A Deterministic Construction and Density Evolution Analysis for Generalized Product Codes

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

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

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

2016 International Zurich Seminar on Communications
March 2, 2016

FORCE
FIBER-OPTIC COMMUNICATIONS
RESEARCH CENTER

AVANCEZ
1829
CHALMERS

Motivation

Motivation

- Error-correcting codes for fiber-optical communications: Generalized product codes with iterative bounded-distance decoding are appealing due to syndrome compression at high code rates (low complexity)

Motivation

- Error-correcting codes for fiber-optical communications: Generalized product codes with iterative bounded-distance decoding are appealing due to syndrome compression at high code rates (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 et al., 2013]

Motivation

- Error-correcting codes for fiber-optical communications: Generalized product codes with iterative bounded-distance decoding are appealing due to syndrome compression at high code rates (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 et al., 2013]
- However, asymptotic analysis is typically based on density evolution using an ensemble argument ([Jian et al., 2012] and [Zhang et al., 2015])

Motivation

- Error-correcting codes for fiber-optical communications: Generalized product codes with iterative bounded-distance decoding are appealing due to syndrome compression at high code rates (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 et al., 2013]
- However, asymptotic analysis is typically based on density evolution using an ensemble argument ([Jian et al., 2012] and [Zhang et al., 2015])
- Exception: asymptotic analysis of product codes in [Schwartz et al., 2005], [Justesen and Høholdt, 2007]

Motivation

- Error-correcting codes for fiber-optical communications: Generalized product codes with iterative bounded-distance decoding are appealing due to syndrome compression at high code rates (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 et al., 2013]
- However, asymptotic analysis is typically based on density evolution using an ensemble argument ([Jian et al., 2012] and [Zhang et al., 2015])
- Exception: asymptotic analysis of product codes in [Schwartz et al., 2005], [Justesen and Høholdt, 2007]

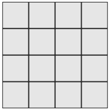
In This Talk ...

- Deterministic code construction that recovers product codes, staircase codes, and block-wise braided codes as special cases
- Rigorous density evolution analysis possible over the binary erasure channel
- Application: Spatially-coupled product codes and symmetric generalized product codes

Introduction: Product Codes and Staircase Codes

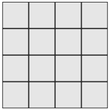
Introduction: Product Codes and Staircase Codes

rectangular array [Elias, 1954]



Introduction: Product Codes and Staircase Codes

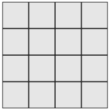
rectangular array [Elias, 1954]



each row/column is a codeword in some component code

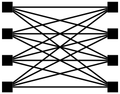
Introduction: Product Codes and Staircase Codes

rectangular array [Elias, 1954]



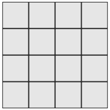
each row/column is a codeword in some component code

Tanner graph



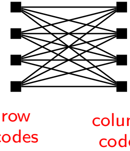
Introduction: Product Codes and Staircase Codes

rectangular array [Elias, 1954]



each row/column is a codeword in some component code

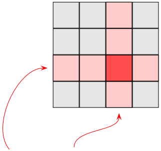
Tanner graph



constraint node (CN) degree = component code length

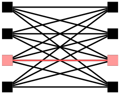
Introduction: Product Codes and Staircase Codes

rectangular array [Elias, 1954]



each row/column is a codeword in some component code

Tanner graph



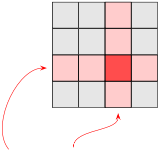
row codes

column codes

constraint node (CN) degree = component code length

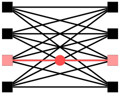
Introduction: Product Codes and Staircase Codes

rectangular array [Elias, 1954]



each row/column is a codeword in some component code

Tanner graph

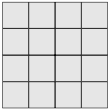


edge = degree-2 variable node (VN)

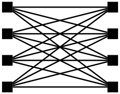
constraint node (CN) degree = component code length

Introduction: Product Codes and Staircase Codes

rectangular array [Elias, 1954]

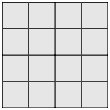


Tanner graph

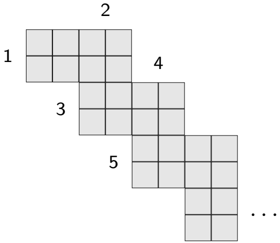


Introduction: Product Codes and Staircase Codes

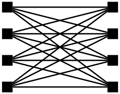
rectangular array [Elias, 1954]

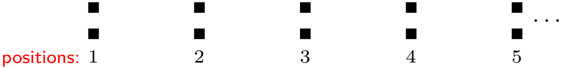


staircase array [Smith et al., 2012]



Tanner graph

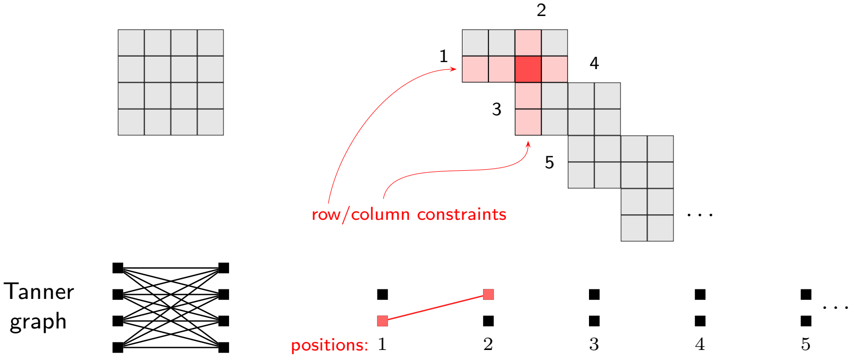




Introduction: Product Codes and Staircase Codes

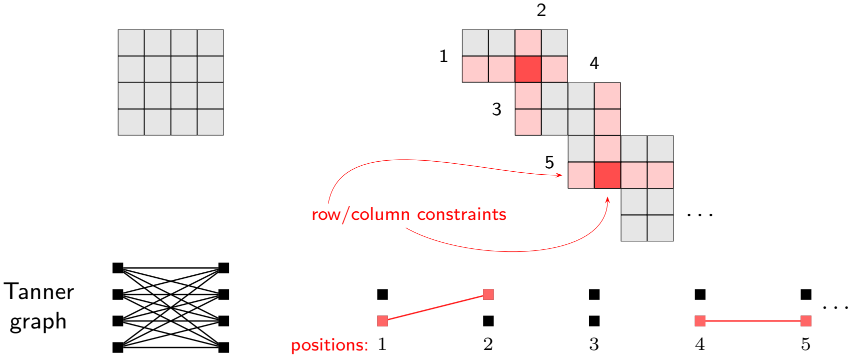
rectangular array [Elias, 1954]

staircase array [Smith et al., 2012]



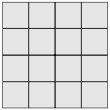
Introduction: Product Codes and Staircase Codes

rectangular array [Elias, 1954]   staircase array [Smith et al., 2012]

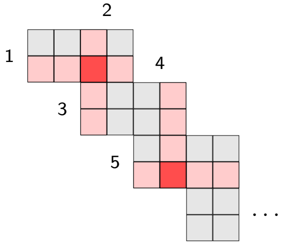


Introduction: Product Codes and Staircase Codes

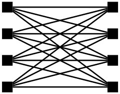
rectangular array [Elias, 1954]

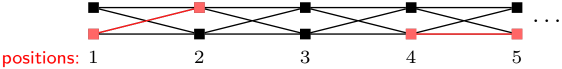


staircase array [Smith et al., 2012]



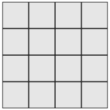
Tanner graph



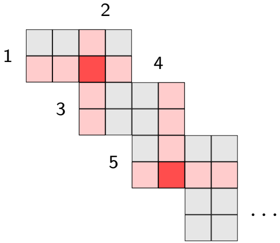


Introduction: Product Codes and Staircase Codes

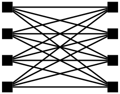
rectangular array [Elias, 1954]

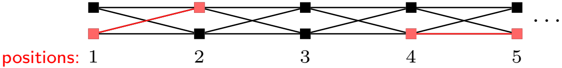


staircase array [Smith et al., 2012]



