Distributed Storage and Data Recovery

Distributed data storage, service and computation

Ago-Erik Riet

Theoretical Computer Science Seminar
Tartu, 29 April 2025

University of Tartu, Tartu, Estonia
Email: agoerik@ut.ee

Distributed Storage System (DSS)

- Cloud storage and computing systems of Google, Amazon, Facebook, Dropbox, AI companies etc. need to be reliable
- Data is stored *redundantly* to prevent data loss: replicated into copies / coded using linear combinations
- Storage node (server/rack) repair from
  - few other nodes (locality), with alternatives (availability)
  - by computing, transferring (bandwidth) a small amount of data
- Updating data economically
- Serving data in parallel from small sets of nodes, with alternatives $\rightarrow$ "distributed service system", service rate region, batch codes
- Private Information Retrieval (PIR) $\rightarrow$ PIR codes
- Multiparty computation (with or without privacy)
- Coded caching
- ...

Errors *vs* erasures

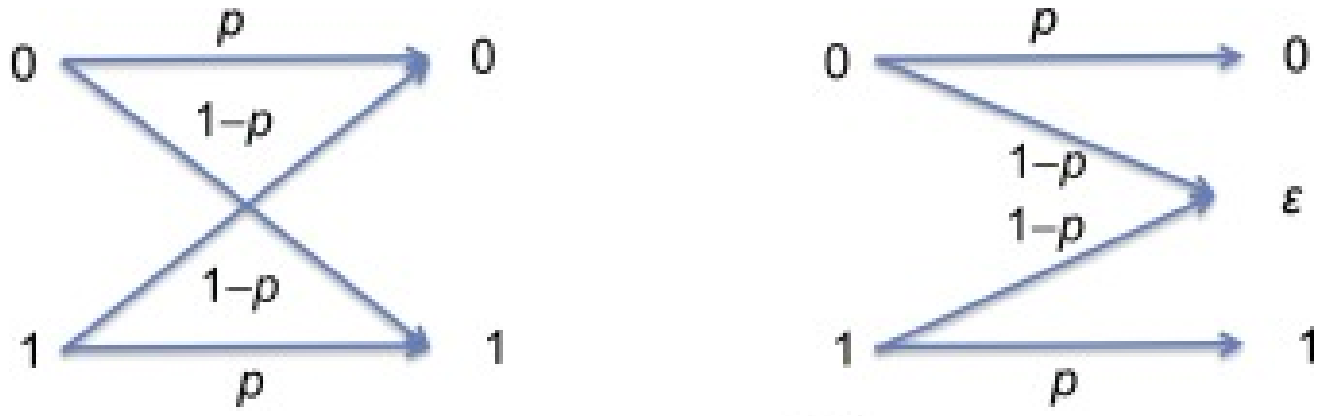


Figure: Encyclopedia of Physical Science and Technology (Third Edition), 2003

- Distributed storage systems (DSSs) use erasure codes, as a defunct node is recognized by the DSS as such
- DSSs are typically linearly coded:
  e.g. $\left(\begin{smallmatrix} a & b \end{smallmatrix}\right)\left(\begin{smallmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{smallmatrix}\right) = \left(\begin{smallmatrix} a & b & a+b \end{smallmatrix}\right)$, i.e.
  data symbols a, b stored in three servers as a, b, a XOR b, so
  $a \leftarrow a$, $a \leftarrow b + (a + b)$, and $b \leftarrow b$, $b \leftarrow a + (a + b)$

Regenerating codes

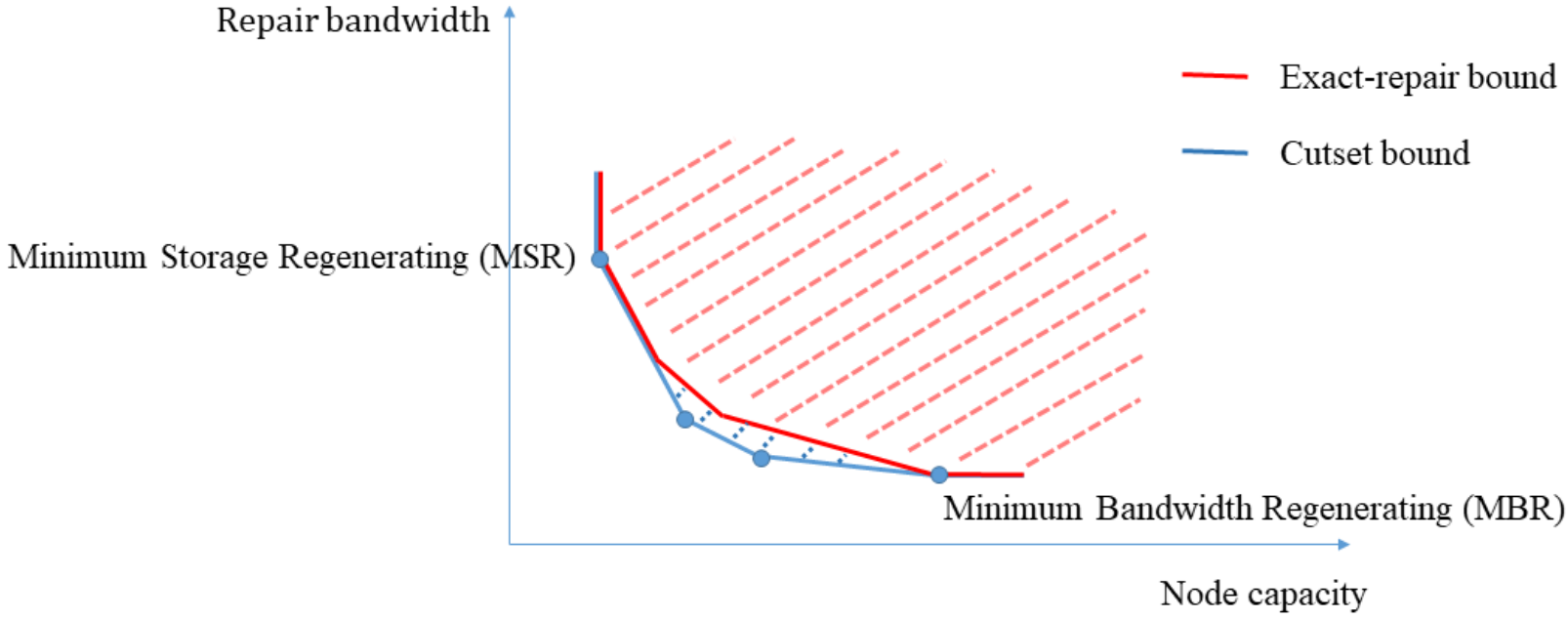
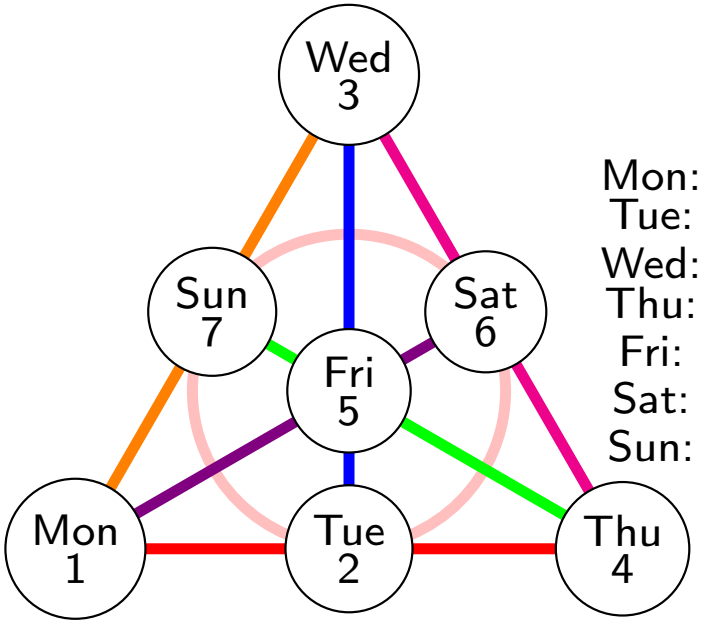


Figure: PhD thesis of Junming Ke

- Trade-off between redundancy and bandwidth of node repair: information-theoretic *cutset bound* (blue curve)
- *Regenerating codes* are optimal, i.e. on the blue curve
- Convex combinations in cutset region by "time-sharing"
- Ke, Hollmann, Riet '24 found a practical, functional-repair regenerating code in blue corner point other than MSR, MBR

Combinatorial Designs, Finite Geometry



Mon:	*Chase*	*Everest*	*Rubble*
Tue:	*Chase*	*Zuma*	*Skye*
Wed:	*Zuma*	*Marshall*	*Rubble*
Thu:	*Chase*	*Marshall*	*Rocky*
Fri:	*Zuma*	*Rocky*	*Everest*
Sat:	*Marshall*	*Everest*	*Skye*
Sun:	*Rocky*	*Skye*	*Rubble*

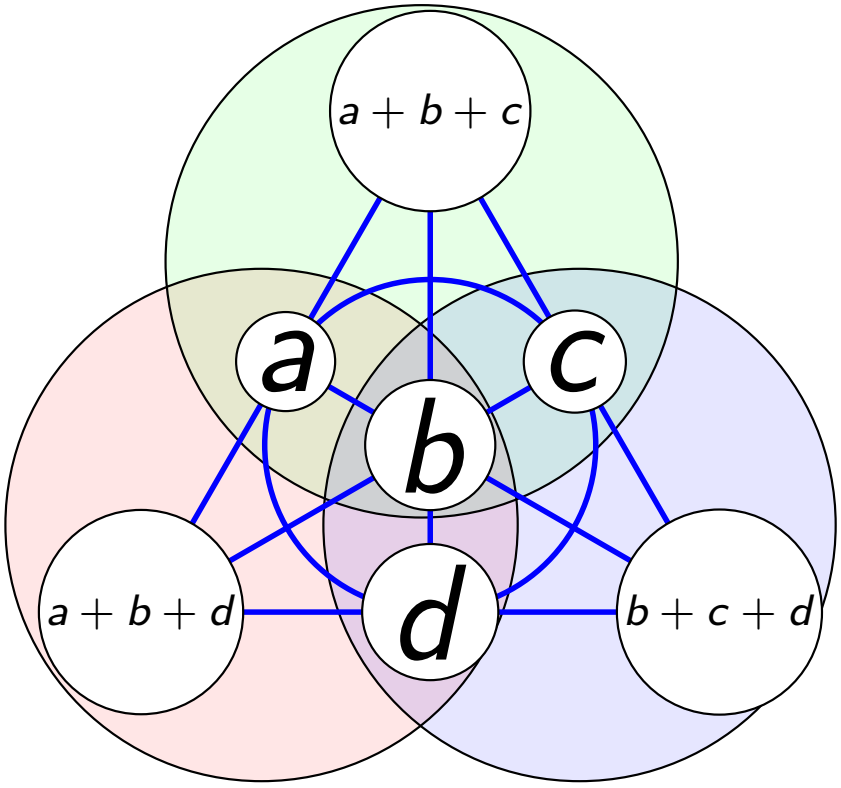
- Fano plane: first *finite projective plane* $\mathrm{PG}(\dim = 2, q = 2)$
- First combinatorial *block design*: $2-(7,3,1)$ design

