UNIVERSITY OF TARTU

Update and Repair Efficient Storage Codes With Availability via Finite Projective Planes

Junming Ke, Ago-Erik Riet

IEEE ISIT 2022

Finite Projective Planes

X: finite set of points,

$\mathcal{L}$: finite set of lines.

- Any two lines intersect in one point;
- Any two points on one line;
- Non-degenerate.

Two parallel lines will intersect at infinity.

$\mathrm{PG}(2, p)$ (Desarguesian finite projective plane):

- Points: 1-dimensional subspaces of $\mathbb{F}_p^3$.
- Lines: 2-dimensional subspaces.
- Each line contains $p + 1$ points, each point on $p + 1$ lines.

Incidence Matrix

Example: An incidence matrix A_3 of $\mathrm{PG}(2,3)$ is

$$A_3 = \begin{pmatrix} 1&1&0&0&1&0&1&0&0&0&0&0&0 \\ 0&1&1&0&0&1&0&1&0&0&0&0&0 \\ 0&0&1&1&0&0&1&0&1&0&0&0&0 \\ 0&0&0&1&1&0&0&1&0&1&0&0&0 \\ 0&0&0&0&1&1&0&0&1&0&1&0&0 \\ 0&0&0&0&0&1&1&0&0&1&0&1&0 \\ 0&0&0&0&0&0&1&1&0&0&1&0&1 \\ 1&0&0&0&0&0&0&1&1&0&0&1&0 \\ 0&1&0&0&0&0&0&0&1&1&0&0&1 \\ 1&0&1&0&0&0&0&0&0&1&1&0&0 \\ 0&1&0&1&0&0&0&0&0&0&1&1&0 \\ 0&0&1&0&1&0&0&0&0&0&0&1&1 \\ 1&0&0&1&0&1&0&0&0&0&0&0&1 \end{pmatrix}.$$

Incidence Matrix:

$$A_p = (a_{ij})$$

$$a_{ij} = \begin{cases} 1, \text{if the point } i \text{ is on the line } j \\ 0, \text{otherwise} \end{cases}$$

***p*-rank (the rank over $\mathbb{F}_p$) of the incidence matrix:**

In $\mathrm{PG}(2,p)$, p prime:

p-rank of the A_p is $\binom{p+1}{2} + 1 = \frac{p(p+1)}{2} + 1.$

The 3-rank of A_3 is $\frac{3(3+1)}{2} + 1 = 7.$

Moorhouse basis

Example: An incidence matrix A_3 of $\mathrm{PG}(2,3)$ is

$$A_3 = \begin{pmatrix} 1&1&0&0&1&0&1&0&0&0&0&0&0 \\ 0&1&1&0&0&1&0&1&0&0&0&0&0 \\ 0&0&1&1&0&0&1&0&1&0&0&0&0 \\ 0&0&0&1&1&0&0&1&0&1&0&0&0 \\ 0&0&0&0&1&1&0&0&1&0&1&0&0 \\ 0&0&0&0&0&1&1&0&0&1&0&1&0 \\ 0&0&0&0&0&0&1&1&0&0&1&0&1 \\ 1&0&0&0&0&0&0&1&1&0&0&1&0 \\ 0&1&0&0&0&0&0&0&1&1&0&0&1 \\ 1&0&1&0&0&0&0&0&0&1&1&0&0 \\ 0&1&0&1&0&0&0&0&0&0&1&1&0 \\ 0&0&1&0&1&0&0&0&0&0&0&1&1 \\ 1&0&0&1&0&1&0&0&0&0&0&0&1 \end{pmatrix}.$$

The rank of A_3 is $\frac{3(3+1)}{2} + 1 = 7$.

- Fix a row arbitrarily (here 1st row is fixed).

- Fix an arbitrary ordering of the positions where there is a one in that row (for example $a_1 = 7, a_2 = 1, a_3 = 5, a_4 = 2$).

- Remove some arbitrary $p + 1 - i$ rows with a one in position a_i (except for the fixed row).

This gives a row basis of the matrix A_p.
Any such row basis is known as a **Moorhouse basis**.

Moorhouse basis

Example: An incidence matrix A_3 of $\mathrm{PG}(2,3)$ is

$$A_3 = \begin{pmatrix} 1&1&0&0&1&0&1&0&0&0&0&0&0 \\ 0&1&1&0&0&1&0&1&0&0&0&0&0 \\ 0&0&1&1&0&0&1&0&1&0&0&0&0 \\ 0&0&0&1&1&0&0&1&0&1&0&0&0 \\ 0&0&0&0&1&1&0&0&1&0&1&0&0 \\ 0&0&0&0&0&1&1&0&0&1&0&1&0 \\ 0&0&0&0&0&0&1&1&0&0&1&0&1 \\ 1&0&0&0&0&0&0&1&1&0&0&1&0 \\ 0&1&0&0&0&0&0&0&1&1&0&0&1 \\ 1&0&1&0&0&0&0&0&0&1&1&0&0 \\ 0&1&0&1&0&0&0&0&0&0&1&1&0 \\ 0&0&1&0&1&0&0&0&0&0&0&1&1 \\ 1&0&0&1&0&1&0&0&0&0&0&0&1 \end{pmatrix}.$$

The rank of A_3 is $\frac{3(3+1)}{2} + 1 = 7$.

- Fix a row arbitrarily (here 1st row is fixed).

- Fix an arbitrary ordering of the positions where there is a one in that row (for example $a_1 = 7, a_2 = 1, a_3 = 5, a_4 = 2$).

- Remove some arbitrary $p + 1 - i$ rows with a one in position a_i (except for the fixed row).

This gives a row basis of the matrix A_p.
Any such row basis is known as a **Moorhouse basis**.