Tanner graph

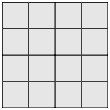


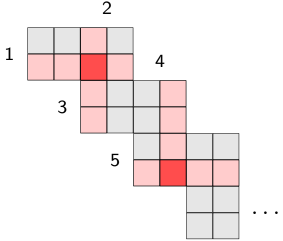


- Deterministic codes with fixed and structured Tanner graph

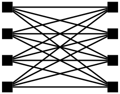
Introduction: Product Codes and Staircase Codes

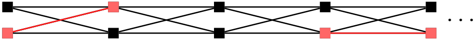
rectangular array [Elias, 1954]   staircase array [Smith et al., 2012]





Tanner graph





- Deterministic codes with fixed and structured Tanner graph
- Our code construction recovers these (and other) codes as special cases

Deterministic Construction for Generalized Product Codes

Deterministic Construction for Generalized Product Codes

- $\boldsymbol{\eta}$: binary symmetric $L \times L$ matrix (defines Tanner graph connectivity)

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

Deterministic Construction for Generalized Product Codes

- $\boldsymbol{\eta}$: binary symmetric $L \times L$ matrix (defines Tanner graph connectivity)
- L: number of positions (i.e., CN classes)

Example: $\boldsymbol{\eta} = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$, $L = 2$

Deterministic Construction for Generalized Product Codes

- $\boldsymbol{\eta}$: binary symmetric $L \times L$ matrix (defines Tanner graph connectivity)
- L: number of positions (i.e., CN classes)
- n: "problem size", proportional to the total number of CNs

Example: $\boldsymbol{\eta} = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$, $L = 2$

$n = 4$

Deterministic Construction for Generalized Product Codes

- $\boldsymbol{\eta}$: binary symmetric $L \times L$ matrix (defines Tanner graph connectivity)
- L: number of positions (i.e., CN classes)
- n: "problem size", proportional to the total number of CNs
- $d \triangleq \gamma n$: block size per spatial position, where γ is some scaling parameter

Example: $\boldsymbol{\eta} = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$, $L = 2$

$n = 4$

Deterministic Construction for Generalized Product Codes

- $\boldsymbol{\eta}$: binary symmetric $L \times L$ matrix (defines Tanner graph connectivity)
- L: number of positions (i.e., CN classes)
- n: "problem size", proportional to the total number of CNs
- $d \triangleq \gamma n$: block size per spatial position, where γ is some scaling parameter

Example: $\boldsymbol{\eta} = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$, $L = 2$, and $\gamma = 1$

$n = 4$

Deterministic Construction for Generalized Product Codes

- $\boldsymbol{\eta}$: binary symmetric $L \times L$ matrix (defines Tanner graph connectivity)
- L: number of positions (i.e., CN classes)
- n: "problem size", proportional to the total number of CNs
- $d \triangleq \gamma n$: block size per spatial position, where γ is some scaling parameter

Example: $\boldsymbol{\eta} = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$, $L = 2$, and $\gamma = 1$

$$n = 4 \implies d = 4$$

Deterministic Construction for Generalized Product Codes

- $\boldsymbol{\eta}$: binary symmetric $L \times L$ matrix (defines Tanner graph connectivity)
- L: number of positions (i.e., CN classes)
- n: "problem size", proportional to the total number of CNs
- $d \triangleq \gamma n$: block size per spatial position, where γ is some scaling parameter

Code Construction for $\mathcal{C}_n(\boldsymbol{\eta})$

Place d CNs at each position

Example: $\boldsymbol{\eta} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, $L = 2$, and $\gamma = 1$

$$n = 4 \implies d = 4$$

Deterministic Construction for Generalized Product Codes

- $\boldsymbol{\eta}$: binary symmetric $L \times L$ matrix (defines Tanner graph connectivity)
- L: number of positions (i.e., CN classes)
- n: "problem size", proportional to the total number of CNs
- $d \triangleq \gamma n$: block size per spatial position, where γ is some scaling parameter

Code Construction for $\mathcal{C}_n(\boldsymbol{\eta})$

Place d CNs at each position

Example: $\boldsymbol{\eta} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, $L = 2$, and $\gamma = 1$

positions: 1 2 $n = 4 \implies d = 4$

Deterministic Construction for Generalized Product Codes

- $\boldsymbol{\eta}$: binary symmetric $L \times L$ matrix (defines Tanner graph connectivity)
- L: number of positions (i.e., CN classes)
- n: "problem size", proportional to the total number of CNs
- $d \triangleq \gamma n$: block size per spatial position, where γ is some scaling parameter

Code Construction for $\mathcal{C}_n(\boldsymbol{\eta})$

Place d CNs at each position

Example: $\boldsymbol{\eta} = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$, $L = 2$, and $\gamma = 1$

positions: 1 2 $n = 4 \implies d = 4$

Deterministic Construction for Generalized Product Codes

- $\boldsymbol{\eta}$: binary symmetric $L \times L$ matrix (defines Tanner graph connectivity)
- L: number of positions (i.e., CN classes)
- n: "problem size", proportional to the total number of CNs
- $d \triangleq \gamma n$: block size per spatial position, where γ is some scaling parameter

Code Construction for $\mathcal{C}_n(\boldsymbol{\eta})$

Place d CNs at each position and connect each CN at position i to each CN at position j (through a VN) iff $\eta_{i,j} = 1$.

Example: $\boldsymbol{\eta} = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$, $L = 2$, and $\gamma = 1$

positions: 1 2 $n = 4 \implies d = 4$

Deterministic Construction for Generalized Product Codes

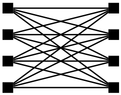
- $\boldsymbol{\eta}$: binary symmetric $L \times L$ matrix (defines Tanner graph connectivity)
- L: number of positions (i.e., CN classes)
- n: "problem size", proportional to the total number of CNs
- $d \triangleq \gamma n$: block size per spatial position, where γ is some scaling parameter

Code Construction for $\mathcal{C}_n(\boldsymbol{\eta})$

Place d CNs at each position and connect each CN at position i to each CN at position j (through a VN) iff $\eta_{i,j} = 1$.

Example: $\boldsymbol{\eta} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, $L = 2$, and $\gamma = 1$

positions: 1 2 $n = 4 \implies d = 4$



Deterministic Construction for Generalized Product Codes

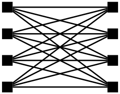
- $\boldsymbol{\eta}$: binary symmetric $L \times L$ matrix (defines Tanner graph connectivity)
- L: number of positions (i.e., CN classes)
- n: "problem size", proportional to the total number of CNs
- $d \triangleq \gamma n$: block size per spatial position, where γ is some scaling parameter

Code Construction for $\mathcal{C}_n(\boldsymbol{\eta})$

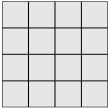
Place d CNs at each position and connect each CN at position i to each CN at position j (through a VN) iff $\eta_{i,j} = 1$.

Example: $\boldsymbol{\eta} = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$, $L = 2$, and $\gamma = 1$ gives a product code with $n \times n$ array.

positions: 1 2



$n = 4 \implies d = 4$



Deterministic Construction for Generalized Product Codes

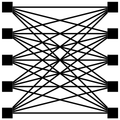
- $\boldsymbol{\eta}$: binary symmetric $L \times L$ matrix (defines Tanner graph connectivity)
- L: number of positions (i.e., CN classes)
- n: "problem size", proportional to the total number of CNs
- $d \triangleq \gamma n$: block size per spatial position, where γ is some scaling parameter

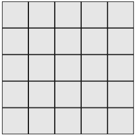
Code Construction for $\mathcal{C}_n(\boldsymbol{\eta})$

Place d CNs at each position and connect each CN at position i to each CN at position j (through a VN) iff $\eta_{i,j} = 1$.

Example: $\boldsymbol{\eta} = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$, $L = 2$, and $\gamma = 1$ gives a product code with $n \times n$ array.

positions: 1 2 $n = 5 \implies d = 5$





Deterministic Construction for Generalized Product Codes

- $\boldsymbol{\eta}$: binary symmetric $L \times L$ matrix (defines Tanner graph connectivity)
- L: number of positions (i.e., CN classes)
- n: "problem size", proportional to the total number of CNs
- $d \triangleq \gamma n$: block size per spatial position, where γ is some scaling parameter

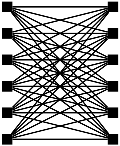
Code Construction for $\mathcal{C}_n(\boldsymbol{\eta})$

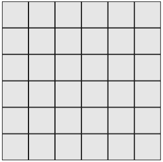
Place d CNs at each position and connect each CN at position i to each CN at position j (through a VN) iff $\eta_{i,j} = 1$.

