Spatial Coupling vs. Block Coding: A Comparison

Daniel J. Costello, Jr.
Dept. of Electrical Engineering,
University of Notre Dame

Coding: From Theory to Practice
UC Berkeley, Feb 9th-13th 2015

Research Collaborators: David Mitchell,
Michael Lentmaier, and Ali Pusane
Outline

- **LDPC Block Codes**
  - Parity-check matrix and Tanner graph representations, minimum distance bounds, iterative decoding thresholds, regular and irregular code designs, protograph-based constructions

- **Spatially Coupled LDPC Codes**
  - Protograph representation, edge-spreading construction, termination
  - Iterative decoding thresholds, threshold saturation, minimum distance

- **Practical Considerations**
  - Finite-length scaling, window decoding, performance, latency, and complexity comparisons to LDPC block codes, implementation aspects
LDPC Block Codes

Definition by parity-check matrix: [Gallager, '62]
Bipartite graph representation: [Tanner, '81]

Code:  \( \{ \mathbf{v} \mid \mathbf{v} \mathbf{H}^T = 0 \} \)

\((J,K)\)-regular LDPC code:  \( R \geq 1 - \frac{J}{K} \)
**LDPC Block Codes**

Definition by parity-check matrix: [Gallager, '62]

Bipartite graph representation: [Tanner, '81]

\[
H = \begin{bmatrix}
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\]

\[H = 15 \times 20\]

\[n = 20 \text{ variable nodes of degree } J = 3\]

\[l = 15 \text{ check nodes of degree } K = 4\]

**Code:** \(\{v \mid vH^T = 0\}\) 

(J,K)-regular LDPC code: \(R \geq 1 - \frac{J}{K}\)

- Graph-based codes can be decoded **iteratively** with **low-complexity** by exchanging messages in the graph using **Belief Propagation (BP)**.
For an **asymptotically good** code ensemble, the minimum distance grows linearly with the block length $n$. 

![Graph showing the growth rate of minimum distance](image-url)
For an asymptotically good code ensemble, the minimum distance grows linearly with the block length $n$.

$(J,K)$-regular block code ensembles are asymptotically good, i.e.,

$$d_{\text{min}} \geq n\delta_{JK}$$

where $\delta_{JK}$ is called the typical minimum distance ratio, or minimum distance growth rate.
For an **asymptotically good** code ensemble, the minimum distance grows linearly with the block length $n$.

- $(J,K)$-regular block code ensembles are asymptotically good, i.e.,

$$d_{\text{min}} \geq n\delta_{JK}$$

where $\delta_{JK}$ is called the **typical minimum distance ratio**, or **minimum distance growth rate**.

- As the density of $(J,K)$-regular ensembles increases, $\delta_{JK}$ approaches the Gilbert-Varshamov bound.
Thresholds of $(J,K)$-regular LDPC Block Code Ensembles

- Iterative decoding thresholds can be calculated for $(J,K)$-regular LDPC block code ensembles using density evolution (DE).

**BEC thresholds**

<table>
<thead>
<tr>
<th>$J$</th>
<th>$K$</th>
<th>Rate</th>
<th>$\varepsilon^*$</th>
<th>$\varepsilon_{Sh}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>6</td>
<td>0.5</td>
<td>0.429</td>
<td>0.5</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>0.5</td>
<td>0.383</td>
<td>0.5</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>0.5</td>
<td>0.341</td>
<td>0.5</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>0.4</td>
<td>0.517</td>
<td>0.6</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>0.333</td>
<td>0.506</td>
<td>0.667</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>0.25</td>
<td>0.647</td>
<td>0.75</td>
</tr>
</tbody>
</table>

**AWGNC thresholds**

<table>
<thead>
<tr>
<th>$J$</th>
<th>$K$</th>
<th>Rate</th>
<th>$(E_b/N_0)^*$</th>
<th>$(E_b/N_0)_{Sh}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>6</td>
<td>0.5</td>
<td>1.11</td>
<td>0.184</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>0.5</td>
<td>1.61</td>
<td>0.184</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>0.5</td>
<td>2.04</td>
<td>0.184</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>0.4</td>
<td>0.96</td>
<td>−0.229</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>0.333</td>
<td>1.67</td>
<td>−0.480</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>0.25</td>
<td>1.00</td>
<td>−0.790</td>
</tr>
</tbody>
</table>

Thresholds of (J,K)-regular LDPC Block Code Ensembles

- Iterative decoding thresholds can be calculated for (J,K)-regular LDPC block code ensembles using density evolution (DE).

**BEC thresholds**

<table>
<thead>
<tr>
<th>J</th>
<th>K</th>
<th>Rate</th>
<th>$\varepsilon^*$</th>
<th>$\varepsilon_{Sh}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>6</td>
<td>0.5</td>
<td>0.429</td>
<td>0.5</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>0.5</td>
<td>0.383</td>
<td>0.5</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>0.5</td>
<td>0.341</td>
<td>0.5</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>0.4</td>
<td>0.517</td>
<td>0.6</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>0.333</td>
<td>0.506</td>
<td>0.667</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>0.25</td>
<td>0.647</td>
<td>0.75</td>
</tr>
</tbody>
</table>

**AWGNC thresholds**

<table>
<thead>
<tr>
<th>J</th>
<th>K</th>
<th>Rate</th>
<th>$(E_b/N_0)^*$</th>
<th>$(E_b/N_0)_{Sh}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>6</td>
<td>0.5</td>
<td>1.11</td>
<td>0.184</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>0.5</td>
<td>1.61</td>
<td>0.184</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>0.5</td>
<td>2.04</td>
<td>0.184</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>0.4</td>
<td>0.96</td>
<td>-0.229</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>0.333</td>
<td>1.67</td>
<td>-0.480</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>0.25</td>
<td>1.00</td>
<td>-0.790</td>
</tr>
</tbody>
</table>

- There exists a relatively **large gap to capacity**.