  $t-(v,k,\lambda)$

  Each t of the v *points* contained in $= \lambda$ of the *blocks*, size k

  lines

Hamming code: update performance

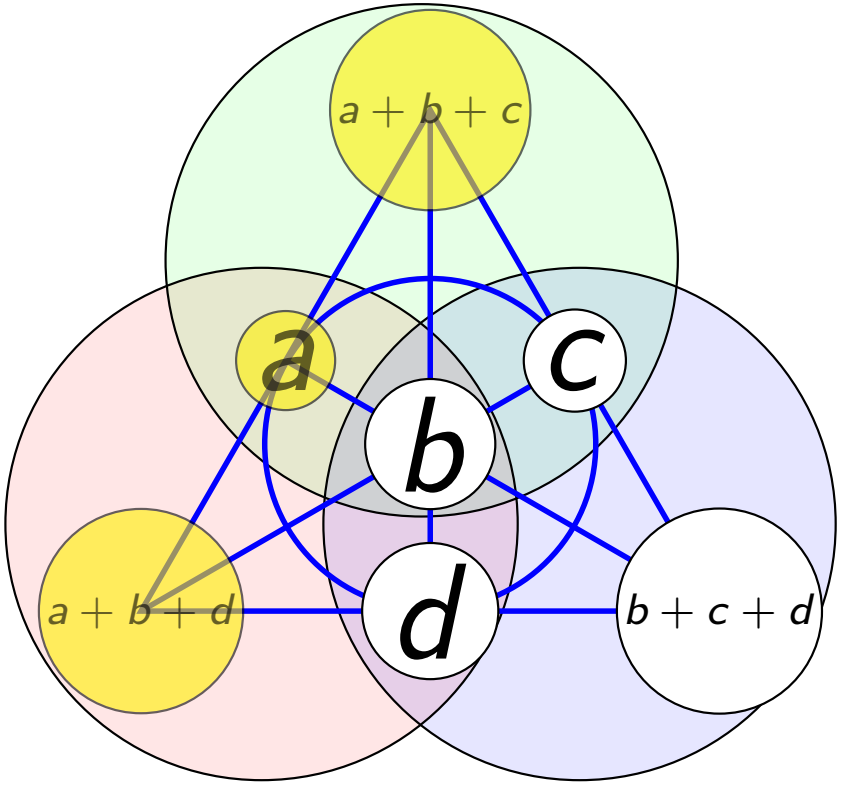


$$\begin{pmatrix} a & b & c & d & a{+}b{+}d & b{+}c{+}d & a{+}b{+}c \end{pmatrix}$$

$$\begin{pmatrix} a & b & c & d \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix} = \begin{pmatrix} a & b & c & d & a{+}b{+}d & b{+}c{+}d & a{+}b{+}c \end{pmatrix}$$

$$\begin{pmatrix} \text{data symbols} \end{pmatrix} \begin{pmatrix} \text{generator} \\ \text{matrix} \end{pmatrix} = \begin{pmatrix} \qquad\qquad \text{coded symbols} \qquad\qquad \end{pmatrix}$$

Hamming code: update performance



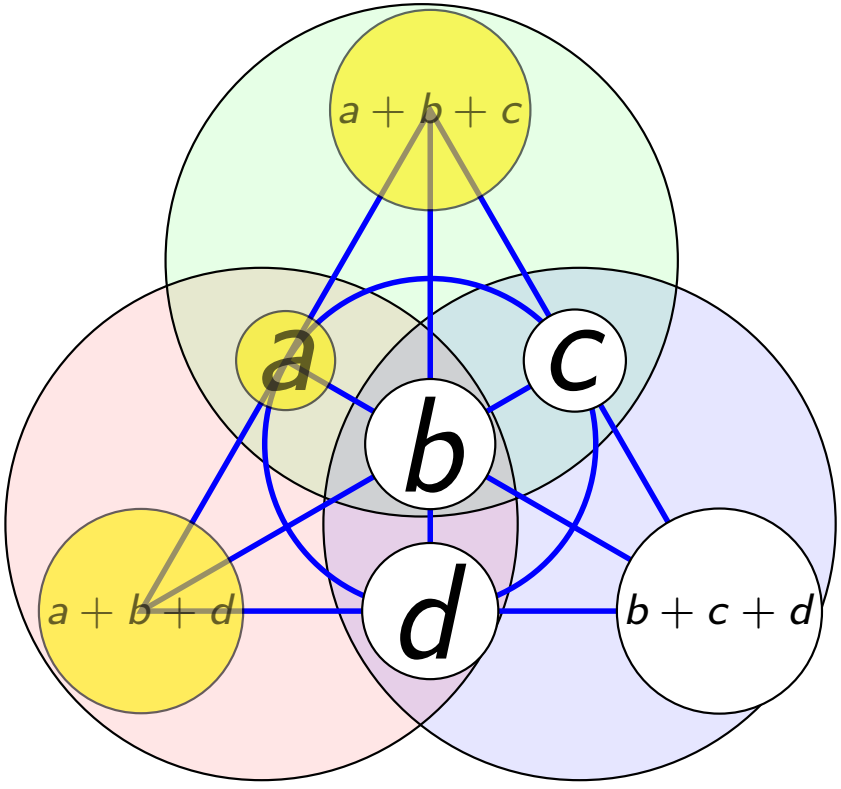
$$(a \;\; b \;\; c \;\; d \;\; a{+}b{+}d \;\; b{+}c{+}d \;\; a{+}b{+}c)$$

Updated a

$$(a \;\; b \;\; c \;\; d) \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix} = (a \;\; b \;\; c \;\; d \;\; a{+}b{+}d \;\; b{+}c{+}d \;\; a{+}b{+}c)$$

$$(\text{data symbols}) \begin{pmatrix} \text{generator} \\ \text{matrix} \end{pmatrix} = (\qquad \text{coded symbols} \qquad)$$

Hamming code: update performance



$$(a \;\; b \;\; c \;\; d \;\; a{+}b{+}d \;\; b{+}c{+}d \;\; a{+}b{+}c)$$

Updated a twice

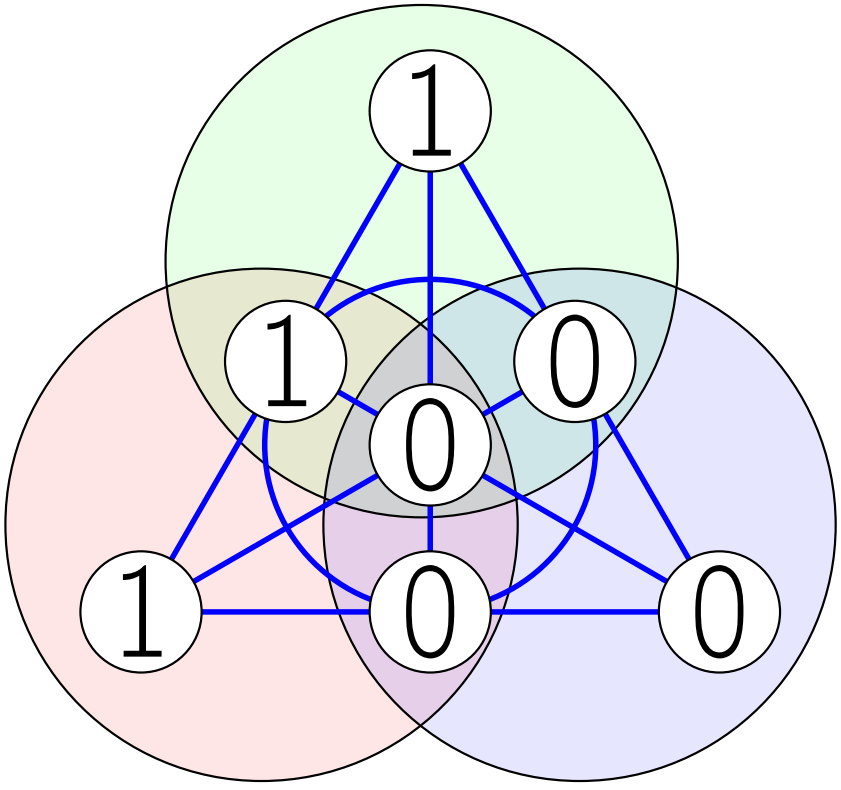
Can aggregate updates:

$a{+}\Delta_1{+}\Delta_2 = a{+}(\Delta_1{+}\Delta_2)$

$$(a \;\; b \;\; c \;\; d) \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix} = (a \;\; b \;\; c \;\; d \;\; a{+}b{+}d \;\; b{+}c{+}d \;\; a{+}b{+}c)$$

$$(\text{data symbols}) \begin{pmatrix} \text{generator} \\ \text{matrix} \end{pmatrix} = (\qquad \text{coded symbols} \qquad)$$

Hamming code: errors and erasures



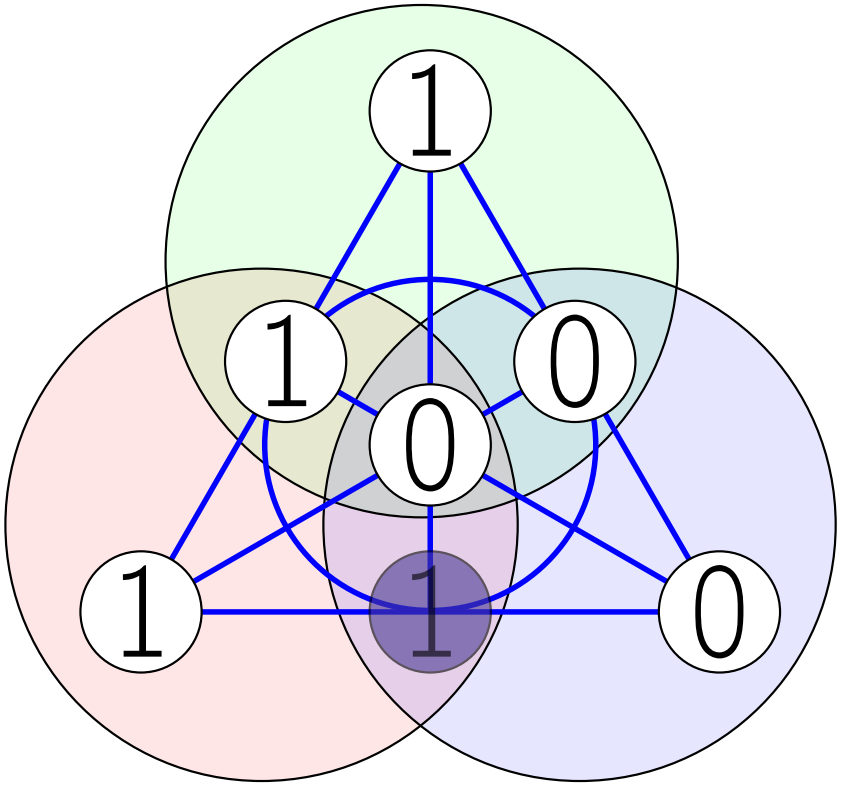
$$\begin{pmatrix} a & b & c & d & a{+}b{+}d & b{+}c{+}d & a{+}b{+}c \end{pmatrix}$$
$$\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 0 \end{pmatrix}$$

Each circle parity-check:
has sum 0 mod 2

$$\begin{pmatrix} a & b & c & d \end{pmatrix} \begin{pmatrix} 1&0&0&0&1&0&1 \\ 0&1&0&0&1&1&1 \\ 0&0&1&0&0&1&1 \\ 0&0&0&1&1&1&0 \end{pmatrix} = \begin{pmatrix} a & b & c & d & a{+}b{+}d & b{+}c{+}d & a{+}b{+}c \end{pmatrix}$$

$$\begin{pmatrix} \text{data symbols} \end{pmatrix} \begin{pmatrix} \text{generator} \\ \text{matrix} \end{pmatrix} = \begin{pmatrix} \qquad\qquad \text{coded symbols} \qquad\qquad \end{pmatrix}$$