Example: $\boldsymbol{\eta} = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$, $L = 2$, and $\gamma = 1$ gives a product code with $n \times n$ array.

positions: 1 2

$n = 6 \implies d = 6$





Deterministic Construction for Generalized Product Codes

- $\boldsymbol{\eta}$: binary symmetric $L \times L$ matrix (defines Tanner graph connectivity)
- L: number of positions (i.e., CN classes)
- n: "problem size", proportional to the total number of CNs
- $d \triangleq \gamma n$: block size per spatial position, where γ is some scaling parameter

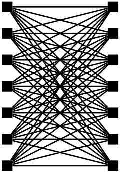
Code Construction for $\mathcal{C}_n(\boldsymbol{\eta})$

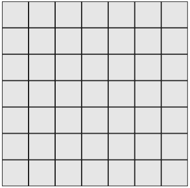
Place d CNs at each position and connect each CN at position i to each CN at position j (through a VN) iff $\eta_{i,j} = 1$.

Example: $\boldsymbol{\eta} = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$, $L = 2$, and $\gamma = 1$ gives a product code with $n \times n$ array.

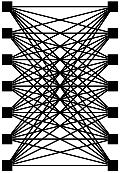
positions: 1 2

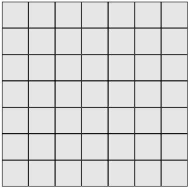
$n = 7 \implies d = 7$



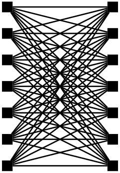


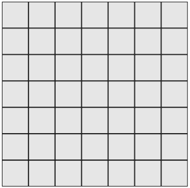
Channel Model and Decoding





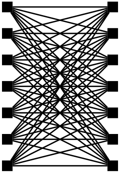
Channel Model and Decoding





- Each CN corresponds to t-erasure correcting component code

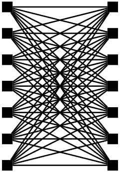
Channel Model and Decoding



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

- Each CN corresponds to t-erasure correcting component code

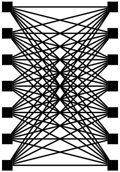
Channel Model and Decoding



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

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

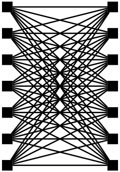
Channel Model and Decoding



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

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

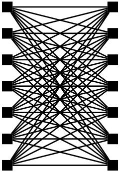
Channel Model and Decoding



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

- Each CN corresponds to t-erasure correcting component code
- Codeword transmission over binary erasure channel with erasure probability p
- ℓ iterations of bounded-distance decoding for all CNs:
  - If weight of an erasure pattern is $\leq t$, correct the pattern
  - If weight is $> t$, declare “failure” and do nothing (in that iteration)

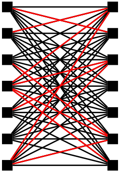
Channel Model and Decoding



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

- Each CN corresponds to t-erasure correcting component code
- Codeword transmission over binary erasure channel with erasure probability p
- ℓ iterations of bounded-distance decoding for all CNs:
  - If weight of an erasure pattern is $\leq t$, correct the pattern
  - If weight is $> t$, declare “failure” and do nothing (in that iteration)
- Residual graph: remove known variable nodes (i.e., edges)

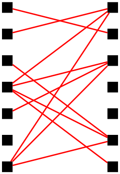
Channel Model and Decoding



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

- Each CN corresponds to t-erasure correcting component code
- Codeword transmission over binary erasure channel with erasure probability p
- ℓ iterations of bounded-distance decoding for all CNs:
  - If weight of an erasure pattern is $\leq t$, correct the pattern
  - If weight is $> t$, declare "failure" and do nothing (in that iteration)
- Residual graph: remove known variable nodes (i.e., edges)

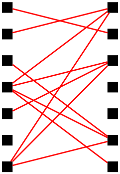
Channel Model and Decoding



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

- Each CN corresponds to t-erasure correcting component code
- Codeword transmission over binary erasure channel with erasure probability p
- ℓ iterations of bounded-distance decoding for all CNs:
  - If weight of an erasure pattern is $\leq t$, correct the pattern
  - If weight is $> t$, declare "failure" and do nothing (in that iteration)
- Residual graph: remove known variable nodes (i.e., edges)

Channel Model and Decoding

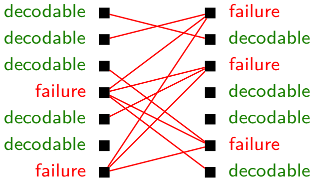


0	?	0	?	0	1	?
?	1	0	1	1	0	1
0	1	0	?	0	?	?
1	1	1	0	1	1	0
0	0	1	0	0	0	1
1	0	?	?	1	1	?
0	1	0	?	0	1	1

- Each CN corresponds to t-erasure correcting component code
- Codeword transmission over binary erasure channel with erasure probability p
- ℓ iterations of bounded-distance decoding for all CNs:
  - If weight of an erasure pattern is $\leq t$, correct the pattern
  - If weight is $> t$, declare “failure” and do nothing (in that iteration)
- Residual graph: remove known variable nodes (i.e., edges)
- Peeling of vertices with degree $\leq t$ (in parallel)

Channel Model and Decoding

1st iteration $(t = 2)$

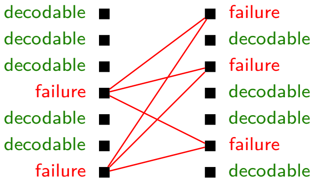


0	?	0	?	0	1	?
?	1	0	1	1	0	1
0	1	0	?	0	?	?
1	1	1	0	1	1	0
0	0	1	0	0	0	1
1	0	?	?	1	1	?
0	1	0	?	0	1	1

- Each CN corresponds to t-erasure correcting component code
- Codeword transmission over binary erasure channel with erasure probability p
- ℓ iterations of bounded-distance decoding for all CNs:
  - If weight of an erasure pattern is $\leq t$, correct the pattern
  - If weight is $> t$, declare "failure" and do nothing (in that iteration)
- Residual graph: remove known variable nodes (i.e., edges)
- Peeling of vertices with degree $\leq t$ (in parallel)

Channel Model and Decoding

1st iteration $(t = 2)$

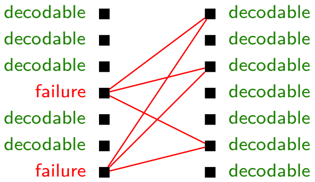


0	1	0	?	0	1	?
0	1	0	1	1	0	1
0	1	0	?	0	1	?
1	1	1	0	1	1	0
0	0	1	0	0	0	1
1	0	0	?	1	1	?
0	1	0	0	0	1	1

- Each CN corresponds to t-erasure correcting component code
- Codeword transmission over binary erasure channel with erasure probability p
- ℓ iterations of bounded-distance decoding for all CNs:
    - If weight of an erasure pattern is $\leq t$, correct the pattern
    - If weight is $> t$, declare “failure” and do nothing (in that iteration)
- Residual graph: remove known variable nodes (i.e., edges)
- Peeling of vertices with degree $\leq t$ (in parallel)

Channel Model and 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

- Each CN corresponds to t-erasure correcting component code
- Codeword transmission over binary erasure channel with erasure probability p
- ℓ iterations of bounded-distance decoding for all CNs:
  - If weight of an erasure pattern is $\leq t$, correct the pattern
  - If weight is $> t$, declare "failure" and do nothing (in that iteration)
- Residual graph: remove known variable nodes (i.e., edges)
- Peeling of vertices with degree $\leq t$ (in parallel)

Channel Model and Decoding

2nd iteration $(t = 2)$

decodable ■	■ decodable
decodable ■	■ decodable
decodable ■	■ decodable
failure ■	■ decodable
decodable ■	■ decodable
decodable ■	■ decodable
failure ■	■ decodable

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

- Each CN corresponds to t-erasure correcting component code
- Codeword transmission over binary erasure channel with erasure probability p
- ℓ iterations of bounded-distance decoding for all CNs:
  - If weight of an erasure pattern is $\leq t$, correct the pattern
  - If weight is $> t$, declare "failure" and do nothing (in that iteration)
- Residual graph: remove known variable nodes (i.e., edges)
- Peeling of vertices with degree $\leq t$ (in parallel)

Density Evolution

Density Evolution

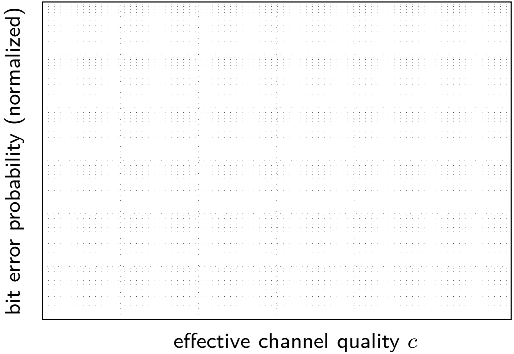
- What happens asymptotically for $n \to \infty$?