Thresholds of \((J,K)\)-regular LDPC Block Code Ensembles

- Iterative decoding thresholds can be calculated for \((J,K)\)-regular LDPC block code ensembles using density evolution (DE).

**BEC thresholds**

<table>
<thead>
<tr>
<th>(J)</th>
<th>(K)</th>
<th>Rate</th>
<th>(\varepsilon^*)</th>
<th>(\varepsilon_{\text{Sh}})</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>6</td>
<td>0.5</td>
<td>0.429</td>
<td>0.5</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>0.5</td>
<td>0.383</td>
<td>0.5</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>0.5</td>
<td>0.341</td>
<td>0.5</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>0.4</td>
<td>0.517</td>
<td>0.6</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>0.333</td>
<td>0.506</td>
<td>0.667</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>0.25</td>
<td>0.647</td>
<td>0.75</td>
</tr>
</tbody>
</table>

**AWGNC thresholds**

<table>
<thead>
<tr>
<th>(J)</th>
<th>(K)</th>
<th>Rate</th>
<th>((E_b/N_0)^*)</th>
<th>((E_b/N_0)_{\text{Sh}})</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>6</td>
<td>0.5</td>
<td>1.11</td>
<td>0.184</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>0.5</td>
<td>1.61</td>
<td>0.184</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>0.5</td>
<td>2.04</td>
<td>0.184</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>0.4</td>
<td>0.96</td>
<td>−0.229</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>0.333</td>
<td>1.67</td>
<td>−0.480</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>0.25</td>
<td>1.00</td>
<td>−0.790</td>
</tr>
</tbody>
</table>

- There exists a relatively large gap to capacity.
- Iterative decoding thresholds get further from capacity as the graph density increases.

**Protograph Representation**

- **Compact representation** of a structured LDPC block code ensemble with code length \( n = Mb_v \) and code design rate \( R \geq (b_v - b_c)/b_v \)

\[
B = \begin{bmatrix}
1 & 1 & 0 & 0 \\
0 & 1 & 1 & 1 \\
1 & 1 & 1 & 0 \\
\end{bmatrix}_{b_c \times b_v}
\]

**Base matrix**

\( b_v \) variable nodes

\( b_c \) check nodes

**Protograph**

Compact representation of a structured LDPC block code ensemble with code length \( n = M b_v \) and code design rate \( R \geq (b_v - b_c)/b_v \)

\[ \mathbf{B} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}_{b_c \times b_v} \]

base matrix

\[ \mathbf{H} = \begin{bmatrix} \Pi_{1,1} & \Pi_{1,2} & 0 & 0 \\ 0 & \Pi_{2,2} & \Pi_{2,3} & \Pi_{2,4} \\ \Pi_{3,1} & \Pi_{3,2} & \Pi_{3,3} & 0 \end{bmatrix}_{M b_c \times M b_v} \]

parity-check matrix

By design, every member of a protograph-based ensemble preserves the structure of the base protograph.
By design, every member of a protograph-based ensemble preserves the structure of the base protograph.

Density evolution analysis can be performed on the protograph, enabling the calculation of the iterative decoding threshold.
By design, every member of a protograph-based ensemble preserves the structure of the base protograph.

Density evolution analysis can be performed on the protograph, enabling the calculation of the iterative decoding threshold.

From the protograph, an expression [Divsalar '06] can be obtained for the ensemble average weight enumerator,

\[ \overline{A}(z) = \sum_{d=0}^{\infty} \overline{A}_d z^d, \]

where \( \overline{A}_d \) is the average number of codewords of weight \( d \), which can be used to test if the ensemble is asymptotically good.
Outline

- **LDPC Block Codes**
  - Parity-check matrix and Tanner graph representations, minimum distance bounds, iterative decoding thresholds, regular and irregular code designs, protograph-based constructions

- **Spatially Coupled LDPC Codes**
  - Protograph representation, edge-spreading construction, termination
  - Iterative decoding thresholds, threshold saturation, minimum distance

- **Practical Considerations**
  - Finite-length properties, window decoding, comparison to block codes, implementation aspects
Consider transmission of consecutive blocks (protograph representation):

\[
\begin{bmatrix}
3 & 3 \\
\end{bmatrix}
\]

(3,6)-regular LDPC-BC base matrix
Spatially Coupled LDPC Code Ensembles

Consider transmission of consecutive blocks (protograph representation):

- Blocks are spatially coupled (introducing memory) by spreading edges over time:

\[
B = \begin{bmatrix}
3 & 3 \\
\end{bmatrix}
\]

(3,6)-regular LDPC-BC base matrix

- (m_s = 2)

