Optimized Bit Mappings for Spatially Coupled LDPC Codes over Parallel Binary Erasure Channels

Christian Häger[1] Alexandre Graell i Amat[1] Alex Alvarado[2]
Fredrik Brännström[1] Erik Agrell[1]

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

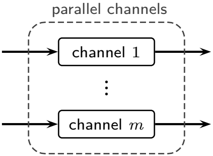
[2]Department of Enginering, University of Cambridge, UK

{christian.haeger, alexandre.graell, fredrik.brannstrom, agrell}@chalmers.se, alex.alvarado@ieee.org

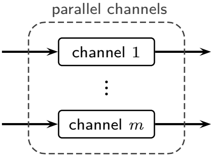
7th IEEE WS AIPWCS, Aalborg, November 14–15, 2013

CHALMERS

Motivation

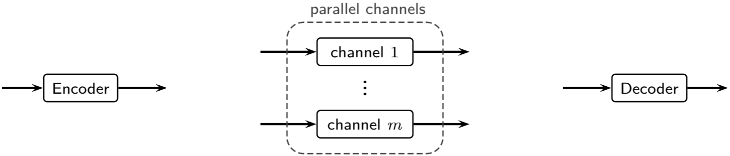


Motivation



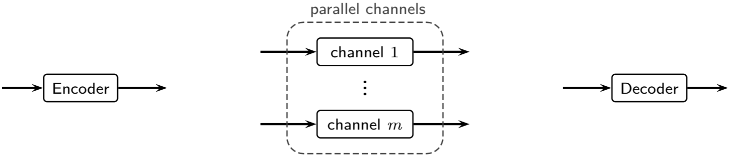
- Parallel channels arise in many practical scenarios: multicarrier transmission, rate-compatible puncturing of turbo-like codes, bit-interleaved coded modulation (BICM), . . .

Motivation



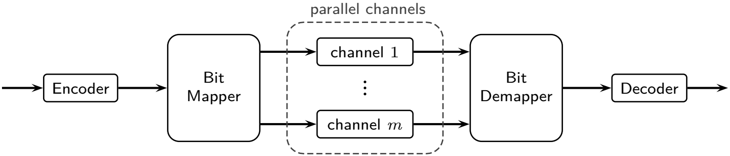
- Parallel channels arise in many practical scenarios: multicarrier transmission, rate-compatible puncturing of turbo-like codes, bit-interleaved coded modulation (BICM), . . .
- One binary encoder/decoder pair is often used for simplicity

Motivation



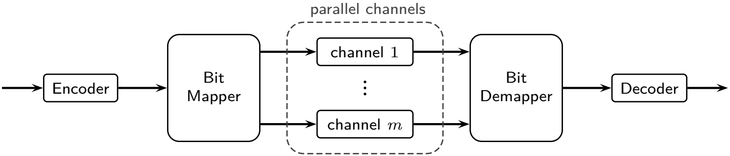
- Parallel channels arise in many practical scenarios: multicarrier transmission, rate-compatible puncturing of turbo-like codes, bit-interleaved coded modulation (BICM), . . .
- One binary encoder/decoder pair is often used for simplicity
- Here: spatially coupled LDPC codes and belief propagation (BP) decoding

Motivation



- Parallel channels arise in many practical scenarios: multicarrier transmission, rate-compatible puncturing of turbo-like codes, bit-interleaved coded modulation (BICM), ...
- One binary encoder/decoder pair is often used for simplicity
- Here: spatially coupled LDPC codes and belief propagation (BP) decoding
- The bit mapper determines the allocation of code bits to the channels

Motivation



- Parallel channels arise in many practical scenarios: multicarrier transmission, rate-compatible puncturing of turbo-like codes, bit-interleaved coded modulation (BICM), . . .
- One binary encoder/decoder pair is often used for simplicity
- Here: spatially coupled LDPC codes and belief propagation (BP) decoding
- The bit mapper determines the allocation of code bits to the channels

Main question

How much gain is possible by optimizing the bit mapper compared to a uniformly random mapper in the asymptotic setting (infinite block length)?

Outline

1. System Model
2. Decoding Threshold and Optimization
3. Results
4. Conclusions

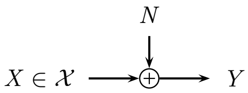
Parallel Channels and BICM

Parallel Channels and BICM

AWGN channel using a labeled signal constellation:

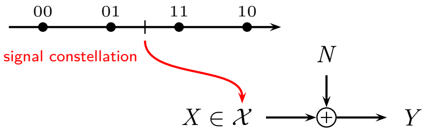
Parallel Channels and BICM

AWGN channel using a labeled signal constellation:



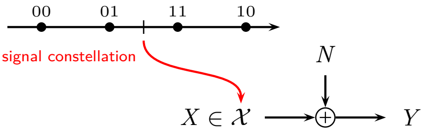
Parallel Channels and BICM

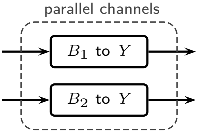
AWGN channel using a labeled signal constellation:



Parallel Channels and BICM

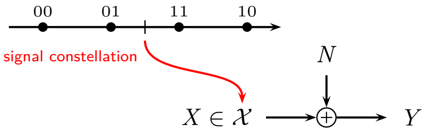
AWGN channel using a labeled signal constellation:

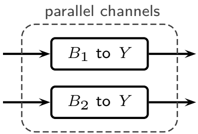




Parallel Channels and BICM

AWGN channel using a labeled signal constellation:

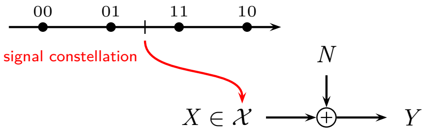


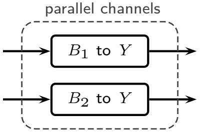


- Individual channel qualities
  $\varepsilon_1 \triangleq 1 - I(B_1; Y)$, $\varepsilon_2 \triangleq 1 - I(B_2; Y)$
  with average $\bar{\varepsilon} = (\varepsilon_1 + \varepsilon_2)/2$.

Parallel Channels and BICM

AWGN channel using a labeled signal constellation:

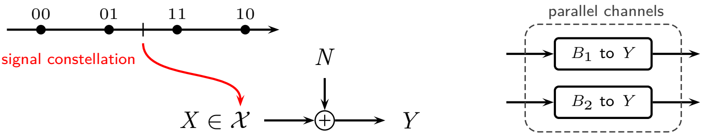


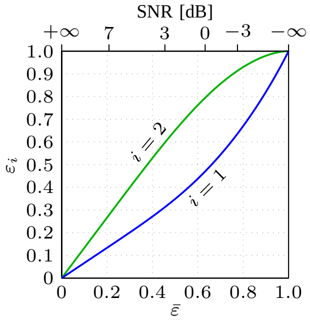


- Individual channel qualities $\varepsilon_1 \triangleq 1 - I(B_1; Y)$, $\varepsilon_2 \triangleq 1 - I(B_2; Y)$ with average $\bar{\varepsilon} = (\varepsilon_1 + \varepsilon_2)/2$.
- $\bar{\varepsilon}$ characterizes the quality of the set of channels

Parallel Channels and BICM

AWGN channel using a labeled signal constellation:

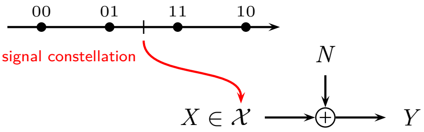


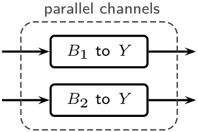


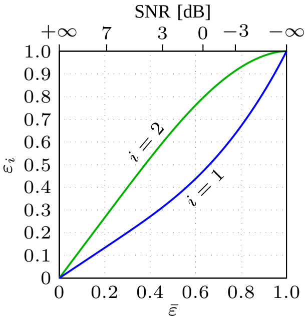
- Individual channel qualities $\varepsilon_1 \triangleq 1 - I(B_1; Y)$, $\varepsilon_2 \triangleq 1 - I(B_2; Y)$ with average $\bar{\varepsilon} = (\varepsilon_1 + \varepsilon_2)/2$.
- $\bar{\varepsilon}$ characterizes the quality of the set of channels

Parallel Channels and BICM

AWGN channel using a labeled signal constellation:



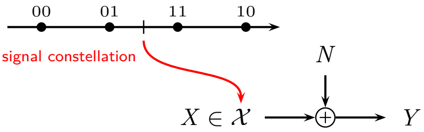


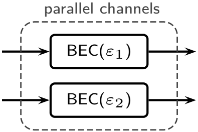


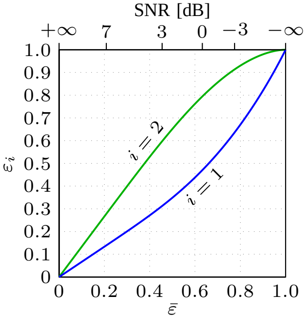
- Individual channel qualities $\varepsilon_1 \triangleq 1 - I(B_1; Y)$, $\varepsilon_2 \triangleq 1 - I(B_2; Y)$ with average $\bar{\varepsilon} = (\varepsilon_1 + \varepsilon_2)/2$.
- $\bar{\varepsilon}$ characterizes the quality of the set of channels
- Simplicity: replace the BICM bit channels by binary erasure channels (BECs)

Parallel Channels and BICM

AWGN channel using a labeled signal constellation:







- Individual channel qualities $\varepsilon_1 \triangleq 1 - I(B_1; Y)$, $\varepsilon_2 \triangleq 1 - I(B_2; Y)$ with average $\bar{\varepsilon} = (\varepsilon_1 + \varepsilon_2)/2$.
- $\bar{\varepsilon}$ characterizes the quality of the set of channels
- Simplicity: replace the BICM bit channels by binary erasure channels (BECs)

Spatially Coupled LDPC Codes

Spatially Coupled LDPC Codes

- Two-sided and circular spatially coupled $(d_\mathsf{v}, d_\mathsf{c}, L, w)$ code ensembles

Spatially Coupled LDPC Codes

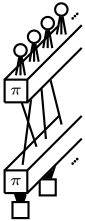
- Two-sided and circular spatially coupled $(d_\mathsf{v}, d_\mathsf{c}, L, w)$ code ensembles
- d_v and d_c denote the VN and CN degrees, L the spatial chain length, and w is a "smoothing"/coupling parameter

Spatially Coupled LDPC Codes

- Two-sided and circular spatially coupled (d_v, d_c, L, w) code ensembles
- d_v and d_c denote the VN and CN degrees, L the spatial chain length, and w is a "smoothing"/coupling parameter
- Example: Tanner graph of the $(3, 6, 5, 2)$ ensemble

Spatially Coupled LDPC Codes

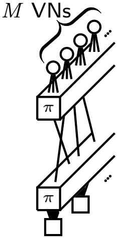
- Two-sided and circular spatially coupled $(d_{\mathsf{v}}, d_{\mathsf{c}}, L, w)$ code ensembles
- d_{v} and d_{c} denote the VN and CN degrees, L the spatial chain length, and w is a "smoothing"/coupling parameter
- Example: Tanner graph of the $(3, 6, 5, 2)$ ensemble



regular (3,6) LDPC ensemble

Spatially Coupled LDPC Codes

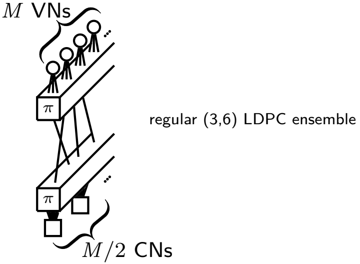
- Two-sided and circular spatially coupled $(d_{\mathsf{v}}, d_{\mathsf{c}}, L, w)$ code ensembles
- d_{v} and d_{c} denote the VN and CN degrees, L the spatial chain length, and w is a "smoothing"/coupling parameter
- Example: Tanner graph of the $(3, 6, 5, 2)$ ensemble



regular (3,6) LDPC ensemble

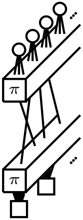
Spatially Coupled LDPC Codes

- Two-sided and circular spatially coupled $(d_\mathsf{v}, d_\mathsf{c}, L, w)$ code ensembles
- d_v and d_c denote the VN and CN degrees, L the spatial chain length, and w is a "smoothing"/coupling parameter
- Example: Tanner graph of the $(3, 6, 5, 2)$ ensemble