Hamming code: errors and erasures



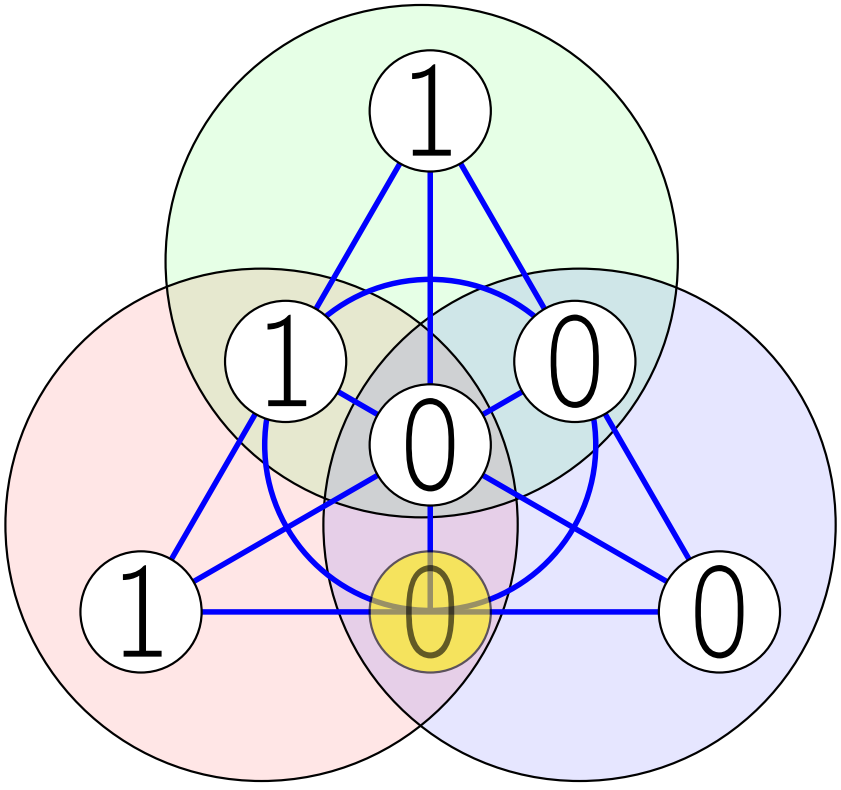
$$\begin{pmatrix} a & b & c & d & a{+}b{+}d & b{+}c{+}d & a{+}b{+}c \end{pmatrix}$$
$$\begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \end{pmatrix}$$

Error: Sum wrong in pink, violet circle

$$\begin{pmatrix} a & b & c & d \end{pmatrix} \begin{pmatrix} 1&0&0&0&1&0&1 \\ 0&1&0&0&1&1&1 \\ 0&0&1&0&0&1&1 \\ 0&0&0&1&1&1&0 \end{pmatrix} = \begin{pmatrix} a & b & c & d & a{+}b{+}d & b{+}c{+}d & a{+}b{+}c \end{pmatrix}$$

$$\begin{pmatrix} \text{data symbols} \end{pmatrix} \begin{pmatrix} \text{generator} \\ \text{matrix} \end{pmatrix} = \begin{pmatrix} \qquad\qquad \text{coded symbols} \qquad\qquad \end{pmatrix}$$

Hamming code: errors and erasures



$$\begin{pmatrix} a & b & c & d & a+b+d & b+c+d & a+b+c \end{pmatrix}$$
$$\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 0 \end{pmatrix}$$

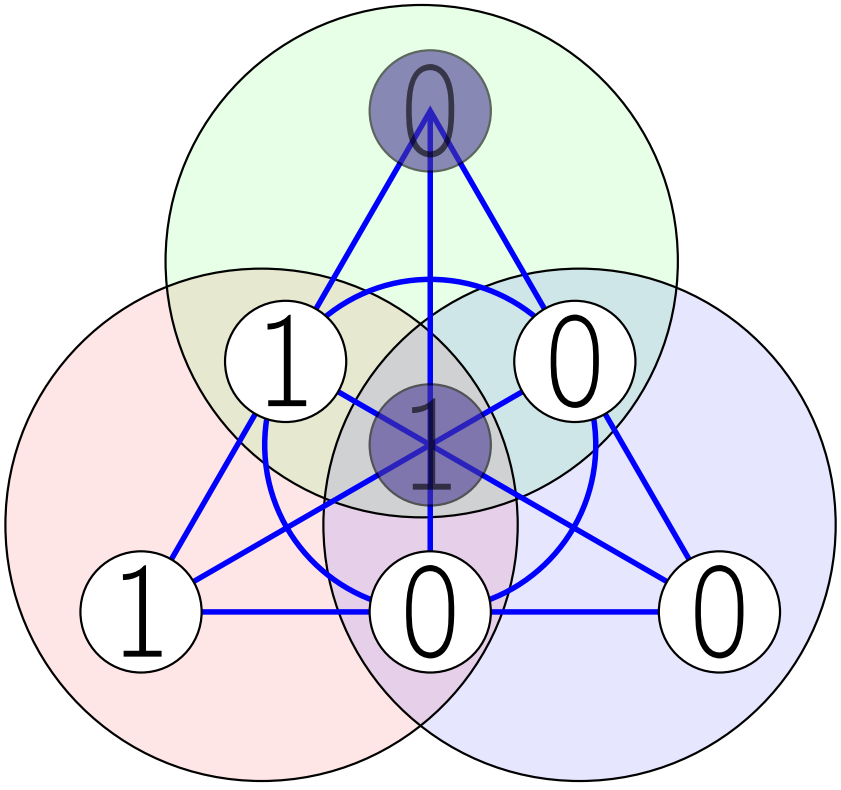
Error: Sum wrong in pink, violet circle

Change bit in pink, violet, not green

$$\begin{pmatrix} a & b & c & d \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix} = \begin{pmatrix} a & b & c & d & a+b+d & b+c+d & a+b+c \end{pmatrix}$$

$$\begin{pmatrix} \text{data symbols} \end{pmatrix} \begin{pmatrix} \text{generator} \\ \text{matrix} \end{pmatrix} = \begin{pmatrix} \qquad\qquad \text{coded symbols} \qquad\qquad \end{pmatrix}$$

Hamming code: errors and erasures



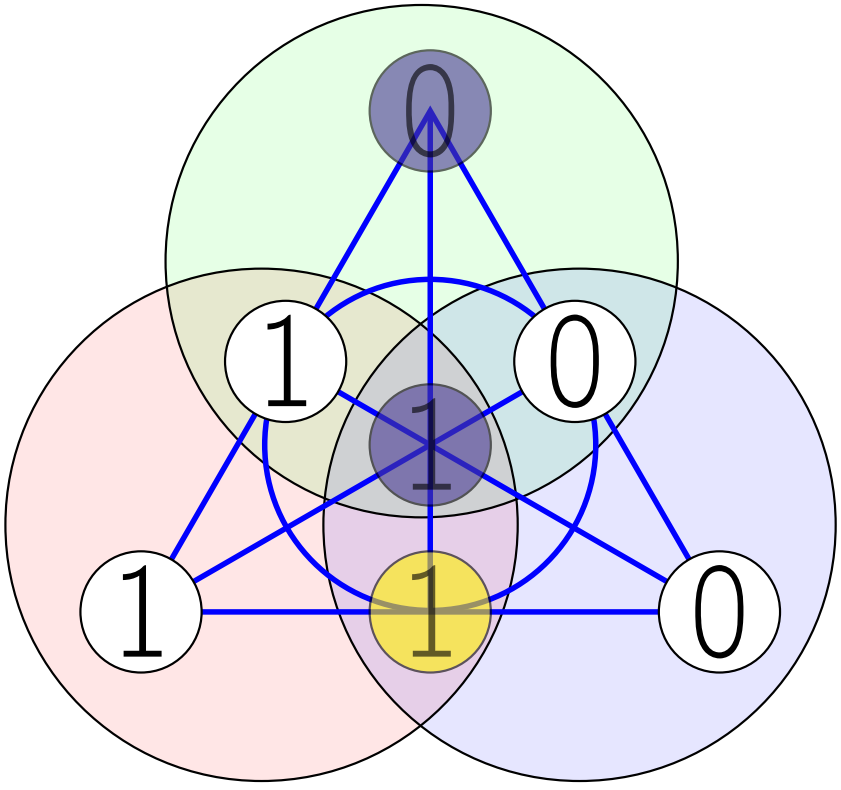
$$\begin{pmatrix} a & b & c & d & a{+}b{+}d & b{+}c{+}d & a{+}b{+}c \end{pmatrix}$$
$$\begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

Two errors. Sum wrong in pink, violet circle

$$\begin{pmatrix} a & b & c & d \end{pmatrix} \begin{pmatrix} 1&0&0&0&1&0&1 \\ 0&1&0&0&1&1&1 \\ 0&0&1&0&0&1&1 \\ 0&0&0&1&1&1&0 \end{pmatrix} = \begin{pmatrix} a & b & c & d & a{+}b{+}d & b{+}c{+}d & a{+}b{+}c \end{pmatrix}$$

$$\begin{pmatrix} \text{data symbols} \end{pmatrix} \begin{pmatrix} \text{generator} \\ \text{matrix} \end{pmatrix} = \begin{pmatrix} \qquad\qquad \text{coded symbols} \qquad\qquad \end{pmatrix}$$

Hamming code: errors and erasures



$$\begin{pmatrix} a & b & c & d & a{+}b{+}d & b{+}c{+}d & a{+}b{+}c \end{pmatrix}$$
$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 1 \end{pmatrix}$$

Two errors. Sum wrong in pink, violet circle

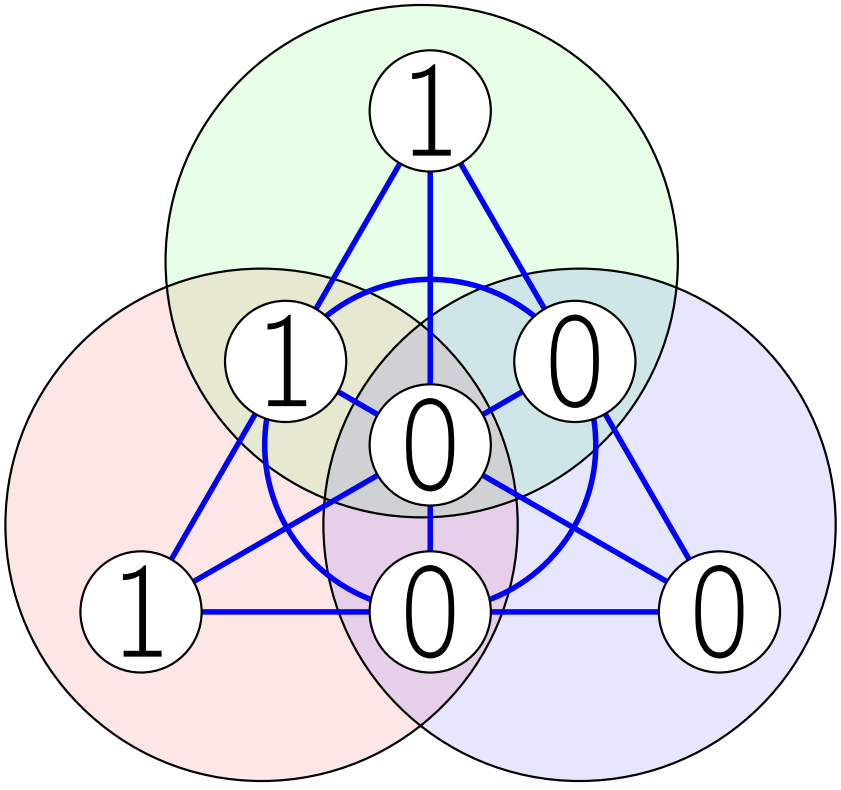
Change bit in pink, violet, not green

Repair failed!