D. J. Costello, Jr., “Spatial Coupling vs. Block Coding: A Comparison”
Spatially Coupled LDPC Code Ensembles

- Consider transmission of consecutive blocks (protograph representation):

![Protograph Diagram]

- Blocks are spatially coupled (introducing memory) by spreading edges over time:

\[ \text{Spreading constraint: } \sum_{i=0}^{m_s} B_i = B \]

\[ B = \begin{bmatrix} 3 & 3 \end{bmatrix} \]

(3,6)-regular LDPC-BC base matrix

\[ \begin{bmatrix} B_0 \\ B_1 \\ B_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix} \]
Transmission of consecutive spatially coupled (SC) blocks results in a convolutional protograph:
Transmission of consecutive spatially coupled (SC) blocks results in a convolutional protograph:
Transmission of consecutive spatially coupled (SC) blocks results in a **convolutional protograph**:
Transmission of consecutive spatially coupled (SC) blocks results in a convolutional protograph:
Transmission of consecutive spatially coupled (SC) blocks results in a convolutional protograph:

\[
\begin{array}{ccccccc}
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
& & & & B_{m_s} & \cdots & B_1 \\
& & & B_{m_s} & \cdots & B_1 & B_0 \\
& & B_{m_s} & \cdots & B_1 & B_0 & \cdots \\
\end{array}
\]

The bi-infinite convolutional protograph corresponds to a bi-infinite convolutional base matrix:

\[
B_{[-\infty, \infty]} = 
\begin{bmatrix}
\ddots & \ddots & \ddots & \ddots \\
& B_{m_s} & \cdots & B_1 \\
& & B_{m_s} & \cdots & B_1 & B_0 \\
& & & \ddots & \ddots & \ddots \\
\end{bmatrix}
\]

- \(B_i\) has size \(b_c \times b_v\)
- **Rate:** \(R = \frac{b_v - b_c}{b_v}\)
- **Constraint length:** \(\nu_s = b_v(m_s + 1)\)
Consider terminating $B_{[-\infty, \infty]}$ to a (block code) base matrix of length $Lb_v$:

$$B_{[0,L-1]} = \begin{bmatrix} B_0 & \cdots & B_0 \\ \vdots & \ddots & \vdots \\ B_{m_s} & \cdots & B_{m_s} \end{bmatrix}$$

($B_i$ is a $b_c \times b_v$ matrix)

**Code rate:**

$$R_L = \frac{Lb_v - (L + m_s)b_c}{Lb_v}.$$
Terminated Spatially Coupled Codes

- Consider terminating $B_{[-\infty, \infty]}$ to a (block code) base matrix of length $Lb_v$:

$$B_{[0,L-1]} = \begin{bmatrix}
B_0 \\
\vdots \\
B_{m_s} \\
\vdots \\
B_0
\end{bmatrix} \quad (B_i \text{ is a } b_c \times b_v \text{ matrix})$$

- For large $L$, $R_L$ approaches the unterminated code rate $R = (b_v - b_c)/b_v$. 

\[
R_L = \frac{Lb_v - (L + m_s)b_c}{Lb_v}.
\]
**Terminated Spatially Coupled Codes**

- Consider terminating $B_{[\infty, \infty]}$ to a (block code) base matrix of length $Lb_v$:

  $B_{[0, L-1]} = \begin{bmatrix} B_0 & \vdots & \vdots & \vdots & B_0 \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & B_{m_s} & \vdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ B_{m_s} \end{bmatrix}$  

  ($B_i$ is a $b_c \times b_v$ matrix)

- For large $L$, $R_L$ approaches the unterminated code rate $R = (b_v - b_c)/b_v$.

- **Example:** $(3, 6)$-regular base matrix $B = \begin{bmatrix} 3 & 3 \end{bmatrix}$, $m_s = 2$, $L = 4$, $R_s = 1/4$

- Code rate:

  $$R_L = \frac{Lb_v - (L + m_s)b_c}{Lb_v}$$

- Consider terminating the base matrix to a (block code) base matrix of length $Lb_v$.

- Example: $(3, 6)$-regular base matrix $B = \begin{bmatrix} 3 & 3 \end{bmatrix}$, $m_s = 2$, $L = 4$, $R_s = 1/4$
Terminated Spatially Coupled Codes

- Consider **terminating** \( B_{[-\infty, \infty]} \) to a (block code) **base matrix** of length \( Lb_v \):

  \[
  B_{[0,L-1]} = \begin{bmatrix}
  B_0 \\
  \vdots \\
  B_{m_s} \\
  \vdots \\
  B_{m_s}
  \end{bmatrix}
  \]

  \((B_i\) is a \( b_c \times b_v \) matrix\)

  \[
  \text{Code rate:} \quad R_L = \frac{Lb_v - (L + m_s)b_c}{Lb_v}.
  \]

- For large \( L \), \( R_L \) approaches the **unterminated** code rate \( R = (b_v - b_c)/b_v \).

- **Example:** \((3,6)\)-regular base matrix \( B = \begin{bmatrix} 3 & 3 \end{bmatrix} \), \( m_s = 2 \), \( L = 4 \), \( R_s = 1/4 \)

  \[
  B_{[0,3]} = \begin{bmatrix}
  1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
  1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
  1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
  0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
  0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
  0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
  0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
  \end{bmatrix}
  \]

  (check node degrees lower at the ends)

- Codes can be **lifted** to different lengths and rates by varying \( M \) and \( L \).
Variable nodes all have the **same degree** as the underlying block code.