Density Evolution

- What happens asymptotically for $n \to \infty$?
- Let $p = c/n$ for $c > 0$, where c is the effective channel quality

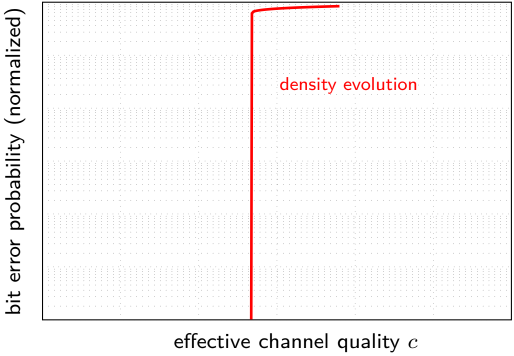
Density Evolution

- What happens asymptotically for $n \to \infty$?
- Let $p = c/n$ for $c > 0$, where c is the effective channel quality



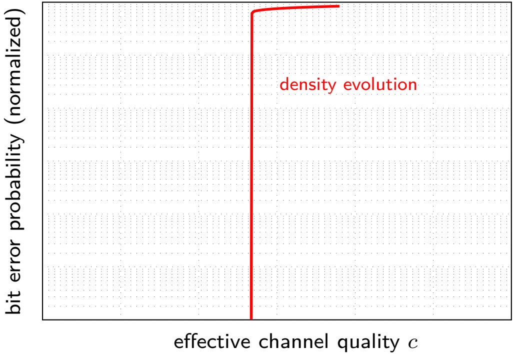
Density Evolution

- What happens asymptotically for $n \to \infty$?
- Let $p = c/n$ for $c > 0$, where c is the effective channel quality



Density Evolution

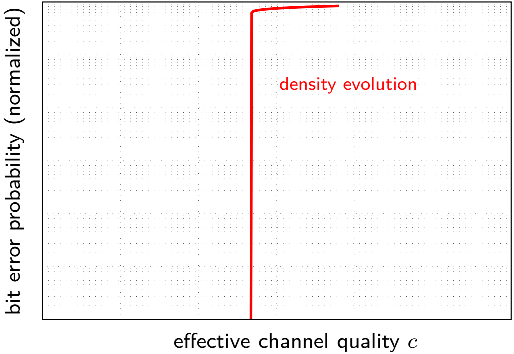
- What happens asymptotically for $n \to \infty$?
- Let $p = c/n$ for $c > 0$, where c is the effective channel quality



$$\boldsymbol{x}^{(\ell)} = \boldsymbol{\Psi}_{\geq t}(c\boldsymbol{B}\boldsymbol{x}^{(\ell-1)})$$

Density Evolution

- What happens asymptotically for $n \to \infty$?
- Let $p = c/n$ for $c > 0$, where c is the effective channel quality

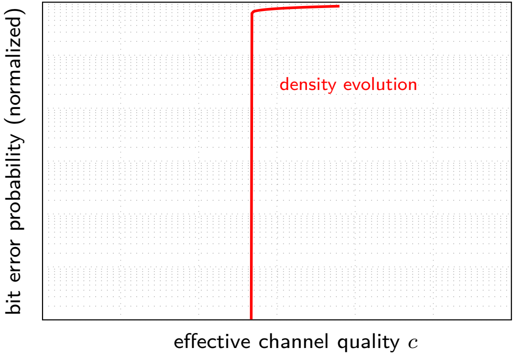


initial condition
$\boldsymbol{x}^{(0)} = (1, \dots, 1)$

$$\boldsymbol{x}^{(\ell)} = \boldsymbol{\Psi}_{\geq t}(c\boldsymbol{B}\boldsymbol{x}^{(\ell-1)})$$

Density Evolution

- What happens asymptotically for $n \to \infty$?
- Let $p = c/n$ for $c > 0$, where c is the effective channel quality



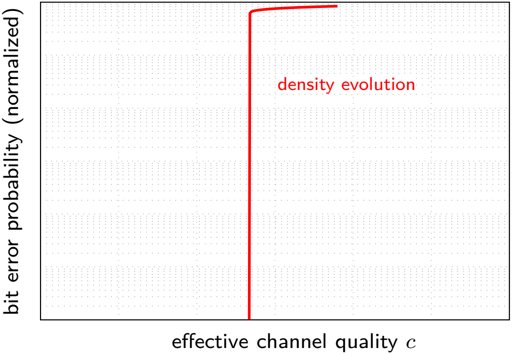
$B \triangleq \gamma \eta$

initial condition $x^{(0)} = (1, \dots, 1)$

$$x^{(\ell)} = \Psi_{\geq t}(c B x^{(\ell-1)})$$

Density Evolution

- What happens asymptotically for $n \to \infty$?
- Let $p = c/n$ for $c > 0$, where c is the effective channel quality



$B \triangleq \gamma \eta$

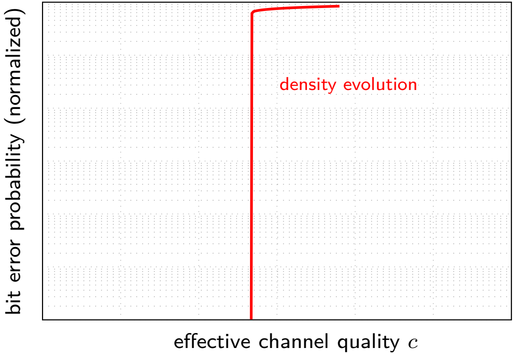
initial condition $x^{(0)} = (1, \dots, 1)$

$$x^{(\ell)} = \Psi_{\geq t}(cBx^{(\ell-1)})$$

element-wise application of $\Psi_{\geq t}(x) \triangleq 1 - \sum_{i=0}^{t-1} \frac{x^i}{i!} e^{-x}$

Density Evolution

- What happens asymptotically for $n \to \infty$?
- Let $p = c/n$ for $c > 0$, where c is the effective channel quality



$B \triangleq \gamma \eta$

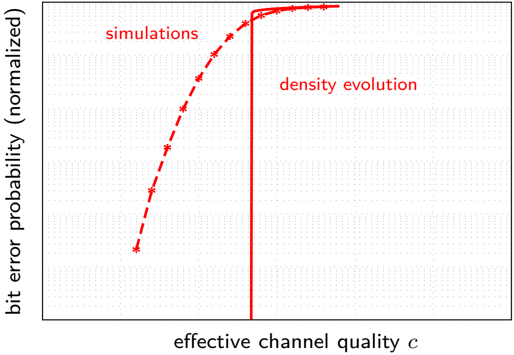
initial condition $x^{(0)} = (1, \dots, 1)$

$$x^{(\ell)} = \Psi_{\geq t}(cBx^{(\ell-1)})$$

element-wise application of $\Psi_{\geq t}(x) \triangleq 1 - \sum_{i=0}^{t-1} \frac{x^i}{i!} e^{-x}$

Density Evolution

- What happens asymptotically for $n \to \infty$?
- Let $p = c/n$ for $c > 0$, where c is the effective channel quality