$$\begin{pmatrix} a & b & c & d \end{pmatrix} \begin{pmatrix} 1&0&0&0&1&0&1 \\ 0&1&0&0&1&1&1 \\ 0&0&1&0&0&1&1 \\ 0&0&0&1&1&1&0 \end{pmatrix} = \begin{pmatrix} a & b & c & d & a{+}b{+}d & b{+}c{+}d & a{+}b{+}c \end{pmatrix}$$

$$\begin{pmatrix} \text{data symbols} \end{pmatrix} \begin{pmatrix} \text{generator} \\ \text{matrix} \end{pmatrix} = \begin{pmatrix} \qquad\qquad \text{coded symbols} \qquad\qquad \end{pmatrix}$$

Hamming code: errors and erasures



$$\begin{pmatrix} a & b & c & d & a{+}b{+}d & b{+}c{+}d & a{+}b{+}c \end{pmatrix}$$
$$\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 0 \end{pmatrix}$$

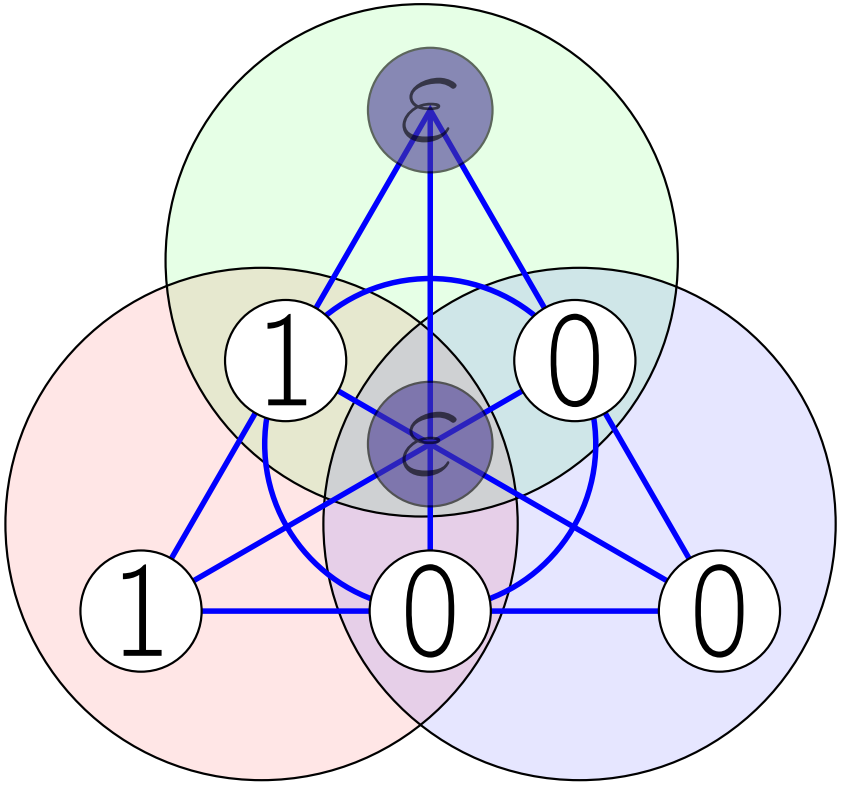
Each circle parity-check:

has sum 0 mod 2

$$\begin{pmatrix} a & b & c & d \end{pmatrix} \begin{pmatrix} 1&0&0&0&1&0&1 \\ 0&1&0&0&1&1&1 \\ 0&0&1&0&0&1&1 \\ 0&0&0&1&1&1&0 \end{pmatrix} = \begin{pmatrix} a & b & c & d & a{+}b{+}d & b{+}c{+}d & a{+}b{+}c \end{pmatrix}$$

$$\begin{pmatrix} \text{data symbols} \end{pmatrix} \begin{pmatrix} \text{generator} \\ \text{matrix} \end{pmatrix} = \begin{pmatrix} \qquad\qquad \text{coded symbols} \qquad\qquad \end{pmatrix}$$

Hamming code: errors and erasures



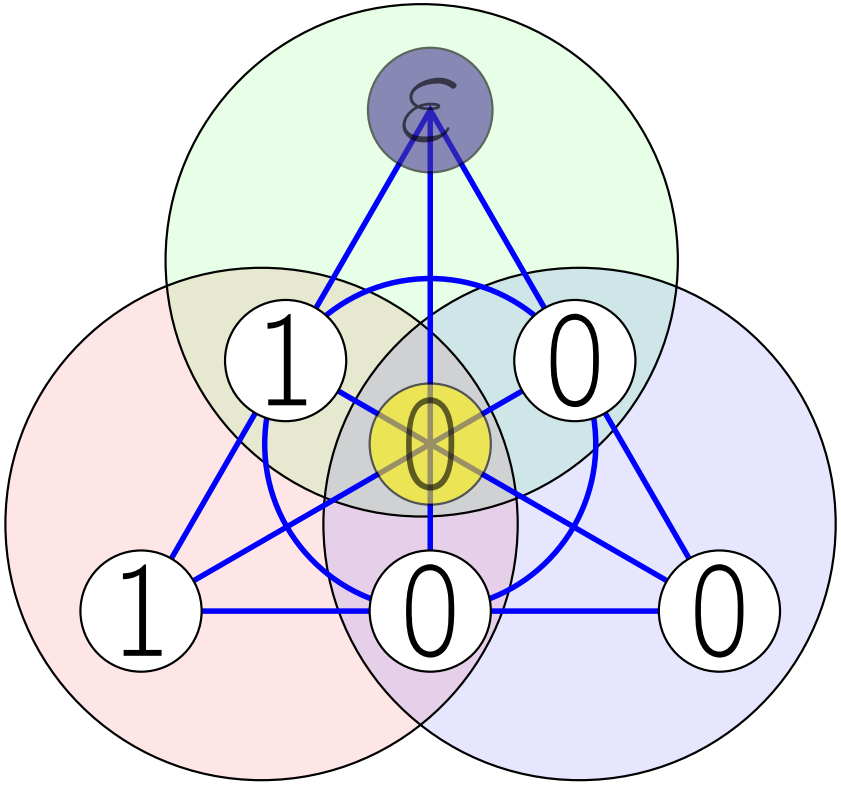
$$\begin{pmatrix} a & b & c & d & a{+}b{+}d & b{+}c{+}d & a{+}b{+}c \end{pmatrix}$$
$$\begin{pmatrix} 1 & \varepsilon & 1 & 0 & 1 & 0 & \varepsilon \end{pmatrix}$$

Two erasures

$$\begin{pmatrix} a & b & c & d \end{pmatrix} \begin{pmatrix} 1&0&0&0&1&0&1 \\ 0&1&0&0&1&1&1 \\ 0&0&1&0&0&1&1 \\ 0&0&0&1&1&1&0 \end{pmatrix} = \begin{pmatrix} a & b & c & d & a{+}b{+}d & b{+}c{+}d & a{+}b{+}c \end{pmatrix}$$

$$\begin{pmatrix} \text{data symbols} \end{pmatrix} \begin{pmatrix} \text{generator} \\ \text{matrix} \end{pmatrix} = \begin{pmatrix} \qquad\qquad \text{coded symbols} \qquad\qquad \end{pmatrix}$$

Hamming code: errors and erasures



$$\begin{pmatrix} a & b & c & d & a{+}b{+}d & b{+}c{+}d & a{+}b{+}c \end{pmatrix}$$
$$\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & \varepsilon \end{pmatrix}$$

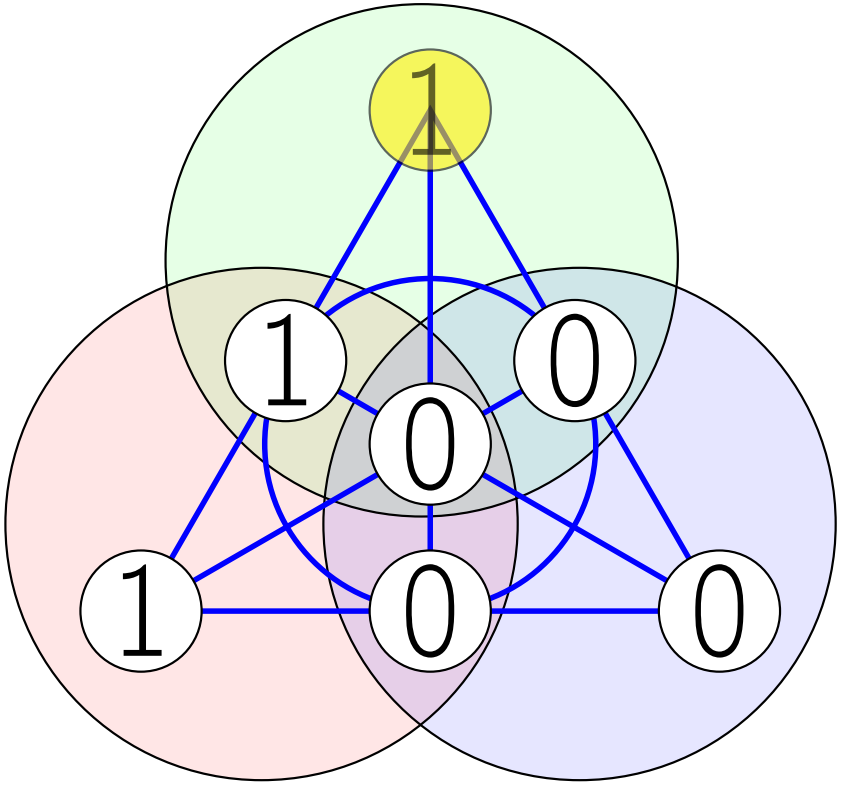
Two erasures

Recover bit from pink

$$\begin{pmatrix} a & b & c & d \end{pmatrix} \begin{pmatrix} 1&0&0&0&1&0&1 \\ 0&1&0&0&1&1&1 \\ 0&0&1&0&0&1&1 \\ 0&0&0&1&1&1&0 \end{pmatrix} = \begin{pmatrix} a & b & c & d & a{+}b{+}d & b{+}c{+}d & a{+}b{+}c \end{pmatrix}$$

$$\begin{pmatrix} \text{data symbols} \end{pmatrix} \begin{pmatrix} \text{generator} \\ \text{matrix} \end{pmatrix} = \begin{pmatrix} \qquad\qquad \text{coded symbols} \qquad\qquad \end{pmatrix}$$

Hamming code: errors and erasures



$$\begin{pmatrix} a & b & c & d & a{+}b{+}d & b{+}c{+}d & a{+}b{+}c \end{pmatrix}$$
$$\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