Check nodes with **lower degrees** (at the ends) improve the BP decoder.
Wave-like Decoding of Terminated Spatially Coupled Codes

- Variable nodes all have the **same degree** as the underlying block code.
- Check nodes with **lower degrees** (at the ends) improve the BP decoder.

Evolution of message probabilities: (3,6)-regular SC-LDPC code \((L = 100)\)
Wave-like Decoding of Terminated Spatially Coupled Codes

- Variable nodes all have the **same degree** as the underlying block code.
- Check nodes with **lower degrees** (at the ends) improve the BP decoder.

Evolution of message probabilities: (3,6)-regular SC-LDPC code \((L = 100)\)
Wave-like Decoding of Terminated Spatially Coupled Codes

- Variable nodes all have the **same degree** as the underlying block code.
- Check nodes with **lower degrees** (at the ends) improve the BP decoder.

Evolution of message probabilities: (3,6)-regular SC-LDPC code ($L = 100$)
Wave-like Decoding of Terminated Spatially Coupled Codes

- Variable nodes all have the **same degree** as the underlying block code.
- Check nodes with **lower degrees** (at the ends) improve the BP decoder.

Evolution of message probabilities: (3,6)-regular SC-LDPC code ($L = 100$)
Wave-like Decoding of Terminated Spatially Coupled Codes

- Variable nodes all have the same degree as the underlying block code.
- Check nodes with lower degrees (at the ends) improve the BP decoder.

Evolution of message probabilities: (3,6)-regular SC-LDPC code ($L = 100$)
Wave-like Decoding of Terminated Spatially Coupled Codes

- Variable nodes all have the **same degree** as the underlying block code.
- Check nodes with **lower degrees** (at the ends) improve the BP decoder.

Evolution of message probabilities: (3,6)-regular SC-LDPC code ($L = 100$)
Wave-like Decoding of Terminated Spatially Coupled Codes

- Variable nodes all have the same degree as the underlying block code.
- Check nodes with lower degrees (at the ends) improve the BP decoder.

Evolution of message probabilities: (3,6)-regular SC-LDPC code ($L = 100$)
Wave-like Decoding of Terminated Spatially Coupled Codes

- Variable nodes all have the **same degree** as the underlying block code.
- Check nodes with **lower degrees** (at the ends) improve the BP decoder.

Evolution of message probabilities: (3,6)-regular SC-LDPC code ($L = 100$)

Note: the **fraction** of lower degree nodes tends to zero as $L \rightarrow \infty$, i.e., the codes are **asymptotically regular**.
Density evolution can be applied to the protograph-based ensembles with $M \to \infty$ [Sridharan et al. '04]:

Example: BEC
Thresholds of Terminated Spatially Coupled Codes

- **Density evolution** can be applied to the protograph-based ensembles with $M \to \infty$ [Sridharan et al. '04]:

**Example:** BEC

$L = 4$, $R = 1/4$

$\varepsilon^* = 0.635$, $\varepsilon_{Sh} = 0.75$

![Graph showing BEC threshold for different termination factors and limits](image)
Density evolution can be applied to the protograph-based ensembles with $M \rightarrow \infty$ [Sridharan et al. '04]:

**Example: BEC**

$L = 4$, $R = 1/4$

$\epsilon^* = 0.635$, $\epsilon_{Sh} = 0.75$

$L = 10$, $R = 2/5$

$\epsilon^* = 0.505$, $\epsilon_{Sh} = 0.6$
Thresholds of Terminated Spatially Coupled Codes

- **Density evolution** can be applied to the protograph-based ensembles with $M \to \infty$ [Sridharan et al. '04]:

**Example:** BEC

$L = 4, R = 1/4$
$\varepsilon^* = 0.635, \varepsilon_{\text{Sh}} = 0.75$

$L = 10, R = 2/5$
$\varepsilon^* = 0.505, \varepsilon_{\text{Sh}} = 0.6$

$L \to \infty, R \to 1/2$
$\varepsilon^* = 0.488, \varepsilon_{\text{Sh}} = 0.5$

(3,6)-regular block code:
$\varepsilon^* = 0.429$
Thresholds of Terminated Spatially Coupled Codes

Iterative decoding thresholds (structured protograph-based ensembles)

<table>
<thead>
<tr>
<th>$(J, K)$</th>
<th>$\epsilon^*_\text{SC}$</th>
<th>$\epsilon^*_\text{blk}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(3,6)</td>
<td>0.488</td>
<td>0.429</td>
</tr>
<tr>
<td>(4,8)</td>
<td>0.497</td>
<td>0.383</td>
</tr>
<tr>
<td>(5,10)</td>
<td>0.499</td>
<td>0.341</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$(J, K)$</th>
<th>$E_b/N_o \text{sc}$</th>
<th>$E_b/N_o \text{blk}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(3,6)</td>
<td>0.46 dB</td>
<td>1.11 dB</td>
</tr>
<tr>
<td>(4,8)</td>
<td>0.26 dB</td>
<td>1.61 dB</td>
</tr>
<tr>
<td>(5,10)</td>
<td>0.21 dB</td>
<td>2.04 dB</td>
</tr>
</tbody>
</table>

- We observe a significant improvement in the thresholds of SC-LDPC codes compared to the associated LDPC block codes (LDPC-BCs) due to the lower degree check nodes at the ends of the graph and wave-like decoding.