$\boldsymbol{B} \triangleq \gamma \boldsymbol{\eta}$

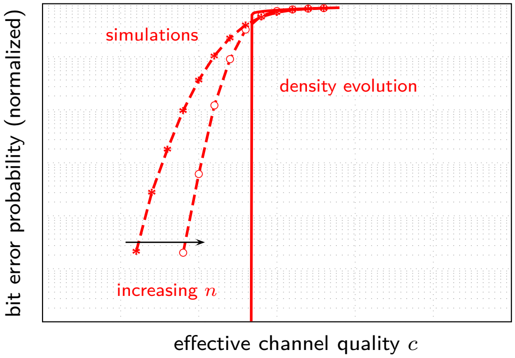
initial condition $\boldsymbol{x}^{(0)} = (1, \dots, 1)$

$$\boldsymbol{x}^{(\ell)} = \boldsymbol{\Psi}_{\geq t}(c\boldsymbol{B}\boldsymbol{x}^{(\ell-1)})$$

element-wise application of $\Psi_{\geq t}(x) \triangleq 1 - \sum_{i=0}^{t-1} \frac{x^i}{i!} e^{-x}$

Density Evolution

- What happens asymptotically for $n \to \infty$?
- Let $p = c/n$ for $c > 0$, where c is the effective channel quality



$B \triangleq \gamma \eta$

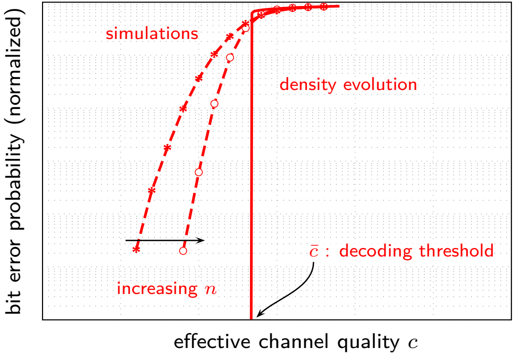
initial condition $x^{(0)} = (1, \dots, 1)$

$$x^{(\ell)} = \Psi_{\geq t}(cBx^{(\ell-1)})$$

element-wise application of $\Psi_{\geq t}(x) \triangleq 1 - \sum_{i=0}^{t-1} \frac{x^i}{i!} e^{-x}$

Density Evolution

- What happens asymptotically for $n \to \infty$?
- Let $p = c/n$ for $c > 0$, where c is the effective channel quality



$\boldsymbol{B} \triangleq \gamma \boldsymbol{\eta}$

initial condition $\boldsymbol{x}^{(0)} = (1, \dots, 1)$

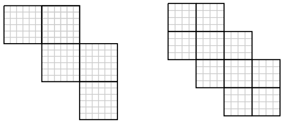
$$\boldsymbol{x}^{(\ell)} = \boldsymbol{\Psi}_{\geq t}(c\boldsymbol{B}\boldsymbol{x}^{(\ell-1)})$$

element-wise application of $\Psi_{\geq t}(x) \triangleq 1 - \sum_{i=0}^{t-1} \frac{x^i}{i!} e^{-x}$

Spatially-Coupled Product Codes

Spatially-Coupled Product Codes

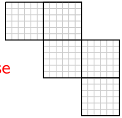
Deterministic

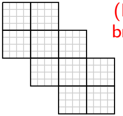


Spatially-Coupled Product Codes

Deterministic

staircase code



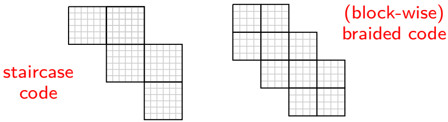


(block-wise) braided code

“convolutional-like” structure

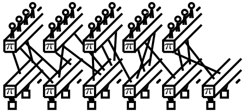
Spatially-Coupled Product Codes

Deterministic



"convolutional-like"
structure

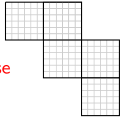
Ensemble-Based [Jian et al., 2012]

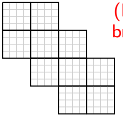


Spatially-Coupled Product Codes

Deterministic

staircase code

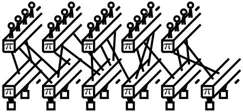




(block-wise) braided code

"convolutional-like" structure

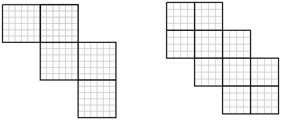
Ensemble-Based [Jian et al., 2012]



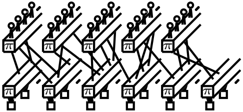
capacity-achieving at high rates over the binary symmetric channel

Spatially-Coupled Product Codes

Deterministic

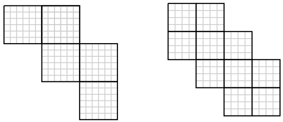


Ensemble-Based [Jian et al., 2012]



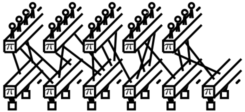
Spatially-Coupled Product Codes

Deterministic



$$\boldsymbol{x}^{(\ell)} = \boldsymbol{\Psi}_{\geq t}(c\boldsymbol{B}\boldsymbol{x}^{(\ell-1)})$$

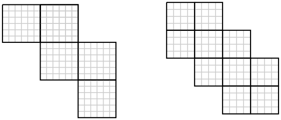
Ensemble-Based [Jian et al., 2012]



$$\boldsymbol{x}^{(\ell)} = \boldsymbol{\Psi}_{\geq t}(c\tilde{\boldsymbol{B}}\boldsymbol{x}^{(\ell-1)})$$

Spatially-Coupled Product Codes

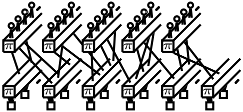
Deterministic



$$\boldsymbol{x}^{(\ell)} = \boldsymbol{\Psi}_{\geq t}(c\boldsymbol{B}\boldsymbol{x}^{(\ell-1)})$$

$$(\boldsymbol{B} = \gamma\boldsymbol{\eta})$$

Ensemble-Based [Jian et al., 2012]



$$\boldsymbol{x}^{(\ell)} = \boldsymbol{\Psi}_{\geq t}(c\tilde{\boldsymbol{B}}\boldsymbol{x}^{(\ell-1)})$$

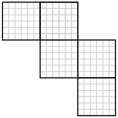
$$(\tilde{\boldsymbol{B}} = \boldsymbol{A}^{\intercal}\boldsymbol{A})$$

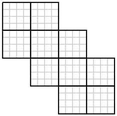
$$\boldsymbol{A} = \frac{1}{w}\begin{pmatrix} 1 & 1 & \cdots & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & \cdots & 1 & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & 1 & 1 & \cdots & 1 \end{pmatrix}$$

w (coupling width)

Spatially-Coupled Product Codes

Deterministic





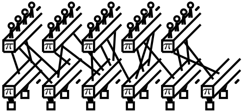
$$\boldsymbol{x}^{(\ell)} = \boldsymbol{\Psi}_{\geq t}(c\boldsymbol{B}\boldsymbol{x}^{(\ell-1)})$$

$(\boldsymbol{B} = \gamma\boldsymbol{\eta})$

$$\frac{1}{2}\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}, \quad \frac{1}{3}\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix},$$

staircase | braided (simplified)

Ensemble-Based [Jian et al., 2012]



$$\boldsymbol{x}^{(\ell)} = \boldsymbol{\Psi}_{\geq t}(c\tilde{\boldsymbol{B}}\boldsymbol{x}^{(\ell-1)})$$

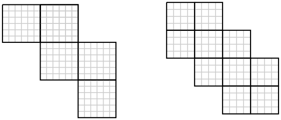
$(\tilde{\boldsymbol{B}} = \boldsymbol{A}^\intercal \boldsymbol{A})$

$$\frac{1}{4}\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 2 & 1 & 0 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 & 0 \\ 0 & 0 & 1 & 2 & 1 & 0 \\ 0 & 0 & 0 & 1 & 2 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}, \quad \frac{1}{9}\begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 2 & 2 & 1 & 0 & 0 \\ 1 & 2 & 3 & 2 & 1 & 0 \\ 0 & 1 & 2 & 3 & 2 & 1 \\ 0 & 0 & 1 & 2 & 2 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

$w = 2$ | $w = 3$

Spatially-Coupled Product Codes

Deterministic



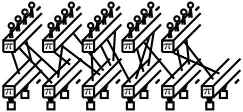
$$\boldsymbol{x}^{(\ell)} = \boldsymbol{\Psi}_{\geq t}(c\boldsymbol{B}\boldsymbol{x}^{(\ell-1)})$$