Spatially Coupled LDPC Codes

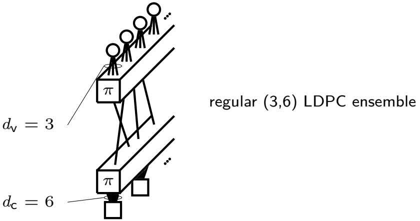
- Two-sided and circular spatially coupled (d_v, d_c, L, w) code ensembles
- d_v and d_c denote the VN and CN degrees, L the spatial chain length, and w is a "smoothing"/coupling parameter
- Example: Tanner graph of the $(3, 6, 5, 2)$ ensemble



regular (3,6) LDPC ensemble

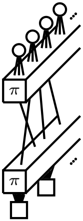
Spatially Coupled LDPC Codes

- Two-sided and circular spatially coupled $(d_{\mathsf{v}}, d_{\mathsf{c}}, L, w)$ code ensembles
- d_{v} and d_{c} denote the VN and CN degrees, L the spatial chain length, and w is a "smoothing"/coupling parameter
- Example: Tanner graph of the $(3, 6, 5, 2)$ ensemble



Spatially Coupled LDPC Codes

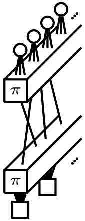
- Two-sided and circular spatially coupled $(d_{\mathsf{v}}, d_{\mathsf{c}}, L, w)$ code ensembles
- d_{v} and d_{c} denote the VN and CN degrees, L the spatial chain length, and w is a "smoothing"/coupling parameter
- Example: Tanner graph of the $(3, 6, 5, 2)$ ensemble



regular (3,6) LDPC ensemble

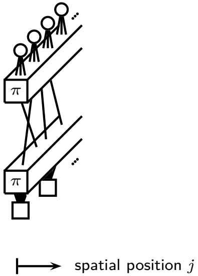
Spatially Coupled LDPC Codes

- Two-sided and circular spatially coupled $(d_\mathsf{v}, d_\mathsf{c}, L, w)$ code ensembles
- d_v and d_c denote the VN and CN degrees, L the spatial chain length, and w is a "smoothing"/coupling parameter
- Example: Tanner graph of the $(3, 6, 5, 2)$ ensemble



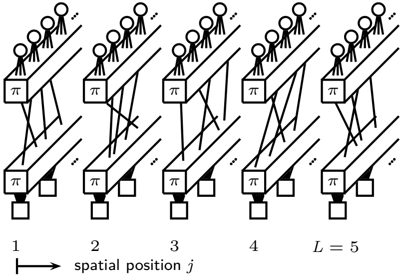
Spatially Coupled LDPC Codes

- Two-sided and circular spatially coupled $(d_\mathsf{v}, d_\mathsf{c}, L, w)$ code ensembles
- d_v and d_c denote the VN and CN degrees, L the spatial chain length, and w is a "smoothing"/coupling parameter
- Example: Tanner graph of the $(3, 6, 5, 2)$ ensemble



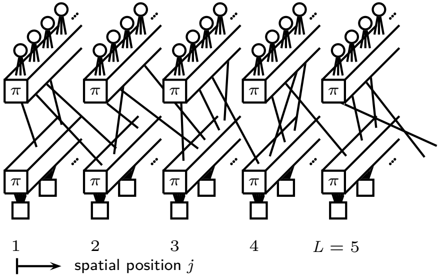
Spatially Coupled LDPC Codes

- Two-sided and circular spatially coupled $(d_\mathsf{v}, d_\mathsf{c}, L, w)$ code ensembles
- d_v and d_c denote the VN and CN degrees, L the spatial chain length, and w is a "smoothing"/coupling parameter
- Example: Tanner graph of the $(3, 6, 5, 2)$ ensemble



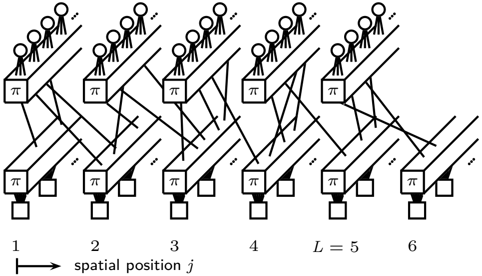
Spatially Coupled LDPC Codes

- Two-sided and circular spatially coupled $(d_\mathsf{v}, d_\mathsf{c}, L, w)$ code ensembles
- d_v and d_c denote the VN and CN degrees, L the spatial chain length, and w is a "smoothing"/coupling parameter
- Example: Tanner graph of the $(3, 6, 5, 2)$ ensemble



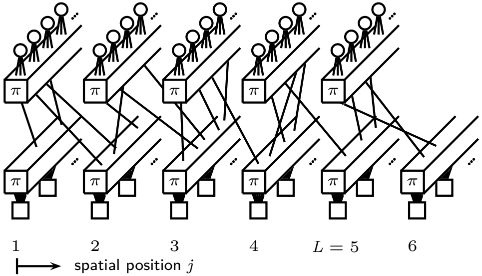
Spatially Coupled LDPC Codes

- Two-sided and circular spatially coupled (d_v, d_c, L, w) code ensembles
- d_v and d_c denote the VN and CN degrees, L the spatial chain length, and w is a "smoothing"/coupling parameter
- Example: Tanner graph of the $(3, 6, 5, 2)$ ensemble



Spatially Coupled LDPC Codes

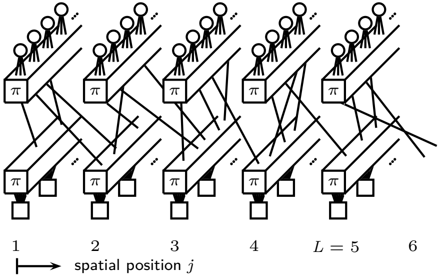
- Two-sided and circular spatially coupled $(d_\mathsf{v}, d_\mathsf{c}, L, w)$ code ensembles
- d_v and d_c denote the VN and CN degrees, L the spatial chain length, and w is a "smoothing"/coupling parameter
- Example: Tanner graph of the $(3, 6, 5, 2)$ ensemble



- Design rate $R = 1 - d_\mathsf{v}/d_\mathsf{c} - R_\mathsf{loss}(L)$, where $R_\mathsf{loss}(L) \to 0$ as $L \to \infty$

Spatially Coupled LDPC Codes

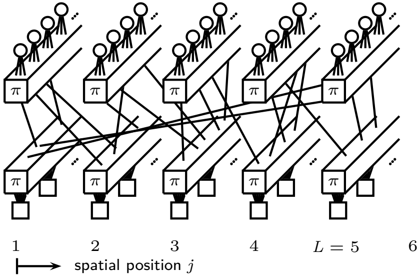
- Two-sided and circular spatially coupled $(d_\mathsf{v}, d_\mathsf{c}, L, w)$ code ensembles
- d_v and d_c denote the VN and CN degrees, L the spatial chain length, and w is a "smoothing"/coupling parameter
- Example: Tanner graph of the $(3, 6, 5, 2)$ ensemble



- Design rate $R = 1 - d_\mathsf{v}/d_\mathsf{c} - R_\mathsf{loss}(L)$, where $R_\mathsf{loss}(L) \to 0$ as $L \to \infty$

Spatially Coupled LDPC Codes

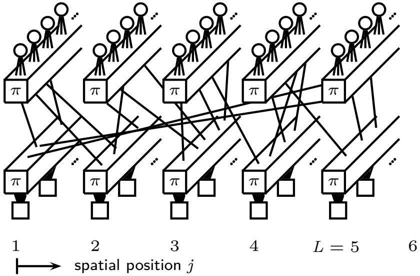
- Two-sided and circular spatially coupled $(d_\mathsf{v}, d_\mathsf{c}, L, w)$ code ensembles
- d_v and d_c denote the VN and CN degrees, L the spatial chain length, and w is a "smoothing"/coupling parameter
- Example: Tanner graph of the $(3, 6, 5, 2)$ ensemble



- Design rate $R = 1 - d_\mathsf{v}/d_\mathsf{c} - R_{\mathsf{loss}}(L)$, where $R_{\mathsf{loss}}(L) \to 0$ as $L \to \infty$

Spatially Coupled LDPC Codes

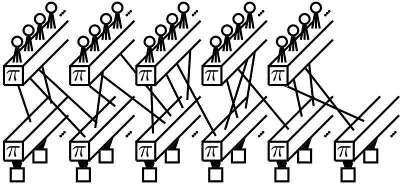
- Two-sided and circular spatially coupled (d_v, d_c, L, w) code ensembles
- d_v and d_c denote the VN and CN degrees, L the spatial chain length, and w is a "smoothing"/coupling parameter
- Example: Tanner graph of the $(3, 6, 5, 2)$ ensemble

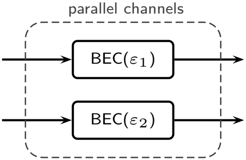


- Design rate $R = 1 - d_v/d_c - R_{loss}(L)$, where $R_{loss}(L) \to 0$ as $L \to \infty$
- For circular ensemble, $R = 1 - d_v/d_c$ (no rate loss)

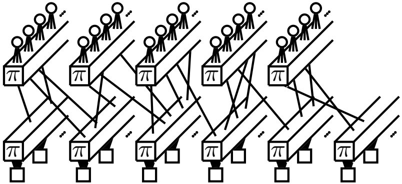
Bit Mapper

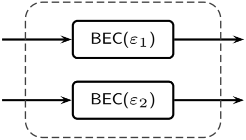
Bit Mapper





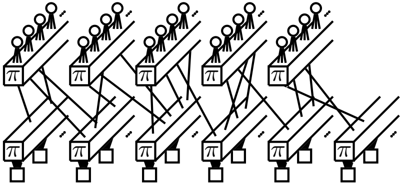
Bit Mapper

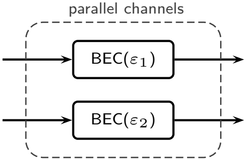




- VNs (i.e., code bits) at different positions belong to different equivalence classes

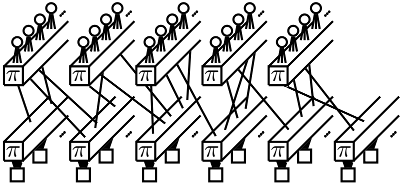
Bit Mapper

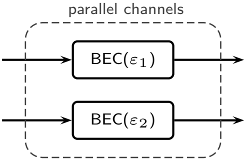




- VNs (i.e., code bits) at different positions belong to different equivalence classes
- Assignment of VN classes to channels via matrix $\mathbf{A} = [a_{i,j}] \in \mathbb{R}^{m \times L}$, where $a_{i,j} \triangleq$ fraction of VNs from position j to be sent over ith BEC