We observe a significant improvement in the thresholds of SC-LDPC codes compared to the associated LDPC block codes (LDPC-BCs) due to the lower degree check nodes at the ends of the graph and wave-like decoding.

In contrast to LDPC-BCs, the iterative decoding thresholds of SC-LDPC codes improve as the graph density increases.

When symbols are perfectly known (BEC), all adjacent edges can be removed from the Tanner graph.
Why are Terminated Spatially Coupled Thresholds Better?

- When symbols are perfectly known (BEC), all adjacent edges can be removed from the Tanner graph.
Why are Terminated Spatially Coupled Thresholds Better?

- When symbols are perfectly known (BEC), all adjacent edges can be removed from the Tanner graph.
Why are Terminated Spatially Coupled Thresholds Better?

- When symbols are perfectly known (BEC), all adjacent edges can be removed from the Tanner graph.
When symbols are perfectly known (BEC), all adjacent edges can be removed from the Tanner graph.
Why are Terminated Spatially Coupled Thresholds Better?

- When symbols are perfectly known (BEC), all adjacent edges can be removed from the Tanner graph.

- The threshold saturates (converges) to a fixed value numerically indistinguishable from the maximum a posteriori (MAP) threshold of the \((J, K)\)-regular LDPC-BC ensemble as \(L \rightarrow \infty\) [LSCZ10].

Why are Terminated Spatially Coupled Thresholds Better?

- When symbols are perfectly known (BEC), all adjacent edges can be removed from the Tanner graph.

- The threshold saturates (converges) to a fixed value numerically indistinguishable from the maximum a posteriori (MAP) threshold of the \((J, K)\)-regular LDPC-BC ensemble as \(L \to \infty\) \([\text{LSCZ10}]\).

- For a more random-like ensemble, this has been proven analytically, first for the BEC \([\text{KRU11}]\), then for all BMS channels \([\text{KRU13}]\).

---


Threshold Saturation (BEC)
Threshold Saturation (BEC)

![Graph showing threshold saturation for different erasure rates and epsilon values. The graph plots bit erasure rate against epsilon. The data points for (5,10) and (3,6) are marked with arrows and circles, respectively.]

D. J. Costello, Jr., “Spatial Coupling vs. Block Coding: A Comparison”
Threshold Saturation (BEC)
By increasing $J$ and $K$, we obtain capacity achieving $(J,K)$-regular SC-LDPC code ensembles with linear minimum distance growth.
Similar results are obtained for the AWGNC

Distance Measures for SC-LDPC Codes

As $L \to \infty$ the minimum distance growth rates of terminated SC-LDPC code ensembles tend to zero. However, the free distance growth rates of the unterminated ensembles remain constant.
As $L \to \infty$ the minimum distance growth rates of terminated SC-LDPC code ensembles tend to zero. However, the free distance growth rates of the unterminated ensembles remain constant.

For large $L$, the strength of unterminated ensembles scales with the constraint length $\nu_s = M(m_s + 1)b_v$ and is independent of $L$. 
Distance Measures for SC-LDPC Codes

- As $L \to \infty$ the minimum distance growth rates of terminated SC-LDPC code ensembles tend to zero. However, the free distance growth rates of the unterminated ensembles remain constant.

- For large $L$, the strength of unterminated ensembles scales with the constraint length $\nu_s = M(m_s + 1)b_v$ and is independent of $L$.

- An appropriate distance measure for 'convolutional-like' terminated ensembles should be independent of $L$. 

An appropriate distance measure for 'convolutional-like' terminated ensembles should be independent of $L$. 

D. J. Costello, Jr., “Spatial Coupling vs. Block Coding: A Comparison”
Outline

- **LDPC Block Codes**
  - Parity-check matrix and Tanner graph representations, minimum distance bounds, iterative decoding thresholds, regular and irregular code designs, protograph-based constructions

- **Spatially Coupled LDPC Codes**
  - Protograph representation, edge-spreading construction, termination
  - Iterative decoding thresholds, threshold saturation, minimum distance

- **Practical Considerations**
  - Finite-length scaling, window decoding, performance, latency, and complexity comparisons to LDPC block codes, implementation aspects
Decoding SC-LDPC Codes

SC-LDPC codes can be decoded with standard iterative decoding schedules.
Decoding SC-LDPC Codes

- SC-LDPC codes can be decoded with standard iterative decoding schedules.

Reliable messages from the ends propagate through the graph toward the center as iterations proceed.
Decoding SC-LDPC Codes

- SC-LDPC codes can be decoded with standard iterative decoding schedules.

![Diagram showing messages passed and reliable messages from the ends propagating through the graph toward the center as iterations proceed.](image-url)
Decoding SC-LDPC Codes

- SC-LDPC codes can be decoded with standard iterative decoding schedules. Reliable messages from the ends propagate through the graph toward the center as iterations proceed.
Decoding SC-LDPC Codes

- SC-LDPC codes can be decoded with standard iterative decoding schedules.

- Reliable messages from the ends propagate through the graph toward the center as iterations proceed.

