1. Background and Contributions

2. Constructions

3. Extensions
Background and Contributions
- Data storage is critical for non volatile memories.

- Number of storage nodes: \( n \)
- Reconstruction parameter: \( k \)
PMDS codes\(^1\) can tolerate node failures and additional symbol failures.

\(^1\)M. Blaum, J. L. Hafner, and S. Hetzler, “Partial-MDS codes and their application to RAID type of architectures”
Background

- PMDS code can only make use of $k$ nodes.
- The rest nodes are wasted.
- Each node downloading all symbols $\Rightarrow$ large latency.
Contributions

- We provide flexible PMDS code.

**Table 1:** An example of $(5, 3, 4, 2)$ flexible PMDS code with \{$(k_1, l_1), (k_2, l_2)$\} = \{(4, 3), (3, 4)$\}.

<table>
<thead>
<tr>
<th></th>
<th>$C_{1,1,1}$</th>
<th>*</th>
<th>$C_{1,1,3}$</th>
<th>*</th>
<th>*</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{1,2,1}$</td>
<td>*</td>
<td></td>
<td>$C_{1,2,3}$</td>
<td>$C_{1,2,4}$</td>
<td>*</td>
</tr>
<tr>
<td>$C_{1,3,1}$</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>$C_{1,3,4}$</td>
<td>*</td>
</tr>
<tr>
<td>$C_{2,1,1}$</td>
<td>*</td>
<td>$C_{2,1,3}$</td>
<td>$C_{2,1,4}$</td>
<td>*</td>
<td></td>
</tr>
</tbody>
</table>

- tolerate up to 2 node failures and 2 symbols failures
• With 1 node failure and 2 symbols failures, we download 3 symbols in each node.

**Table 2:** An example of (5, 3, 4, 2) flexible PMDS code with \(\{(k_1, \ell_1), (k_2, \ell_2)\} = \{(4, 3), (3, 4)\} \).

<table>
<thead>
<tr>
<th></th>
<th>(C_{1,1,1})</th>
<th>(C_{1,1,2})</th>
<th>(C_{1,1,3})</th>
<th>*</th>
<th>*</th>
</tr>
</thead>
<tbody>
<tr>
<td>(C_{1,2,1})</td>
<td>(C_{1,2,2})</td>
<td>(C_{1,2,3})</td>
<td>(C_{1,2,4})</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>(C_{1,3,1})</td>
<td>(C_{1,3,2})</td>
<td>*</td>
<td>(C_{1,3,4})</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>(C_{2,1,1})</td>
<td>(C_{2,1,2})</td>
<td>(C_{2,1,3})</td>
<td>(C_{2,1,4})</td>
<td>*</td>
<td>*</td>
</tr>
</tbody>
</table>
• [Blaum-Hafiner-Hetzler, 2013], first work for PMDS codes.
• [Calis-Koyluoglu, 2016], general constructions for all parameters.
• [Gabrys-Yaakobi-Blaum-Siegel, 2019], constructions for PMDS codes over small fields.
• [Blaum, 2020], a hierarchical architecture that can tolerate different number of symbols failures in different layers.
Constructions
- Based on [Calis-Koyluoglu, 2016].
- Original information $\Rightarrow$ Gabidulin codeword symbols $\Rightarrow$ MDS code in each layer

$\begin{array}{cccccc}
C_{1,1} & C_{1,2} & \cdots & \cdots & C_{1,k_1} & \text{parity} \\
C_{2,1} & C_{2,2} & \cdots & C_{2,k_2} & \text{parity} & \text{parity} \\
\vdots & \vdots & \ddots & \vdots & \text{parity} & \text{parity} \\
C_{a,1} & \cdots & C_{a,k_a} & \text{parity} & \text{parity} & \text{parity}
\end{array}$

- $n$: number of nodes. $k$: reconstruction parameter.
- $\ell$: number of symbols in each node.
- $k\ell = k_1\ell_1 = k_2\ell_2 = \ldots = k_a\ell_a$.
- $s$: number of additional symbol failures.
**Code Constructions**

\[
\begin{bmatrix}
C_{1,1,1} & C_{1,1,2} & \cdots & C_{1,1,n} \\
C_{1,2,1} & C_{1,2,2} & \cdots & C_{1,2,n} \\
\vdots & \vdots & \ddots & \vdots \\
C_{1,\ell_1,1} & C_{1,\ell_1,2} & \cdots & C_{1,\ell_1,n} \\
\vdots & \vdots & \ddots & \vdots \\
C_{a,1,1} & C_{a,1,2} & \cdots & C_{a,1,n} \\
C_{a,2,1} & C_{a,2,2} & \cdots & C_{a,2,n} \\
\vdots & \vdots & \ddots & \vdots \\
C_{a,\ell_a-\ell_a-1,1} & C_{a,\ell_a-\ell_a-1,2} & \cdots & C_{a,\ell_a-\ell_a-1,n}
\end{bmatrix}, \quad (1)
\]

- **Layer** $j$: $(\ell_{j-1} + 1)$-th row to $\ell_j$-th row.
- $C_{j,m_j,i}, j$: layer index, $m_j$: row index, $i$: node index.
Encoding

- **Original information → Gabidulin codeword symbols:**
  - $K = k\ell - s$ original information symbols.
  - $N = \sum_{j=1}^{a} k_j(\ell_j - \ell_{j-1})$ Gabidulin codeword symbols.