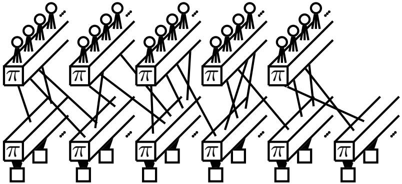
Bit Mapper

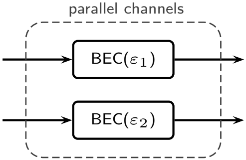




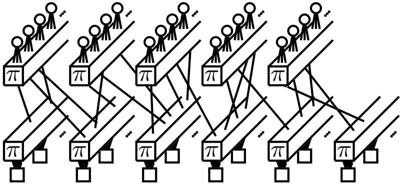
- VNs (i.e., code bits) at different positions belong to different equivalence classes
- Assignment of VN classes to channels via matrix $\mathbf{A} = [a_{i,j}] \in \mathbb{R}^{m \times L}$, where $a_{i,j} \triangleq$ fraction of VNs from position j to be sent over ith BEC
- Resulting VN erasure probablities $(\varepsilon^1, \ldots, \varepsilon^L) = (\varepsilon_1, \ldots, \varepsilon_m) \cdot \mathbf{A}$

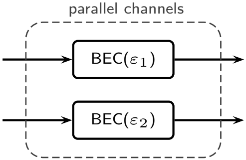
Bit Mapper





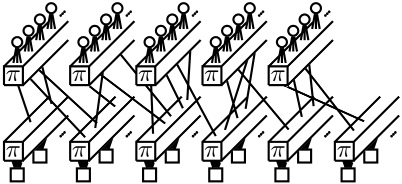
Bit Mapper

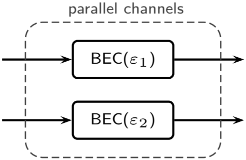




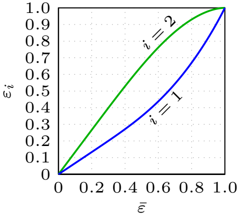
- Example: $\mathbf{A} = \begin{pmatrix} 1.0 & 0.0 & 0.5 & 0.75 & 0.25 \\ 0.0 & 1.0 & 0.5 & 0.25 & 0.75 \end{pmatrix}$

Bit Mapper

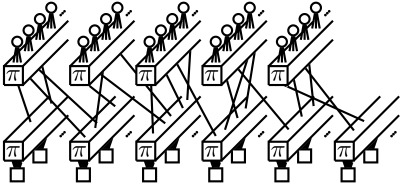


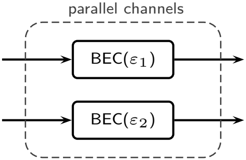


- Example: $\mathbf{A} = \begin{pmatrix} 1.0 & 0.0 & 0.5 & 0.75 & 0.25 \\ 0.0 & 1.0 & 0.5 & 0.25 & 0.75 \end{pmatrix}$

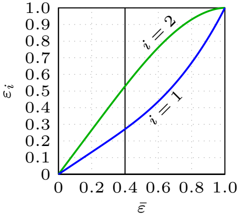


Bit Mapper



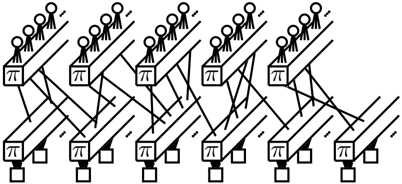


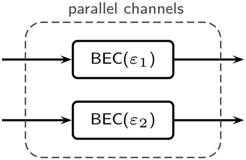
- Example: $\mathbf{A} = \begin{pmatrix} 1.0 & 0.0 & 0.5 & 0.75 & 0.25 \\ 0.0 & 1.0 & 0.5 & 0.25 & 0.75 \end{pmatrix}$



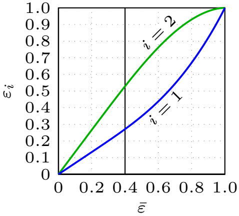
Fix $\bar{\varepsilon} = 0.4$

Bit Mapper



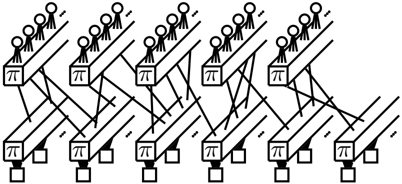


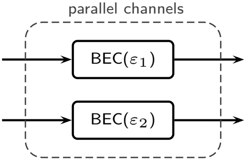
- Example: $\mathbf{A} = \begin{pmatrix} 1.0 & 0.0 & 0.5 & 0.75 & 0.25 \\ 0.0 & 1.0 & 0.5 & 0.25 & 0.75 \end{pmatrix}$



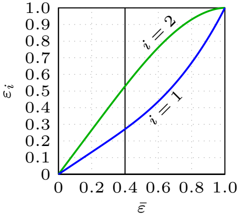
Fix $\bar{\varepsilon} = 0.4 \Rightarrow (\varepsilon_1, \varepsilon_2) = (0.27, 0.53)$

Bit Mapper





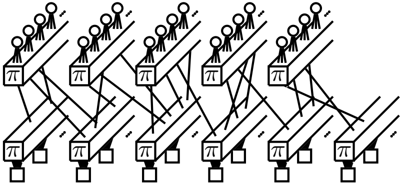
- Example: $\mathbf{A} = \begin{pmatrix} 1.0 & 0.0 & 0.5 & 0.75 & 0.25 \\ 0.0 & 1.0 & 0.5 & 0.25 & 0.75 \end{pmatrix}$

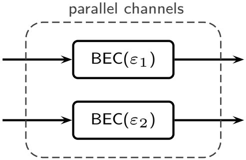


Fix $\bar{\varepsilon} = 0.4 \Rightarrow (\varepsilon_1, \varepsilon_2) = (0.27, 0.53)$

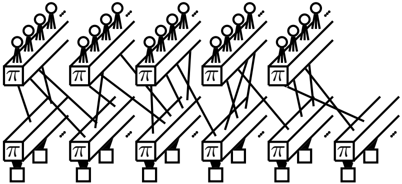
$$(0.27, 0.53) \cdot \mathbf{A} = (0.27, 0.53, 0.40, 0.335, 0.465)$$

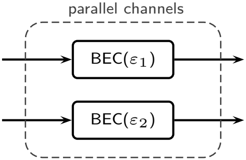
Bit Mapper





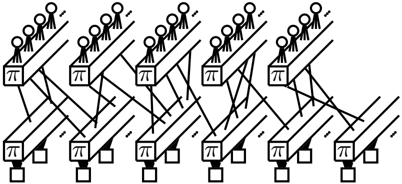
Bit Mapper

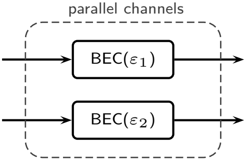




- Set of valid assignment matrices $\mathcal{A}^{m \times L}$: columns sum to 1 (all VNs are assigned), rows sum to L/m (all channels are used equally often)

Bit Mapper





- Set of valid assignment matrices $\mathcal{A}^{m\times L}$: columns sum to 1 (all VNs are assigned), rows sum to L/m (all channels are used equally often)
- Baseline bit mapper $\mathbf{A}_{\text{uni}}$ with $a_{i,j} = 1/m$, $\forall i, j$

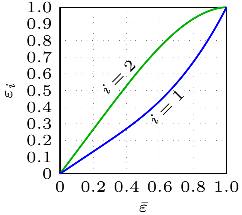
Density Evolution for BECs

Density Evolution for BECs

- For a fixed $\mathbf{A}$, density evolution provides a threshold $\bar{\varepsilon}^*(\mathbf{A})$ that divides the parameter range of $\bar{\varepsilon}$ into an admissable and a non-admissable region

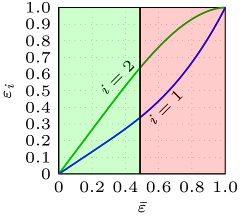
Density Evolution for BECs

- For a fixed $\mathbf{A}$, density evolution provides a threshold $\bar{\varepsilon}^*(\mathbf{A})$ that divides the parameter range of $\bar{\varepsilon}$ into an admissable and a non-admissable region



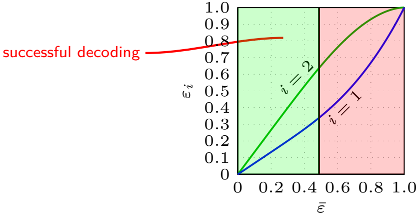
Density Evolution for BECs

- For a fixed $\mathbf{A}$, density evolution provides a threshold $\bar{\varepsilon}^*(\mathbf{A})$ that divides the parameter range of $\bar{\varepsilon}$ into an admissable and a non-admissable region



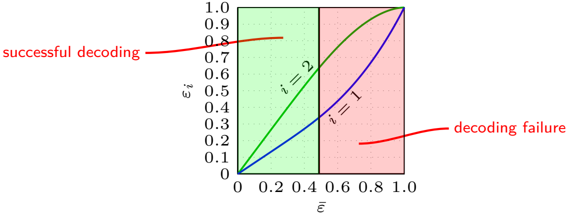
Density Evolution for BECs

- For a fixed $\mathbf{A}$, density evolution provides a threshold $\bar{\varepsilon}^*(\mathbf{A})$ that divides the parameter range of $\bar{\varepsilon}$ into an admissable and a non-admissable region



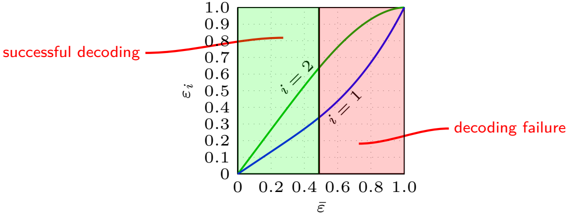
Density Evolution for BECs

- For a fixed $\mathbf{A}$, density evolution provides a threshold $\bar{\varepsilon}^*(\mathbf{A})$ that divides the parameter range of $\bar{\varepsilon}$ into an admissable and a non-admissable region



Density Evolution for BECs

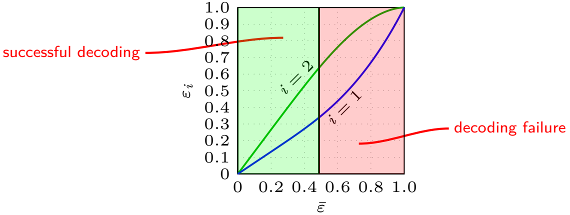
- For a fixed $\mathbf{A}$, density evolution provides a threshold $\bar{\varepsilon}^*(\mathbf{A})$ that divides the parameter range of $\bar{\varepsilon}$ into an admissable and a non-admissable region



- For BECs, density evolution is simple: Track the evolution of the VN erasure probabilities at all positions j.