Moorhouse basis

Example: An incidence matrix A_3 of PG$(2,3)$ is

$$A_3 = \begin{pmatrix} 1&1&0&0&1&0&1&0&0&0&0&0&0 \\ 0&1&1&0&0&1&0&1&0&0&0&0&0 \\ 0&0&1&1&0&0&1&0&1&0&0&0&0 \\ 0&0&0&1&1&0&0&1&0&1&0&0&0 \\ 0&0&0&0&1&1&0&0&1&0&1&0&0 \\ 0&0&0&0&0&1&1&0&0&1&0&1&0 \\ 0&0&0&0&0&0&1&1&0&0&1&0&1 \\ 1&0&0&0&0&0&0&1&1&0&0&1&0 \\ 0&1&0&0&0&0&0&0&1&1&0&0&1 \\ 1&0&1&0&0&0&0&0&0&1&1&0&0 \\ 0&1&0&1&0&0&0&0&0&0&1&1&0 \\ 0&0&1&0&1&0&0&0&0&0&0&1&1 \\ 1&0&0&1&0&1&0&0&0&0&0&0&1 \end{pmatrix}.$$

The rank of A_3 is $\frac{3(3+1)}{2} + 1 = 7$.

A Moorhouse basis

$$H_3 = \begin{pmatrix} 1&1&0&0&1&0&1&0&0&0&0&0&0 \\ 0&1&1&0&0&1&0&1&0&0&0&0&0 \\ 0&0&0&1&1&0&0&1&0&1&0&0&0 \\ 0&0&0&0&1&1&0&0&1&0&1&0&0 \\ 1&0&0&0&0&0&0&1&1&0&0&1&0 \\ 0&1&0&0&0&0&0&0&1&1&0&0&1 \\ 0&1&0&1&0&0&0&0&0&0&1&1&0 \end{pmatrix}$$

- Fix a row arbitrarily (here 1st row is fixed).
- Fix an arbitrary ordering of the positions where there is a one in that row (for example $a_1 = 7, a_2 = 1, a_3 = 5, a_4 = 2$).
- Remove some arbitrary $p + 1 - i$ rows with a one in position a_i (except for the fixed row).

This gives a row basis of the matrix A_p.
Any such row basis is known as a **Moorhouse basis**.

Moorhouse basis

Example: An incidence matrix A_3 of PG$(2,3)$ is

$$A_3 = \begin{pmatrix} 1&1&0&0&1&0&1&0&0&0&0&0&0\\ 0&1&1&0&0&1&0&1&0&0&0&0&0\\ 0&0&1&1&0&0&1&0&1&0&0&0&0\\ 0&0&0&1&1&0&0&1&0&1&0&0&0\\ 0&0&0&0&1&1&0&0&1&0&1&0&0\\ 0&0&0&0&0&1&1&0&0&1&0&1&0\\ 0&0&0&0&0&0&1&1&0&0&1&0&1\\ 1&0&0&0&0&0&0&1&1&0&0&1&0\\ 0&1&0&0&0&0&0&0&1&1&0&0&1\\ 1&0&1&0&0&0&0&0&0&1&1&0&0\\ 0&1&0&1&0&0&0&0&0&0&1&1&0\\ 0&0&1&0&1&0&0&0&0&0&0&1&1\\ 1&0&0&1&0&1&0&0&0&0&0&0&1 \end{pmatrix}.$$

The rank of A_3 is $\frac{3(3+1)}{2} + 1 = 7$.

Another Moorhouse basis

$$H_3 = \begin{pmatrix} 1&1&0&0&1&0&1&0&0&0&0&0&0\\ 0&1&1&0&0&1&0&1&0&0&0&0&0\\ 0&0&1&1&0&0&1&0&1&0&0&0&0\\ 0&0&0&1&1&0&0&1&0&1&0&0&0\\ 0&0&0&0&1&1&0&0&1&0&1&0&0\\ 0&0&0&0&0&1&1&0&0&1&0&1&0\\ 0&0&0&0&0&0&1&1&0&0&1&0&1 \end{pmatrix}.$$

- Fix a row arbitrarily (here 1st row is fixed).

- Fix an arbitrary ordering of the positions where there is a one in that row (for example $a_1 = 1, a_2 = 2, a_3 = 5, a_4 = 7$).

- Remove some arbitrary $p + 1 - i$ rows with a one in position a_i (except for the fixed row).

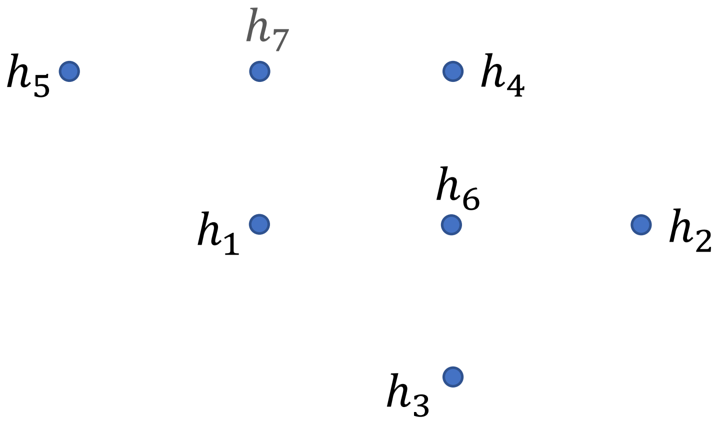
This gives a row basis of the matrix A_p.
Any such row basis is known as a **Moorhouse basis**.

Generator matrix G

$$H_3 = \begin{pmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \end{pmatrix} = \begin{pmatrix} 1&1&0&0&1&0&1&0&0&0&0&0&0 \\ 0&1&1&0&0&1&0&1&0&0&0&0&0 \\ 0&0&1&1&0&0&1&0&1&0&0&0&0 \\ 0&0&0&1&1&0&0&1&0&1&0&0&0 \\ 0&0&0&0&1&1&0&0&1&0&1&0&0 \\ 0&0&0&0&0&1&1&0&0&1&0&1&0 \\ 0&0&0&0&0&0&1&1&0&0&1&0&1 \end{pmatrix}$$

