase Codes

rectangular array [Elias, 1954]



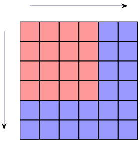
Introduction: Product Codes and Staircase Codes

rectangular array [Elias, 1954]



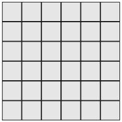
Introduction: Product Codes and Staircase Codes

rectangular array [Elias, 1954]



Introduction: Product Codes and Staircase Codes

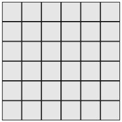
rectangular array [Elias, 1954]



each row/column is a codeword in some component code

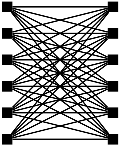
Introduction: Product Codes and Staircase Codes

rectangular array [Elias, 1954]



each row/column is a codeword in some component code

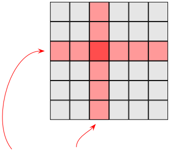
Tanner graph



constraint node (CN) degree = component code length

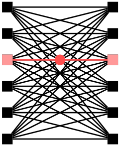
Introduction: Product Codes and Staircase Codes

rectangular array [Elias, 1954]



each row/column is a codeword in
some component code

Tanner
graph

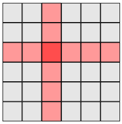


edge = degree-2 variable node (VN)

constraint node (CN) degree = component code length

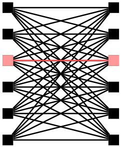
Introduction: Product Codes and Staircase Codes

rectangular array [Elias, 1954]



each row/column is a codeword in some component code

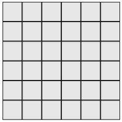
Tanner graph



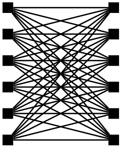
constraint node (CN) degree = component code length

Introduction: Product Codes and Staircase Codes

rectangular array [Elias, 1954]

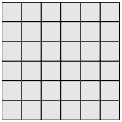


Tanner graph

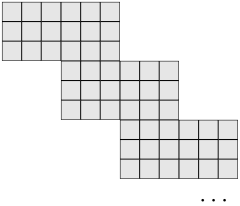


Introduction: Product Codes and Staircase Codes

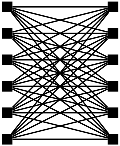
rectangular array [Elias, 1954]



staircase array [Smith et al., 2012]

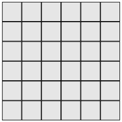


Tanner graph

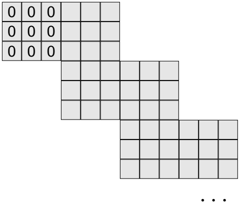


Introduction: Product Codes and Staircase Codes

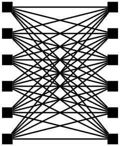
rectangular array [Elias, 1954]



staircase array [Smith et al., 2012]

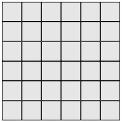


Tanner graph

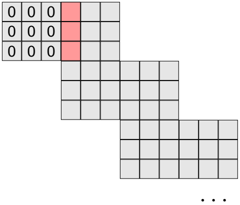


Introduction: Product Codes and Staircase Codes

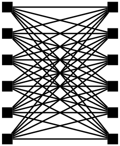
rectangular array [Elias, 1954]



staircase array [Smith et al., 2012]



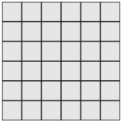
Tanner graph

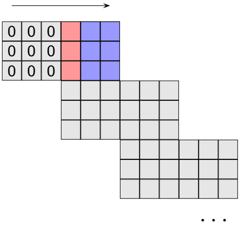


Introduction: Product Codes and Staircase Codes

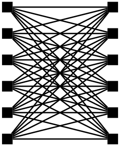
rectangular array [Elias, 1954]

staircase array [Smith et al., 2012]





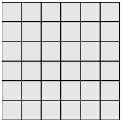
Tanner graph

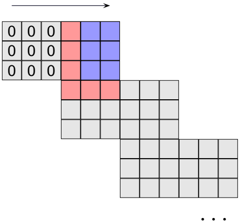


Introduction: Product Codes and Staircase Codes

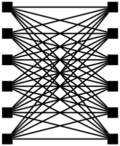
rectangular array [Elias, 1954]

staircase array [Smith et al., 2012]



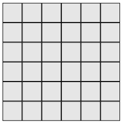


Tanner graph

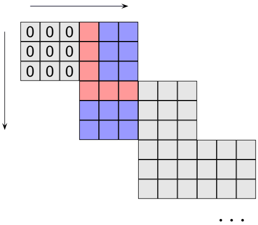


Introduction: Product Codes and Staircase Codes

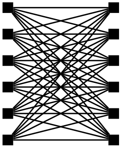
rectangular array [Elias, 1954]



staircase array [Smith et al., 2012]

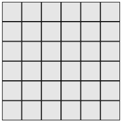


Tanner graph

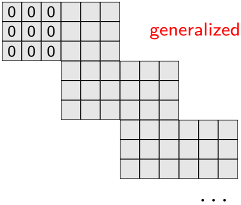


Introduction: Product Codes and Staircase Codes

rectangular array [Elias, 1954]

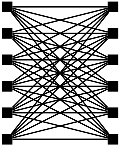


staircase array [Smith et al., 2012]



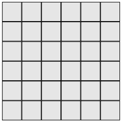
generalized product code (GPC)

Tanner graph

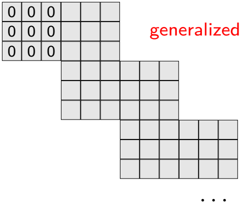


Introduction: Product Codes and Staircase Codes

rectangular array [Elias, 1954]

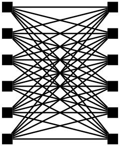


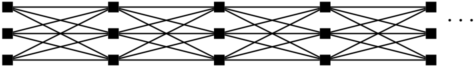
staircase array [Smith et al., 2012]



generalized product code (GPC)

Tanner graph

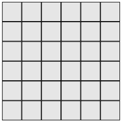


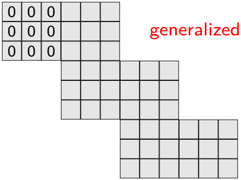


Introduction: Product Codes and Staircase Codes

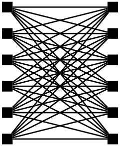
rectangular array [Elias, 1954]

staircase array [Smith et al., 2012]

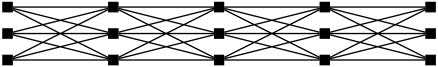




Tanner graph

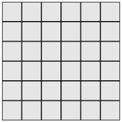


spatially-coupled code

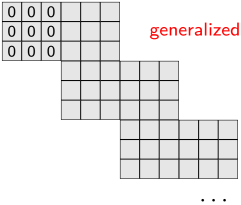


Introduction: Product Codes and Staircase Codes

rectangular array [Elias, 1954]

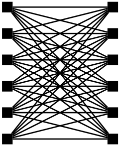


staircase array [Smith et al., 2012]

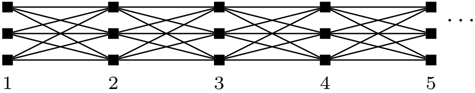


generalized product code (GPC)

Tanner graph



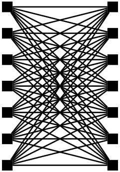
spatially-coupled code

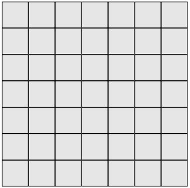


positions:

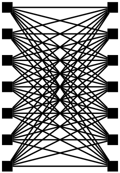
- Deterministic codes with fixed and structured Tanner graph

Iterative Bounded-Distance Decoding



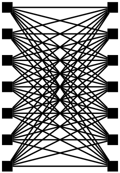


Iterative Bounded-Distance Decoding



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

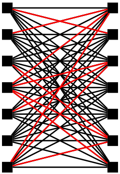
Iterative Bounded-Distance Decoding



0	?	0	?	0	1	?
?	1	0	1	1	0	1
0	1	0	?	0	?	?
1	1	1	0	1	1	0
0	0	1	0	0	0	1
1	0	?	?	1	1	?
0	1	0	?	0	1	1

- Codeword transmission over the BEC with erasure probability p

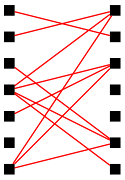
Iterative Bounded-Distance Decoding



0	?	0	?	0	1	?
?	1	0	1	1	0	1
0	1	0	?	0	?	?
1	1	1	0	1	1	0
0	0	1	0	0	0	1
1	0	?	?	1	1	?
0	1	0	?	0	1	1

- Codeword transmission over the BEC with erasure probability p

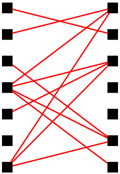
Iterative Bounded-Distance Decoding



0	?	0	?	0	1	?
?	1	0	1	1	0	1
0	1	0	?	0	?	?
1	1	1	0	1	1	0
0	0	1	0	0	0	1
1	0	?	?	1	1	?
0	1	0	?	0	1	1

- Codeword transmission over the BEC with erasure probability p

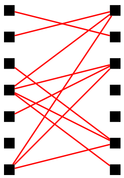
Iterative Bounded-Distance Decoding


residual graph

0	?	0	?	0	1	?
?	1	0	1	1	0	1
0	1	0	?	0	?	?
1	1	1	0	1	1	0
0	0	1	0	0	0	1
1	0	?	?	1	1	?
0	1	0	?	0	1	1