Density Evolution for BECs

- For a fixed $\mathbf{A}$, density evolution provides a threshold $\bar{\varepsilon}^*(\mathbf{A})$ that divides the parameter range of $\bar{\varepsilon}$ into an admissable and a non-admissable region

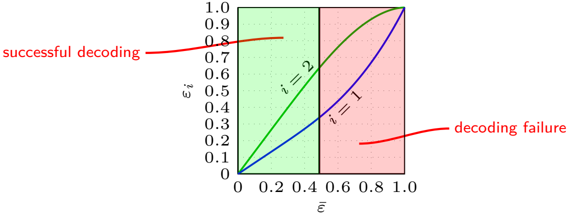


- For BECs, density evolution is simple: Track the evolution of the VN erasure probabilities at all positions j. For the lth BP iteration we have:

$$p_j^{(l)} = \varepsilon^j \left(\frac{1}{w} \sum_{a=0}^{w} \left(1 - \left(1 - \frac{1}{w} \sum_{b=0}^{w} p_{j+a-b}^{(l-1)} \right)^{d_{\mathsf{c}}-1} \right) \right)^{d_{\mathsf{v}}-1}$$

Density Evolution for BECs

- For a fixed $\mathbf{A}$, density evolution provides a threshold $\bar{\varepsilon}^*(\mathbf{A})$ that divides the parameter range of $\bar{\varepsilon}$ into an admissable and a non-admissable region



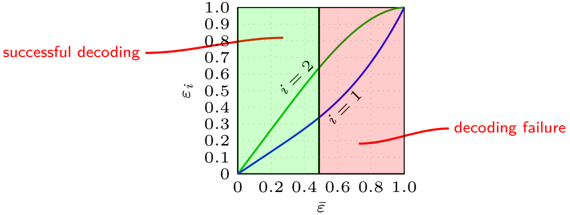
- For BECs, density evolution is simple: Track the evolution of the VN erasure probabilities at all positions j. For the lth BP iteration we have:

initial VN erasure probability

$$p_j^{(l)} = \varepsilon^j \left(\frac{1}{w} \sum_{a=0}^{w} \left(1 - \left(1 - \frac{1}{w} \sum_{b=0}^{w} p_{j+a-b}^{(l-1)} \right)^{d_\mathsf{c}-1} \right) \right)^{d_\mathsf{v}-1}$$

Density Evolution for BECs

- For a fixed $\mathbf{A}$, density evolution provides a threshold $\bar{\varepsilon}^*(\mathbf{A})$ that divides the parameter range of $\bar{\varepsilon}$ into an admissable and a non-admissable region

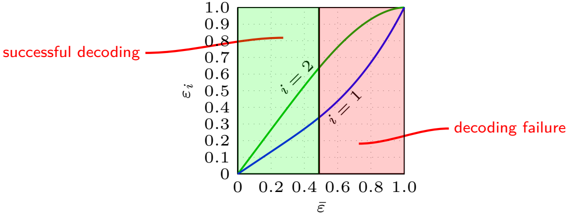


- For BECs, density evolution is simple: Track the evolution of the VN erasure probabilities at all positions j. For the lth BP iteration we have:

$$p_j^{(l)} = \varepsilon^j \left(\frac{1}{w} \sum_{a=0}^{w} \left(1 - \left(1 - \frac{1}{w} \sum_{b=0}^{w} p_{j+a-b}^{(l-1)} \right)^{d_c - 1} \right) \right)^{d_v - 1}$$

Density Evolution for BECs

- For a fixed $\mathbf{A}$, density evolution provides a threshold $\bar{\varepsilon}^*(\mathbf{A})$ that divides the parameter range of $\bar{\varepsilon}$ into an admissable and a non-admissable region



- For BECs, density evolution is simple: Track the evolution of the VN erasure probabilities at all positions j. For the lth BP iteration we have:

$$p_j^{(l)} = \varepsilon^j \left(\frac{1}{w} \sum_{a=0}^{w} \left(1 - \left(1 - \frac{1}{w} \sum_{b=0}^{w} p_{j+a-b}^{(l-1)} \right)^{d_\mathsf{c}-1} \right) \right)^{d_\mathsf{v}-1}$$

- $\bar{\varepsilon}^*(\mathbf{A}) \triangleq$ largest $\bar{\varepsilon} \in [0,1]$ such that $\lim_{l \to \infty} p_j^{(l)} \to 0$, $\forall j$

Example for Baseline Bit Mapper $\mathbf{A}_{\text{uni}}$

Example for Baseline Bit Mapper $\mathbf{A}_{\text{uni}}$

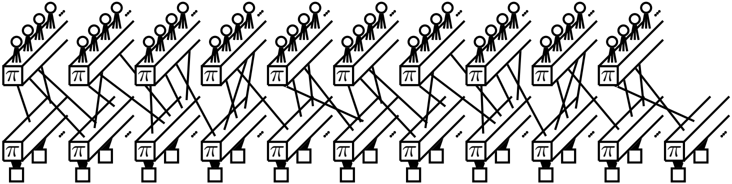
- For $\mathbf{A}_{\text{uni}}$, we have $\varepsilon^j = 0.5\varepsilon_1 + 0.5\varepsilon_2 = \bar{\varepsilon}$, $\forall j$

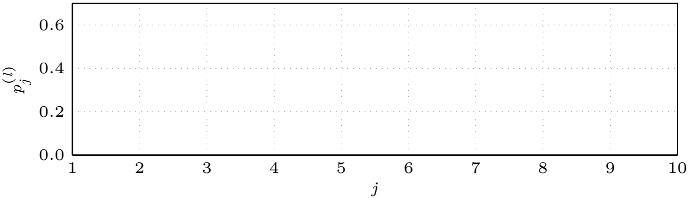
Example for Baseline Bit Mapper $\mathbf{A}_{\text{uni}}$

- For $\mathbf{A}_{\text{uni}}$, we have $\varepsilon^j = 0.5\varepsilon_1 + 0.5\varepsilon_2 = \bar{\varepsilon}$, $\forall j$
- For two-sided $(3, 6, 10, 2)$ ensemble, $\bar{\varepsilon}^*(\mathbf{A}) \approx 0.488$

Example for Baseline Bit Mapper $\mathbf{A}_{\text{uni}}$

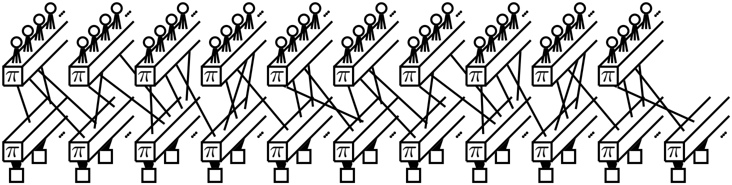
- For $\mathbf{A}_{\text{uni}}$, we have $\varepsilon^j = 0.5\varepsilon_1 + 0.5\varepsilon_2 = \bar{\varepsilon}$, $\forall j$
- For two-sided $(3, 6, 10, 2)$ ensemble, $\bar{\varepsilon}^*(\mathbf{A}) \approx 0.488$

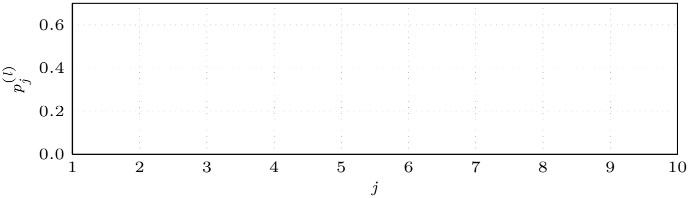




Example for Baseline Bit Mapper $\mathbf{A}_{\text{uni}}$

- For $\mathbf{A}_{\text{uni}}$, we have $\varepsilon^j = 0.5\varepsilon_1 + 0.5\varepsilon_2 = \bar{\varepsilon}$, $\forall j$
- For two-sided $(3, 6, 10, 2)$ ensemble, $\bar{\varepsilon}^*(\mathbf{A}) \approx 0.488$

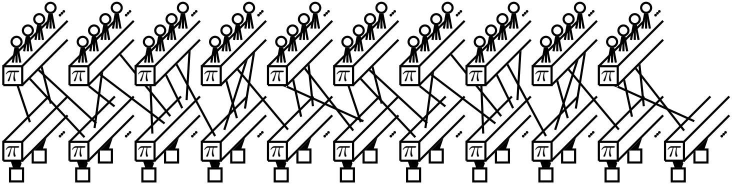


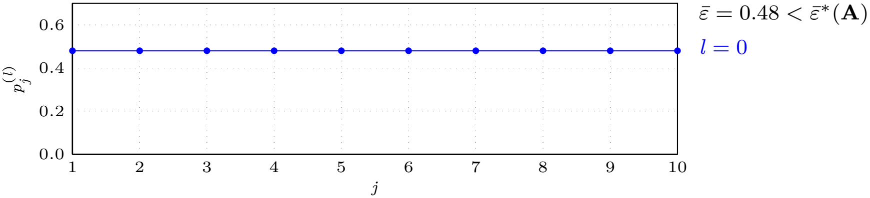


$\bar{\varepsilon} = 0.48 < \bar{\varepsilon}^*(\mathbf{A})$

Example for Baseline Bit Mapper $\mathbf{A}_{\text{uni}}$

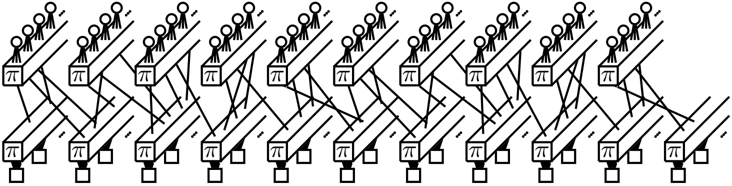
- For $\mathbf{A}_{\text{uni}}$, we have $\varepsilon^j = 0.5\varepsilon_1 + 0.5\varepsilon_2 = \bar{\varepsilon}$, $\forall j$
- For two-sided $(3, 6, 10, 2)$ ensemble, $\bar{\varepsilon}^*(\mathbf{A}) \approx 0.488$

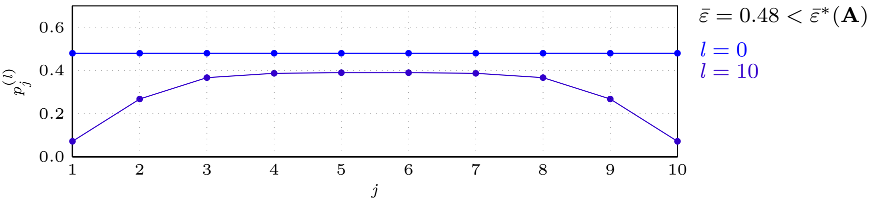




Example for Baseline Bit Mapper $\mathbf{A}_{\text{uni}}$

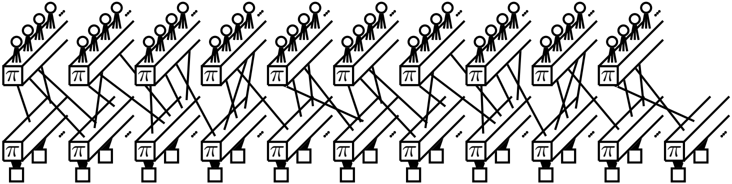
- For $\mathbf{A}_{\text{uni}}$, we have $\varepsilon^j = 0.5\varepsilon_1 + 0.5\varepsilon_2 = \bar{\varepsilon}$, $\forall j$
- For two-sided $(3, 6, 10, 2)$ ensemble, $\bar{\varepsilon}^*(\mathbf{A}) \approx 0.488$

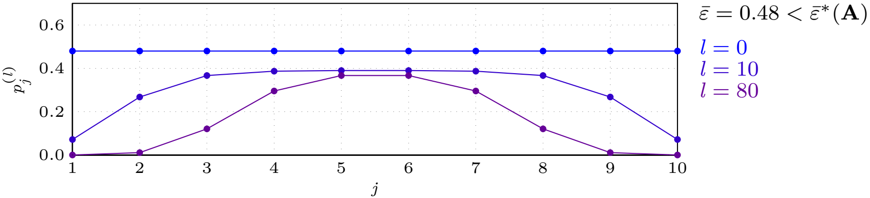




Example for Baseline Bit Mapper $\mathbf{A}_{\text{uni}}$

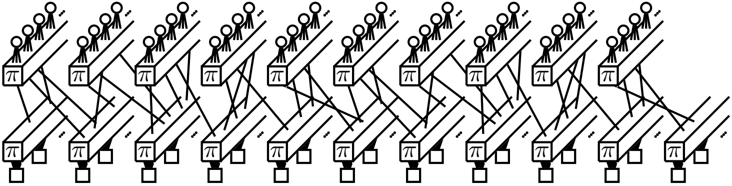
- For $\mathbf{A}_{\text{uni}}$, we have $\varepsilon^j = 0.5\varepsilon_1 + 0.5\varepsilon_2 = \bar{\varepsilon}$, $\forall j$
- For two-sided $(3, 6, 10, 2)$ ensemble, $\bar{\varepsilon}^*(\mathbf{A}) \approx 0.488$