H_3 used to construct generator matrix G_3

1. The vertex set of the graph is the set of rows of H.

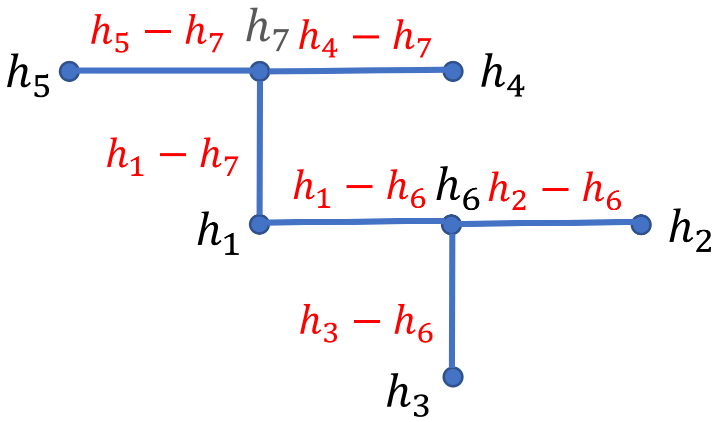


Generator matrix G

$$H_3 = \begin{pmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \end{pmatrix} = \begin{pmatrix} 1&1&0&0&1&0&1&0&0&0&0&0&0 \\ 0&1&1&0&0&1&0&1&0&0&0&0&0 \\ 0&0&1&1&0&0&1&0&1&0&0&0&0 \\ 0&0&0&1&1&0&0&1&0&1&0&0&0 \\ 0&0&0&0&1&1&0&0&1&0&1&0&0 \\ 0&0&0&0&0&1&1&0&0&1&0&1&0 \\ 0&0&0&0&0&0&1&1&0&0&1&0&1 \end{pmatrix}$$

H_3 used to construct generator matrix G_3

1. The vertex set of the graph is the set of rows of H.
2. Fix an arbitrary tree on the vertex set, i.e. a connected graph without cycles.

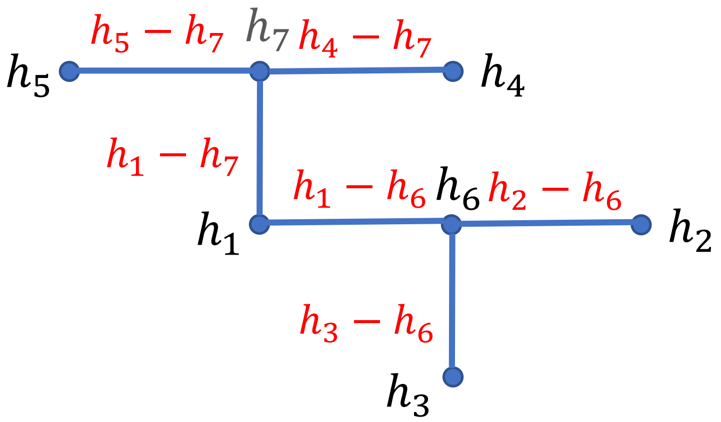


Generator matrix G

$$H_3 = \begin{pmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \end{pmatrix} = \begin{pmatrix} 1&1&0&0&1&0&1&0&0&0&0&0&0 \\ 0&1&1&0&0&1&0&1&0&0&0&0&0 \\ 0&0&1&1&0&0&1&0&1&0&0&0&0 \\ 0&0&0&1&1&0&0&1&0&1&0&0&0 \\ 0&0&0&0&1&1&0&0&1&0&1&0&0 \\ 0&0&0&0&0&1&1&0&0&1&0&1&0 \\ 0&0&0&0&0&0&1&1&0&0&1&0&1 \end{pmatrix}$$

1. The vertex set of the graph is the set of rows of H.
2. Fix an arbitrary tree on the vertex set, i.e. a connected graph without cycles.
3. A row of G corresponds to an edge, and it is the difference of its endpoints (rows of H).

H_3 used to construct generator matrix G_3



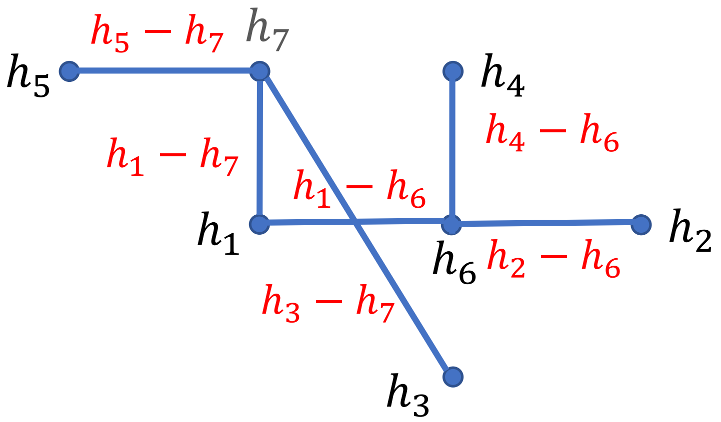
$$G_3^{(1)} = \begin{pmatrix} h_1 - h_6 \\ h_1 - h_7 \\ h_2 - h_6 \\ h_3 - h_6 \\ h_4 - h_7 \\ h_5 - h_7 \end{pmatrix} = \begin{pmatrix} 1&1&0&0&1&-1&0&0&0&-1&0&-1&0 \\ 1&1&0&0&1&0&0&-1&0&0&-1&0&-1 \\ 0&1&1&0&0&0&-1&1&0&-1&0&-1&0 \\ 0&0&1&1&0&-1&0&0&1&-1&0&-1&0 \\ 0&0&0&1&1&0&-1&0&0&1&-1&0&-1 \\ 0&0&0&0&1&1&-1&-1&1&0&0&0&-1 \end{pmatrix}$$

Generator matrix G

$$H_3 = \begin{pmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \end{pmatrix} = \begin{pmatrix} 1&1&0&0&1&0&1&0&0&0&0&0&0 \\ 0&1&1&0&0&1&0&1&0&0&0&0&0 \\ 0&0&1&1&0&0&1&0&1&0&0&0&0 \\ 0&0&0&1&1&0&0&1&0&1&0&0&0 \\ 0&0&0&0&1&1&0&0&1&0&1&0&0 \\ 0&0&0&0&0&1&1&0&0&1&0&1&0 \\ 0&0&0&0&0&0&1&1&0&0&1&0&1 \end{pmatrix}$$