$(\boldsymbol{B} = \gamma\boldsymbol{\eta})$

$$\frac{1}{2}\begin{pmatrix} 0&1&0&0&0&0\\ 1&0&1&0&0&0\\ 0&1&0&1&0&0\\ 0&0&1&0&1&0\\ 0&0&0&1&0&1\\ 0&0&0&0&1&0 \end{pmatrix}, \quad \frac{1}{3}\begin{pmatrix} 1&1&0&0&0&0\\ 1&1&1&0&0&0\\ 0&1&1&1&0&0\\ 0&0&1&1&1&0\\ 0&0&0&1&1&1\\ 0&0&0&0&1&1 \end{pmatrix},$$

staircase — braided (simplified)

Ensemble-Based [Jian et al., 2012]



$$\boldsymbol{x}^{(\ell)} = \boldsymbol{\Psi}_{\geq t}(c\tilde{\boldsymbol{B}}\boldsymbol{x}^{(\ell-1)})$$

$(\tilde{\boldsymbol{B}} = \boldsymbol{A}^{\intercal}\boldsymbol{A})$

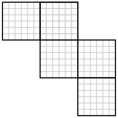
$$\frac{1}{4}\begin{pmatrix} 1&1&0&0&0&0\\ 1&2&1&0&0&0\\ 0&1&2&1&0&0\\ 0&0&1&2&1&0\\ 0&0&0&1&2&1\\ 0&0&0&0&1&1 \end{pmatrix}, \quad \frac{1}{9}\begin{pmatrix} 1&1&1&0&0&0\\ 1&2&2&1&0&0\\ 1&2&3&2&1&0\\ 0&1&2&3&2&1\\ 0&0&1&2&2&1\\ 0&0&0&1&1&1 \end{pmatrix}$$

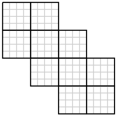
$w = 2$ — $w = 3$

- Equations have the same form, but different averaging matrices $\boldsymbol{B}$ and $\tilde{\boldsymbol{B}}$

Spatially-Coupled Product Codes

Deterministic





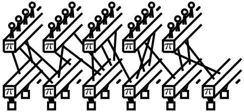
$$\boldsymbol{x}^{(\ell)} = \boldsymbol{\Psi}_{\geq t}(c\boldsymbol{B}\boldsymbol{x}^{(\ell-1)})$$

$$(\boldsymbol{B} = \gamma\boldsymbol{\eta})$$

$$\frac{1}{2}\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}, \quad \frac{1}{3}\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix},$$

staircase | braided (simplified)

Ensemble-Based [Jian et al., 2012]



$$\boldsymbol{x}^{(\ell)} = \boldsymbol{\Psi}_{\geq t}(c\tilde{\boldsymbol{B}}\boldsymbol{x}^{(\ell-1)})$$

$$(\tilde{\boldsymbol{B}} = \boldsymbol{A}^\intercal\boldsymbol{A})$$

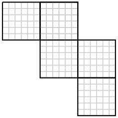
$$\frac{1}{4}\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 2 & 1 & 0 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 & 0 \\ 0 & 0 & 1 & 2 & 1 & 0 \\ 0 & 0 & 0 & 1 & 2 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}, \quad \frac{1}{9}\begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 2 & 2 & 1 & 0 & 0 \\ 1 & 2 & 3 & 2 & 1 & 0 \\ 0 & 1 & 2 & 3 & 2 & 1 \\ 0 & 0 & 1 & 2 & 2 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

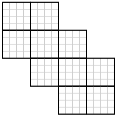
$w = 2$ | $w = 3$

- Equations have the same form, but different averaging matrices $\boldsymbol{B}$ and $\tilde{\boldsymbol{B}}$
- One can show that ensemble performance can be “emulated”

Spatially-Coupled Product Codes

Deterministic





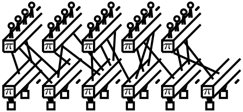
$$\boldsymbol{x}^{(\ell)} = \boldsymbol{\Psi}_{\geq t}(c\boldsymbol{B}\boldsymbol{x}^{(\ell-1)})$$

$(\boldsymbol{B} = \gamma\boldsymbol{\eta})$

$$\frac{1}{2}\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}, \quad \frac{1}{3}\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix},$$

staircase — braided (simplified)

Ensemble-Based [Jian et al., 2012]



$$\boldsymbol{x}^{(\ell)} = \boldsymbol{\Psi}_{\geq t}(c\tilde{\boldsymbol{B}}\boldsymbol{x}^{(\ell-1)})$$

$(\tilde{\boldsymbol{B}} = \boldsymbol{A}^{\intercal}\boldsymbol{A})$

$$\frac{1}{4}\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 2 & 1 & 0 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 & 0 \\ 0 & 0 & 1 & 2 & 1 & 0 \\ 0 & 0 & 0 & 1 & 2 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}, \quad \frac{1}{9}\begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 2 & 2 & 1 & 0 & 0 \\ 1 & 2 & 3 & 2 & 1 & 0 \\ 0 & 1 & 2 & 3 & 2 & 1 \\ 0 & 0 & 1 & 2 & 2 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

$w = 2$ — $w = 3$

- Equations have the same form, but different averaging matrices $\boldsymbol{B}$ and $\tilde{\boldsymbol{B}}$
- One can show that ensemble performance can be "emulated"
- $\implies$ ensemble threshold bounds in [Jian et al., 2012] apply to deterministic codes!

Symmetric Generalized Product Codes

Symmetric Generalized Product Codes

- So far, $\eta_{i,i} = 0$ for all $i \in \{1, 2, \ldots, L\}$. What about $\eta_{i,i} = 1$?

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

product code

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

staircase code

$$\begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}$$

(block-wise) braided code

Symmetric Generalized Product Codes

- So far, $\eta_{i,i} = 0$ for all $i \in \{1, 2, \dots, L\}$. What about $\eta_{i,i} = 1$?

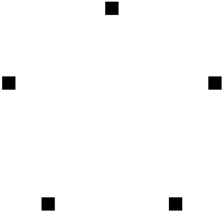
Example: $L = 1$, $\boldsymbol{\eta} = 1$, and $\gamma = 1$

Symmetric Generalized Product Codes

- So far, $\eta_{i,i} = 0$ for all $i \in \{1, 2, \ldots, L\}$. What about $\eta_{i,i} = 1$?

Example: $L = 1$, $\boldsymbol{\eta} = 1$, and $\gamma = 1$

$n = 5 \implies d = 5$

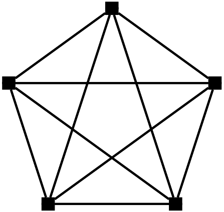


Symmetric Generalized Product Codes

- So far, $\eta_{i,i} = 0$ for all $i \in \{1, 2, \ldots, L\}$. What about $\eta_{i,i} = 1$?

Example: $L = 1$, $\boldsymbol{\eta} = 1$, and $\gamma = 1$

$n = 5 \implies d = 5$

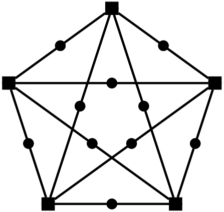


Symmetric Generalized Product Codes

- So far, $\eta_{i,i} = 0$ for all $i \in \{1, 2, \ldots, L\}$. What about $\eta_{i,i} = 1$?

Example: $L = 1$, $\boldsymbol{\eta} = 1$, and $\gamma = 1$

$n = 5 \implies d = 5$

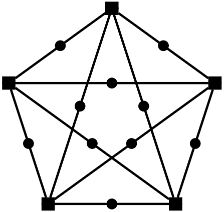


Symmetric Generalized Product Codes

- So far, $\eta_{i,i} = 0$ for all $i \in \{1, 2, \ldots, L\}$. What about $\eta_{i,i} = 1$?

Example: $L = 1$, $\boldsymbol{\eta} = 1$, and $\gamma = 1$

$n = 5 \implies d = 5$



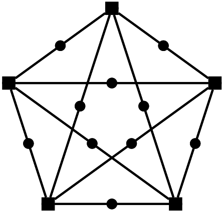
array representation?