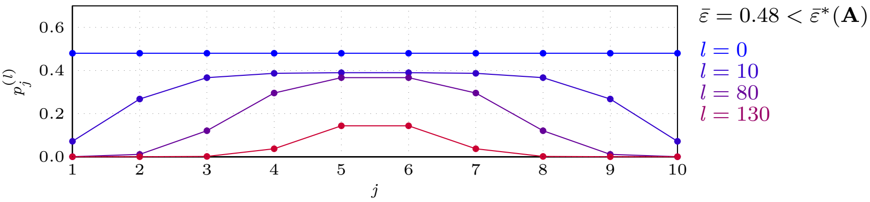




Example for Baseline Bit Mapper $\mathbf{A}_{\text{uni}}$

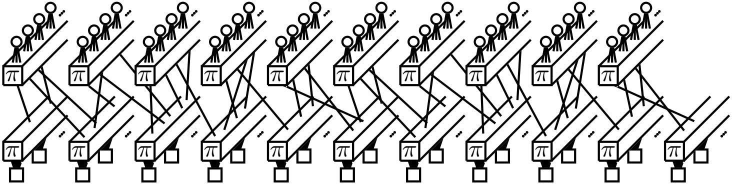
- For $\mathbf{A}_{\text{uni}}$, we have $\varepsilon^j = 0.5\varepsilon_1 + 0.5\varepsilon_2 = \bar{\varepsilon}$, $\forall j$
- For two-sided $(3, 6, 10, 2)$ ensemble, $\bar{\varepsilon}^*(\mathbf{A}) \approx 0.488$

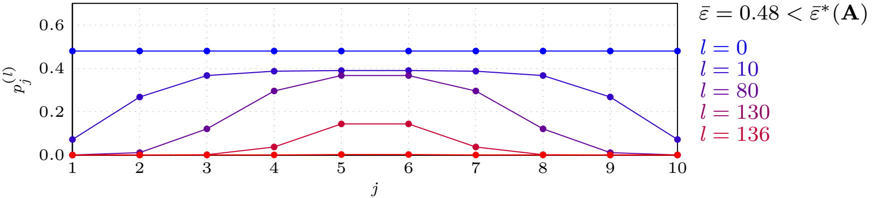




Example for Baseline Bit Mapper $\mathbf{A}_{\text{uni}}$

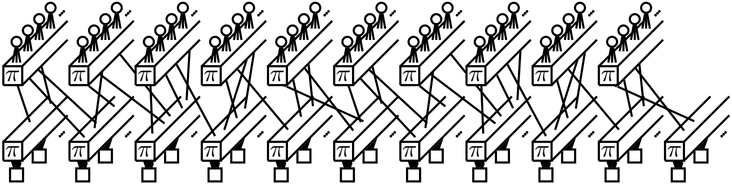
- For $\mathbf{A}_{\text{uni}}$, we have $\varepsilon^j = 0.5\varepsilon_1 + 0.5\varepsilon_2 = \bar{\varepsilon}$, $\forall j$
- For two-sided $(3, 6, 10, 2)$ ensemble, $\bar{\varepsilon}^*(\mathbf{A}) \approx 0.488$

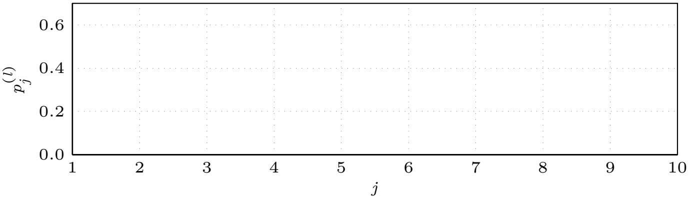




Example for Baseline Bit Mapper $\mathbf{A}_{\text{uni}}$

- For $\mathbf{A}_{\text{uni}}$, we have $\varepsilon^j = 0.5\varepsilon_1 + 0.5\varepsilon_2 = \bar{\varepsilon}$, $\forall j$
- For two-sided $(3, 6, 10, 2)$ ensemble, $\bar{\varepsilon}^*(\mathbf{A}) \approx 0.488$

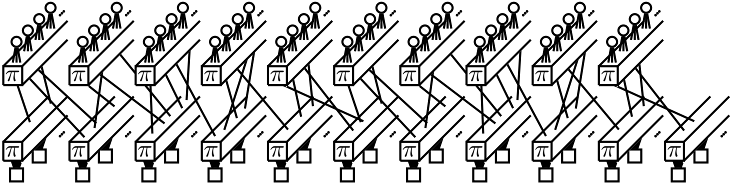


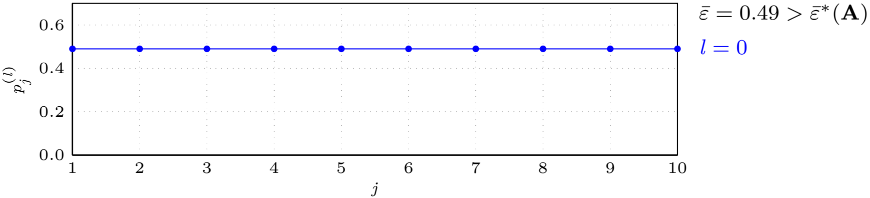


$\bar{\varepsilon} = 0.49 > \bar{\varepsilon}^*(\mathbf{A})$

Example for Baseline Bit Mapper $\mathbf{A}_{\text{uni}}$

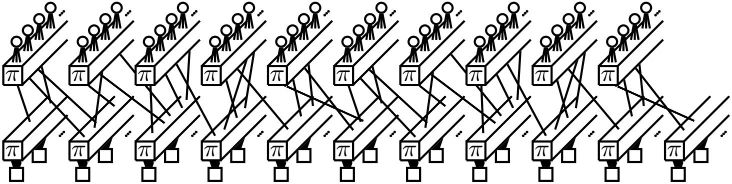
- For $\mathbf{A}_{\text{uni}}$, we have $\varepsilon^j = 0.5\varepsilon_1 + 0.5\varepsilon_2 = \bar{\varepsilon}$, $\forall j$
- For two-sided $(3, 6, 10, 2)$ ensemble, $\bar{\varepsilon}^*(\mathbf{A}) \approx 0.488$

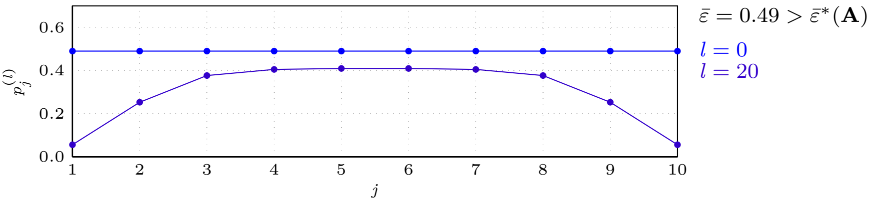




Example for Baseline Bit Mapper $\mathbf{A}_{\text{uni}}$

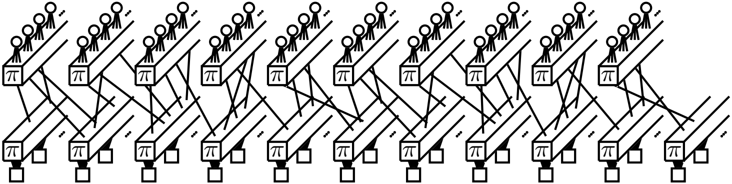
- For $\mathbf{A}_{\text{uni}}$, we have $\varepsilon^j = 0.5\varepsilon_1 + 0.5\varepsilon_2 = \bar{\varepsilon}, \forall j$
- For two-sided $(3, 6, 10, 2)$ ensemble, $\bar{\varepsilon}^*(\mathbf{A}) \approx 0.488$

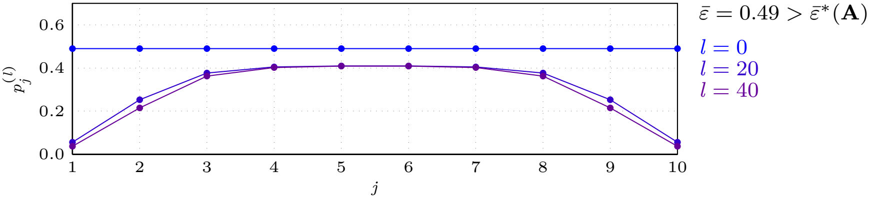




Example for Baseline Bit Mapper $\mathbf{A}_{\text{uni}}$

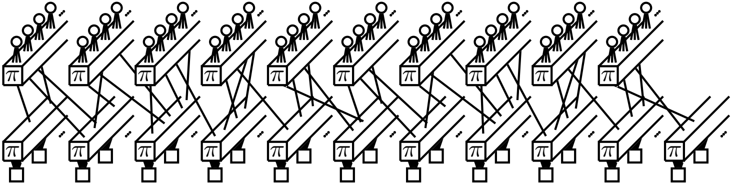
- For $\mathbf{A}_{\text{uni}}$, we have $\varepsilon^j = 0.5\varepsilon_1 + 0.5\varepsilon_2 = \bar{\varepsilon}$, $\forall j$
- For two-sided $(3, 6, 10, 2)$ ensemble, $\bar{\varepsilon}^*(\mathbf{A}) \approx 0.488$

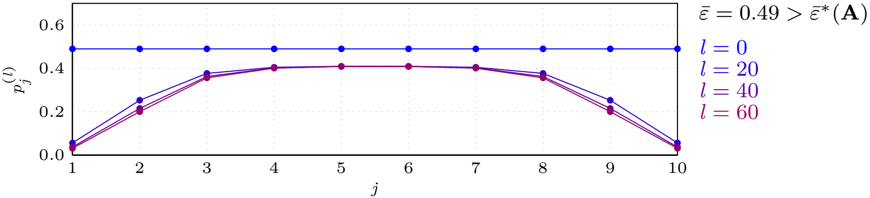




Example for Baseline Bit Mapper $\mathbf{A}_{\text{uni}}$

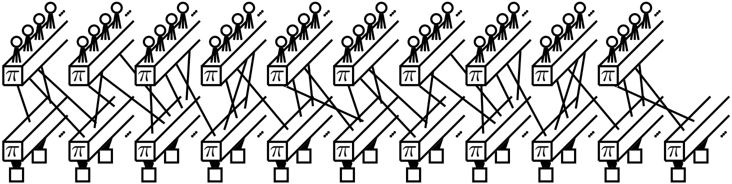
- For $\mathbf{A}_{\text{uni}}$, we have $\varepsilon^j = 0.5\varepsilon_1 + 0.5\varepsilon_2 = \bar{\varepsilon}$, $\forall j$
- For two-sided $(3, 6, 10, 2)$ ensemble, $\bar{\varepsilon}^*(\mathbf{A}) \approx 0.488$

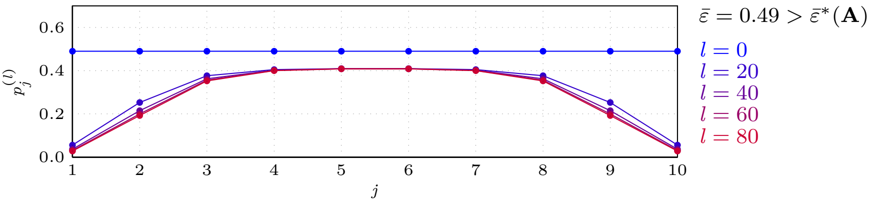




Example for Baseline Bit Mapper $\mathbf{A}_{\text{uni}}$

- For $\mathbf{A}_{\text{uni}}$, we have $\varepsilon^j = 0.5\varepsilon_1 + 0.5\varepsilon_2 = \bar{\varepsilon}$, $\forall j$
- For two-sided $(3, 6, 10, 2)$ ensemble, $\bar{\varepsilon}^*(\mathbf{A}) \approx 0.488$