Two erasures

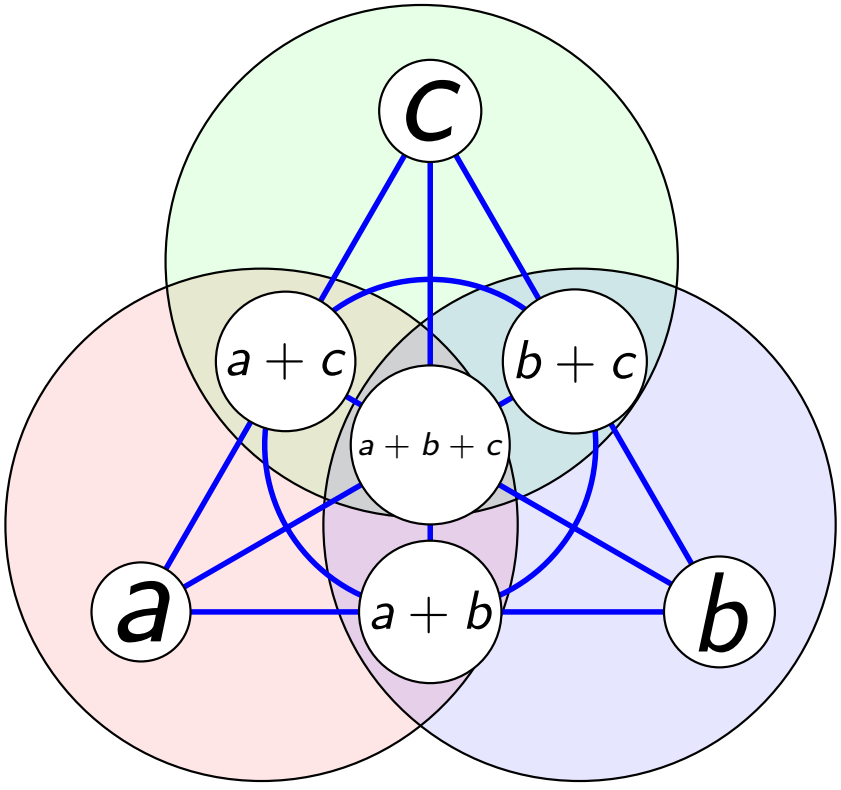
Recover bit from pink

Recover bit from green

$$\begin{pmatrix} a & b & c & d \end{pmatrix} \begin{pmatrix} 1&0&0&0&1&0&1 \\ 0&1&0&0&1&1&1 \\ 0&0&1&0&0&1&1 \\ 0&0&0&1&1&1&0 \end{pmatrix} = \begin{pmatrix} a & b & c & d & a{+}b{+}d & b{+}c{+}d & a{+}b{+}c \end{pmatrix}$$

$$\begin{pmatrix} \text{data symbols} \end{pmatrix} \begin{pmatrix} \text{generator} \\ \text{matrix} \end{pmatrix} = \begin{pmatrix} \qquad\qquad \text{coded symbols} \qquad\qquad \end{pmatrix}$$

Simplex code: the dual of Hamming code



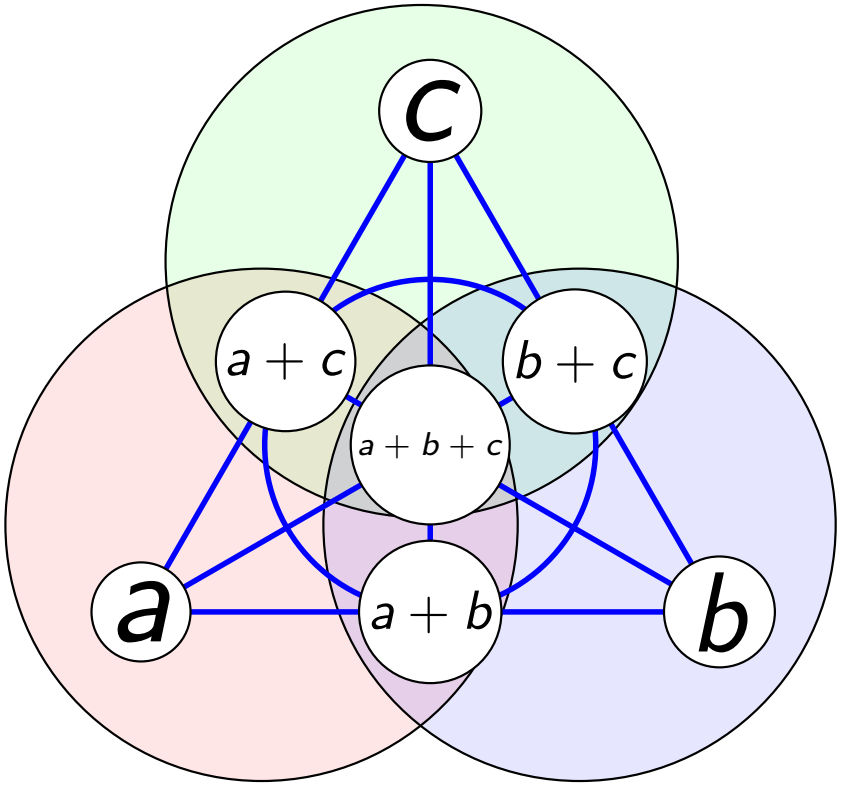
$$\begin{pmatrix} a & b & c & b{+}c & a{+}c & a{+}b & a{+}b{+}c \end{pmatrix}$$

More parity-checks

$$\begin{pmatrix} a & b & c \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{pmatrix} = \begin{pmatrix} a & b & c & b{+}c & a{+}c & a{+}b & a{+}b{+}c \end{pmatrix}$$

Parity-checks $\begin{pmatrix} \text{row1: } 1\ 1\ 0\ 1\ 0\ 0\ 0 \\ \textit{row2}\text{: } 0\ 1\ 1\ 0\ 1\ 0\ 0 \\ \textit{row3}\text{: } 0\ 0\ 1\ 1\ 0\ 1\ 0 \\ \textit{row4}\text{: } 0\ 0\ 0\ 1\ 1\ 0\ 1 \\ \textit{row5}\text{: } 1\ 0\ 0\ 0\ 1\ 1\ 0 \\ \textit{row6}\text{: } 0\ 1\ 0\ 0\ 0\ 1\ 1 \\ \textit{row7}\text{: } 1\ 0\ 1\ 0\ 0\ 0\ 1 \end{pmatrix}$, generator matrix $\begin{pmatrix} \text{row1}-\text{row4: } 1\ 1\ 0\ 0\ 1\ 0\ 1 \\ \text{row2}-\text{row4: } 0\ 1\ 1\ 1\ 0\ 0\ 1 \\ \text{row3}-\text{row4: } 0\ 0\ 1\ 0\ 1\ 1\ 1 \end{pmatrix}$

Simplex code: the dual of Hamming code



$$\begin{pmatrix} a & b & c & b{+}c & a{+}c & a{+}b & a{+}b{+}c \end{pmatrix}$$

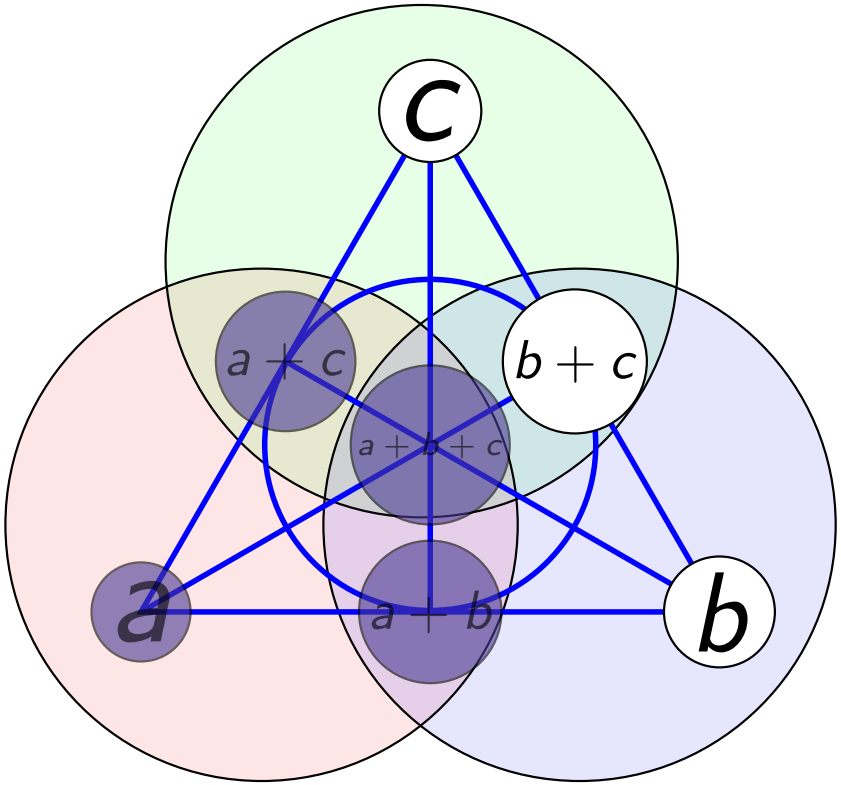
More parity-checks

Corrects any 3 erasures

$$\begin{pmatrix} a & b & c \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{pmatrix} = \begin{pmatrix} a & b & c & b{+}c & a{+}c & a{+}b & a{+}b{+}c \end{pmatrix}$$

Parity-checks $\begin{pmatrix} \text{row1: } 1\ 1\ 0\ 1\ 0\ 0\ 0 \\ \textit{row2}\text{: } 0\ 1\ 1\ 0\ 1\ 0\ 0 \\ \textit{row3}\text{: } 0\ 0\ 1\ 1\ 0\ 1\ 0 \\ \textit{row4}\text{: } 0\ 0\ 0\ 1\ 1\ 0\ 1 \\ \textit{row5}\text{: } 1\ 0\ 0\ 0\ 1\ 1\ 0 \\ \textit{row6}\text{: } 0\ 1\ 0\ 0\ 0\ 1\ 1 \\ \textit{row7}\text{: } 1\ 0\ 1\ 0\ 0\ 0\ 1 \end{pmatrix}$, generator matrix $\begin{pmatrix} \text{row1}-\text{row4: } 1\ 1\ 0\ 0\ 1\ 0\ 1 \\ \text{row2}-\text{row4: } 0\ 1\ 1\ 1\ 0\ 0\ 1 \\ \text{row3}-\text{row4: } 0\ 0\ 1\ 0\ 1\ 1\ 1 \end{pmatrix}$

Simplex code: the dual of Hamming code



$$\begin{pmatrix} a & b & c & b{+}c & a{+}c & a{+}b & a{+}b{+}c \end{pmatrix}$$

More parity-checks

Corrects any 3 erasures

Stopping set / set without tangents

$$\begin{pmatrix} a & b & c \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{pmatrix} = \begin{pmatrix} a & b & c & b{+}c & a{+}c & a{+}b & a{+}b{+}c \end{pmatrix}$$

Parity-checks $\begin{pmatrix} \text{row1: } 1\ 1\ 0\ 1\ 0\ 0\ 0 \\ \textit{row2}\text{: } 0\ 1\ 1\ 0\ 1\ 0\ 0 \\ \textit{row3}\text{: } 0\ 0\ 1\ 1\ 0\ 1\ 0 \\ \textit{row4}\text{: } 0\ 0\ 0\ 1\ 1\ 0\ 1 \\ \textit{row5}\text{: } 1\ 0\ 0\ 0\ 1\ 1\ 0 \\ \textit{row6}\text{: } 0\ 1\ 0\ 0\ 0\ 1\ 1 \\ \textit{row7}\text{: } 1\ 0\ 1\ 0\ 0\ 0\ 1 \end{pmatrix}$, generator matrix $\begin{pmatrix} \text{row1}-\text{row4: } 1\ 1\ 0\ 0\ 1\ 0\ 1 \\ \text{row2}-\text{row4: } 0\ 1\ 1\ 1\ 0\ 0\ 1 \\ \text{row3}-\text{row4: } 0\ 0\ 1\ 0\ 1\ 1\ 1 \end{pmatrix}$