- Codeword transmission over the BEC with erasure probability p
- Each CN corresponds to t-erasure correcting component code

Iterative Bounded-Distance Decoding



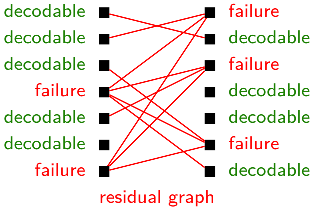
residual graph

0	?	0	?	0	1	?
?	1	0	1	1	0	1
0	1	0	?	0	?	?
1	1	1	0	1	1	0
0	0	1	0	0	0	1
1	0	?	?	1	1	?
0	1	0	?	0	1	1

- Codeword transmission over the BEC with erasure probability p
- Each CN corresponds to t-erasure correcting component code
- ℓ iterations of bounded-distance decoding $=$ peeling of vertices with degree $\leq t$ (in parallel)

Iterative Bounded-Distance Decoding

1st iteration $(t = 2)$

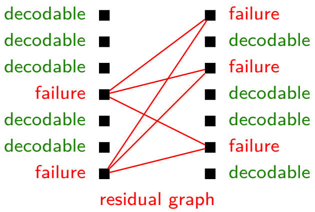


0	?	0	?	0	1	?
?	1	0	1	1	0	1
0	1	0	?	0	?	?
1	1	1	0	1	1	0
0	0	1	0	0	0	1
1	0	?	?	1	1	?
0	1	0	?	0	1	1

- Codeword transmission over the BEC with erasure probability p
- Each CN corresponds to t-erasure correcting component code
- ℓ iterations of bounded-distance decoding $=$ peeling of vertices with degree $\leq t$ (in parallel)

Iterative Bounded-Distance Decoding

1st iteration $(t = 2)$

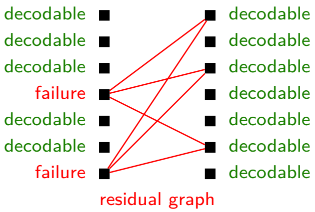


0	1	0	?	0	1	?
0	1	0	1	1	0	1
0	1	0	?	0	1	?
1	1	1	0	1	1	0
0	0	1	0	0	0	1
1	0	0	?	1	1	?
0	1	0	0	0	1	1

- Codeword transmission over the BEC with erasure probability p
- Each CN corresponds to t-erasure correcting component code
- ℓ iterations of bounded-distance decoding $=$ peeling of vertices with degree $\leq t$ (in parallel)

Iterative Bounded-Distance Decoding

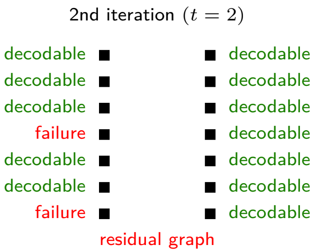
2nd iteration $(t = 2)$



0	1	0	?	0	1	?
0	1	0	1	1	0	1
0	1	0	?	0	1	?
1	1	1	0	1	1	0
0	0	1	0	0	0	1
1	0	0	?	1	1	?
0	1	0	0	0	1	1

- Codeword transmission over the BEC with erasure probability p
- Each CN corresponds to t-erasure correcting component code
- ℓ iterations of bounded-distance decoding $=$ peeling of vertices with degree $\leq t$ (in parallel)

Iterative Bounded-Distance Decoding

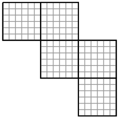


0	1	0	1	0	1	0
0	1	0	1	1	0	1
0	1	0	1	0	1	0
1	1	1	0	1	1	0
0	0	1	0	0	0	1
1	0	0	0	1	1	1
0	1	0	0	0	1	1

- Codeword transmission over the BEC with erasure probability p
- Each CN corresponds to t-erasure correcting component code
- ℓ iterations of bounded-distance decoding = peeling of vertices with degree $\leq t$ (in parallel)

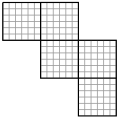
Asymptotic Performance Prediction

Asymptotic Performance Prediction



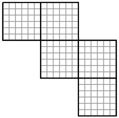
- Example: staircase code with a fixed component code

Asymptotic Performance Prediction

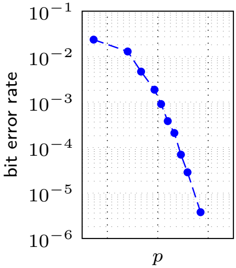


- Example: staircase code with a fixed component code
- Use simulations to predict performance $\rightarrow$ computationally intensive

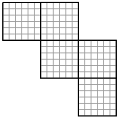
Asymptotic Performance Prediction



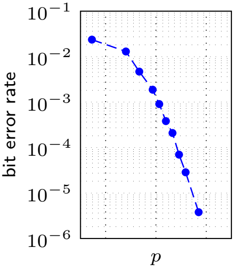
- Example: staircase code with a fixed component code
- Use simulations to predict performance $\rightarrow$ computationally intensive



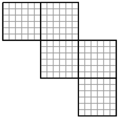
Asymptotic Performance Prediction



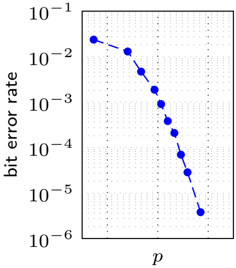
- Example: staircase code with a fixed component code
- Use simulations to predict performance $\rightarrow$ computationally intensive
- Efficient asymptotic analysis possible via density evolution without ensemble argument [Häger et al., 2015]



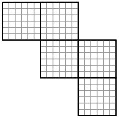
Asymptotic Performance Prediction



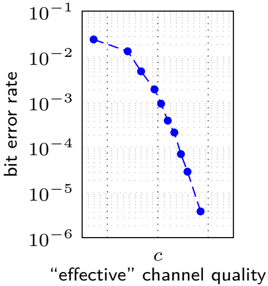
- Example: staircase code with a fixed component code
- Use simulations to predict performance → computationally intensive
- Efficient asymptotic analysis possible via density evolution without ensemble argument [Häger et al., 2015]
- Analysis for transmission over the BEC where $p = c/n$ for $c > 0$ and $n \to \infty$



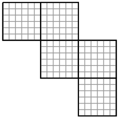
Asymptotic Performance Prediction



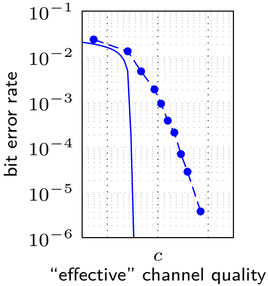
- Example: staircase code with a fixed component code
- Use simulations to predict performance → computationally intensive
- Efficient asymptotic analysis possible via density evolution without ensemble argument [Häger et al., 2015]
- Analysis for transmission over the BEC where $p = c/n$ for $c > 0$ and $n \to \infty$



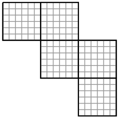
Asymptotic Performance Prediction

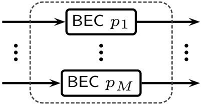


- Example: staircase code with a fixed component code
- Use simulations to predict performance → computationally intensive
- Efficient asymptotic analysis possible via density evolution without ensemble argument [Häger et al., 2015]
- Analysis for transmission over the BEC where $p = c/n$ for $c > 0$ and $n \to \infty$

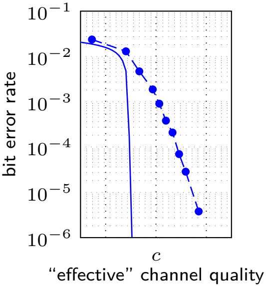


Asymptotic Performance Prediction





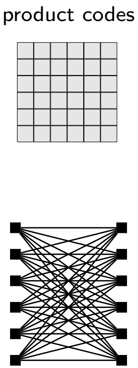
- Example: staircase code with a fixed component code
- Use simulations to predict performance → computationally intensive
- Efficient asymptotic analysis possible via density evolution without ensemble argument [Häger et al., 2015]
- Analysis for transmission over the BEC where $p = c/n$ for $c > 0$ and $n \to \infty$

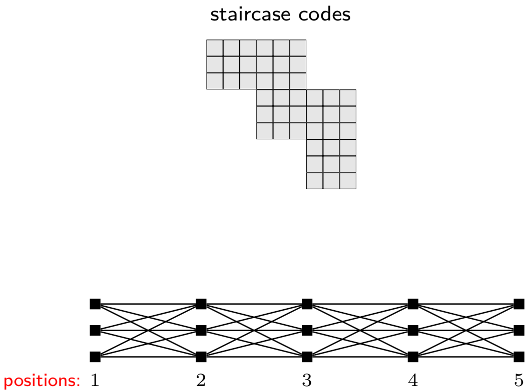


Main contribution

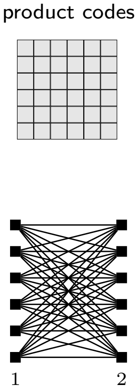
Generalization to parallel BECs with different erasure probabilities

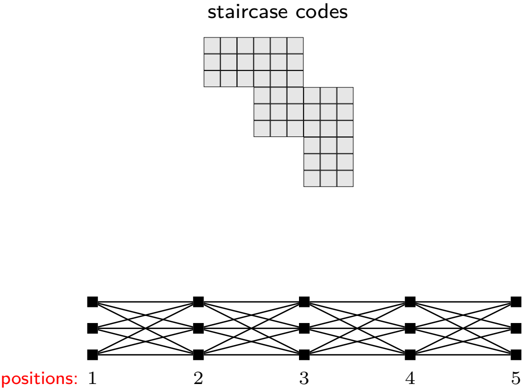
Parametrized Construction of Generalized Product Codes