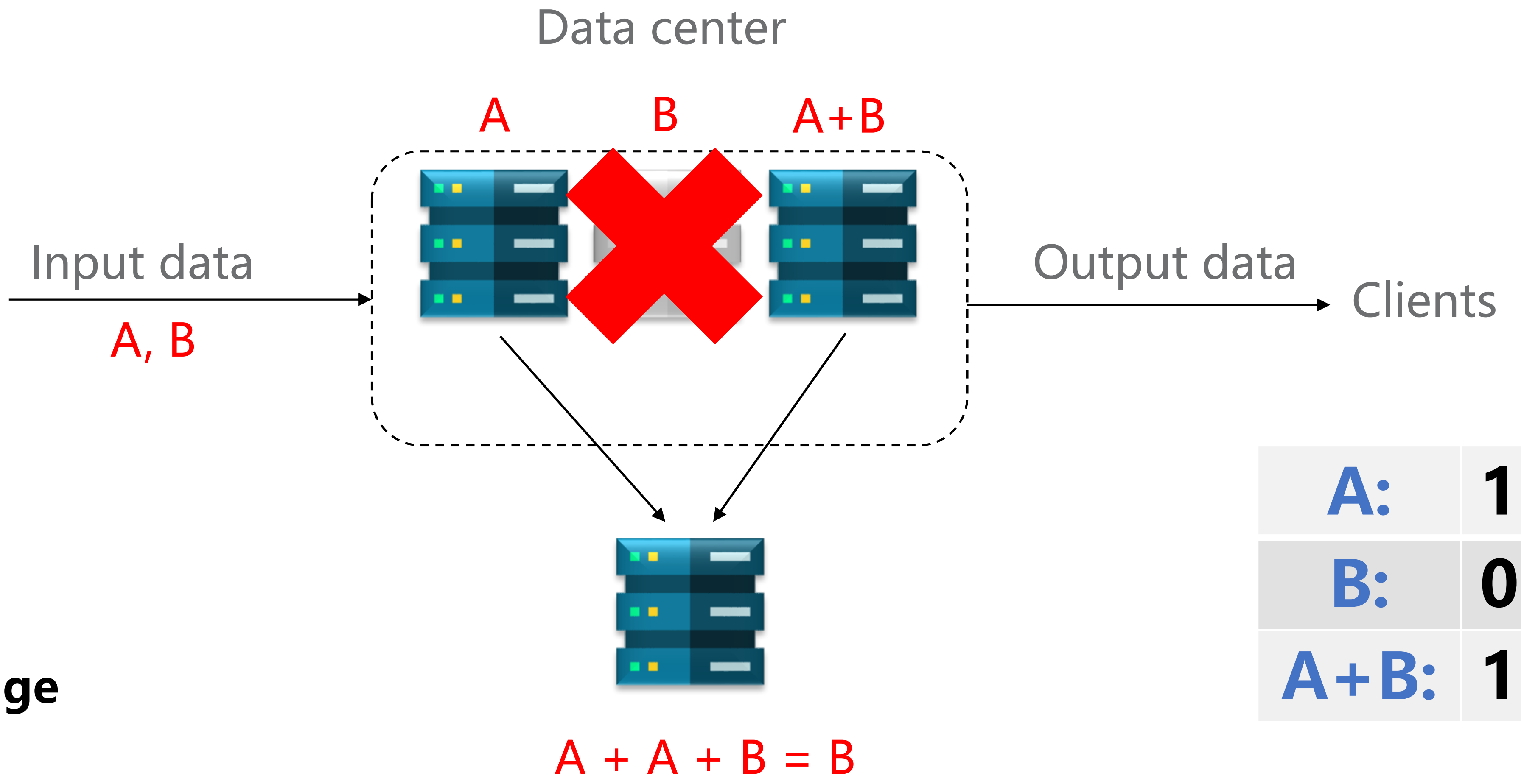
1. The vertex set of the graph is the set of rows of H.
2. Fix an arbitrary tree on the vertex set, i.e. a connected graph without cycles.
3. A row of G corresponds to an edge, and it is the difference of its endpoints (rows of H).

H_3 used to construct generator matrix G_3



$$G_3^{(2)} = \begin{pmatrix} h_1 - h_6 \\ h_1 - h_7 \\ h_2 - h_6 \\ h_4 - h_6 \\ h_3 - h_7 \\ h_5 - h_7 \end{pmatrix} = \begin{pmatrix} 1&1&0&0&1&-1&0&0&0&-1&0&-1&0 \\ 1&1&0&0&1&0&0&-1&0&0&-1&0&-1 \\ 0&1&1&0&0&0&-1&1&0&-1&0&-1&0 \\ 0&0&1&1&0&0&0&-1&1&0&-1&0&-1 \\ 0&0&0&1&1&-1&-1&1&0&0&0&-1&0 \\ 0&0&0&0&1&1&-1&-1&1&0&0&0&-1 \end{pmatrix}$$

Repair in coded storage



Coded Storage

A:	10
B:	01
A+B:	11

Repair locality

Repair group for a position: set of other positions that can reconstruct its information.

Repair locality of a position: size of its smallest repair group.

For a code from $\mathrm{PG}(2, p)$:

Codeword length is $p^2 + p + 1$.

$p + 1$ points on a line means repair locality of every symbol is p.