Distributed service system

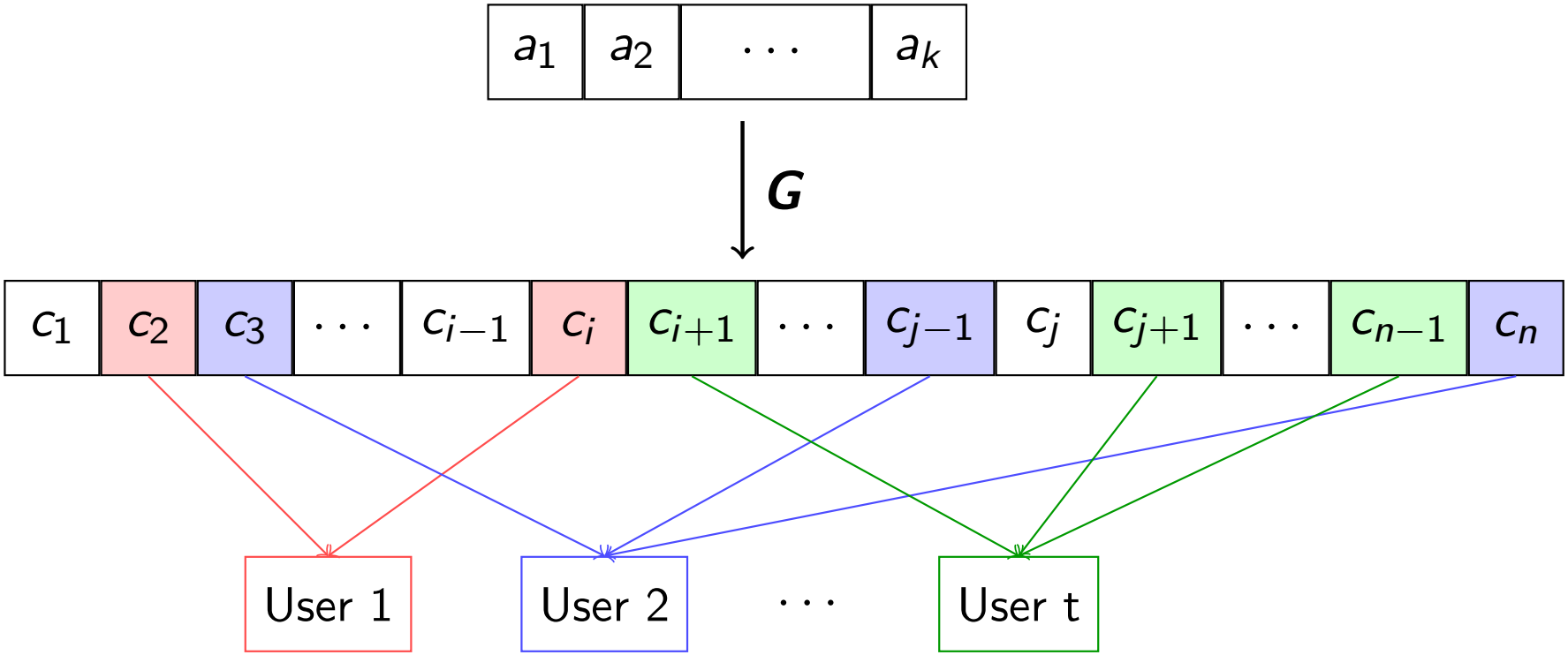


Figure: Vitaly Skachek

Service rate region

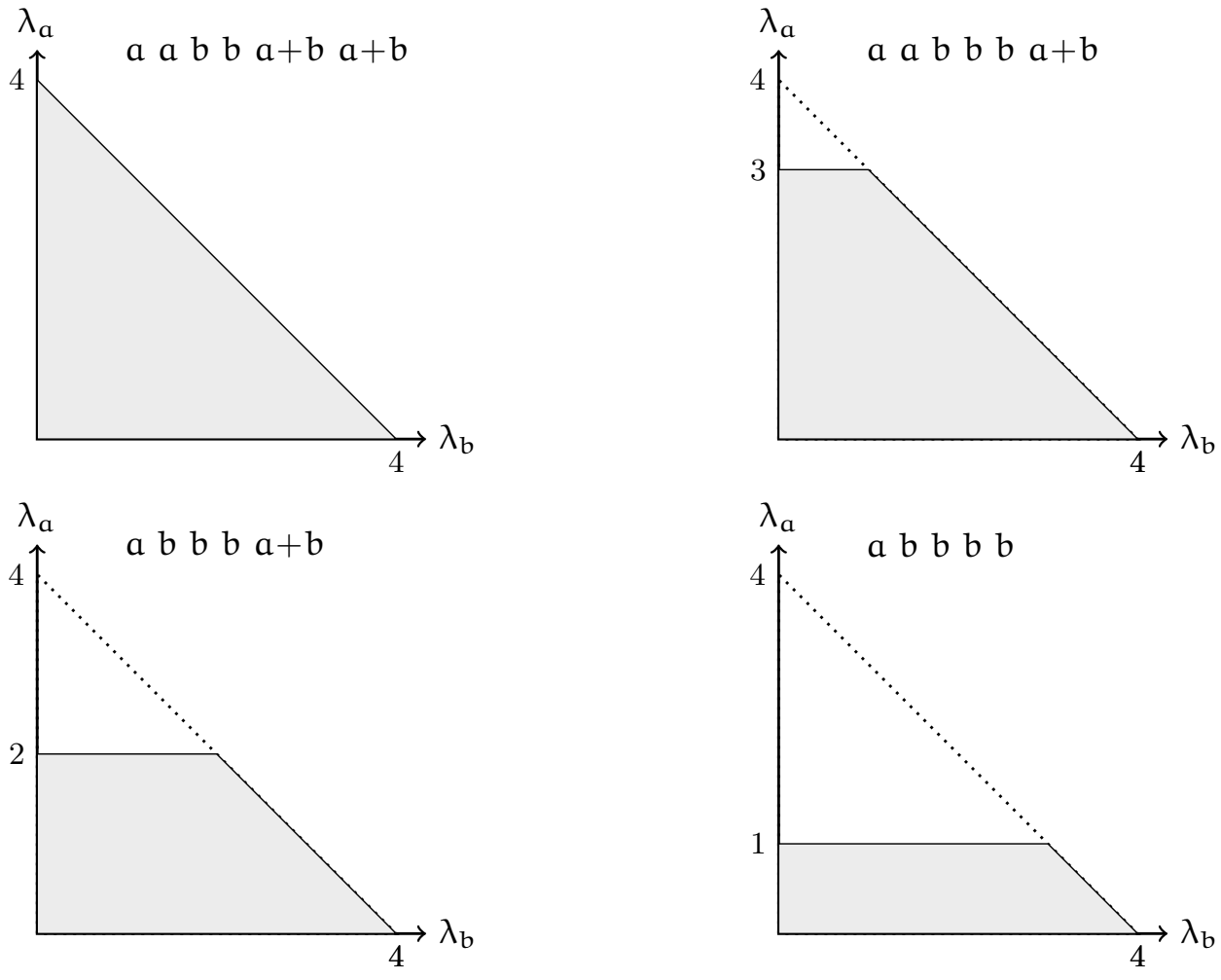


Figure: *Service Rate Region: A New Aspect of Coded Distributed System Design*, TIT, 2021, by Aktaş, Joshi, Kadhe, Kazemi, Soljanin

Definition of Batch Codes

- Proposed in the crypto community for:
  - Load balancing
  - Private information retrieval

Definition [Ishai, Kushilevitz, Ostrovsky, Sahai 2004]

An $(k, N, t, n, \nu)_\Sigma$ batch code over Σ encodes any $\boldsymbol{a} = (a_1, a_2, \cdots, a_k) \in \Sigma^k$ into n strings (buckets) $\boldsymbol{c}_1, \boldsymbol{c}_2, \cdots, \boldsymbol{c}_n$ over Σ of total length N, such that $\forall i_1, i_2, \cdots, i_t \in [k]$, t users can retrieve $a_{i_1}, a_{i_2}, \cdots, a_{i_t}$, resp., by reading $\leq \nu$ symbols from each bucket, s.t. a_{i_ℓ} is recovered from symbols read by ℓ-th user alone

- • **Y. Ishai, E. Kushilevitz, R. Ostrovsky, and A. Sahai,** "Batch codes and their applications,", *Proc. 36th ACM Symposium on Theory of Computing (STOC)*, June 2004, Chicago, IL.

Motivation

- Private information retrieval (PIR) codes for multi-server private information retrieval to reduce the storage overhead [Fazeli, Vardy, Yaakobi]
- Batch codes for server load balancing for hot data [Ishai, Kushilevitz, Ostrovsky, Sahai]
- Primitive multiset batch codes / PIR codes: $\leq t$ clients access pairwise disjoint sets of servers to retrieve
  - any $(a_{i_1}, \ldots, a_{i_t})$, $a_{i_j} \in [k]$ for batch codes
  - any $(a_i, \ldots, a_i)$, $i \in [k]$ for PIR codes

• **A. Fazeli, A. Vardy, and E. Yaakobi,** "PIR with low storage overhead: coding instead of replication", *2015 IEEE International Symposium on Information Theory (ISIT)*, Hong Kong, pp. 2852-2856, 2015.

• **Y. Ishai, E. Kushilevitz, R. Ostrovsky, and A. Sahai,** "Batch codes and their applications,", *Proc. 36th ACM Symposium on Theory of Computing (STOC)*, June 2004, Chicago, IL.

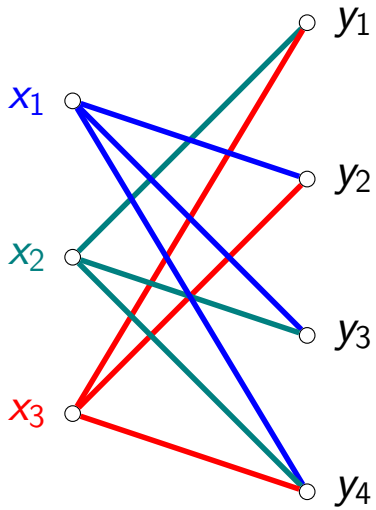
Example: simplex code $k = 3$

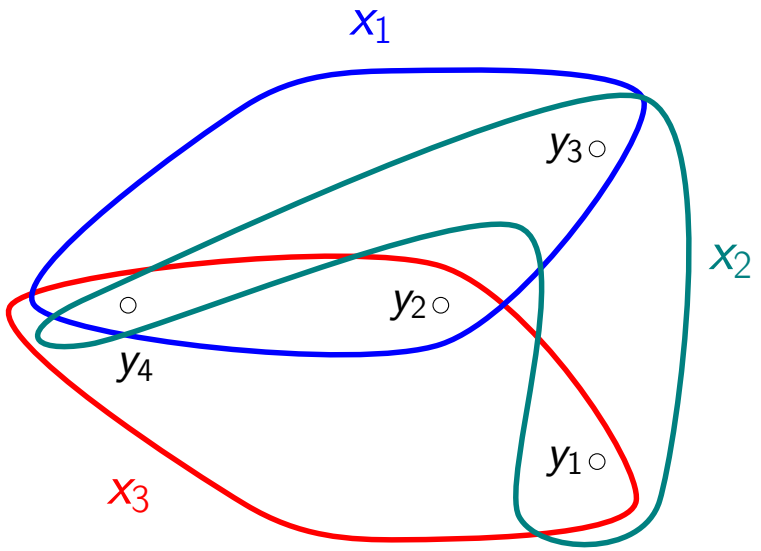
Generator matrix $\rightarrow$

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

Code $\downarrow$

$$\begin{bmatrix} x_1 & x_2 & x_3 & x_2 \oplus x_3 & x_1 \oplus x_3 & x_1 \oplus x_2 & x_1 \oplus x_2 \oplus x_3 \\ & & & = y_1 & = y_2 & = y_3 & = y_4 \end{bmatrix}$$