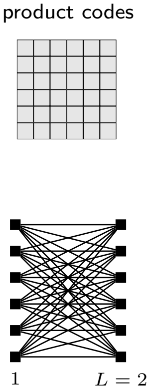


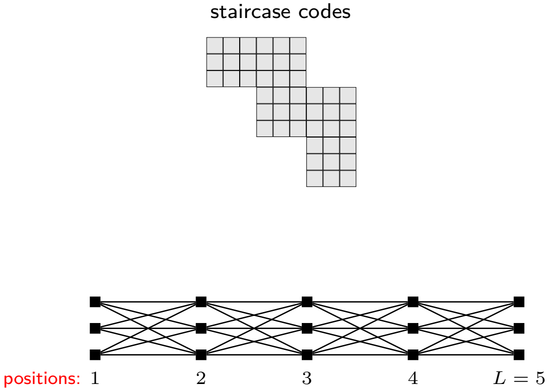
Parametrized Construction of Generalized Product Codes



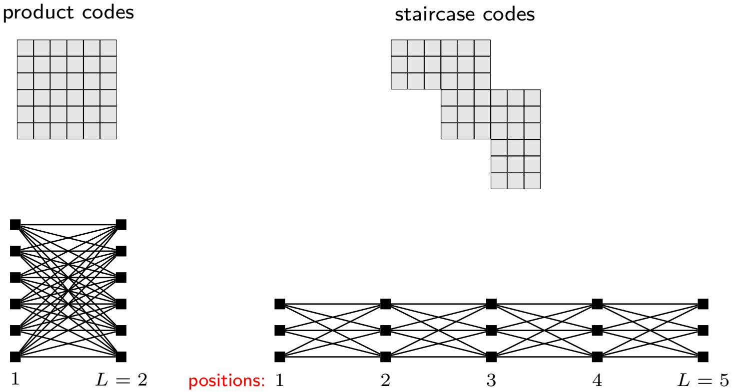


Parametrized Construction of Generalized Product Codes





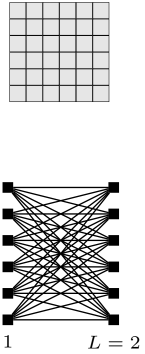
Parametrized Construction of Generalized Product Codes



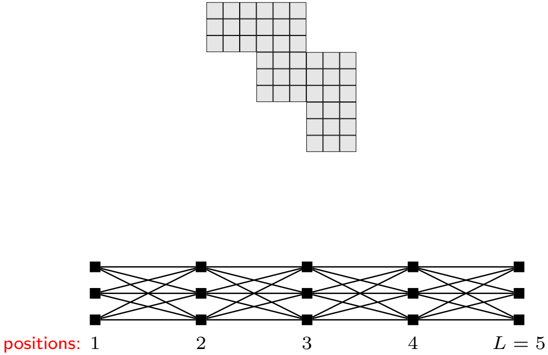
$\boldsymbol{\eta}$: symmetric $L \times L$ matrix that defines graph connectivity

Parametrized Construction of Generalized Product Codes

product codes



staircase codes

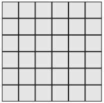


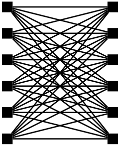
$$\boldsymbol{\eta} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

$\boldsymbol{\eta}$: symmetric $L \times L$ matrix that defines graph connectivity

Parametrized Construction of Generalized Product Codes

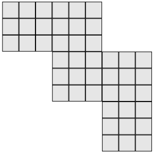
product codes

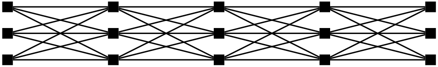




$1 \qquad L = 2$

staircase codes





positions: $1 \qquad 2 \qquad 3 \qquad 4 \qquad L = 5$

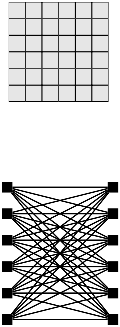
$$\boldsymbol{\eta} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

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

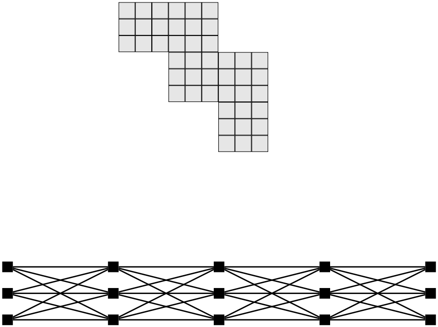
$\boldsymbol{\eta}$: symmetric $L \times L$ matrix that defines graph connectivity

Parametrized Construction of Generalized Product Codes

product codes

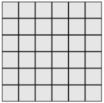


staircase codes

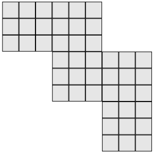


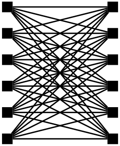
Parametrized Construction of Generalized Product Codes

product codes

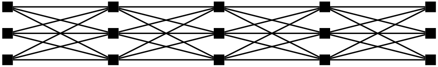


staircase codes



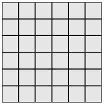


n: "problem size", proportional to the number of constraint nodes

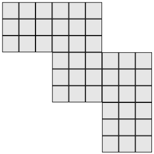


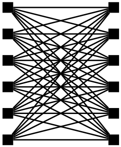
Parametrized Construction of Generalized Product Codes

product codes

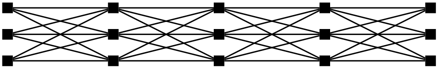


staircase codes





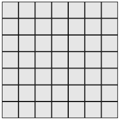
n: "problem size", proportional to the number of constraint nodes



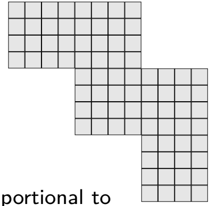
increasing n

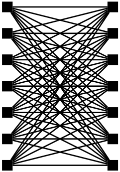
Parametrized Construction of Generalized Product Codes

product codes

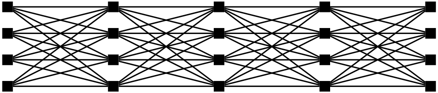


staircase codes



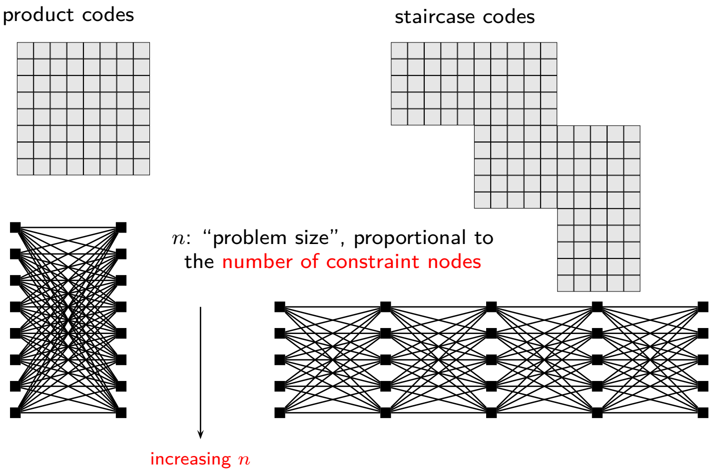


n: "problem size", proportional to the number of constraint nodes



increasing n

Parametrized Construction of Generalized Product Codes



Component Code Mixtures

Component Code Mixtures

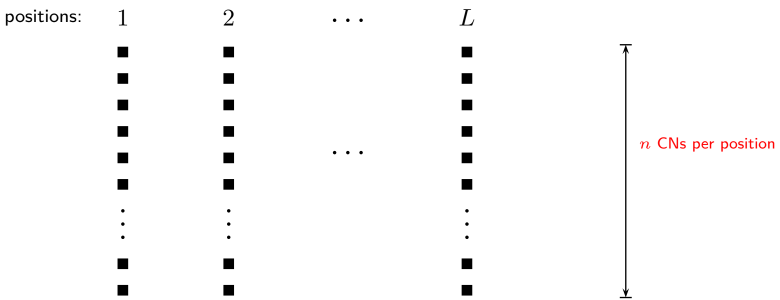
- For each position $i \in \{1, 2, \ldots, L\}$, let $\boldsymbol{\tau}(i) = (\tau_1(i), \tau_2(i), \ldots, \tau_{t_{\mathsf{max}}}(i))^\intercal$ be a probability vector/distribution ($\sum_t \tau_t(i) = 1$ and $\tau_t(i) \geq 0$ for all i)

Component Code Mixtures

- For each position $i \in \{1, 2, \ldots, L\}$, let $\boldsymbol{\tau}(i) = (\tau_1(i), \tau_2(i), \ldots, \tau_{t_{\mathsf{max}}}(i))^\intercal$ be a probability vector/distribution ($\sum_t \tau_t(i) = 1$ and $\tau_t(i) \geq 0$ for all i)
- $\tau_t(i)$: fraction of component codes at position i that correct t erasures