Example: An incidence matrix A_3 of $\mathrm{PG}(2,3)$ is

$$A_3 = \begin{pmatrix} 1&1&0&0&1&0&1&0&0&0&0&0&0 \\ 0&1&1&0&0&1&0&1&0&0&0&0&0 \\ 0&0&1&1&0&0&1&0&1&0&0&0&0 \\ 0&0&0&1&1&0&0&1&0&1&0&0&0 \\ 0&0&0&0&1&1&0&0&1&0&1&0&0 \\ 0&0&0&0&0&1&1&0&0&1&0&1&0 \\ 0&0&0&0&0&0&1&1&0&0&1&0&1 \\ 1&0&0&0&0&0&0&1&1&0&0&1&0 \\ 0&1&0&0&0&0&0&0&1&1&0&0&1 \\ 1&0&1&0&0&0&0&0&0&1&1&0&0 \\ 0&1&0&1&0&0&0&0&0&0&1&1&0 \\ 0&0&1&0&1&0&0&0&0&0&0&1&1 \\ 1&0&0&1&0&1&0&0&0&0&0&0&1 \end{pmatrix}.$$

$(b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_9, b_{10}, b_{11}, b_{12}, b_{13})$

From the first row,
For coded symbol b_5, a repair group is $\{b_1, b_2, b_7\}$;
For coded symbol b_1, a repair group is $\{b_2, b_5, b_7\}$;
For coded symbol b_7, a repair group is $\{b_1, b_2, b_5\}$;
$\ldots$

Repair availability

Repair availability: maximum number of disjoint repair groups.

The $p+1$ lines through a point form $p+1$ **disjoint** repair groups for the corresponding position.

Example: An incidence matrix A_3 of $\mathrm{PG}(2,3)$ is

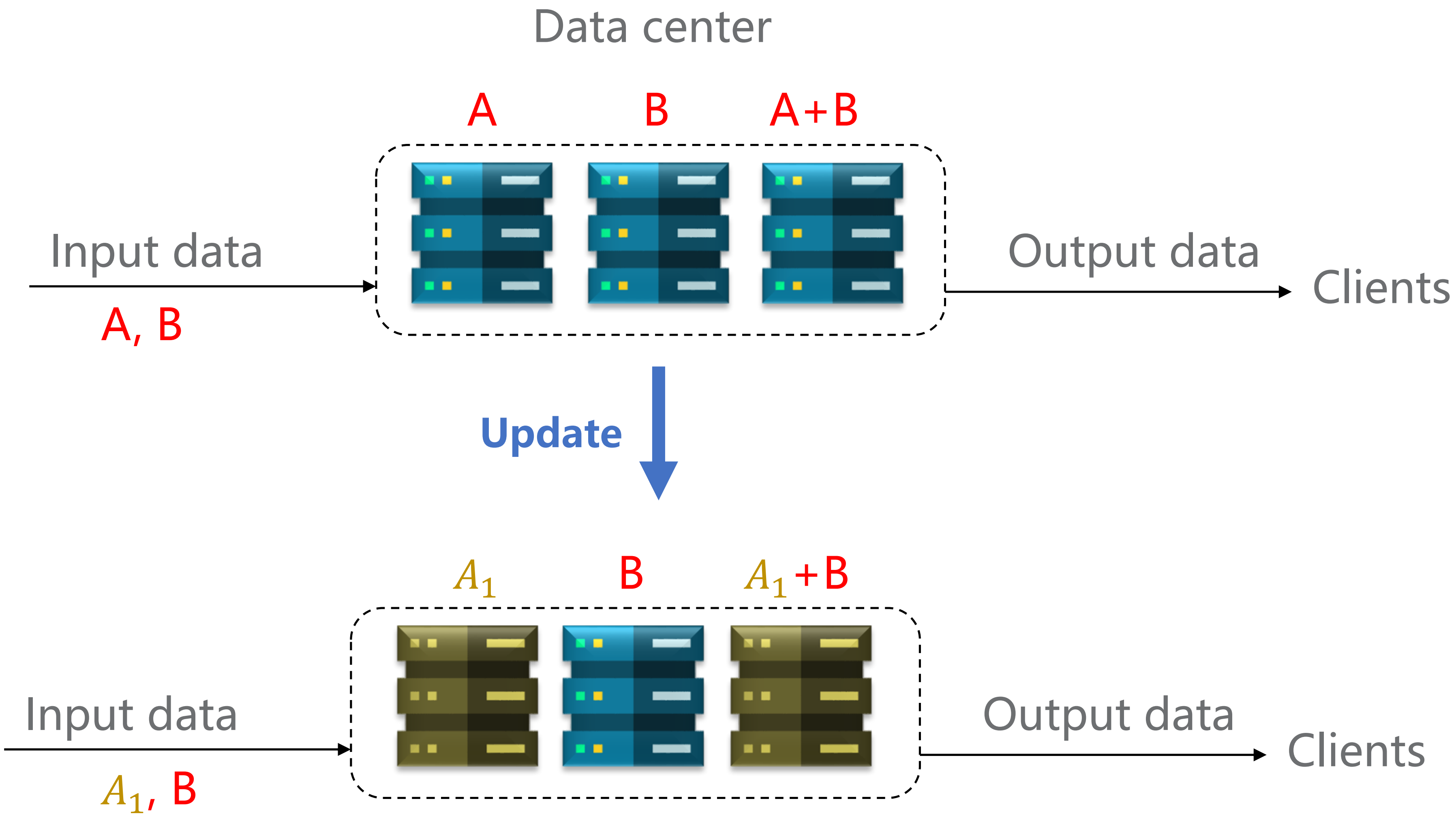
$$A_3 = \begin{pmatrix}
\boxed{1\ 1\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0} \\
\boxed{0\ 1\ 1\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 0\ 0} \\
0\ 0\ 1\ 1\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 0 \\
0\ 0\ 0\ 1\ 1\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 0 \\
0\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 1\ 0\ 1\ 0\ 0 \\
0\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 1\ 0\ 1\ 0 \\
0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 1\ 0\ 1 \\
1\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 1\ 0 \\
\boxed{0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 1} \\
1\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0 \\
\boxed{0\ 1\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0} \\
0\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1 \\
1\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 1
\end{pmatrix}.$$

$(b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_9, b_{10}, b_{11}, b_{12}, b_{13})$

For coded symbol b_2,
The repair groups are:
$\{b_1, b_5, b_7\}$ by looking at row 1;
$\{b_3, b_6, b_8\}$ by looking at row 2;
$\{b_9, b_{10}, b_{13}\}$ by looking at row 9;
$\{b_4, b_{11}, b_{12}\}$ by looking at row 11.