Optimization Routine

Optimization Routine

- Ideally, we would like to solve the problem

$$\mathbf{A}_{\text{opt}} = \underset{\mathbf{A} \in \mathcal{A}^{m \times L}}{\text{argmax}} \quad \bar{\varepsilon}^*(\mathbf{A}).$$

Optimization Routine

- Ideally, we would like to solve the problem

$$\mathbf{A}_{\text{opt}} = \underset{\mathbf{A}\in\mathcal{A}^{m\times L}}{\text{argmax}} \quad \bar{\varepsilon}^*(\mathbf{A}).$$

- Difficult, due to computational cost of one threshold computation

Optimization Routine

- Ideally, we would like to solve the problem

$$\mathbf{A}_{\text{opt}} = \underset{\mathbf{A} \in \mathcal{A}^{m \times L}}{\text{argmax}} \quad \bar{\varepsilon}^*(\mathbf{A}).$$

- Difficult, due to computational cost of one threshold computation
- Alternative iterative optimization to find good bit mappers:

Optimization Routine

- Ideally, we would like to solve the problem

$$\mathbf{A}_{\text{opt}} = \underset{\mathbf{A} \in \mathcal{A}^{m \times L}}{\text{argmax}} \quad \bar{\varepsilon}^*(\mathbf{A}).$$

- Difficult, due to computational cost of one threshold computation
- Alternative iterative optimization to find good bit mappers:
  1. Initialize $\bar{\varepsilon}$ to the threshold of the baseline bit mapper $\bar{\varepsilon}^*(\mathbf{A}_{\text{uni}})$

Optimization Routine

- Ideally, we would like to solve the problem

$$\mathbf{A}_{\text{opt}} = \underset{\mathbf{A}\in\mathcal{A}^{m\times L}}{\text{argmax}} \quad \bar{\varepsilon}^*(\mathbf{A}).$$

- Difficult, due to computational cost of one threshold computation
- Alternative iterative optimization to find good bit mappers:
  1. Initialize $\bar{\varepsilon}$ to the threshold of the baseline bit mapper $\bar{\varepsilon}^*(\mathbf{A}_{\text{uni}})$
  2. Find $\mathbf{A}^*$ that minimizes the number of decoding iterations until convergence. Here, we use Differential Evolution

Optimization Routine

- Ideally, we would like to solve the problem

$$\mathbf{A}_{\text{opt}} = \underset{\mathbf{A} \in \mathcal{A}^{m \times L}}{\text{argmax}} \quad \bar{\varepsilon}^*(\mathbf{A}).$$

- Difficult, due to computational cost of one threshold computation
- Alternative iterative optimization to find good bit mappers:
  1. Initialize $\bar{\varepsilon}$ to the threshold of the baseline bit mapper $\bar{\varepsilon}^*(\mathbf{A}_{\text{uni}})$
  2. Find $\mathbf{A}^*$ that minimizes the number of decoding iterations until convergence. Here, we use Differential Evolution
  3. Calculate the new threshold. If it did not improve, stop, otherwise, go to step 2

Optimization Routine

- Ideally, we would like to solve the problem

$$\mathbf{A}_{\mathsf{opt}} = \underset{\mathbf{A}\in\mathcal{A}^{m\times L}}{\mathsf{argmax}} \quad \bar{\varepsilon}^*(\mathbf{A}).$$

- Difficult, due to computational cost of one threshold computation
- Alternative iterative optimization to find good bit mappers:
  1. Initialize $\bar{\varepsilon}$ to the threshold of the baseline bit mapper $\bar{\varepsilon}^*(\mathbf{A}_{\mathsf{uni}})$
  2. Find $\mathbf{A}^*$ that minimizes the number of decoding iterations until convergence. Here, we use Differential Evolution
  3. Calculate the new threshold. If it did not improve, stop, otherwise, go to step 2
- Significantly reduced computational complexity, however, $\mathbf{A}_{\mathsf{opt}} \neq \mathbf{A}^*$ in general

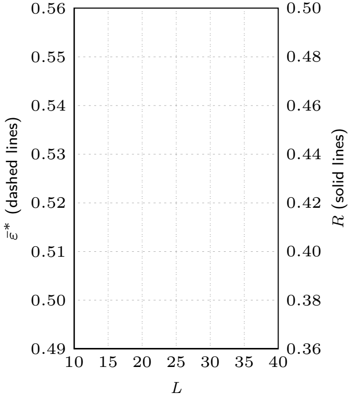
Two-Sided Ensembles

Two-Sided Ensembles

- Two-sided $(4, 8, L, w)$ ensemble, where $L \in \{10, 15, ..., 40\}$ and $w \in \{2, 4\}$

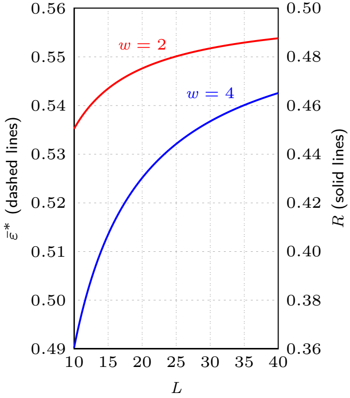
Two-Sided Ensembles

- Two-sided $(4, 8, L, w)$ ensemble, where $L \in \{10, 15, ..., 40\}$ and $w \in \{2, 4\}$



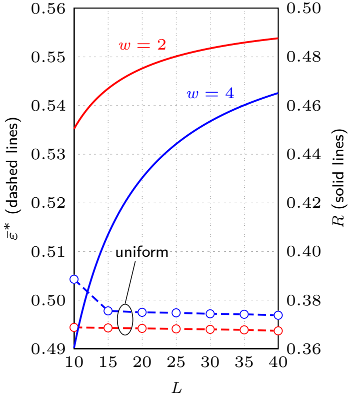
Two-Sided Ensembles

- Two-sided $(4, 8, L, w)$ ensemble, where $L \in \{10, 15, ..., 40\}$ and $w \in \{2, 4\}$



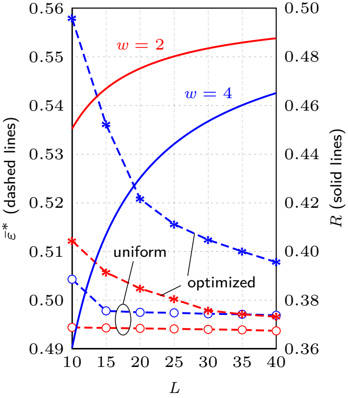
Two-Sided Ensembles

- Two-sided $(4, 8, L, w)$ ensemble, where $L \in \{10, 15, ..., 40\}$ and $w \in \{2, 4\}$



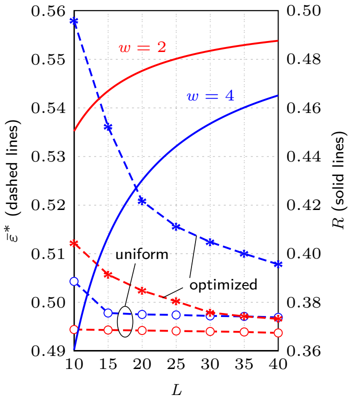
Two-Sided Ensembles

- Two-sided $(4, 8, L, w)$ ensemble, where $L \in \{10, 15, ..., 40\}$ and $w \in \{2, 4\}$



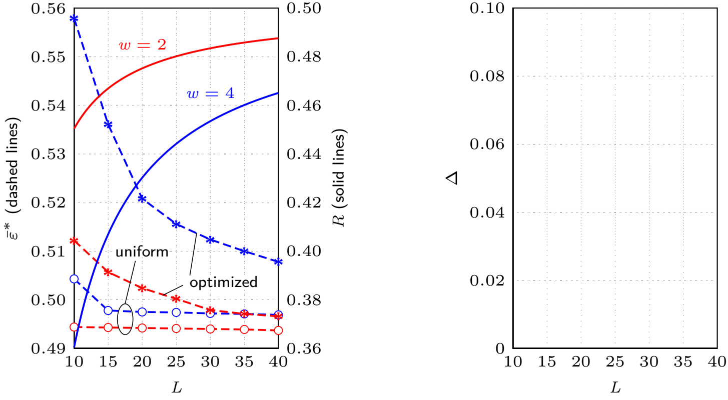
Two-Sided Ensembles

- Two-sided $(4, 8, L, w)$ ensemble, where $L \in \{10, 15, ..., 40\}$ and $w \in \{2, 4\}$
- Gap to capacity $\Delta \triangleq 1 - \bar{\varepsilon}^*(\mathbf{A}) - R$



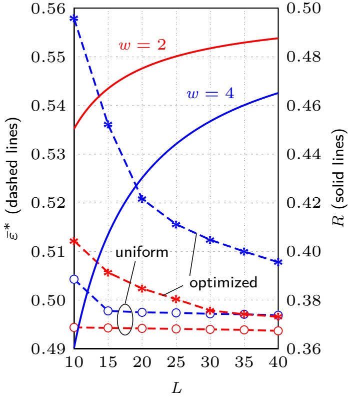
Two-Sided Ensembles

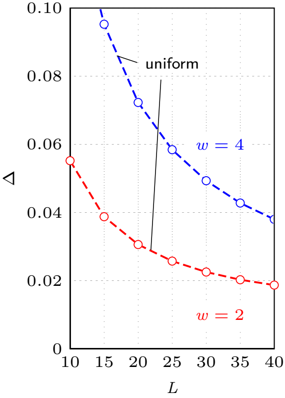
- Two-sided $(4, 8, L, w)$ ensemble, where $L \in \{10, 15, ..., 40\}$ and $w \in \{2, 4\}$
- Gap to capacity $\Delta \triangleq 1 - \bar{\varepsilon}^*(\mathbf{A}) - R$



Two-Sided Ensembles

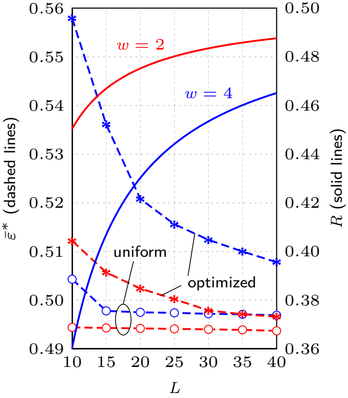
- Two-sided $(4, 8, L, w)$ ensemble, where $L \in \{10, 15, ..., 40\}$ and $w \in \{2, 4\}$
- Gap to capacity $\Delta \triangleq 1 - \bar{\varepsilon}^*(\mathbf{A}) - R$

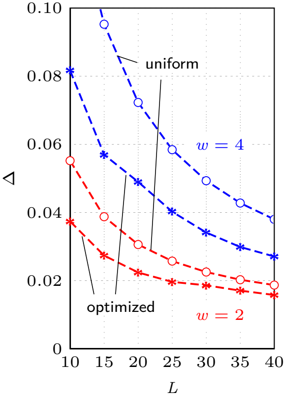




Two-Sided Ensembles

- Two-sided $(4, 8, L, w)$ ensemble, where $L \in \{10, 15, ..., 40\}$ and $w \in \{2, 4\}$
- Gap to capacity $\Delta \triangleq 1 - \bar{\varepsilon}^*(\mathbf{A}) - R$