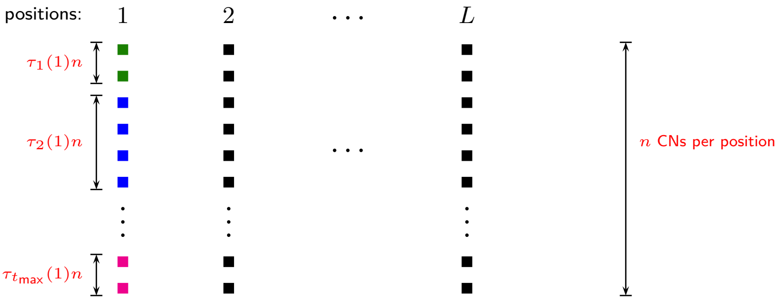
Component Code Mixtures

- For each position $i \in \{1, 2, \ldots, L\}$, let $\boldsymbol{\tau}(i) = (\tau_1(i), \tau_2(i), \ldots, \tau_{t_{\mathsf{max}}}(i))^\intercal$ be a probability vector/distribution ($\sum_t \tau_t(i) = 1$ and $\tau_t(i) \geq 0$ for all i)
- $\tau_t(i)$: fraction of component codes at position i that correct t erasures



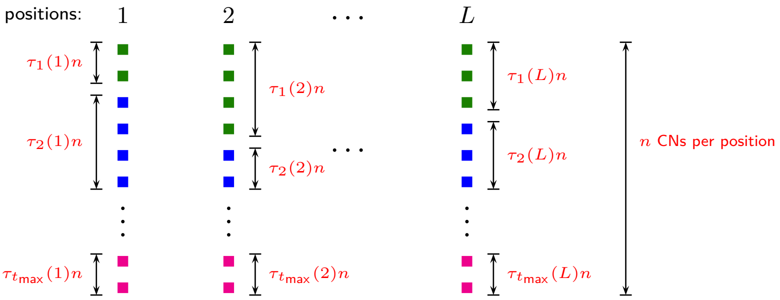
Component Code Mixtures

- For each position $i \in \{1, 2, \ldots, L\}$, let $\boldsymbol{\tau}(i) = (\tau_1(i), \tau_2(i), \ldots, \tau_{t_{\max}}(i))^\intercal$ be a probability vector/distribution ($\sum_t \tau_t(i) = 1$ and $\tau_t(i) \geq 0$ for all i)
- $\tau_t(i)$: fraction of component codes at position i that correct t erasures



Component Code Mixtures

- For each position $i \in \{1, 2, \ldots, L\}$, let $\boldsymbol{\tau}(i) = (\tau_1(i), \tau_2(i), \ldots, \tau_{t_{\mathsf{max}}}(i))^{\intercal}$ be a probability vector/distribution ($\sum_t \tau_t(i) = 1$ and $\tau_t(i) \geq 0$ for all i)
- $\tau_t(i)$: fraction of component codes at position i that correct t erasures



Component Code Mixtures

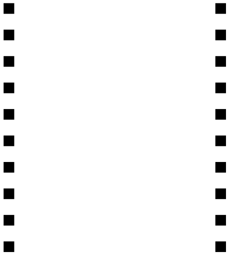
- For each position $i \in \{1, 2, \ldots, L\}$, let $\boldsymbol{\tau}(i) = (\tau_1(i), \tau_2(i), \ldots, \tau_{t_{\mathsf{max}}}(i))^\intercal$ be a probability vector/distribution ($\sum_t \tau_t(i) = 1$ and $\tau_t(i) \geq 0$ for all i)
- $\tau_t(i)$: fraction of component codes at position i that correct t erasures

Component Code Mixtures

- For each position $i \in \{1, 2, \ldots, L\}$, let $\boldsymbol{\tau}(i) = (\tau_1(i), \tau_2(i), \ldots, \tau_{t_{\mathsf{max}}}(i))^\intercal$ be a probability vector/distribution ($\sum_t \tau_t(i) = 1$ and $\tau_t(i) \geq 0$ for all i)
- $\tau_t(i)$: fraction of component codes at position i that correct t erasures
- Example: irregular product codes, where $\boldsymbol{\eta} = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$ (i.e., $L = 2$) and
  - $\tau_{t_1}(1) = 0.2$, $\tau_{t_2}(1) = 0.3$, $\tau_{t_3}(1) = 0.5$
  - $\tau_{t_4}(2) = 0.3$, $\tau_{t_5}(2) = 0.7$

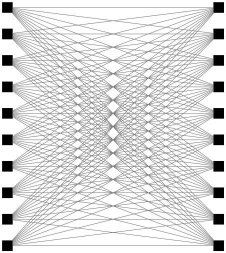
Component Code Mixtures

- For each position $i \in \{1, 2, \ldots, L\}$, let $\boldsymbol{\tau}(i) = (\tau_1(i), \tau_2(i), \ldots, \tau_{t_{\mathsf{max}}}(i))^\mathsf{T}$ be a probability vector/distribution ($\sum_t \tau_t(i) = 1$ and $\tau_t(i) \geq 0$ for all i)
- $\tau_t(i)$: fraction of component codes at position i that correct t erasures
- Example: irregular product codes, where $\boldsymbol{\eta} = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$ (i.e., $L = 2$) and
  - $\tau_{t_1}(1) = 0.2$, $\tau_{t_2}(1) = 0.3$, $\tau_{t_3}(1) = 0.5$
  - $\tau_{t_4}(2) = 0.3$, $\tau_{t_5}(2) = 0.7$



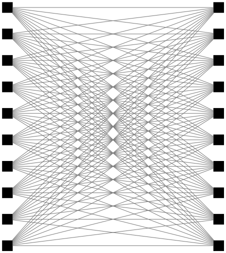
Component Code Mixtures

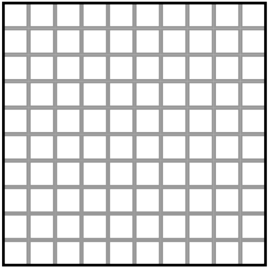
- For each position $i \in \{1, 2, \ldots, L\}$, let $\boldsymbol{\tau}(i) = (\tau_1(i), \tau_2(i), \ldots, \tau_{t_{\max}}(i))^\intercal$ be a probability vector/distribution ($\sum_t \tau_t(i) = 1$ and $\tau_t(i) \geq 0$ for all i)
- $\tau_t(i)$: fraction of component codes at position i that correct t erasures
- Example: irregular product codes, where $\boldsymbol{\eta} = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$ (i.e., $L = 2$) and
  - $\tau_{t_1}(1) = 0.2$, $\tau_{t_2}(1) = 0.3$, $\tau_{t_3}(1) = 0.5$
  - $\tau_{t_4}(2) = 0.3$, $\tau_{t_5}(2) = 0.7$



Component Code Mixtures

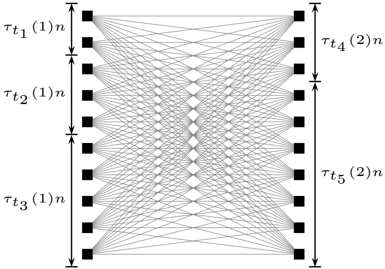
- For each position $i \in \{1, 2, \dots, L\}$, let $\boldsymbol{\tau}(i) = (\tau_1(i), \tau_2(i), \dots, \tau_{t_{\text{max}}}(i))^\intercal$ be a probability vector/distribution ($\sum_t \tau_t(i) = 1$ and $\tau_t(i) \geq 0$ for all i)
- $\tau_t(i)$: fraction of component codes at position i that correct t erasures
- Example: irregular product codes, where $\boldsymbol{\eta} = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$ (i.e., $L = 2$) and
  - $\tau_{t_1}(1) = 0.2$, $\tau_{t_2}(1) = 0.3$, $\tau_{t_3}(1) = 0.5$
  - $\tau_{t_4}(2) = 0.3$, $\tau_{t_5}(2) = 0.7$

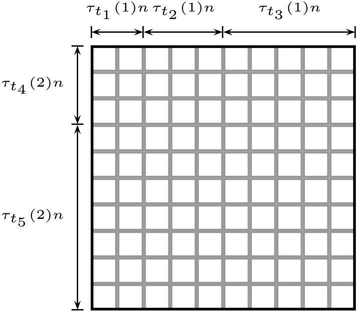




Component Code Mixtures

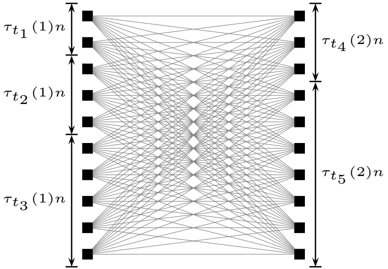
- For each position $i \in \{1, 2, \ldots, L\}$, let $\boldsymbol{\tau}(i) = (\tau_1(i), \tau_2(i), \ldots, \tau_{t_{\max}}(i))^\intercal$ be a probability vector/distribution ($\sum_t \tau_t(i) = 1$ and $\tau_t(i) \geq 0$ for all i)
- $\tau_t(i)$: fraction of component codes at position i that correct t erasures
- Example: irregular product codes, where $\boldsymbol{\eta} = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$ (i.e., $L = 2$) and
  - $\tau_{t_1}(1) = 0.2$, $\tau_{t_2}(1) = 0.3$, $\tau_{t_3}(1) = 0.5$
  - $\tau_{t_4}(2) = 0.3$, $\tau_{t_5}(2) = 0.7$