UNIVERSITY OF TARTU 1632

Update in coded storage



A:	10
B:	01
A+B:	11
A_1:	11
A_1+B:	10

Update efficiency

Update efficiency: the number of coded symbols that need to be updated when updating a data symbol.

The update efficiency of the code: the maximum update efficiency for each data symbol.

For a code from PG$(2, p)$:

Codeword length is $p^2 + p + 1$.

The update efficiency is $2p$.

$$G_3^{(1)} = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & -1 & 0 & -1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & -1 \\ 0 & 1 & 1 & 0 & 0 & 0 & -1 & 1 & 0 & -1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 1 & 0 & -1 & 0 & 0 & 1 & -1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & -1 & 0 & 0 & 1 & -1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & 1 & 0 & 0 & 0 & -1 \end{pmatrix}$$

Delta update:

Data symbols: $(a_1, a_2, a_3, a_4, a_5, a_6)$

Coded symbols: Data symbols$\times$ $G_3^{(1)} =$

$(b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_9, b_{10}, b_{11}, b_{12}, b_{13})$

Updated data symbols:

$(a_1, a_2, a_3, a_4, a_5, a_6) + (\Delta_1, 0,0,0,0,0)$

Updated Coded symbols:

$(b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_9, b_{10}, b_{11}, b_{12}, b_{13})$

$+(\Delta_1, \Delta_1, 0,0, \Delta_1, -\Delta_1, 0,0,0, -\Delta_1, 0, -\Delta_1, 0)$

UNIVERSITY OF TARTU

Choice of G

$$H_3 = \begin{pmatrix} 1&1&0&0&1&0&1&0&0&0&0&0&0 \\ 0&1&1&0&0&1&0&1&0&0&0&0&0 \\ 0&0&1&1&0&0&1&0&1&0&0&0&0 \\ 0&0&0&1&1&0&0&1&0&1&0&0&0 \\ 0&0&0&0&1&1&0&0&1&0&1&0&0 \\ 0&0&0&0&0&1&1&0&0&1&0&1&0 \\ 0&0&0&0&0&0&1&1&0&0&1&0&1 \end{pmatrix}$$

$$G_3^{(1)} = \begin{pmatrix} 1&1&0&0&1&-1&0&0&0&-1&0&-1&0 \\ 1&1&0&0&1&0&0&-1&0&0&-1&0&-1 \\ 0&1&1&0&0&0&-1&1&0&-1&0&-1&0 \\ 0&0&1&1&0&-1&0&0&1&-1&0&-1&0 \\ 0&0&0&1&1&0&-1&0&0&1&-1&0&-1 \\ 0&0&0&0&1&1&-1&-1&1&0&0&0&-1 \end{pmatrix}$$

$$G_3^{(2)} = \begin{pmatrix} 1&1&0&0&1&-1&0&0&0&-1&0&-1&0 \\ 1&1&0&0&1&0&0&-1&0&0&-1&0&-1 \\ 0&1&1&0&0&0&-1&1&0&-1&0&-1&0 \\ 0&0&1&1&0&0&0&-1&1&0&-1&0&-1 \\ 0&0&0&1&1&-1&-1&1&0&0&0&-1&0 \\ 0&0&0&0&1&1&-1&-1&1&0&0&0&-1 \end{pmatrix}$$

Data symbols	$(a_1, a_2, a_3, a_4, a_5, a_6)$
Update frequency of data symbols during a time unit	1) (2,2,2,2,2,2)
	2) (1,4,1,4,1,1)
	3) (4,1,1,4,1,1)

For example, (1,4,1,4,1,1) means
$\boldsymbol{a_1}$ updates once,
$\boldsymbol{a_2}$ updates four times,
$\boldsymbol{a_3}$ updates once,
$\boldsymbol{a_4}$ updates four times,
$\boldsymbol{a_5}$ updates once,
$\boldsymbol{a_6}$ updates once,
during a time unit.

Update frequencies (coded symbols) for (1,4,1,4,1,1):
$G_3^{(1)}$: $(5, 6, 5, 5, \mathbf{7}, 6, 3, 6, 5, \mathbf{7}, 5, 6, 6)$,
$G_3^{(2)}$: $(5, 6, 5, 5, 7, 3, 3, \mathbf{11}, 5, 2, 8, 3, 9)$.

Choice of G

$$H_3 = \begin{pmatrix} 1&1&0&0&1&0&1&0&0&0&0&0&0\\ 0&1&1&0&0&1&0&1&0&0&0&0&0\\ 0&0&1&1&0&0&1&0&1&0&0&0&0\\ 0&0&0&1&1&0&0&1&0&1&0&0&0\\ 0&0&0&0&1&1&0&0&1&0&1&0&0\\ 0&0&0&0&0&1&1&0&0&1&0&1&0\\ 0&0&0&0&0&0&1&1&0&0&1&0&1 \end{pmatrix}$$

$$G_3^{(1)} = \begin{pmatrix} 1&1&0&0&1&-1&0&0&0&-1&0&-1&0\\ 1&1&0&0&1&0&0&-1&0&0&-1&0&-1\\ 0&1&1&0&0&0&-1&1&0&-1&0&-1&0\\ 0&0&1&1&0&-1&0&0&1&-1&0&-1&0\\ 0&0&0&1&1&0&-1&0&0&1&-1&0&-1\\ 0&0&0&0&1&1&-1&-1&1&0&0&0&-1 \end{pmatrix}$$

$$G_3^{(2)} = \begin{pmatrix} 1&1&0&0&1&-1&0&0&0&-1&0&-1&0\\ 1&1&0&0&1&0&0&-1&0&0&-1&0&-1\\ 0&1&1&0&0&0&-1&1&0&-1&0&-1&0\\ 0&0&1&1&0&0&0&-1&1&0&-1&0&-1\\ 0&0&0&1&1&-1&-1&1&0&0&0&-1&0\\ 0&0&0&0&1&1&-1&-1&1&0&0&0&-1 \end{pmatrix}$$

Data symbols	$(a_1, a_2, a_3, a_4, a_5, a_6)$
Update frequency of data symbols during a time unit	1) (2,2,2,2,2,2)
	2) (1,4,1,4,1,1)
	3) (4,1,1,4,1,1)

The maximum update frequencies of coded symbols

$G_3^{(1)}$:

1) 8 (preferable)

2) 7 (preferable)

3) 10

$G_3^{(2)}$:

1) 10

2) 11

3) 8 (preferable)

Circulant structure

Circulant structure:

Rows of the matrix are the cyclic shifts of one row.