Symmetric Generalized Product Codes

- So far, $\eta_{i,i} = 0$ for all $i \in \{1, 2, \ldots, L\}$. What about $\eta_{i,i} = 1$?

Example: $L = 1$, $\boldsymbol{\eta} = 1$, and $\gamma = 1$

$n = 5 \implies d = 5$



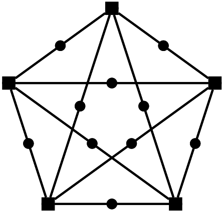
0	$*$	$*$	$*$	$*$
c_1	0	$*$	$*$	$*$
c_2	c_3	0	$*$	$*$
c_4	c_5	c_6	0	$*$
c_7	c_8	c_9	c_{10}	0

Symmetric Generalized Product Codes

- So far, $\eta_{i,i} = 0$ for all $i \in \{1, 2, \dots, L\}$. What about $\eta_{i,i} = 1$?

Example: $L = 1$, $\boldsymbol{\eta} = 1$, and $\gamma = 1$

$n = 5 \implies d = 5$



0	*	*	*	*
c_1	0	*	*	*
c_2	c_3	0	*	*
c_4	c_5	c_6	0	*
c_7	c_8	c_9	c_{10}	0

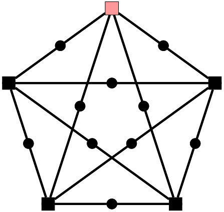
$*$ = not used

Symmetric Generalized Product Codes

- So far, $\eta_{i,i} = 0$ for all $i \in \{1, 2, \ldots, L\}$. What about $\eta_{i,i} = 1$?

Example: $L = 1$, $\boldsymbol{\eta} = 1$, and $\gamma = 1$

$n = 5 \implies d = 5$



0	*	*	*	*
c_1	0	*	*	*
c_2	c_3	0	*	*
c_4	c_5	c_6	0	*
c_7	c_8	c_9	c_{10}	0

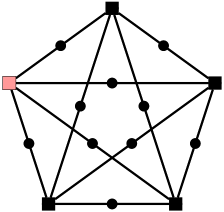
$*$ = not used

Symmetric Generalized Product Codes

- So far, $\eta_{i,i} = 0$ for all $i \in \{1, 2, \dots, L\}$. What about $\eta_{i,i} = 1$?

Example: $L = 1$, $\boldsymbol{\eta} = 1$, and $\gamma = 1$

$n = 5 \implies d = 5$



0	$*$	$*$	$*$	$*$
c_1	0	$*$	$*$	$*$
c_2	c_3	0	$*$	$*$
c_4	c_5	c_6	0	$*$
c_7	c_8	c_9	c_{10}	0

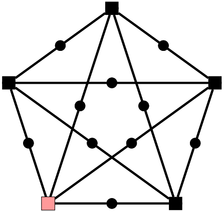
$*$ = not used

Symmetric Generalized Product Codes

- So far, $\eta_{i,i} = 0$ for all $i \in \{1, 2, \dots, L\}$. What about $\eta_{i,i} = 1$?

Example: $L = 1$, $\boldsymbol{\eta} = 1$, and $\gamma = 1$

$n = 5 \implies d = 5$



0	$*$	$*$	$*$	$*$
c_1	0	$*$	$*$	$*$
c_2	c_3	0	$*$	$*$
c_4	c_5	c_6	0	$*$
c_7	c_8	c_9	c_{10}	0

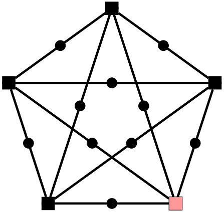
$*$ = not used

Symmetric Generalized Product Codes

- So far, $\eta_{i,i} = 0$ for all $i \in \{1, 2, \dots, L\}$. What about $\eta_{i,i} = 1$?

Example: $L = 1$, $\boldsymbol{\eta} = 1$, and $\gamma = 1$

$n = 5 \implies d = 5$



0	$*$	$*$	$*$	$*$
c_1	0	$*$	$*$	$*$
c_2	c_3	0	$*$	$*$
c_4	c_5	c_6	0	$*$
c_7	c_8	c_9	c_{10}	0

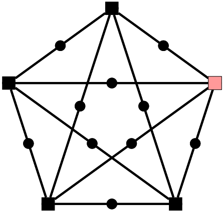
$*$ = not used

Symmetric Generalized Product Codes

- So far, $\eta_{i,i} = 0$ for all $i \in \{1, 2, \dots, L\}$. What about $\eta_{i,i} = 1$?

Example: $L = 1$, $\boldsymbol{\eta} = 1$, and $\gamma = 1$

$n = 5 \implies d = 5$



0	$*$	$*$	$*$	$*$
c_1	0	$*$	$*$	$*$
c_2	c_3	0	$*$	$*$
c_4	c_5	c_6	0	$*$
c_7	c_8	c_9	c_{10}	0

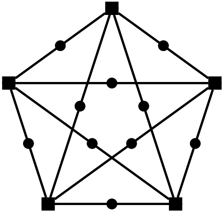
$*$ = not used

Symmetric Generalized Product Codes

- So far, $\eta_{i,i} = 0$ for all $i \in \{1, 2, \ldots, L\}$. What about $\eta_{i,i} = 1$?

Example: $L = 1$, $\boldsymbol{\eta} = 1$, and $\gamma = 1$

$n = 5 \implies d = 5$



0	*	*	*	*
c_1	0	*	*	*
c_2	c_3	0	*	*
c_4	c_5	c_6	0	*
c_7	c_8	c_9	c_{10}	0

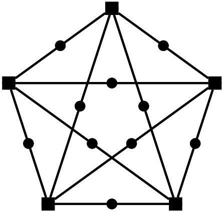
$*$ = not used

Symmetric Generalized Product Codes

- So far, $\eta_{i,i} = 0$ for all $i \in \{1, 2, \dots, L\}$. What about $\eta_{i,i} = 1$?

Example: $L = 1$, $\boldsymbol{\eta} = 1$, and $\gamma = 1$

$n = 5 \implies d = 5$



0	c_1	c_2	c_4	c_7
c_1	0	c_3	c_5	c_8
c_2	c_3	0	c_6	c_9
c_4	c_5	c_6	0	c_{10}
c_7	c_8	c_9	c_{10}	0

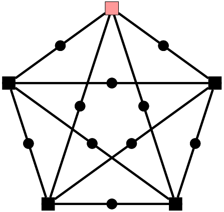
symmetric array

Symmetric Generalized Product Codes

- So far, $\eta_{i,i} = 0$ for all $i \in \{1, 2, \dots, L\}$. What about $\eta_{i,i} = 1$?

Example: $L = 1$, $\boldsymbol{\eta} = 1$, and $\gamma = 1$

$n = 5 \implies d = 5$



0	c_1	c_2	c_4	c_7
c_1	0	c_3	c_5	c_8
c_2	c_3	0	c_6	c_9
c_4	c_5	c_6	0	c_{10}
c_7	c_8	c_9	c_{10}	0

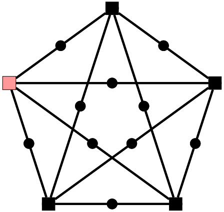
symmetric array

Symmetric Generalized Product Codes

- So far, $\eta_{i,i} = 0$ for all $i \in \{1, 2, \dots, L\}$. What about $\eta_{i,i} = 1$?

Example: $L = 1$, $\boldsymbol{\eta} = 1$, and $\gamma = 1$

$n = 5 \implies d = 5$



0	c_1	c_2	c_4	c_7
c_1	0	c_3	c_5	c_8
c_2	c_3	0	c_6	c_9
c_4	c_5	c_6	0	c_{10}
c_7	c_8	c_9	c_{10}	0

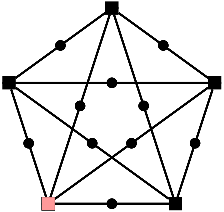
symmetric array

Symmetric Generalized Product Codes

- So far, $\eta_{i,i} = 0$ for all $i \in \{1, 2, \ldots, L\}$. What about $\eta_{i,i} = 1$?