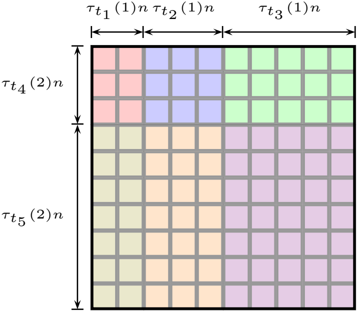




Component Code Mixtures

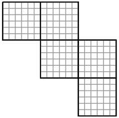
- For each position $i \in \{1, 2, \ldots, L\}$, let $\boldsymbol{\tau}(i) = (\tau_1(i), \tau_2(i), \ldots, \tau_{t_{\max}}(i))^\intercal$ be a probability vector/distribution ($\sum_t \tau_t(i) = 1$ and $\tau_t(i) \geq 0$ for all i)
- $\tau_t(i)$: fraction of component codes at position i that correct t erasures
- Example: irregular product codes, where $\boldsymbol{\eta} = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$ (i.e., $L = 2$) and
  - $\tau_{t_1}(1) = 0.2$, $\tau_{t_2}(1) = 0.3$, $\tau_{t_3}(1) = 0.5$
  - $\tau_{t_4}(2) = 0.3$, $\tau_{t_5}(2) = 0.7$

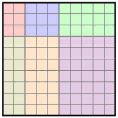


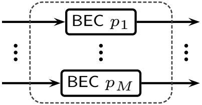


$K = 6$ distinct VN classes

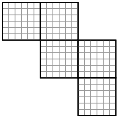
Bit Mapper

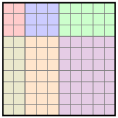


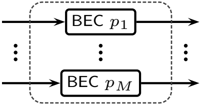




Bit Mapper

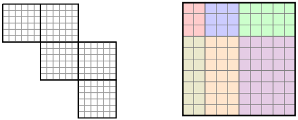


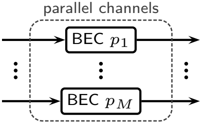




- Transmission over parallel BECs with erasure probabilities $p_1, \ldots, p_M$

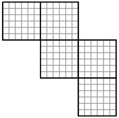
Bit Mapper

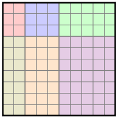


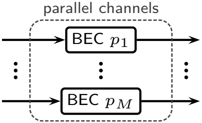


- Transmission over parallel BECs with erasure probabilities $p_1, \ldots, p_M$
- Bit mapper $\boldsymbol{A}$ determines the allocation of coded bits to channels

Bit Mapper

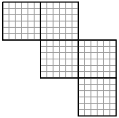


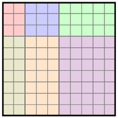


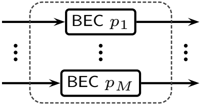


- Transmission over parallel BECs with erasure probabilities $p_1, \ldots, p_M$
- Bit mapper $\boldsymbol{A}$ determines the allocation of coded bits to channels
- $\boldsymbol{A}$: $K \times M$ matrix, where entries $a_{k,q}$ denote the fraction of bits from the k-th VN class that are allocated to the q-th BEC

Bit Mapper



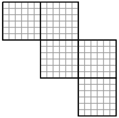


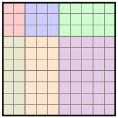


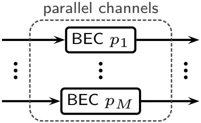
- Transmission over parallel BECs with erasure probabilities $p_1, \dots, p_M$
- Bit mapper $\boldsymbol{A}$ determines the allocation of coded bits to channels
- $\boldsymbol{A}$: $K \times M$ matrix, where entries $a_{k,q}$ denote the fraction of bits from the k-th VN class that are allocated to the q-th BEC
- Effectively, coded bits from different VN classes are transmitted through "virtual" BECs with erasure probability $\tilde{p}_k$, where

$$\tilde{p}_k = \sum_{q=1}^{M} a_{k,q} p_q$$

Bit Mapper







- Transmission over parallel BECs with erasure probabilities $p_1, \ldots, p_M$
- Bit mapper $\boldsymbol{A}$ determines the allocation of coded bits to channels
- $\boldsymbol{A}$: $K \times M$ matrix, where entries $a_{k,q}$ denote the fraction of bits from the k-th VN class that are allocated to the q-th BEC
- Effectively, coded bits from different VN classes are transmitted through "virtual" BECs with erasure probability $\tilde{p}_k$, where

$$\tilde{p}_k = \sum_{q=1}^{M} a_{k,q} p_q$$

- Example: baseline/uniform bit mapper, where $a_{k,q} = 1/M$ for all k, q
  $\implies$ all virtual BECs are the same

Density Evolution

Density Evolution

- What happens asymptotically for $n \to \infty$?

Density Evolution

- What happens asymptotically for $n \to \infty$?
- Let $p_k = c_k/n$ for $c_k > 0$ and $k \in \{1, 2, \ldots, M\}$, where c_k is the effective channel quality of the k-th BEC

Density Evolution

- What happens asymptotically for $n \to \infty$?
- Let $p_k = c_k/n$ for $c_k > 0$ and $k \in \{1, 2, \dots, M\}$, where c_k is the effective channel quality of the k-th BEC
- Bit mapper transforms these into effective channel qualities for the virtual BECs of each VN class: $\tilde{c}_{t,t'}(i,j) \leftrightarrow \tilde{c}_k = \sum_{q=1}^{M} a_{k,q} c_q$

Density Evolution

- What happens asymptotically for $n \to \infty$?
- Let $p_k = c_k/n$ for $c_k > 0$ and $k \in \{1, 2, \ldots, M\}$, where c_k is the effective channel quality of the k-th BEC
- Bit mapper transforms these into effective channel qualities for the virtual BECs of each VN class: $\tilde{c}_{t,t'}(i,j) \leftrightarrow \tilde{c}_k = \sum_{q=1}^{M} a_{k,q} c_q$
- One parameter $x_{i,t}^{(\ell)}$ is tracked per CN class : (asymptotic) probability that a randomly chosen erased bit corresponding to a t-erasure correcting component code at position i is not recovered

Density Evolution

- What happens asymptotically for $n \to \infty$?
- Let $p_k = c_k/n$ for $c_k > 0$ and $k \in \{1, 2, \ldots, M\}$, where c_k is the effective channel quality of the k-th BEC
- Bit mapper transforms these into effective channel qualities for the virtual BECs of each VN class: $\tilde{c}_{t,t'}(i,j) \leftrightarrow \tilde{c}_k = \sum_{q=1}^{M} a_{k,q} c_q$
- One parameter $x_{i,t}^{(\ell)}$ is tracked per CN class : (asymptotic) probability that a randomly chosen erased bit corresponding to a t-erasure correcting component code at position i is not recovered

$$x_{i,t}^{(\ell)} = \Psi_{\geq t}\left(\frac{1}{L}\sum_{j=1}^{L}\eta_{i,j}\sum_{t'=1}^{t_{\mathsf{max}}}\tilde{c}_{t,t'}(i,j)\tau_{t'}(j)x_{j,t'}^{(\ell-1)}\right)$$

Density Evolution

- What happens asymptotically for $n \to \infty$?
- Let $p_k = c_k/n$ for $c_k > 0$ and $k \in \{1, 2, \dots, M\}$, where c_k is the effective channel quality of the k-th BEC
- Bit mapper transforms these into effective channel qualities for the virtual BECs of each VN class: $\tilde{c}_{t,t'}(i,j) \leftrightarrow \tilde{c}_k = \sum_{q=1}^{M} a_{k,q} c_q$
- One parameter $x_{i,t}^{(\ell)}$ is tracked per CN class : (asymptotic) probability that a randomly chosen erased bit corresponding to a t-erasure correcting component code at position i is not recovered

initial condition
$x_{i,t}^{(0)} = 1$ for all i, t

$$x_{i,t}^{(\ell)} = \Psi_{\geq t}\left(\frac{1}{L}\sum_{j=1}^{L}\eta_{i,j}\sum_{t'=1}^{t_{\max}}\tilde{c}_{t,t'}(i,j)\tau_{t'}(j)x_{j,t'}^{(\ell-1)}\right)$$

Density Evolution

- What happens asymptotically for $n \to \infty$?
- Let $p_k = c_k / n$ for $c_k > 0$ and $k \in \{1, 2, \dots, M\}$, where c_k is the effective channel quality of the k-th BEC
- Bit mapper transforms these into effective channel qualities for the virtual BECs of each VN class: $\tilde{c}_{t,t'}(i,j) \leftrightarrow \tilde{c}_k = \sum_{q=1}^{M} a_{k,q} c_q$
- One parameter $x_{i,t}^{(\ell)}$ is tracked per CN class : (asymptotic) probability that a randomly chosen erased bit corresponding to a t-erasure correcting component code at position i is not recovered

initial condition
$x_{i,t}^{(0)} = 1$ for all i, t

$$x_{i,t}^{(\ell)} = \Psi_{\geq t}\left(\frac{1}{L} \sum_{j=1}^{L} \eta_{i,j} \sum_{t'=1}^{t_{\mathsf{max}}} \tilde{c}_{t,t'}(i,j) \tau_{t'}(j) x_{j,t'}^{(\ell-1)} \right)$$

code construction parameters

Density Evolution

- What happens asymptotically for $n \to \infty$?
- Let $p_k = c_k/n$ for $c_k > 0$ and $k \in \{1, 2, \dots, M\}$, where c_k is the effective channel quality of the k-th BEC
- Bit mapper transforms these into effective channel qualities for the virtual BECs of each VN class: $\tilde{c}_{t,t'}(i,j) \leftrightarrow \tilde{c}_k = \sum_{q=1}^{M} a_{k,q} c_q$
- One parameter $x_{i,t}^{(\ell)}$ is tracked per CN class : (asymptotic) probability that a randomly chosen erased bit corresponding to a t-erasure correcting component code at position i is not recovered