The incidence matrix of $\mathrm{PG}(2, p)$ always has the circulant structure (from the Singer cycle).

Example with $p = 2$:

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

Short description (sparse $0-1$ vector):

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

Repair algorithm

Algorithm A:

While possible, do:

- Find a projective line with exactly one erased coded symbol,
- Repair this coded symbol from the other points of the line.

Specifically, the coded symbol is repaired to minus the sum of the other coded symbols of the line (over $\mathbb{F}_p$).

$$A_3 = \begin{pmatrix} 1&1&0&0&1&0&1&0&0&0&0&0&0 \\ 0&1&1&0&0&1&0&1&0&0&0&0&0 \\ 0&0&1&1&0&0&1&0&1&0&0&0&0 \\ 0&0&0&1&1&0&0&1&0&1&0&0&0 \\ 0&0&0&0&1&1&0&0&1&0&1&0&0 \\ 0&0&0&0&0&1&1&0&0&1&0&1&0 \\ 0&0&0&0&0&0&1&1&0&0&1&0&1 \\ 1&0&0&0&0&0&0&1&1&0&0&1&0 \\ 0&1&0&0&0&0&0&0&1&1&0&0&1 \\ 1&0&1&0&0&0&0&0&0&1&1&0&0 \\ 0&1&0&1&0&0&0&0&0&0&1&1&0 \\ 0&0&1&0&1&0&0&0&0&0&0&1&1 \\ 1&0&0&1&0&1&0&0&0&0&0&0&1 \end{pmatrix}.$$

$(b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_9, b_{10}, b_{11}, b_{12}, b_{13})$

Example:

Erased coded symbols is

$\{b_1, b_2, b_3, b_4, b_8\}$

UNIVERSITY OF TARTU

1632

Repair algorithm

Algorithm A:

While possible, do:

- Find a projective line with exactly one erased coded symbol,
- Repair this coded symbol from the other points of the line.

Specifically, the coded symbol is repaired to minus the sum of the other coded symbols of the line (over $\mathbb{F}_p$).

$$A_3 = \begin{pmatrix} 1&1&0&0&1&0&1&0&0&0&0&0&0\\ 0&1&1&0&0&1&0&1&0&0&0&0&0\\ 0&0&1&1&0&0&1&0&1&0&0&0&0\\ 0&0&0&1&1&0&0&1&0&1&0&0&0\\ 0&0&0&0&1&1&0&0&1&0&1&0&0\\ 0&0&0&0&0&1&1&0&0&1&0&1&0\\ 0&0&0&0&0&0&1&1&0&0&1&0&1\\ 1&0&0&0&0&0&0&1&1&0&0&1&0\\ 0&1&0&0&0&0&0&0&1&1&0&0&1\\ 1&0&1&0&0&0&0&0&0&1&1&0&0\\ 0&1&0&1&0&0&0&0&0&0&1&1&0\\ 0&0&1&0&1&0&0&0&0&0&0&1&1\\ 1&0&0&1&0&1&0&0&0&0&0&0&1 \end{pmatrix}.$$

$(b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_9, b_{10}, b_{11}, b_{12}, b_{13})$

At first,

b_1 cannot be repaired, (each repair group contains an erased coded symbol, see the red arrows.)

Similarly, b_4 cannot be repaired.

Repair algorithm

Algorithm A:

While possible, do:

- Find a projective line with exactly one erased coded symbol,
- Repair this coded symbol from the other points of the line.

Specifically, the coded symbol is repaired to minus the sum of the other coded symbols of the line (over $\mathbb{F}_p$).

$$A_3 = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

$(b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_9, b_{10}, b_{11}, b_{12}, b_{13})$

Repair algorithm

Algorithm A:

While possible, do:

- Find a projective line with exactly one erased coded symbol,
- Repair this coded symbol from the other points of the line.

Specifically, the coded symbol is repaired to minus the sum of the other coded symbols of the line (over $\mathbb{F}_p$).

$$A_3 = \begin{pmatrix}
1&1&0&0&1&0&1&0&0&0&0&0&0\\
0&1&1&0&0&1&0&1&0&0&0&0&0\\
0&0&1&1&0&0&1&0&1&0&0&0&0\\
0&0&0&1&1&0&0&1&0&1&0&0&0\\
0&0&0&0&1&1&0&0&1&0&1&0&0\\
0&0&0&0&0&1&1&0&0&1&0&1&0\\
0&0&0&0&0&0&1&1&0&0&1&0&1\\
1&0&0&0&0&0&0&1&1&0&0&1&0\\
0&1&0&0&0&0&0&0&1&1&0&0&1\\
1&0&1&0&0&0&0&0&0&1&1&0&0\\
0&1&0&1&0&0&0&0&0&0&1&1&0\\
0&0&1&0&1&0&0&0&0&0&0&1&1\\
1&0&0&1&0&1&0&0&0&0&0&0&1
\end{pmatrix}.$$

$(b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_9, b_{10}, b_{11}, b_{12}, b_{13})$

Repair algorithm

Algorithm A:

While possible, do:

- Find a projective line with exactly one erased coded symbol,
- Repair this coded symbol from the other points of the line.