- The **highly localized (convolutional) structure** is well-suited for efficient decoding schedules that reduce memory and latency requirements.
Decoding SC-LDPC Codes

- SC-LDPC codes can be decoded with standard iterative decoding schedules. Reliable messages from the ends propagate through the graph toward the center as iterations proceed.

- The **highly localized (convolutional) structure** is well-suited for efficient decoding schedules that reduce memory and latency requirements.

- **Sliding window decoding (WD)** updates nodes only within a localized window and then the window shifts across the graph [Lentmaier et al '10, Iyengar et al '12].
Decoding SC-LDPC Codes

- SC-LDPC codes can be decoded with standard iterative decoding schedules.
- Reliable messages from the ends propagate through the graph toward the center as iterations proceed.
- The highly localized (convolutional) structure is well-suited for efficient decoding schedules that reduce memory and latency requirements.
- Sliding window decoding (WD) updates nodes only within a localized window and then the window shifts across the graph [Lentmaier et al '10, Iyengar et al '12].
Decoding SC-LDPC Codes

- SC-LDPC codes can be decoded with standard iterative decoding schedules.
  - Reliable messages from the ends propagate through the graph toward the center as iterations proceed.

- The **highly localized (convolutional) structure** is well-suited for efficient decoding schedules that reduce memory and latency requirements.
  - **Sliding window decoding** (WD) updates nodes only within a localized window and then the window shifts across the graph [Lentmaier et al '10, Iyengar et al '12].

D. J. Costello, Jr., “Spatial Coupling vs. Block Coding: A Comparison”
Window Decoding Performance

Latencies:
LDPC: $6M_{BC}$
SC-LDPC: $2M_{SC}W$

$M_{BC} = 1000$

Window Decoding Performance

Latencies:
- LDPC: $6M_{BC}$
- SC-LDPC: $2M_{SC}W$

For equal lifting factors, SC-LDPC codes display a large convolutional gain at the cost of increased latency.

Window Decoding Performance

Latencies:
- LDPC: $6M_{BC}$
- SC-LDPC: $2M_{SC}W$

For equal lifting factors, SC-LDPC codes display a large convolutional gain at the cost of increased latency.

For equal latency, SC-LDPC codes still display a significant performance gain.

Window Decoding Performance

Latencies:
LDPC: $6M_{BC}$
SC-LDPC: $2M_{SC}W$

- For equal lifting factors, SC-LDPC codes display a large **convolutional gain** at the cost of increased latency.
- For equal latency, SC-LDPC codes still display a significant **performance gain**.

For window decoding, SC-LDPC codes typically display a **large convolutional gain** at the cost of increased latency. For equal latency, SC-LDPC codes still display a significant **performance gain** over LDPC codes.

---

For the BEC, approximate analytical expressions obtained for the error probability of SC-LDPC codes compare well to simulated results.

For the BEC, approximate analytical expressions obtained for the error probability of SC-LDPC codes compare well to simulated results.

The scaling law is a useful engineering tool to gain insight into the design of SC-LDPC codes.

Equal Latency Comparison for (3,6)-Regular LDPC Codes

- Required $E_b/N_0$ to achieve a BER of $10^{-5}$ as a function of latency:

- **Latencies:**
  - LDPC: $4M_{BC}$
  - SC-LDPC: $2M_{SC}W$

- decreases as $W$ (and thus the latency) increases.
- does not decrease significantly beyond a certain $W$ ($W \approx 10$).
Equal Latency Comparison for (3,6)-Regular LDPC Codes

- Required $E_b/N_0$ to achieve a BER of $10^{-5}$ as a function of latency:

![Graph showing the required $E_b/N_0$ vs. decoding latency for different latencies.]

**Latencies:**
- LDPC: $4M_{BC}$
- SC-LDPC: $2M_{SC}W$

- decreases as $W$ (and thus the latency) increases.
- does not decrease significantly beyond a certain $W$ ($W \approx 10$).
Equal Latency Comparison for (3,6)-Regular LDPC Codes

- Required $E_b/N_0$ to achieve a BER of $10^{-5}$ as a function of latency:

Latencies:
- LDPC: $4M_{BC}$
- SC-LDPC: $2M_{SC}W$

- decreases as $W$ (and thus the latency) increases.
- does not decrease significantly beyond a certain $W$ ($W \approx 10$).
Equal Latency Comparison for (3,6)-Regular LDPC Codes

Required $E_b/N_0$ to achieve a BER of $10^{-5}$ as a function of latency:

Latencies:
- LDPC: $4M_{BC}$
- SC-LDPC: $2M_{SC}W$

- decreases as $W$ (and thus the latency) increases.
- does not decrease significantly beyond a certain $W$ ($W \approx 10$).
Equal Latency Comparison for (3,6)-Regular LDPC Codes

Required $E_b/N_0$ to achieve a BER of $10^{-5}$ as a function of latency:

Latencies:
- LDPC: $4M_{BC}$
- SC-LDPC: $2M_{SC}W$

- decreases as $W$ (and thus the latency) increases.
- does not decrease significantly beyond a certain $W$ ($W \approx 10$).
Required $E_b/N_0$ to achieve a BER of $10^{-5}$ as a function of latency:

**Latencies:**
- LDPC: $4M_{BC}$
- SC-LDPC: $2M_{SC}W$

- decreases as $W$ (and thus the latency) increases.
- does not decrease significantly beyond a certain $W$ ($W \approx 10$).
Required $E_b/N_0$ to achieve a BER of $10^{-5}$ as a function of latency:

- **Latencies:**
  - LDPC: $4M_{BC}$
  - SC-LDPC: $2M_{SC}W$

- As $W$ (and thus the latency) increases, it decreases.
- It does not decrease significantly beyond a certain $W$ ($W \approx 10$).