initial condition $x_{i,t}^{(0)} = 1$ for all i, t

effective channel qualities for VN classes

$$x_{i,t}^{(\ell)} = \Psi_{\geq t}\left(\frac{1}{L} \sum_{j=1}^{L} \eta_{i,j} \sum_{t'=1}^{t_{\text{max}}} \tilde{c}_{t,t'}(i,j) \tau_{t'}(j) x_{j,t'}^{(\ell-1)} \right)$$

code construction parameters

Density Evolution

- What happens asymptotically for $n \to \infty$?
- Let $p_k = c_k/n$ for $c_k > 0$ and $k \in \{1, 2, \ldots, M\}$, where c_k is the effective channel quality of the k-th BEC
- Bit mapper transforms these into effective channel qualities for the virtual BECs of each VN class: $\tilde{c}_{t,t'}(i,j) \leftrightarrow \tilde{c}_k = \sum_{q=1}^{M} a_{k,q} c_q$
- One parameter $x_{i,t}^{(\ell)}$ is tracked per CN class : (asymptotic) probability that a randomly chosen erased bit corresponding to a t-erasure correcting component code at position i is not recovered

effective channel qualities for VN classes

initial condition $x_{i,t}^{(0)} = 1$ for all i, t

$$x_{i,t}^{(\ell)} = \Psi_{\geq t}\left(\frac{1}{L}\sum_{j=1}^{L}\eta_{i,j}\sum_{t'=1}^{t_{\text{max}}}\tilde{c}_{t,t'}(i,j)\tau_{t'}(j)x_{j,t'}^{(\ell-1)}\right)$$

Poisson tail probability $\Psi_{\geq t}(x) \triangleq 1 - \sum_{i=0}^{t-1}\frac{x^i}{i!}e^{-x}$

code construction parameters

Decoding Thresholds

Decoding Thresholds

- Density evolution depends on effective channel qualities $\boldsymbol{c} = (c_1, \dots, c_M)$ of the parallel BECs

Decoding Thresholds

- Density evolution depends on effective channel qualities $\boldsymbol{c} = (c_1, \dots, c_M)$ of the parallel BECs
- $\boldsymbol{c}$ is admissable if $\lim_{\ell \to \infty} x_{i,t}^{(\ell)} = 0$ for all i, t (successful decoding with high probability)

Decoding Thresholds

- Density evolution depends on effective channel qualities $\boldsymbol{c} = (c_1, \dots, c_M)$ of the parallel BECs
- $\boldsymbol{c}$ is admissable if $\lim_{\ell \to \infty} x_{i,t}^{(\ell)} = 0$ for all i, t (successful decoding with high probability)
- For a given code and bit mapper, one may then define threshold regions

Decoding Thresholds

- Density evolution depends on effective channel qualities $\boldsymbol{c} = (c_1, \dots, c_M)$ of the parallel BECs
- $\boldsymbol{c}$ is admissable if $\lim_{\ell \to \infty} x_{i,t}^{(\ell)} = 0$ for all i, t (successful decoding with high probability)
- For a given code and bit mapper, one may then define threshold regions
- Simplification: effective channel qualities are linearly parameterized by a single parameter c (think "signal-to-noise ratio"), i.e., for $k \in \{1, \dots, M\}$

$$c_k = c b_k,$$

with fixed b_k

Decoding Thresholds

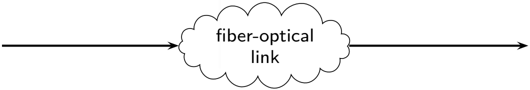
- Density evolution depends on effective channel qualities $\boldsymbol{c} = (c_1, \ldots, c_M)$ of the parallel BECs
- $\boldsymbol{c}$ is admissable if $\lim_{\ell\to\infty} x_{i,t}^{(\ell)} = 0$ for all i, t (successful decoding with high probability)
- For a given code and bit mapper, one may then define threshold regions
- Simplification: effective channel qualities are linearly parameterized by a single parameter c (think "signal-to-noise ratio"), i.e., for $k \in \{1, \ldots, M\}$

$$c_k = cb_k,$$

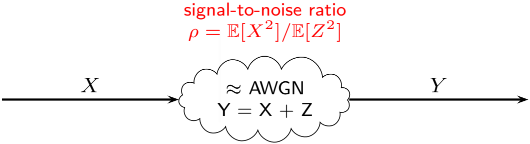
  with fixed b_k
- In this case, one may define decoding thresholds as usual:

$$\bar{c} = \sup\{c > 0 \,|\, \lim_{\ell\to\infty} x_{i,t}^{(\ell)} = 0 \text{ for all } i, t\}$$

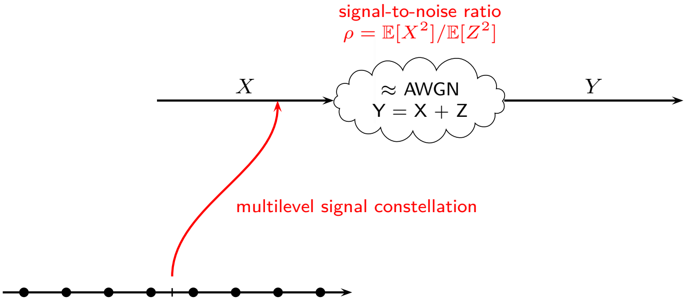
Spectrally-Efficient Communication



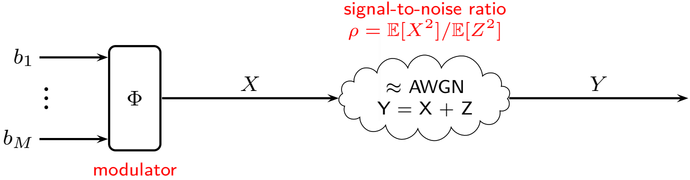
Spectrally-Efficient Communication

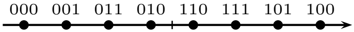


Spectrally-Efficient Communication

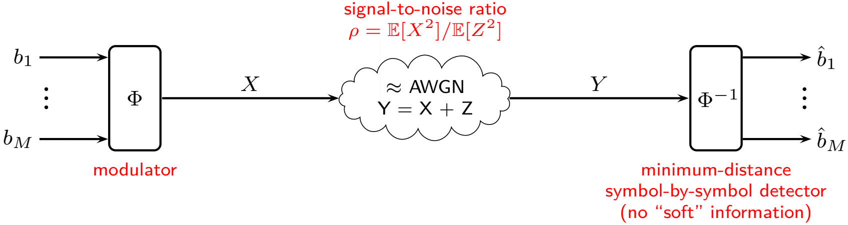


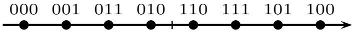
Spectrally-Efficient Communication



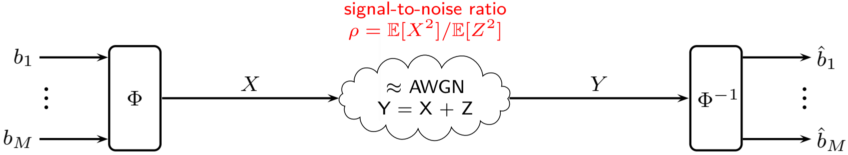


Spectrally-Efficient Communication

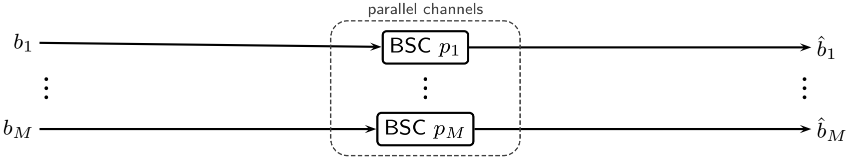




Spectrally-Efficient Communication

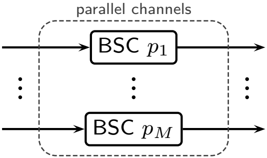


Spectrally-Efficient Communication



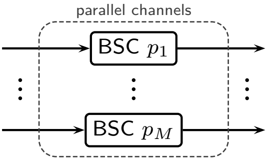
- Approximate setup: parallel binary symmetric channels (BSCs) with different crossover probabilities $p_1, \dots, p_M$

Spectrally-Efficient Communication



- Approximate setup: parallel binary symmetric channels (BSCs) with different crossover probabilities $p_1, \ldots, p_M$

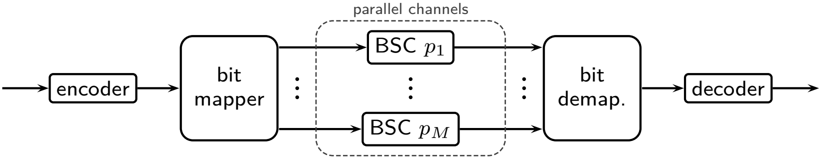
Spectrally-Efficient Communication



- Approximate setup: parallel binary symmetric channels (BSCs) with different crossover probabilities $p_1, \dots, p_M$
- Nearest-neighbor approximation: $p_k \approx b_k \bar{p}(\rho)$, $b_k = M 2^{k-1}/(2^M - 1)$,

$$\bar{p}(\rho) = \frac{2^M - 1}{M 2^{M-1}} Q\left(\sqrt{\frac{3\rho}{2^{2M} - 1}}\right)$$