Specifically, the coded symbol is repaired to minus the sum of the other coded symbols of the line (over $\mathbb{F}_q$).

$$A_3 = \begin{pmatrix} 1&1&0&0&1&0&1&0&0&0&0&0&0 \\ 0&1&1&0&0&1&0&1&0&0&0&0&0 \\ 0&0&1&1&0&0&1&0&1&0&0&0&0 \\ 0&0&0&1&1&0&0&1&0&1&0&0&0 \\ 0&0&0&0&1&1&0&0&1&0&1&0&0 \\ 0&0&0&0&0&1&1&0&0&1&0&1&0 \\ 0&0&0&0&0&0&1&1&0&0&1&0&1 \\ 1&0&0&0&0&0&0&1&1&0&0&1&0 \\ 0&1&0&0&0&0&0&0&1&1&0&0&1 \\ 1&0&1&0&0&0&0&0&0&1&1&0&0 \\ 0&1&0&1&0&0&0&0&0&0&1&1&0 \\ 0&0&1&0&1&0&0&0&0&0&0&1&1 \\ 1&0&0&1&0&1&0&0&0&0&0&0&1 \end{pmatrix}.$$

$(b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_9, b_{10}, b_{11}, b_{12}, b_{13})$

Repair algorithm

Algorithm A:

While possible, do:

- Find a projective line with exactly one erased coded symbol,
- Repair this coded symbol from the other points of the line.

Specifically, the coded symbol is repaired to minus the sum of the other coded symbols of the line (over $\mathbb{F}_p$).

$$A_3 = \begin{pmatrix}
1&1&0&0&1&0&1&0&0&0&0&0&0\\
0&1&1&0&0&1&0&1&0&0&0&0&0\\
0&0&1&1&0&0&1&0&1&0&0&0&0\\
0&0&0&1&1&0&0&1&0&1&0&0&0\\
0&0&0&0&1&1&0&0&1&0&1&0&0\\
0&0&0&0&0&1&1&0&0&1&0&1&0\\
0&0&0&0&0&0&1&1&0&0&1&0&1\\
1&0&0&0&0&0&0&1&1&0&0&1&0\\
0&1&0&0&0&0&0&0&1&1&0&0&1\\
1&0&1&0&0&0&0&0&0&1&1&0&0\\
0&1&0&1&0&0&0&0&0&0&1&1&0\\
0&0&1&0&1&0&0&0&0&0&0&1&1\\
1&0&0&1&0&1&0&0&0&0&0&0&1
\end{pmatrix}.$$

$(b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_9, b_{10}, b_{11}, b_{12}, b_{13})$

Repair algorithm

Algorithm A:

While possible, do:

- Find a projective line with exactly one erased coded symbol,
- Repair this coded symbol from the other points of the line.

Specifically, the coded symbol is repaired to minus the sum of the other coded symbols of the line (over $\mathbb{F}_p$).

$$A_3 = \begin{pmatrix} 1&1&0&0&1&0&1&0&0&0&0&0&0 \\ 0&1&1&0&0&1&0&1&0&0&0&0&0 \\ 0&0&1&1&0&0&1&0&1&0&0&0&0 \\ 0&0&0&1&1&0&0&1&0&1&0&0&0 \\ 0&0&0&0&1&1&0&0&1&0&1&0&0 \\ 0&0&0&0&0&1&1&0&0&1&0&1&0 \\ 0&0&0&0&0&0&1&1&0&0&1&0&1 \\ 1&0&0&0&0&0&0&1&1&0&0&1&0 \\ 0&1&0&0&0&0&0&0&1&1&0&0&1 \\ 1&0&1&0&0&0&0&0&0&1&1&0&0 \\ 0&1&0&1&0&0&0&0&0&0&1&1&0 \\ 0&0&1&0&1&0&0&0&0&0&0&1&1 \\ 1&0&0&1&0&1&0&0&0&0&0&0&1 \end{pmatrix}.$$

$(b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_9, b_{10}, b_{11}, b_{12}, b_{13})$

UNIVERSITY OF TARTU

Stopping sets

The **stopping sets (sets without tangents)** are the sets of projective points intersecting no line in exactly one point.

Example:

Erased coded symbols: $\{b_2, b_3, b_5, b_6, b_7, b_8\}$.

It is the union of two lines through b_1, without b_1.

Algorithm A never manages to repair any of those servers.

$$A_3 = \begin{pmatrix} 1&1&0&0&1&0&1&0&0&0&0&0&0\\ 0&1&1&0&0&1&0&1&0&0&0&0&0\\ 0&0&1&1&0&0&1&0&1&0&0&0&0\\ 0&0&0&1&1&0&0&1&0&1&0&0&0\\ 0&0&0&0&1&1&0&0&1&0&1&0&0\\ 0&0&0&0&0&1&1&0&0&1&0&1&0\\ 0&0&0&0&0&0&1&1&0&0&1&0&1\\ 1&0&0&0&0&0&0&1&1&0&0&1&0\\ 0&1&0&0&0&0&0&0&1&1&0&0&1\\ 1&0&1&0&0&0&0&0&0&1&1&0&0\\ 0&1&0&1&0&0&0&0&0&0&1&1&0\\ 0&0&1&0&1&0&0&0&0&0&0&1&1\\ 1&0&0&1&0&1&0&0&0&0&0&0&1 \end{pmatrix}.$$

The smallest size s_p of a set without tangents or stopping set:

$$p + \frac{1}{4}\sqrt{2p} + 2 \leq s_p \leq 2p - 2$$

For small odd p:

$s_3 = 6$,
$s_5 = 10$,
$s_7 = 12$,
$s_9 = 15$,
$s_{11} = 18$.

While $s_{11} = 18$, the smallest stopping sets our experiments encountered for $p = 11$ had size 27.
This means smaller stopping sets are very rare.

Conclusion

Explicit codes with efficient updates, good repair locality and good repair availability.

Allows different choices for G, i.e. support for unequal update frequencies.

1. We recalled the $\mathbb{F}_p$-linear code with parity-checks given by the rows of the incidence matrix of $\mathrm{PG}(2, p)$. A parity-check matrix is obtained e.g. via a Moorhouse basis.
2. We propose a method to obtain the generator matrix via tree construction.
3. The repair locality is p, and the repair availability is $p + 1$, while the number of coded symbols is $p^2 + p + 1$.
4. We can use different generator matrices depending on update frequencies of symbols, to reduce the maximum update frequency of a coded symbol.
5. The incidence matrix is circulant.
6. We designed the repair algorithm from the incidence matrix and discussed stopping sets.

Open problem:

The recovery of any individual data symbols from a small number of coded symbols.