$\uparrow$ $\uparrow$

(3-left-regular) bipartite graph $G(E, V, S)$

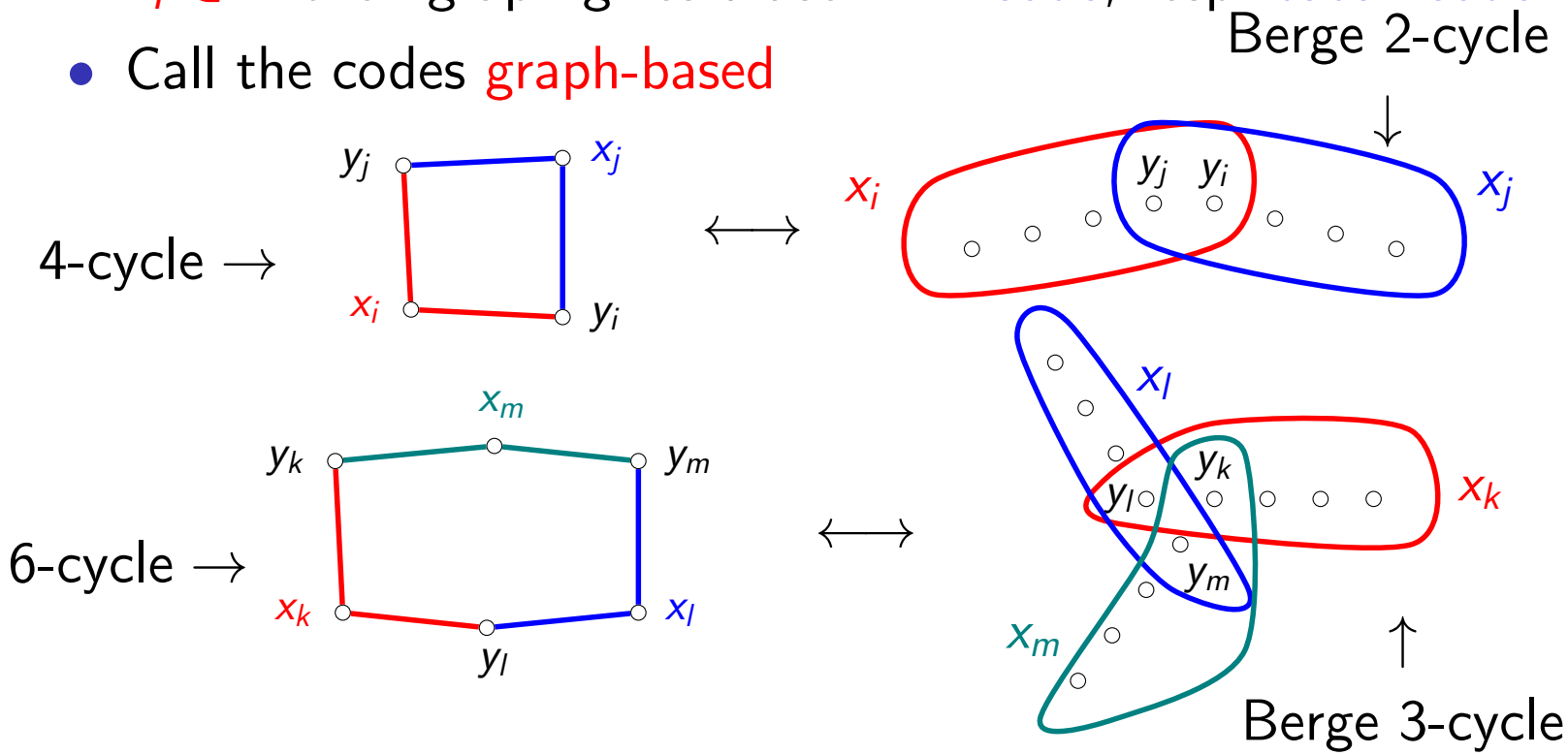
(3-uniform) hypergraph $\mathcal{H}(V, E)$

Linear (computational) PIR/batch codes

- $\boldsymbol{x} = (x_1, x_2, \cdots, x_k)$ information string
- $\boldsymbol{y} = (y_1, y_2, \cdots, y_{n-k})$ encoding of $\boldsymbol{x}$ - redundant symbols
- write y_i, $i \in \{1, \cdots, n-k\}$ as $y_i = \sum_{j=1}^{k} g_{j,i} x_j$
- Generator matrix: $[\boldsymbol{I}|\boldsymbol{G}]$ where $\boldsymbol{G} = \left(g_{j,i}\right)_{j\in[k],i\in[n-k]}$ the encoding is $[\boldsymbol{x}|\boldsymbol{y}] = \boldsymbol{x}[\boldsymbol{I}|\boldsymbol{G}]$
- $E = \{x_1, \ldots, x_k\}$ (information symbols)
- $V = \{y_1, \ldots, y_{n-k}\}$ (redundant symbols)
- $\{x_j, y_i\} \in S$ (an edge) iff $g_{j,i} \neq 0$
- bipartite graph $G(E, V, S)$ (left part E, right part V, edge set S); hypergraph $\mathcal{H}(V, E)$

Graph-based PIR/batch codes

- [Rawat, Song, Dimakis, Gal]: if $G(E, V, S)$ has girth (length of shortest cycle) ≥ 6, resp. ≥ 8 and $\min \deg(x_i) \geq t - 1$ for $x_i \in E$ then graph gives t-user PIR code, resp. batch code
- Call the codes graph-based



• A.S. Rawat, Z. Song, A.G. Dimakis, and A. Gal, "Batch codes through dense graphs without short cycles", *IEEE Trans. Information Theory*, vol. 62, no. 4, pp. 1592-1604, 2016.

Asynchronous batch code model

- Synchronous (classical) - serve t requests at same time
- Asynchronous (R., Skachek, Thomas) - stream of requests
  Can start serving any new request whenever $< t$ requests are currently being served
- For any code C we have $t_{\max}^{\text{asynch}}(C) \leq t_{\max}^{\text{synch}}(C)$
- $C =$ (simplex code $k = 3$)
  $$\Rightarrow t_{\max}^{\text{synch}}(C) = 4;\ t_{\max}^{\text{asynch}}(C) = 2$$
- For graph-based batch (and PIR) codes
  $t_{\max}^{\text{synch}} \geq$min_left_degree+1 [Rawat, Song, Dimakis, Gal]
- Observation (R., Skachek, Thomas): Graph-based (PIR and) batch codes have $t_{\max}^{\text{asynch}} \geq$min_left_degree+1

• A.S. Rawat, Z. Song, A.G. Dimakis, and A. Gal, "Batch codes through dense graphs without short cycles", *IEEE Trans. Information Theory*, vol. 62, no. 4, pp. 1592-1604, 2016.

Asynchronous batch code model - examples

Ex. 1: $\not{1}\ \not{1}\ \not{2}\ \not{3}\ \not{2}\ \not{3}\ 1\ \not{3}\ \not{3}\ \not{1}\ 1\ 3\ \not{2}\ \not{1}\ \not{3}\ \not{1}\ 2\ \ldots$

Ex. 2: Code $[x_1 \ \ x_2 \ \ x_1 \oplus x_2]$ (simplex code for $k = 2$).

- $t_{\text{max}}^{\text{synch}} = 2$, indeed:
  - can serve x_1 and $x_1 = x_2 \oplus (x_1 \oplus x_2)$;
  - can serve x_1 and x_2;
  - can serve x_2 and $x_2 = x_1 \oplus (x_1 \oplus x_2)$;
  - cannot serve x_1, x_1 and x_1.
- But $t_{\text{max}}^{\text{asynch}} = 1$, indeed:
  - stream of requests: $112\ldots$;
  - serve x_1 (1st server);
  - serve $x_1 = x_2 \oplus (x_1 \oplus x_2)$ (2nd and 3rd server);
  - 1st server finishes serving;
  - but incoming request x_2 cannot be served by 1st server only.

Hypergraphs and graph-based PIR/batch codes

- Remove edges to make $G(E, V, S)$ $(t-1)$-left-regular
  Girth cannot decrease - no new cycles
- Corresponding $(t-1)$-uniform (multi)hypergraph
  ("$(t-1)$-graph") $\mathcal{H}(V, E)$: identify $e \in E$ with $\{v \mid \{e, v\} \in I\}$
- For PIR codes (bipartite girth ≥ 6 or Berge girth ≥ 3)
  equivalently a 2-$(|V|, t-1, 1)$ packing design. Asymptotically optimal packing designs exist (for large enough $|V|$).
- For batch codes (bipartite girth ≥ 8 or Berge girth ≥ 4) use solution of "$(3(t-1)-3, 3)$-problem" (cf. "Ruzsa-Szemerédi $(6, 3)$-problem"):

Hypergraph $(6,3)$-problem

- [Brown, Erdős and Sós] $H^{(t-1)}(n-k;\kappa,s)$: max. number of hyperedges of an $(t-1)$-graph on $n-k$ vertices whose no set of κ vertices contains s or more hyperedges (fixed t,κ,s)
- [Ruzsa and Szemerédi] essentially solved first open case $H^{(3)}(n-k;6,3)$, known as the $(6,3)$-problem
- [Erdős, Frankl and Rödl] essentially found $H^{(t-1)}(n-k;3(t-1)-3,3)$, solved $(3(t-1)-3,3)$-problem
  We applied their solution to graph-based batch codes

• **W. G. Brown, P. Erdős, and V.T. Sós,** "Some extremal problems on r-graphs", *New Directions in the Theory of Graphs, 3rd Ann. Arbor Conference on Graph Theory*, Academic Press, pp. 55–63, 1973.

• **W. G. Brown, P. Erdős, and V.T. Sós,** "On the existence of triangulated spheres in 3-graphs and related problems", *Periodica Mathematica Hungaria*, vol. 3, pp. 221–228, 1973.

• **I.Z. Ruzsa, E. Szemerédi,** "Triple systems with no six points carrying three triangles", *Coll. Math. Soc. Janos Bolyai*, no. 18, pp. 939–945, 1978.

• **P. Erdös, P. Frankl, V. Rödl,** "The asymptotic number of graphs not containing a fixed subgraph and a problem for hypergraphs having no exponent", *Graphs and Combinatorics*, vol. 2, No. 1, pp. 113–121, 1986.

Hypergraph $(6, 3)$-problem

- PROPOSITION (Riet, Skachek, Thomas):

$$H^{(t-1)}(n-k;\, 3(t-1)-3,\, 3) = B^{(t-1)}(n-k;\, 4)$$

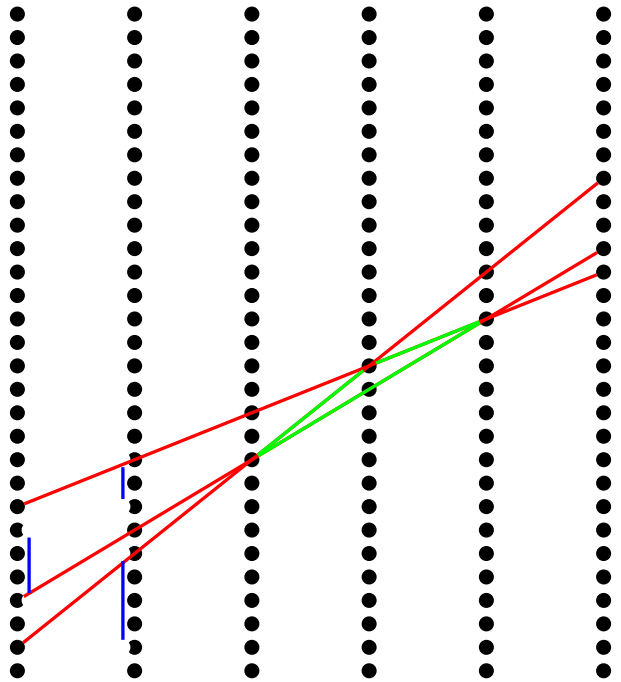
  where $B^{(t-1)}(n-k;\, 4)$ is max. number of hyperedges in a $(t-1)$-graph on $(n-k)$ vertices with Berge girth ≥ 4

- **Proof outline.** A $(t-1)$-graph with no 3 hyperedges contained in any set of $3(t-1)-3$ vertices, can be modified slightly to have no Berge 2- or 3-cycle. A $(t-1)$-graph with no Berge 2- or 3-cycles already has no 3 hyperedges contained in any set of $3(t-1)-3$ vertices

• **P. Erdös, P. Frankl, V. Rödl,** "The asymptotic number of graphs not containing a fixed subgraph and a problem for hypergraphs having no exponent", *Graphs and Combinatorics*, vol. 2, No. 1, pp. 113–121, 1986.