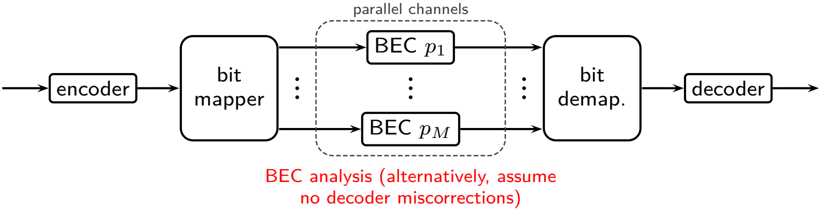
Spectrally-Efficient Communication



- Approximate setup: parallel binary symmetric channels (BSCs) with different crossover probabilities $p_1, \ldots, p_M$
- Nearest-neighbor approximation: $p_k \approx b_k \bar{p}(\rho)$, $b_k = M 2^{k-1}/(2^M - 1)$,

$$\bar{p}(\rho) = \frac{2^M - 1}{M 2^{M-1}} Q\left(\sqrt{\frac{3\rho}{2^{2M} - 1}}\right)$$

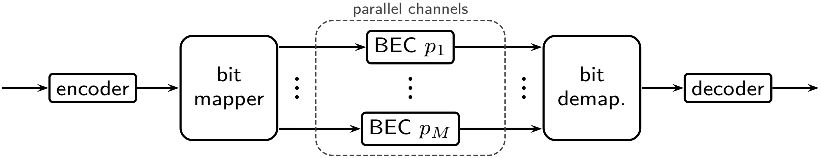
Spectrally-Efficient Communication



- Approximate setup: parallel binary symmetric channels (BSCs) with different crossover probabilities $p_1, \ldots, p_M$
- Nearest-neighbor approximation: $p_k \approx b_k \bar{p}(\rho)$, $b_k = M2^{k-1}/(2^M - 1)$,

$$\bar{p}(\rho) = \frac{2^M - 1}{M2^{M-1}} Q\left(\sqrt{\frac{3\rho}{2^{2M} - 1}}\right)$$

Spectrally-Efficient Communication



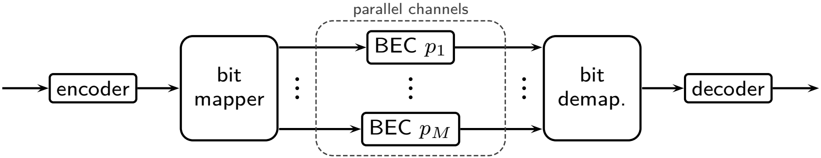
BEC analysis (alternatively, assume no decoder miscorrections)

- Approximate setup: parallel binary symmetric channels (BSCs) with different crossover probabilities $p_1, \dots, p_M$
- Nearest-neighbor approximation: $p_k \approx b_k \bar{p}(\rho)$, $b_k = M 2^{k-1}/(2^M - 1)$,

$$\bar{p}(\rho) = \frac{2^M - 1}{M 2^{M-1}} Q\left(\sqrt{\frac{3\rho}{2^{2M} - 1}}\right)$$

- Recall linear parametrization: $p_k = c b_k / n \implies \rho = \bar{p}^{-1}(c/n)$

Spectrally-Efficient Communication



BEC analysis (alternatively, assume no decoder miscorrections)

- Approximate setup: parallel binary symmetric channels (BSCs) with different crossover probabilities $p_1, \ldots, p_M$
- Nearest-neighbor approximation: $p_k \approx b_k \bar{p}(\rho)$, $b_k = M2^{k-1}/(2^M - 1)$,

$$\bar{p}(\rho) = \frac{2^M - 1}{M2^{M-1}} Q\left(\sqrt{\frac{3\rho}{2^{2M} - 1}}\right)$$

- Recall linear parametrization: $p_k = cb_k/n \implies \rho = \bar{p}^{-1}(c/n)$
- Example: threshold $\bar{c} = 10.63$ and $M = 4$. For $n = 1600$, waterfall region expected at $\rho = \bar{p}^{-1}(c/n) \approx 26.11\,\text{dB}$

Bit Mapper Optimization

Bit Mapper Optimization

Problem Formulation ([Richter et al., 2007], [Cheng et al., 2012], . . .)

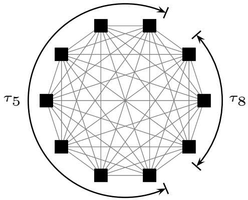
Optimize the bit mapper $\boldsymbol{A}$ for a given code and signal constellation

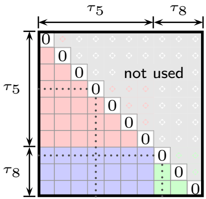
Bit Mapper Optimization

Problem Formulation ([Richter et al., 2007], [Cheng et al., 2012], ...)

Optimize the bit mapper $\boldsymbol{A}$ for a given code and signal constellation

- For illustration purposes, we consider irregular half-product codes, where $\boldsymbol{\eta} = 1$, $\tau_5 = 0.667$, $\tau_8 = 0.333 \implies K = 3$ VN classes



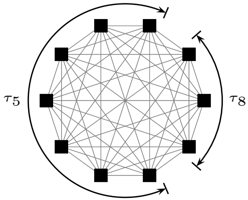


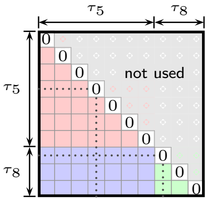
Bit Mapper Optimization

Problem Formulation ([Richter et al., 2007], [Cheng et al., 2012], . . .)

Optimize the bit mapper $\boldsymbol{A}$ for a given code and signal constellation

- For illustration purposes, we consider irregular half-product codes, where $\boldsymbol{\eta} = 1$, $\tau_5 = 0.667$, $\tau_8 = 0.333 \implies K = 3$ VN classes





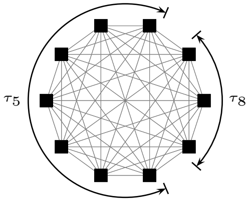
- Signal constellation: 16-PAM (256-QAM) $\implies M = 4$ distinct channels

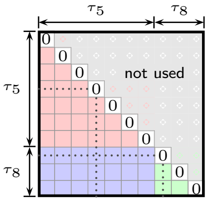
Bit Mapper Optimization

Problem Formulation ([Richter et al., 2007], [Cheng et al., 2012], . . .)

Optimize the bit mapper $\boldsymbol{A}$ for a given code and signal constellation

- For illustration purposes, we consider irregular half-product codes, where $\boldsymbol{\eta} = 1$, $\tau_5 = 0.667$, $\tau_8 = 0.333 \implies K = 3$ VN classes





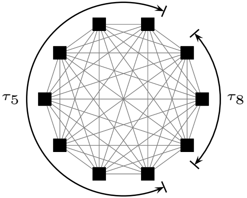
- Signal constellation: 16-PAM (256-QAM) $\implies M = 4$ distinct channels
- Heuristic threshold maximization via differential evolution algorithm

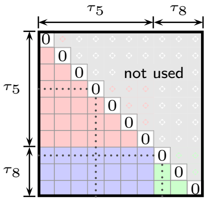
Bit Mapper Optimization

Problem Formulation ([Richter et al., 2007], [Cheng et al., 2012], . . .)

Optimize the bit mapper $\boldsymbol{A}$ for a given code and signal constellation

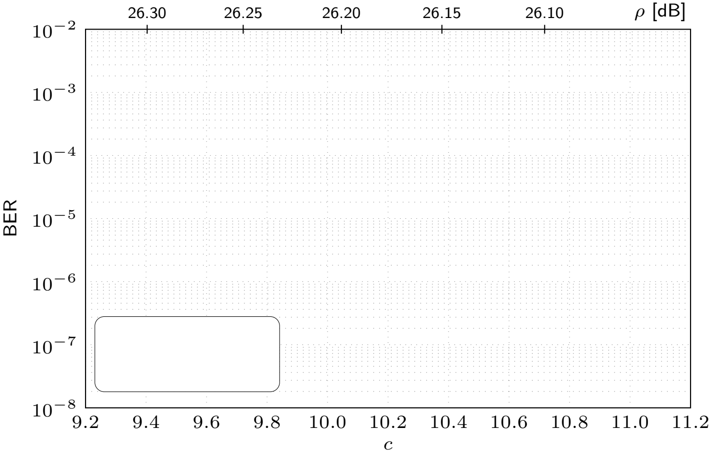
- For illustration purposes, we consider irregular half-product codes, where $\boldsymbol{\eta} = 1$, $\tau_5 = 0.667$, $\tau_8 = 0.333 \implies K = 3$ VN classes



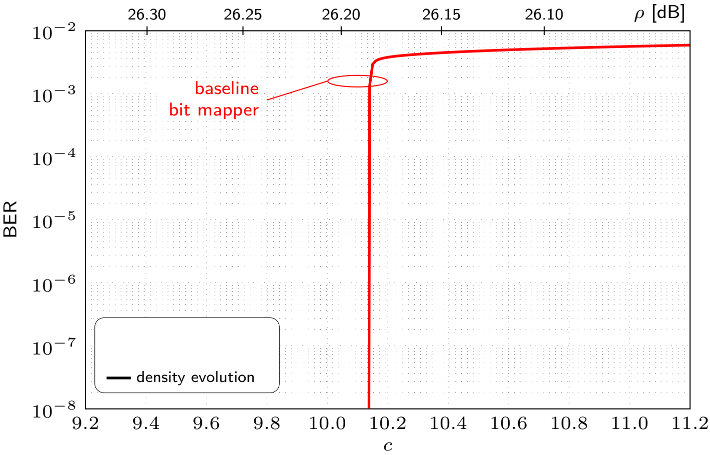


- Signal constellation: 16-PAM (256-QAM) $\implies M = 4$ distinct channels
- Heuristic threshold maximization via differential evolution algorithm
- Validation with $n = 1600$ and $\ell = 50$ iterations