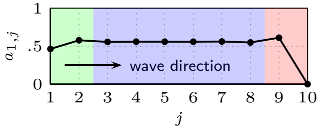
Optimized Bit Mappers for $w = 2$

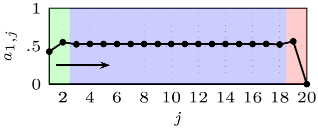
Optimized Bit Mappers for $w = 2$

- First row ($i = 1$) of the optimized assignment matrices $\mathbf{A}^*$ (fraction of VNs at a particular position to be sent over the good channel)

Optimized Bit Mappers for $w = 2$

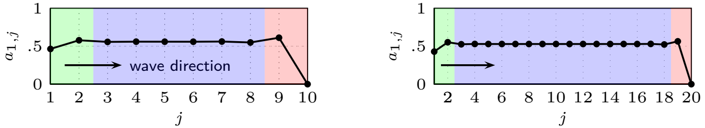
- First row ($i = 1$) of the optimized assignment matrices $\mathbf{A}^*$ (fraction of VNs at a particular position to be sent over the good channel)





Optimized Bit Mappers for $w = 2$

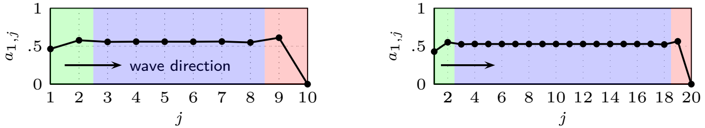
- First row ($i = 1$) of the optimized assignment matrices $\mathbf{A}^*$ (fraction of VNs at a particular position to be sent over the good channel)



- Shaded regions correspond to the part where a decoding wave will start (green), end (red), and propagate at roughly constant speed (blue)

Optimized Bit Mappers for $w = 2$

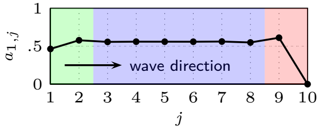
- First row ($i = 1$) of the optimized assignment matrices $\mathbf{A}^*$ (fraction of VNs at a particular position to be sent over the good channel)

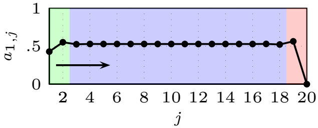


- Shaded regions correspond to the part where a decoding wave will start (green), end (red), and propagate at roughly constant speed (blue)
- Illustration of the iterative decoding behavior:

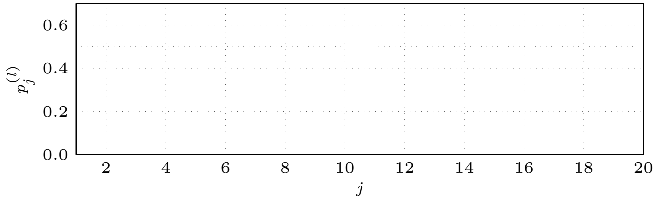
Optimized Bit Mappers for $w = 2$

- First row ($i = 1$) of the optimized assignment matrices $\mathbf{A}^*$ (fraction of VNs at a particular position to be sent over the good channel)



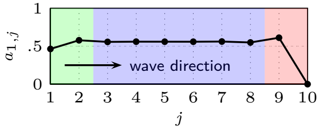


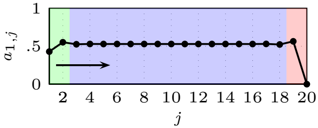
- Shaded regions correspond to the part where a decoding wave will start (green), end (red), and propagate at roughly constant speed (blue)
- Illustration of the iterative decoding behavior:



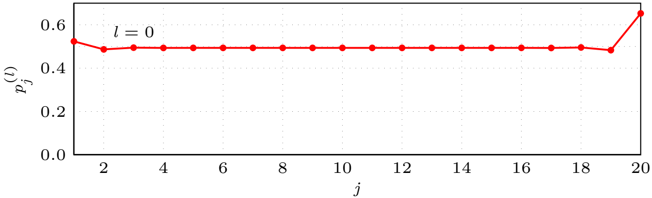
Optimized Bit Mappers for $w = 2$

- First row ($i = 1$) of the optimized assignment matrices $\mathbf{A}^*$ (fraction of VNs at a particular position to be sent over the good channel)



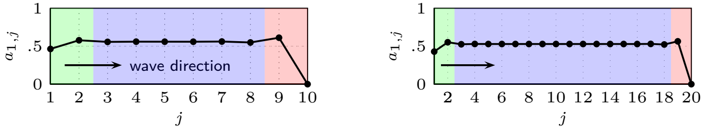


- Shaded regions correspond to the part where a decoding wave will start (green), end (red), and propagate at roughly constant speed (blue)
- Illustration of the iterative decoding behavior:

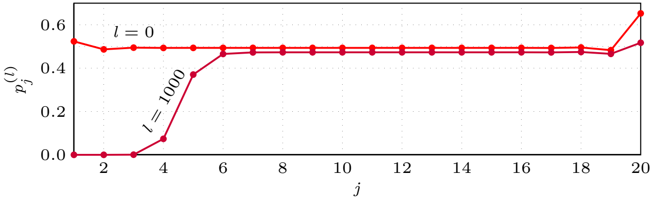


Optimized Bit Mappers for $w = 2$

- First row ($i = 1$) of the optimized assignment matrices $\mathbf{A}^*$ (fraction of VNs at a particular position to be sent over the good channel)

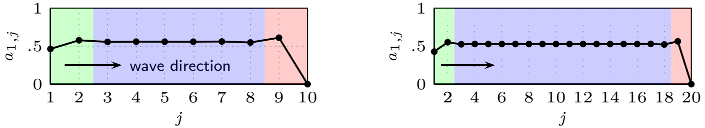


- Shaded regions correspond to the part where a decoding wave will start (green), end (red), and propagate at roughly constant speed (blue)
- Illustration of the iterative decoding behavior:

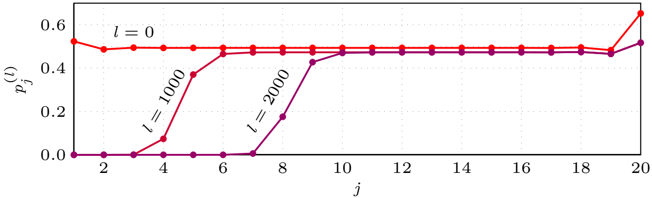


Optimized Bit Mappers for $w = 2$

- First row ($i = 1$) of the optimized assignment matrices $\mathbf{A}^*$ (fraction of VNs at a particular position to be sent over the good channel)

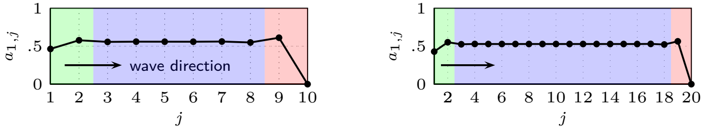


- Shaded regions correspond to the part where a decoding wave will start (green), end (red), and propagate at roughly constant speed (blue)
- Illustration of the iterative decoding behavior:

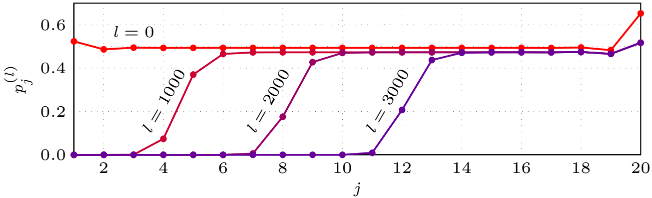


Optimized Bit Mappers for $w = 2$

- First row ($i = 1$) of the optimized assignment matrices $\mathbf{A}^*$ (fraction of VNs at a particular position to be sent over the good channel)

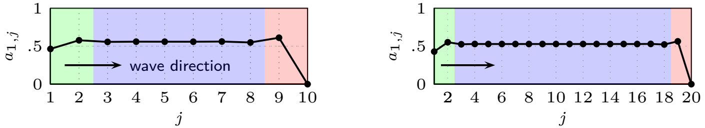


- Shaded regions correspond to the part where a decoding wave will start (green), end (red), and propagate at roughly constant speed (blue)
- Illustration of the iterative decoding behavior:

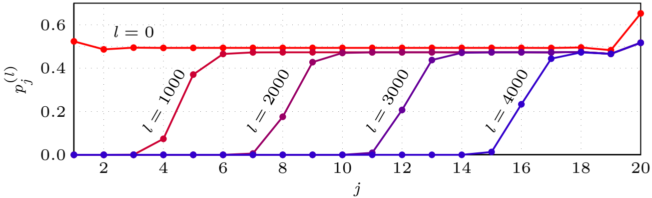


Optimized Bit Mappers for $w = 2$

- First row ($i = 1$) of the optimized assignment matrices $\mathbf{A}^*$ (fraction of VNs at a particular position to be sent over the good channel)

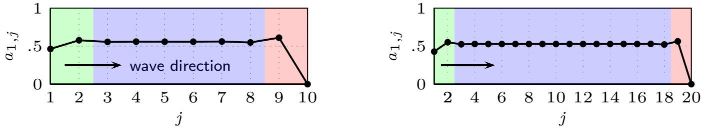


- Shaded regions correspond to the part where a decoding wave will start (green), end (red), and propagate at roughly constant speed (blue)
- Illustration of the iterative decoding behavior:

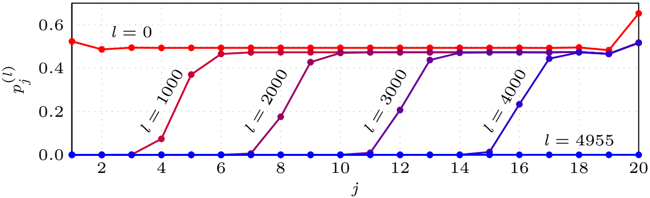


Optimized Bit Mappers for $w = 2$

- First row ($i = 1$) of the optimized assignment matrices $\mathbf{A}^*$ (fraction of VNs at a particular position to be sent over the good channel)



- Shaded regions correspond to the part where a decoding wave will start (green), end (red), and propagate at roughly constant speed (blue)
- Illustration of the iterative decoding behavior:



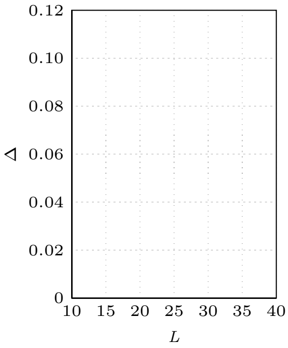
Circular Ensembles

Circular Ensembles

- Circular $(4, 8, L, w)$ ensemble, where $L \in \{10, 15, ..., 40\}$, $w \in \{2, 4\}$

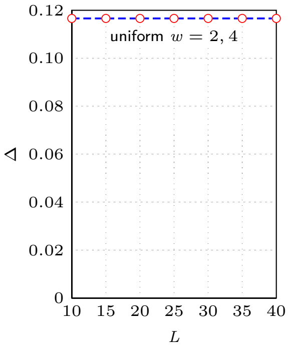
Circular Ensembles

- Circular $(4, 8, L, w)$ ensemble, where $L \in \{10, 15, ..., 40\}$, $w \in \{2, 4\}$



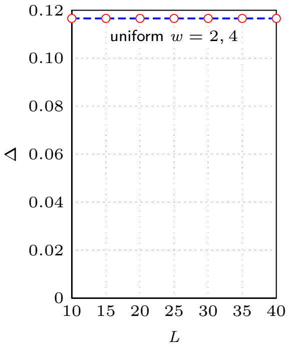
Circular Ensembles

- Circular $(4, 8, L, w)$ ensemble, where $L \in \{10, 15, ..., 40\}$, $w \in \{2, 4\}$