- When choosing parameters:
  - Large $M_{SC}$ improves code performance.
  - Large $W$ improves decoder performance.
For equal latency, SC-LDPC codes display a performance gain compared to the underlying LDPC-BCs.
Complexity Tradeoffs

- For equal latency, SC-LDPC codes display a performance gain compared to the underlying LDPC-BCs.

- With standard stopping rules, the computational complexity is higher for SC-LDPC codes.
For equal latency, SC-LDPC codes display a performance gain compared to the underlying LDPC-BCs.

With standard stopping rules, the computational complexity is higher for SC-LDPC codes.

LDPC-BCs cannot achieve equal performance by increasing the number of iterations.
For equal performance (BER of $10^{-5}$ at 1.5dB), SC-LDPC codes display a large reduction in latency compared to LDPC-BCs for similar complexity (including non-binary codes).
For equal performance (BER of $10^{-5}$ at 1.5dB), SC-LDPC codes display a large reduction in latency compared to LDPC-BCs for similar complexity.

With increasing (small) field sizes $q$, latency decreases for increasing complexity.
For equal performance (BER of $10^{-5}$ at 1.5dB), SC-LDPC codes display a large reduction in latency compared to LDPC-BCs for similar complexity.

- With increasing (small) field sizes $q$, latency decreases for increasing complexity.
- For larger $q$, both latency and complexity increase (for both SC-LDPC codes and LDPC-BCs)!

(including non-binary codes)
For equal performance (BER of $10^{-5}$ at 1.5dB), SC-LDPC codes display a large reduction in latency compared to LDPC-BCs for similar complexity.

- With increasing (small) field sizes $q$, latency decreases for increasing complexity.
- For larger $q$, both latency and complexity increase (for both SC-LDPC codes and LDPC-BCs)!
- SC-LDPC codes over GF(4) offer a good balance between complexity and latency.

(including non-binary codes)
Consider a comparison of a (3,6)-regular SC-LDPC code vs. an optimized irregular LDPC code with degree distribution

\[
\lambda(x) = 0.409x + 0.202x^2 + 0.0768x^3 + 0.1971x^6 + 0.1151x^7
\]

\[
\rho(x) = x^5
\]
Consider a comparison of a (3,6)-regular SC-LDPC code vs. an optimized irregular LDPC code with degree distribution

\[
\lambda(x) = 0.409x + 0.202x^2 + 0.0768x^3 + 0.1971x^6 + 0.1151x^7
\]
\[
\rho(x) = x^5
\]

The irregular ensemble has rate R=0.5004, BEC threshold \( \epsilon^* \approx 0.4810 \), and AWGNC threshold \( (E_b/N_0)^* \approx 0.4333 \text{ dB} \).
Consider a comparison of a (3,6)-regular SC-LDPC code vs. an optimized irregular LDPC code with degree distribution

\[ \lambda(x) = 0.409x + 0.202x^2 + 0.0768x^3 + 0.1971x^6 + 0.1151x^7 \]

\[ \rho(x) = x^5 \]

The irregular ensemble has rate R=0.5004, BEC threshold \( \epsilon^* \approx 0.4810 \), and AWGNC threshold \( (E_b/N_0)^* \approx 0.4333 \) dB.

We will compare this to a (3,6)-regular SC-LDPC code with L=50 and R=0.49. The corresponding window decoding thresholds are \( \epsilon^* \approx 0.4758 \) and \( (E_b/N_0)^* \approx 0.5925 \) dB.
On an equal latency basis, the regular SC-LDPC code outperforms the irregular LDPC-BC at BERs below $10^{-3}$. 

- SC-LDPC code
  - $W = 8000$
  - $I_{\text{max}} = 10$

- LDPC-BC
  - $n = 8000$
  - $I_{\text{max}} = 10$
  - $I_{\text{max}} = 100$
On an equal latency basis, the regular SC-LDPC code outperforms the irregular LDPC-BC at BERs below $10^{-3}$. The asymptotically good regular SC-LDPC code shows no sign of an error floor.
On an equal latency basis, the regular SC-LDPC code outperforms the irregular LDPC-BC at BERs below $10^{-3}$.

The asymptotically good regular SC-LDPC code shows no sign of an error floor.

The regular structure has implementation advantages.
Randomly Punctured LDPC Codes

- Solid line: SC-LDPC code, $I_{\text{max}} = 10$
- Dashed line: LDPC-BC, $I_{\text{max}} = 100$

Random puncturing can be applied to LDPC code ensembles to increase the rate.
Randomly Punctured LDPC Codes

- Random puncturing can be applied to LDPC code ensembles to increase the rate.
- Equal latency performance comparisons are consistent for higher rate ensembles.
- Regular SC-LDPC codes display robust decoding performance compared to irregular LDPC-BCs.