- **Gabidulin codeword symbols → MDS code in each layer:**
  - $(n, k_j)$ MDS code in each row.

$$\begin{bmatrix} C_{j,m_j,k_j+1}, \ldots, C_{j,m_j,n} \end{bmatrix} = \begin{bmatrix} C_{j,m_j,1}, \ldots, C_{j,m_j,k_j} \end{bmatrix} G_{n,k_j}$$
Table 3: An example of (5, 3, 4, 2) flexible PMDS code with 
\(\{(k_1, \ell_1), (k_2, \ell_2)\} = \{(4, 3), (3, 4)\}\).

<table>
<thead>
<tr>
<th></th>
<th>(C_{1,1,1})</th>
<th>(C_{1,1,2})</th>
<th>(C_{1,1,3})</th>
<th>(C_{1,1,4})</th>
<th>(C_{1,1,5})</th>
</tr>
</thead>
<tbody>
<tr>
<td>(C_{1,2,1}) &amp; (C_{1,2,2}) &amp; (C_{1,2,3}) &amp; (C_{1,2,4}) &amp; (C_{1,2,5})</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(C_{1,3,1}) &amp; (C_{1,3,2}) &amp; (C_{1,3,3}) &amp; (C_{1,3,4}) &amp; (C_{1,3,5})</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(C_{2,1,1}) &amp; (C_{2,1,2}) &amp; (C_{2,1,3}) &amp; (C_{2,1,4}) &amp; (C_{2,1,5})</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- 10 information symbols \(\rightarrow\) 15 Gabidulin codeword symbols.
- (5, 4) MDS code in row 1 \(\sim\) 3. (5, 3) MDS code in row 4.
Decoding

- $n - k_J$ node failures, $s$ symbol failures.
- MDS code in each layer $\rightarrow$ Gabidulin codeword symbols:
  - $t_{m_j} \leq k_J$ Gabidulin codeword symbols decoded in row $m_j$ of layer $j$.
  - Totally $K$ Gabidulin codeword symbols are decoded:
    \[
    \sum_{j=1}^{J} \sum_{m_j=1}^{\ell_j - \ell_{j-1}} t_{m_j} = \ell_J k_J - s = K.
    \]
- Gabidulin codeword symbols $\rightarrow$ Original information:
  - Original information can be decoded from $K$ Gabidulin codeword symbols.
Table 4: An example of \((5, 3, 4, 2)\) flexible PMDS code with \(\{(k_1, \ell_1), (k_2, \ell_2)\} = \{(4, 3), (3, 4)\}\).

<table>
<thead>
<tr>
<th>(C_{1,1,1})</th>
<th>(C_{1,1,2})</th>
<th>(C_{1,1,3})</th>
<th>*</th>
<th>*</th>
</tr>
</thead>
<tbody>
<tr>
<td>(C_{1,2,1})</td>
<td>(C_{1,2,2})</td>
<td>(C_{1,2,3})</td>
<td>(C_{1,2,4})</td>
<td>*</td>
</tr>
<tr>
<td>(C_{1,3,1})</td>
<td>(C_{1,3,2})</td>
<td>*</td>
<td>(C_{1,3,4})</td>
<td>*</td>
</tr>
<tr>
<td>(C_{2,1,1})</td>
<td>(C_{2,1,2})</td>
<td>(C_{2,1,3})</td>
<td>(C_{2,1,4})</td>
<td>*</td>
</tr>
</tbody>
</table>

- 1 node failure, row 1 \(\sim 3\) are read in each node.
- 10 Gabidulin codeword symbols can be decoded.
Decoding

Table 5: An example of $(5, 3, 4, 2)$ flexible PMDS code with 
$\{(k_1, \ell_1), (k_2, \ell_2)\} = \{(4, 3), (3, 4)\}$.

<table>
<thead>
<tr>
<th></th>
<th>$C_{1,1,1}$</th>
<th>*</th>
<th>$C_{1,1,3}$</th>
<th>*</th>
<th>*</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{1,2,1}$</td>
<td>*</td>
<td>$C_{1,2,3}$</td>
<td>$C_{1,2,4}$</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>$C_{1,3,1}$</td>
<td>*</td>
<td>*</td>
<td>$C_{1,3,4}$</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>$C_{2,1,1}$</td>
<td>*</td>
<td>$C_{2,1,3}$</td>
<td>$C_{2,1,4}$</td>
<td>*</td>
<td></td>
</tr>
</tbody>
</table>

- 2 node failures, all symbols are read in each node.
- 10 Gabidulin codeword symbols can be decoded.
• Assume 15 nodes, each node has failure probability $p = 0.2$, download 1 symbol takes time $t$.

<table>
<thead>
<tr>
<th>Fixed $(k, \ell)$</th>
<th>Fixed $(k, \ell)$</th>
<th>Fixed $(k, \ell)$</th>
<th>Flexible</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(8, 15)$</td>
<td>$(10, 12)$</td>
<td>$(12, 10)$</td>
<td>$(k_1, \ell_1), (k_2, \ell_2), (k_3, \ell_3)$</td>
</tr>
<tr>
<td>Probability of success</td>
<td>99.58%</td>
<td>93.89%</td>
<td>64.82%</td>
</tr>
<tr>
<td>Average latency</td>
<td>$15t$</td>
<td>$12t$</td>
<td>$10t$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$10.82t$</td>
</tr>
</tbody>
</table>
Extensions
• Similar construction can be applied to MDS codes, with optimal repair meeting MSR bound.
• LRC codes are also applied to the construction, with locality and flexible recovering nodes.