Results

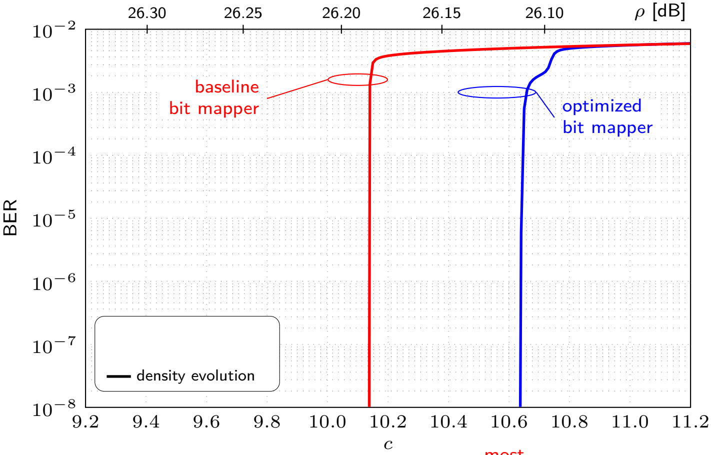

26.30
26.25
26.20
26.15
26.10
ρ [dB]
BER
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
10^{-8}
9.2
9.4
9.6
9.8
10.0
10.2
10.4
10.6
10.8
11.0
11.2
c

Results



$$\boldsymbol{A}_{\text{base}} = \begin{pmatrix} 0.3333 & 0.3333 & 0.3333 & 0.3333 \\ 0.3333 & 0.3333 & 0.3333 & 0.3333 \\ 0.3333 & 0.3333 & 0.3333 & 0.3333 \end{pmatrix}$$

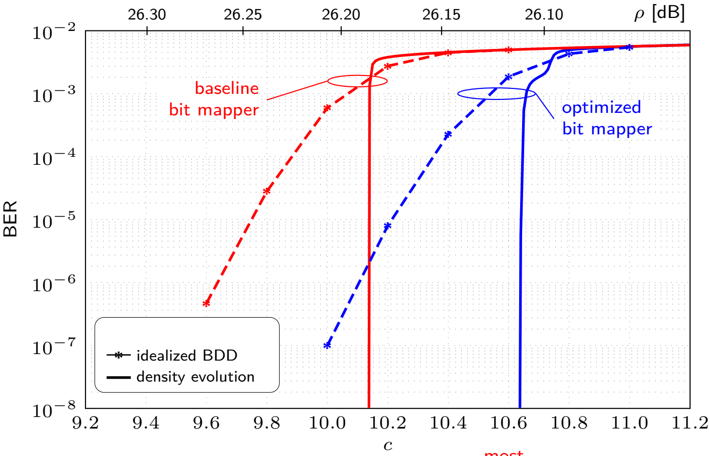
Results



$$\boldsymbol{A}_{\text{base}} = \begin{pmatrix} 0.3333 & 0.3333 & 0.3333 & 0.3333 \\ 0.3333 & 0.3333 & 0.3333 & 0.3333 \\ 0.3333 & 0.3333 & 0.3333 & 0.3333 \end{pmatrix}$$

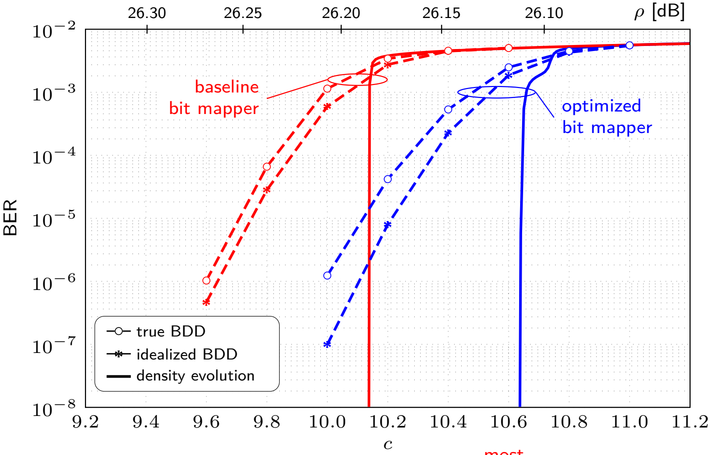
$$\begin{array}{cc} & \begin{array}{cccc} \text{most reliable} & & & \text{least reliable} \end{array} \\ \boldsymbol{A}^* = & \begin{pmatrix} 0.2640 & 0.1056 & 0.2984 & 0.3320 \\ 0.2976 & 0.4508 & 0.2453 & 0.0063 \\ 0.0031 & 0.0249 & 0.0746 & 0.8973 \end{pmatrix} \end{array}$$

Results



$$A_{\text{base}} = \begin{pmatrix} 0.3333 & 0.3333 & 0.3333 & 0.3333 \\ 0.3333 & 0.3333 & 0.3333 & 0.3333 \\ 0.3333 & 0.3333 & 0.3333 & 0.3333 \end{pmatrix} \qquad A^* = \begin{array}{c} \begin{array}{cccc} \text{most reliable} & & & \text{least reliable} \end{array} \\ \begin{pmatrix} 0.2640 & 0.1056 & 0.2984 & 0.3320 \\ 0.2976 & 0.4508 & 0.2453 & 0.0063 \\ 0.0031 & 0.0249 & 0.0746 & 0.8973 \end{pmatrix} \end{array}$$

Results



$$
\boldsymbol{A}_{\text{base}} = \begin{pmatrix} 0.3333 & 0.3333 & 0.3333 & 0.3333 \\ 0.3333 & 0.3333 & 0.3333 & 0.3333 \\ 0.3333 & 0.3333 & 0.3333 & 0.3333 \end{pmatrix}
$$

$$
\begin{array}{cccc} \text{most reliable} & & & \text{least reliable} \end{array}
$$

$$
\boldsymbol{A}^{*} = \begin{pmatrix} 0.2640 & 0.1056 & 0.2984 & 0.3320 \\ 0.2976 & 0.4508 & 0.2453 & 0.0063 \\ 0.0031 & 0.0249 & 0.0746 & 0.8973 \end{pmatrix}
$$

Conclusions

Conclusions

- Certain deterministic codes can be analyzed with density evolution over the BEC and over parallel BECs.

Conclusions

- Certain deterministic codes can be analyzed with density evolution over the BEC and over parallel BECs.
- Analysis can be used to predict the performance and optimize bit mappers in coded modulation systems with a hard-decision symbol detector.

Conclusions

- Certain deterministic codes can be analyzed with density evolution over the BEC and over parallel BECs.
- Analysis can be used to predict the performance and optimize bit mappers in coded modulation systems with a hard-decision symbol detector.
- Bit mapper optimization typically leads to moderate performance improvements, albeit at almost no increased system complexity cost.

Conclusions

- Certain deterministic codes can be analyzed with density evolution over the BEC and over parallel BECs.
- Analysis can be used to predict the performance and optimize bit mappers in coded modulation systems with a hard-decision symbol detector.
- Bit mapper optimization typically leads to moderate performance improvements, albeit at almost no increased system complexity cost.
- Future work should consider the joint design of the code and bit mapper.

Conclusions

- Certain deterministic codes can be analyzed with density evolution over the BEC and over parallel BECs.
- Analysis can be used to predict the performance and optimize bit mappers in coded modulation systems with a hard-decision symbol detector.
- Bit mapper optimization typically leads to moderate performance improvements, albeit at almost no increased system complexity cost.
- Future work should consider the joint design of the code and bit mapper.

Thank you!

FORCE
FIBER-OPTIC COMMUNICATIONS RESEARCH CENTER

References I

Cheng, T., Peng, K., Song, J., and Yan, K. (2012).
EXIT-aided bit mapping design for LDPC coded modulation with APSK constellations.
IEEE Commun. Lett., 16(6):777–780.

Elias, P. (1954).
Error-free coding.
IRE Trans. Inf. Theory, 4(4):29–37.

Häger, C., Pfister, H. D., Graell i Amat, A., and Brännström, F. (2015).
Density evolution for deterministic generalized product codes on the binary erasure channel.
submitted to IEEE Trans. Inf. Theory.

Jian, Y.-Y. (2013).
On The Analysis of Spatially-Coupled GLDPC Codes and the Weighted Min-Sum Algorithm.
PhD thesis, Texas A&M University.

Justesen, J., Larsen, K. J., and Pedersen, L. A. (2010).
Error correcting coding for OTN.
IEEE Commun. Mag., 59(9):70–75.

Richter, G., Hof, A., and Bossert, M. (2007).
On the mapping of low-density parity-check codes for bit-interleaved coded modulation.
In *Proc. IEEE Int. Symp. Information Theory (ISIT)*, Nice, Italy.

Smith, B. P., Farhood, A., Hunt, A., Kschischang, F. R., and Lodge, J. (2012).
Staircase codes: FEC for 100 Gb/s OTN.
J. Lightw. Technol., 30(1):110–117.