Example: $L = 1$, $\boldsymbol{\eta} = 1$, and $\gamma = 1$

$n = 5 \implies d = 5$



0	c_1	c_2	c_4	c_7
c_1	0	c_3	c_5	c_8
c_2	c_3	0	c_6	c_9
c_4	c_5	c_6	0	c_{10}
c_7	c_8	c_9	c_{10}	0

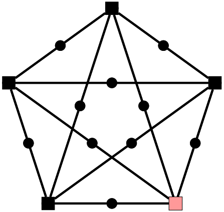
symmetric array

Symmetric Generalized Product Codes

- So far, $\eta_{i,i} = 0$ for all $i \in \{1, 2, \dots, L\}$. What about $\eta_{i,i} = 1$?

Example: $L = 1$, $\boldsymbol{\eta} = 1$, and $\gamma = 1$

$n = 5 \implies d = 5$



0	c_1	c_2	c_4	c_7
c_1	0	c_3	c_5	c_8
c_2	c_3	0	c_6	c_9
c_4	c_5	c_6	0	c_{10}
c_7	c_8	c_9	c_{10}	0

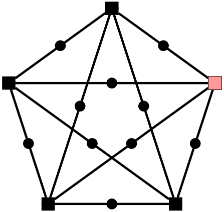
symmetric array

Symmetric Generalized Product Codes

- So far, $\eta_{i,i} = 0$ for all $i \in \{1, 2, \dots, L\}$. What about $\eta_{i,i} = 1$?

Example: $L = 1$, $\boldsymbol{\eta} = 1$, and $\gamma = 1$

$n = 5 \implies d = 5$



0	c_1	c_2	c_4	c_7
c_1	0	c_3	c_5	c_8
c_2	c_3	0	c_6	c_9
c_4	c_5	c_6	0	c_{10}
c_7	c_8	c_9	c_{10}	0

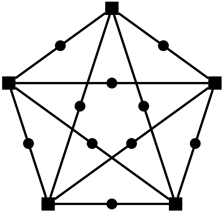
symmetric array

Symmetric Generalized Product Codes

- So far, $\eta_{i,i} = 0$ for all $i \in \{1, 2, \dots, L\}$. What about $\eta_{i,i} = 1$?

Example: $L = 1$, $\boldsymbol{\eta} = 1$, and $\gamma = 1$

$n = 5 \implies d = 5$



0	c_1	c_2	c_4	c_7
c_1	0	c_3	c_5	c_8
c_2	c_3	0	c_6	c_9
c_4	c_5	c_6	0	c_{10}
c_7	c_8	c_9	c_{10}	0

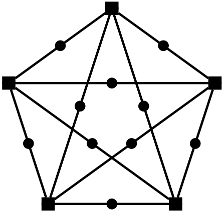
symmetric array

Symmetric Generalized Product Codes

- So far, $\eta_{i,i} = 0$ for all $i \in \{1, 2, \ldots, L\}$. What about $\eta_{i,i} = 1$?

Example: $L = 1$, $\boldsymbol{\eta} = 1$, and $\gamma = 1$ gives a half-product code [Justesen, 2011]

$n = 5 \implies d = 5$



0	c_1	c_2	c_4	c_7
c_1	0	c_3	c_5	c_8
c_2	c_3	0	c_6	c_9
c_4	c_5	c_6	0	c_{10}
c_7	c_8	c_9	c_{10}	0

symmetric array

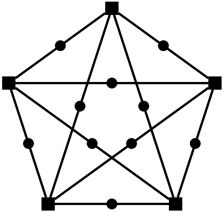
Symmetric Generalized Product Codes

- So far, $\eta_{i,i} = 0$ for all $i \in \{1, 2, \dots, L\}$. What about $\eta_{i,i} = 1$?

Example: $L = 1$, $\boldsymbol{\eta} = 1$, and $\gamma = 1$ gives a half-product code [Justesen, 2011]

$n = 5 \implies d = 5$

Graph appears already in [Tanner, 1981]



0	c_1	c_2	c_4	c_7
c_1	0	c_3	c_5	c_8
c_2	c_3	0	c_6	c_9
c_4	c_5	c_6	0	c_{10}
c_7	c_8	c_9	c_{10}	0

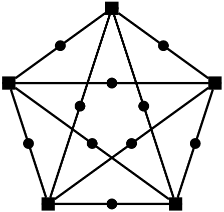
symmetric array

Symmetric Generalized Product Codes

- So far, $\eta_{i,i} = 0$ for all $i \in \{1, 2, \ldots, L\}$. What about $\eta_{i,i} = 1$?

Example: $L = 1$, $\boldsymbol{\eta} = 1$, and $\gamma = 1$ gives a half-product code [Justesen, 2011]

$n = 5 \implies d = 5$



Graph appears already in [Tanner, 1981]

0	c_1	c_2	c_4	c_7
c_1	0	c_3	c_5	c_8
c_2	c_3	0	c_6	c_9
c_4	c_5	c_6	0	c_{10}
c_7	c_8	c_9	c_{10}	0

symmetric array

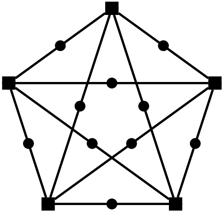
- A half-product code has the same threshold as a product code, but less than half the block length

Symmetric Generalized Product Codes

- So far, $\eta_{i,i} = 0$ for all $i \in \{1, 2, \ldots, L\}$. What about $\eta_{i,i} = 1$?

Example: $L = 1$, $\boldsymbol{\eta} = 1$, and $\gamma = 1$ gives a half-product code [Justesen, 2011]

$n = 5 \implies d = 5$

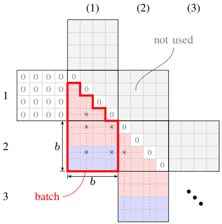


Graph appears already in [Tanner, 1981]

0	c_1	c_2	c_4	c_7
c_1	0	c_3	c_5	c_8
c_2	c_3	0	c_6	c_9
c_4	c_5	c_6	0	c_{10}
c_7	c_8	c_9	c_{10}	0

symmetric array

- A half-product code has the same threshold as a product code, but less than half the block length
- Half-braided codes can outperform staircase and braided codes in the waterfall region, at a lower error floor and decoding delay [Häger et al., 2016]



Conclusions and Future Work

Conclusions and Future Work

- Density evolution can be applied to a large class of deterministic generalized product codes.

Conclusions and Future Work

- Density evolution can be applied to a large class of deterministic generalized product codes.
- There exists a family of (deterministic) codes that performs asymptotically equivalent to a previously studied spatially-coupled code ensemble.

Conclusions and Future Work

- Density evolution can be applied to a large class of deterministic generalized product codes.
- There exists a family of (deterministic) codes that performs asymptotically equivalent to a previously studied spatially-coupled code ensemble.
- Symmetric generalized product codes can outperform their nonsymmetric counterparts.

Conclusions and Future Work

- Density evolution can be applied to a large class of deterministic generalized product codes.
- There exists a family of (deterministic) codes that performs asymptotically equivalent to a previously studied spatially-coupled code ensemble.
- Symmetric generalized product codes can outperform their nonsymmetric counterparts.

Thank you!

FORCE
FIBER-OPTIC COMMUNICATIONS RESEARCH CENTER

References

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. (2016).
Density evolution and error floor analysis of staircase and braided codes.
In *Proc. Optical Fiber Communication Conf. (OFC)*, Anaheim, CA.

Jian, Y.-Y., Pfister, H. D., and Narayanan, K. R. (2012).
Approaching capacity at high rates with iterative hard-decision decoding.
In *Proc. IEEE Int. Symp. Information Theory (ISIT)*, Cambridge, MA.

Jian, Y.-Y., Pfister, H. D., Narayanan, K. R., Rao, R., and Mazahreh, R. (2013).
Iterative hard-decision decoding of braided BCH codes for high-speed optical communication.
In *Proc. IEEE Glob. Communication Conf. (GLOBECOM)*, Atlanta, GA.

Justesen, J. (2011).
Performance of product codes and related structures with iterated decoding.
IEEE Trans. Commun., 59(2):407–415.

Justesen, J. and Høholdt, T. (2007).
Analysis of iterated hard decision decoding of product codes with Reed-Solomon component codes.
In *Proc. IEEE Information Theory Workshop (ITW)*, Tahoe City, CA.

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