**Graph:**
- Solid line: SC-LDPC code, $I_{\text{max}} = 10$
- Dashed line: LDPC-BC, $I_{\text{max}} = 100$

**Legend:**
- $\alpha = 0, 0.10$
- $\alpha = 0.10$
- $\alpha = 0.26$
- $\alpha = 0.35$
Alternatively, we can couple irregular codes to construct an irregular SC-LDPC code ensemble. Consider the ARJA LDPC-BC protograph:

\[
\begin{bmatrix}
1 & 2 & 0 & 0 & 0 \\
0 & 3 & 1 & 1 & 1 \\
0 & 1 & 2 & 1 & 2
\end{bmatrix}
\]
An Irregular Example

- Alternatively, we can couple irregular codes to construct an irregular SC-LDPC code ensemble. Consider the ARJA LDPC-BC protograph:

\[
x \quad B = \begin{bmatrix}
1 & 2 & 0 & 0 & 0 \\
0 & 3 & 1 & 1 & 1 \\
0 & 1 & 2 & 1 & 2
\end{bmatrix}
\]

- A spatially coupled version can be created by edge spreading:
An Irregular Example

- Alternatively, we can couple irregular codes to construct an irregular SC-LDPC code ensemble. Consider the ARJA LDPC-BC protograph:

\[
B = \begin{bmatrix}
1 & 2 & 0 & 0 & 0 \\
0 & 3 & 1 & 1 & 1 \\
0 & 1 & 2 & 1 & 2
\end{bmatrix}
\]

- A spatially coupled version can be created by edge spreading:
An Irregular Example

- Alternatively, we can couple irregular codes to construct an irregular SC-LDPC code ensemble. Consider the ARJA LDPC-BC protograph:

\[
B = \begin{bmatrix}
1 & 2 & 0 & 0 & 0 \\
0 & 3 & 1 & 1 & 1 \\
0 & 1 & 2 & 1 & 2
\end{bmatrix}
\]

- A spatially coupled version can be created by edge spreading:
An Irregular Example

- Alternatively, we can couple irregular codes to construct an irregular SC-LDPC code ensemble. Consider the ARJA LDPC-BC protograph:

\[
B = \begin{bmatrix}
x & 1 & 2 & 0 & 0 & 0 \\
0 & 3 & 1 & 1 & 1 \\
0 & 1 & 2 & 1 & 2 \\
\end{bmatrix}
\]

- A spatially coupled version can be created by edge spreading:
An Irregular Example

- Alternatively, we can couple irregular codes to construct an irregular SC-LDPC code ensemble. Consider the ARJA LDPC-BC protograph:

\[ B = \begin{bmatrix}
1 & 2 & 0 & 0 & 0 \\
0 & 3 & 1 & 1 & 1 \\
0 & 1 & 2 & 1 & 2
\end{bmatrix} \]

- A spatially coupled version can be created by edge spreading:
An Irregular Example

- Alternatively, we can couple irregular codes to construct an irregular SC-LDPC code ensemble. Consider the ARJA LDPC-BC protograph:

\[
x \quad B = \begin{bmatrix}
1 & 2 & 0 & 0 & 0 \\
0 & 3 & 1 & 1 & 1 \\
0 & 1 & 2 & 1 & 2
\end{bmatrix}
\]

- A spatially coupled version can be created by edge spreading:
An Irregular Example

- Alternatively, we can couple irregular codes to construct an irregular SC-LDPC code ensemble. Consider the ARJA LDPC-BC protograph:

\[ B = \begin{bmatrix}
1 & 2 & 0 & 0 & 0 \\
0 & 3 & 1 & 1 & 1 \\
0 & 1 & 2 & 1 & 2
\end{bmatrix} \]

- A spatially coupled version can be created by edge spreading:
Irregular SC-LDPC code ensembles also display excellent properties.

Implementation Aspects

- As a result of their excellent performance and simple structure, regular SC-LDPC codes may be attractive for future coding standards. Several key features will require further investigation:
  - Hardware advantages of QC designs obtained by circulant liftings
  - Hardware advantages of the 'asymptotically-regular' structure
  - Design advantages of the flexible frame length feature obtained by varying $L$
  - Flexible rate feature obtained by puncturing
Implementation Aspects

- As a result of their excellent performance and simple structure, regular SC-LDPC codes may be attractive for future coding standards. Several key features will require further investigation:
  - Hardware advantages of QC designs obtained by circulant liftings
  - Hardware advantages of the 'asymptotically-regular' structure
  - Design advantages of the flexible frame length feature obtained by varying $L$
  - Flexible rate feature obtained by puncturing

- Of particular importance for applications requiring extremely low decoded bit error rates (e.g., optical communication, data storage) is an investigation of error floor issues related to stopping sets, trapping sets, and absorbing sets.
Conclusions

- Spatially coupled LDPC code ensembles achieve threshold saturation, i.e., their iterative decoding thresholds (for large $L$ and $M$) approach the MAP decoding thresholds of the underlying LDPC block code ensembles.

- The threshold saturation and linear minimum distance growth properties of $(J,K)$-regular SC-LDPC codes combine the best asymptotic features of both regular and irregular LDPC-BCs.

- With window decoding, SC-LDPC codes also compare favorably to LDPC-BCs in the finite-length regime, providing flexible tradeoffs between BER performance, decoding latency, and decoder complexity.