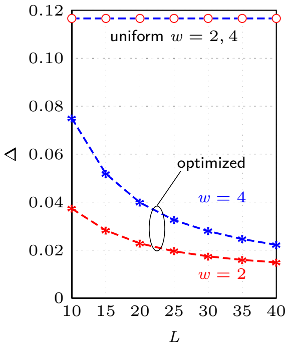
Circular Ensembles

- Circular $(4, 8, L, w)$ ensemble, where $L \in \{10, 15, ..., 40\}$, $w \in \{2, 4\}$
- Design rate is $R = 1/2$ and the threshold for uniform mapper is $\bar{\varepsilon}^*(\mathbf{A}_{\text{uni}}) = 0.3834$, independent of L and w



Circular Ensembles

- Circular $(4, 8, L, w)$ ensemble, where $L \in \{10, 15, ..., 40\}$, $w \in \{2, 4\}$
- Design rate is $R = 1/2$ and the threshold for uniform mapper is $\bar{\varepsilon}^*(\mathbf{A}_{\text{uni}}) = 0.3834$, independent of L and w



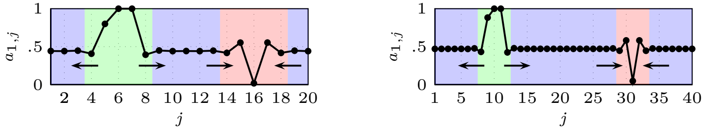
Optimized Bit Mappers for $w = 2$

Optimized Bit Mappers for $w = 2$

- First row ($i = 1$) of the optimized assignment matrices $\mathbf{A}^*$ (fraction of VNs at a particular position to be sent over the good channel)

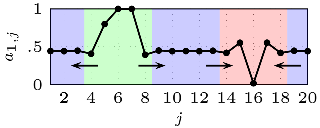
Optimized Bit Mappers for $w = 2$

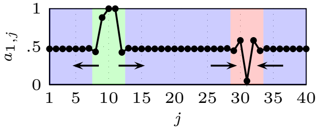
- First row ($i = 1$) of the optimized assignment matrices $\mathbf{A}^*$ (fraction of VNs at a particular position to be sent over the good channel)



Optimized Bit Mappers for $w = 2$

- First row ($i = 1$) of the optimized assignment matrices $\mathbf{A}^*$ (fraction of VNs at a particular position to be sent over the good channel)

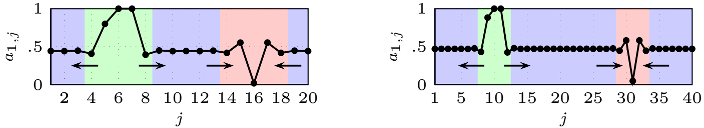




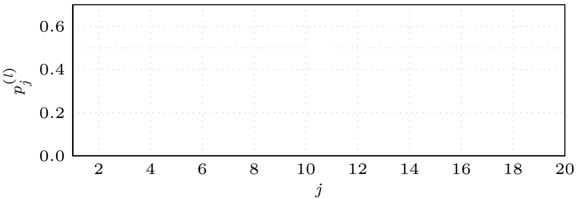
- Different channel qualities can be exploited to create a boundary-like termination effect

Optimized Bit Mappers for $w = 2$

- First row ($i = 1$) of the optimized assignment matrices $\mathbf{A}^*$ (fraction of VNs at a particular position to be sent over the good channel)

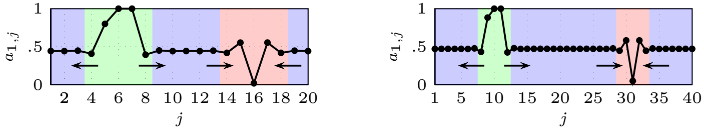


- Different channel qualities can be exploited to create a boundary-like termination effect

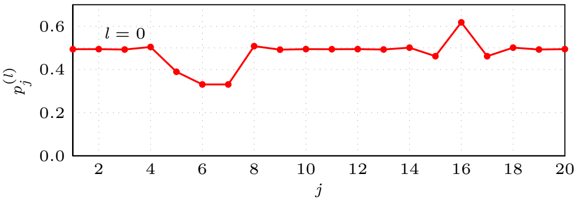


Optimized Bit Mappers for $w = 2$

- First row ($i = 1$) of the optimized assignment matrices $\mathbf{A}^*$ (fraction of VNs at a particular position to be sent over the good channel)

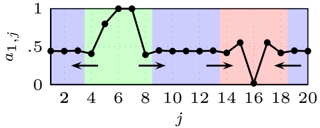


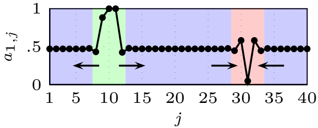
- Different channel qualities can be exploited to create a boundary-like termination effect



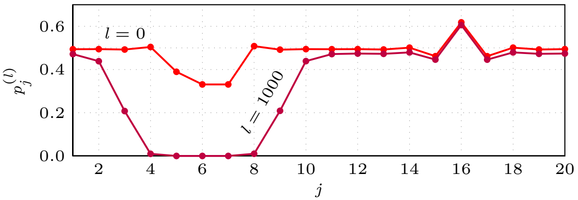
Optimized Bit Mappers for $w = 2$

- First row ($i = 1$) of the optimized assignment matrices $\mathbf{A}^*$ (fraction of VNs at a particular position to be sent over the good channel)



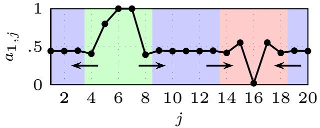


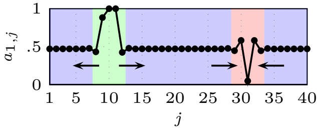
- Different channel qualities can be exploited to create a boundary-like termination effect



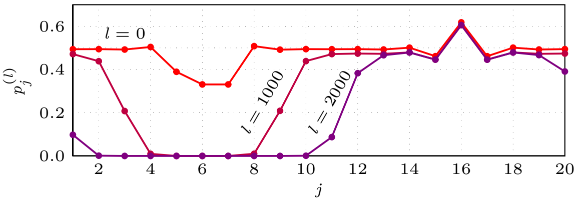
Optimized Bit Mappers for $w = 2$

- First row ($i = 1$) of the optimized assignment matrices $\mathbf{A}^*$ (fraction of VNs at a particular position to be sent over the good channel)



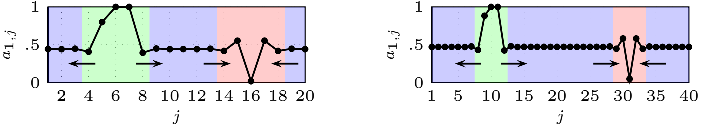


- Different channel qualities can be exploited to create a boundary-like termination effect

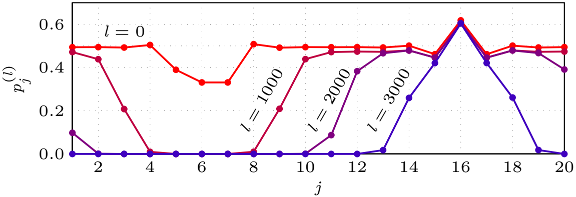


Optimized Bit Mappers for $w = 2$

- First row ($i = 1$) of the optimized assignment matrices $\mathbf{A}^*$ (fraction of VNs at a particular position to be sent over the good channel)

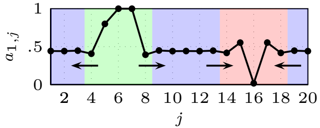


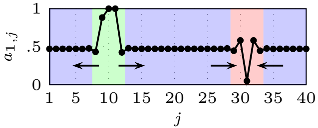
- Different channel qualities can be exploited to create a boundary-like termination effect



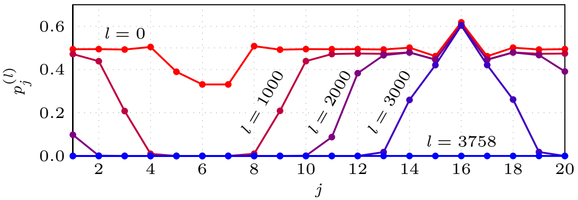
Optimized Bit Mappers for $w = 2$

- First row ($i = 1$) of the optimized assignment matrices $\mathbf{A}^*$ (fraction of VNs at a particular position to be sent over the good channel)





- Different channel qualities can be exploited to create a boundary-like termination effect



BICM Verification

BICM Verification

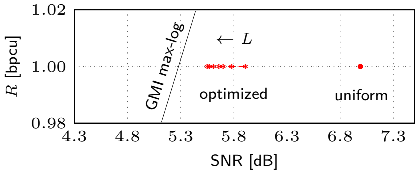
- Without going into details: BICM assuming max-log approximation for LLR computation and LLR symmetrization

BICM Verification

- Without going into details: BICM assuming max-log approximation for LLR computation and LLR symmetrization
- Discretized density evolution thresholds for circular $(4, 8, L, 2)$ ensembles using bit mappers optimized for BECs

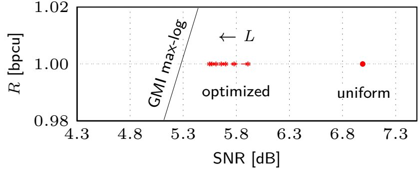
BICM Verification

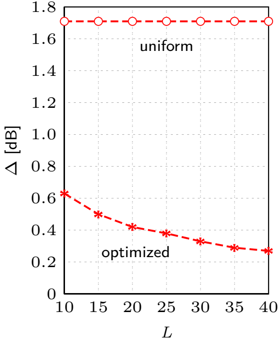
- Without going into details: BICM assuming max-log approximation for LLR computation and LLR symmetrization
- Discretized density evolution thresholds for circular $(4, 8, L, 2)$ ensembles using bit mappers optimized for BECs



BICM Verification

- Without going into details: BICM assuming max-log approximation for LLR computation and LLR symmetrization
- Discretized density evolution thresholds for circular $(4, 8, L, 2)$ ensembles using bit mappers optimized for BECs





Conclusions and Future Work

Conclusions and Future Work

1. Decoding threshold can be improved over a uniform random bit mapper, or, alternatively, the spatial chain length can be reduced.

Conclusions and Future Work

1. Decoding threshold can be improved over a uniform random bit mapper, or, alternatively, the spatial chain length can be reduced.
2. For circular ensembles, different channel qualities can be exploited to obtain wave-like decoding behavior similar to terminated ensembles

Conclusions and Future Work

1. Decoding threshold can be improved over a uniform random bit mapper, or, alternatively, the spatial chain length can be reduced.
2. For circular ensembles, different channel qualities can be exploited to obtain wave-like decoding behavior similar to terminated ensembles

Future work include study of protograph-based ensembles (for finite length code design).

Conclusions and Future Work

1. Decoding threshold can be improved over a uniform random bit mapper, or, alternatively, the spatial chain length can be reduced.
2. For circular ensembles, different channel qualities can be exploited to obtain wave-like decoding behavior similar to terminated ensembles

Future work include study of protograph-based ensembles (for finite length code design).

Paper version available on Arxiv (submitted to ICC 2014): Häger, Graell i Amat, Alvarado, Brännström, Agrell - Optimized Bit Mappers for Spatially Coupled LDPC Codes over Parallel Erasure Channels, Sep. 2013

Conclusions and Future Work

1. Decoding threshold can be improved over a uniform random bit mapper, or, alternatively, the spatial chain length can be reduced.
2. For circular ensembles, different channel qualities can be exploited to obtain wave-like decoding behavior similar to terminated ensembles

Future work include study of protograph-based ensembles (for finite length code design).

Paper version available on Arxiv (submitted to ICC 2014): Häger, Graell i Amat, Alvarado, Bränström, Agrell - Optimized Bit Mappers for Spatially Coupled LDPC Codes over Parallel Erasure Channels, Sep. 2013

Thank you!