Erdős, Frankl and Rödl construction

- Arrange vertices as $\lfloor (n-k)/(t-1) \rfloor$-by-$(t-1)$-grid.
  Hyperedges are lines of $t-1$ points with restricted slopes:
  Avoid 3-cycles (triangles):



3 slopes, 3 elements (here 4,3,2) of $(t-1)$-element arithmetic progression might give rise to a Berge 3-cycle

Erdős, Frankl and Rödl construction

- [Behrend] constructed a big subset of $\{1, 2, \ldots, N\}$ containing no 3-term arithmetic progression $(a, a + b, a + 2b)$
- [Erdős, Frankl and Rödl] modified the contstruction, giving a big subset $A \subseteq \{1, 2, \ldots, N\}$, containing no 3 terms of any $(t - 1)$-term arithmetic progression
- In the grid, use only lines with slopes from A
- Obtain hypergraph $\mathcal{H}(V, E)$ without (Berge) 2- or 3-cycles
- The respective bipartite graph $G(E, V, S)$ has girth ≥ 8, producing a graph-based batch code

• **F.A. Behrend** "On sets of integers which contain no three elements in arithmetic progression", *Nat. Acad. Sci.*, no. 23, pp. 331–332, 1946.

• **P. Erdös, P. Frankl, V. Rödl,** "The asymptotic number of graphs not containing a fixed subgraph and a problem for hypergraphs having no exponent", *Graphs and Combinatorics*, vol. 2, No. 1, pp. 113–121, 1986.

Erdős, Frankl and Rödl construction and bound

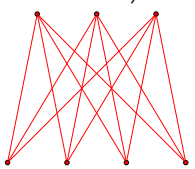
- Gives $k = \Omega((n-k)^{2-\epsilon})$ hyperedges for any $\epsilon > 0$
- Reminder: there are $n-k$ redundant symbols/vertices and $k = (n-k)^{2-\epsilon}$ information symbols/hyperedges and $t - 1 \geq 3$
- Redundancy $r = n - k = O(k^{1/(2-\epsilon)})$
- [Erdős, Frankl and Rödl] use (early version) Szemerédi's Regularity Lemma to bound number of hyperdges to $o((n-k)^2)$, so $\lim \frac{r}{\sqrt{k}} \to \infty$ for redundancy r of graph-based batch codes

• **J. Komlós, A. Shokoufandeh, M. Simonovits, and E. Szemerédi** "The Regularity Lemma and Its Applications in Graph Theory", in: G. Khosrovshahi, A. Shokoufandeh, and A. Shokrollahi(eds) "Theoretical Aspects of Computer Science", Springer, pp. 84–112, 2002.

• **P. Erdös, P. Frankl, V. Rödl,** "The asymptotic number of graphs not containing a fixed subgraph and a problem for hypergraphs having no exponent", *Graphs and Combinatorics*, vol. 2, No. 1, pp. 113–121, 1986.

Case $t - 1 = 2$

- For 2-graphs (graphs) we need to avoid multiple-edges and 3-cycles
- Max. number of edges in a graph on $n - k$ vertices with no triangles is $\lfloor \frac{(n-k)^2}{4} \rfloor$ by Mantel's (Turán's) Theorem
- The complete bipartite graph $\mathcal{H}(V, E) = K_{\lfloor (n-k)/2 \rfloor, \lceil (n-k)/2 \rceil}$ attains this bound. Example: $K_{3,4}$:



- Can construct respective bipartite graph $G(E, V, S)$ which is 2-left-regular and has girth ≥ 8
- For graph-based batch codes for $t = 3$ thus redundancy

$$r = n - k = \Theta(\sqrt{k})$$

Open questions

- [Rao, Vardy] proved redundancy $r = \Omega(\sqrt{k})$ for PIR codes for t=3. Thus this holds for $t \geq 3$ and for batch codes for $t \geq 3$
- [Vardy, Yaakobi] showed for batch codes $r = O(\sqrt{k})$ for $t = 3, 4$ and $r = O(\sqrt{k} \log k)$ for $t \geq 5$
- Note gap for $t = 4$ between $O(\sqrt{k})$ and $\omega(\sqrt{k})$ general/graph-based batch codes
- For $t \geq 4$ find asymptotics of optimal redundancy for graph-based batch codes
- Is there a gap for $t \geq 5$ between optimal redundancy of batch codes ($O(\sqrt{k} \log k)$) and graph-based batch codes ($O(k^{1/(2-\epsilon)})$ and $\omega(\sqrt{k})$)?
- Find good asynch. batch codes other than graph-based

• **S. Rao and A. Vardy,** "Lower Bound on the Redundancy of PIR Codes", arXiv:1605.01869, May 2016.

• **A. Vardy and E. Yaakobi,** "Constructions of batch codes with near-optimal redundancy", *Proc. ISIT*, Barcelona, pp. 1197-1201, July 2016.

Functional Batch Code Conjecture

On the Functional Batch Code Conjecture:

- Introduction: data recovery, definitions of batch-type codes
- Simplex code as batch code
- Functional Batch Code Conjecture and related results
- Two results as corollary of Marshall Hall, Jr. 1952, "A combinatorial problem on abelian groups"
- Rainbow matchings
- Future directions

Data recovery

Encoding: data $\boldsymbol{a} = (a_1, \ldots, a_k)$ stored as $\boldsymbol{c} = (c_1, \ldots, c_n) = \boldsymbol{aG}$

Recovering data:

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{g}_1^\top & \boldsymbol{g}_2^\top & \cdots & \boldsymbol{g}_k^\top \end{bmatrix}$$

If $\sum_{i \in I} \boldsymbol{g}_i = \boldsymbol{e}_j = (0, \ldots, 0, 1, 0, \ldots, 0)$,

that is, $\boldsymbol{G} \cdot \chi_I = \boldsymbol{e}_j^\top$, then $\boldsymbol{c} \cdot \chi_I = \boldsymbol{a} \cdot \boldsymbol{G} \cdot \chi_I = \boldsymbol{a} \cdot \boldsymbol{e}_j^\top$,

so $\sum_{i \in I} c_i = a_j$.

So property of interest $\sum_{i \in I} \boldsymbol{g}_i = \boldsymbol{e}_j$ depends on $\boldsymbol{G}$ and not on data!

Serving a request

Definition

$k \times n$ matrix $\boldsymbol{G}$ can serve request sequence/multiset $\boldsymbol{r}_1, \ldots, \boldsymbol{r}_t$ of (not necessarily distinct) vectors in $\mathbb{F}_2^k$ if
$\exists$ **p/w disjoint** *column index sets $I_1, \ldots, I_t$ s.t. $\boldsymbol{r}_j \in \langle$col's I_j of $\boldsymbol{G}\rangle$*

$\sum_{i \in I_j} \boldsymbol{g}_i = \boldsymbol{r}_j$, *for $j = 1, \ldots, t$ over $\mathbb{F}_2$*

Codes are functional: $\forall$ individual request $\boldsymbol{r}$ is for lin. comb.

$$\boldsymbol{a}\boldsymbol{r}^\top = a_1 r_1 + \cdots + a_k r_k$$

of the data symbols

Example

$$\boldsymbol{G}_2 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{e}_1{}^\top & \boldsymbol{e}_2{}^\top & (\boldsymbol{e}_1 + \boldsymbol{e}_2)^\top \end{bmatrix}$$

$$\boldsymbol{a} = (a_1, a_2) \longrightarrow \boldsymbol{c} = \boldsymbol{a}\boldsymbol{G}_2 = (c_1, c_2, c_3) = (a_1, a_2, a_1 + a_2)$$

2-PIR:

$\boldsymbol{e}_1, \boldsymbol{e}_1 \longleftarrow \{1\}, \{2, 3\}$ two same requests: one+two columns

2-batch:

$\boldsymbol{e}_1, \boldsymbol{e}_2 \longleftarrow \{1\}, \{2\}$ two different requests: one+one column

2-functional-batch:

$\boldsymbol{e}_1, \boldsymbol{e}_1 + \boldsymbol{e}_2 \longleftarrow \{1\}, \{3\}$

$\boldsymbol{e}_1 + \boldsymbol{e}_2, \boldsymbol{e}_1 + \boldsymbol{e}_2 \longleftarrow \{1, 2\}, \{3\}$

PIR/batch code definitions

An encoder (= a generator matrix $\boldsymbol{G}$) is

1. t-PIR 2. t-batch 3. t-odd-batch 4. t-functional-batch	if $\boldsymbol{G}$ can serve any request sequence consisting of	1. a t-fold repetition of a unit vector 2. t unit vectors 3. t odd-weight (not in hyperplane) vectors 4. t arbitrary vectors

Proposition

Any t-PIR code $\boldsymbol{G}$ (the most general, thus each of the previous) generates a code with minimum distance at least t.

Simplex code

Binary simplex code, dimension k, length $n = 2^k - 1$,
generator matrix $\boldsymbol{G}_k \in F_2^{k \times (2^k - 1)}$

$$\boldsymbol{G}_k = \begin{bmatrix} 1 & 0 & 1 & 0 & & 0 & 1 \\ 0 & 1 & 1 & 0 & & 1 & 1 \\ 0 & 0 & 0 & 1 & \cdots & 1 & 1 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & 0 & & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1^\top & 2^\top & 3^\top & \cdots & (2^k - 1)^\top \end{bmatrix}$$

Has minimal distance 2^{k-1}, so "optimal" would be $t = 2^{k-1}$.

Greedy does not always work (using only recovery sets size ≤ 2)

$$\boldsymbol{G}_3 = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

$t = 2^{k-1} = 4$.

Request: $\boldsymbol{r}_1 = 1, \boldsymbol{r}_2 = 2, \boldsymbol{r}_3 = 1, \boldsymbol{r}_4 = 2$.

Serve $\boldsymbol{r}_1 = 1$ and $\boldsymbol{r}_2 = 2$ by $\{1\}$ and $\{2\}$ (cheapest).

Then $\boldsymbol{r}_3 = 1$ by $\{4, 5\}$ or $\{6, 7\}$;

and $\boldsymbol{r}_4 = 2$ by $\{4, 6\}$ or $\{5, 7\}$.

Not possible!

Use service $\{1\}, \{4, 6\}, \{2, 3\}, \{5, 7\}$.

Balister, Győri, Schelp 2011 Conj.: $\forall$ multiset of vectors of $\mathbb{F}_2^k$, size 2^{k-1}, with sum 0, $\exists$ partition of $\mathbb{F}_2^k$ into pairs with sums the elements of the multiset. Still OPEN! Equivalent to:

Yamawaki, Kamabe, Lu 2017; Yohananov, Yaakobi 2022 "Hadamard solution version" Functional Batch Code Conj.: $\forall$ 2^{k-1} requests from $\mathbb{F}_2^k$ can be served with recovery sets size ≤ 2, all but ≤ 2 size $= 2$. I.e. $\boldsymbol{G}_k$ is 2^{k-1}-functional-batch.

Hollmann, Khathuria, R., Skachek 2023: $\boldsymbol{G}_k$ can serve every request of 2^{k-1} vectors of odd weight (or outside a hyperplane).

Lemma in **Lember, R. 2024**: $[\boldsymbol{G}_k \boldsymbol{G}_k]$ can serve every request of 2^k vectors of any weight.

Last two also corollaries to **M. Hall, Jr. 1952 "A combinatorial problem on abelian groups"**:

Marshall Hall, Jr.'s theorem (1952) on abelian groups

Theorem **Marshall Hall, Jr. 1952**: $\forall$ sequence (r_i), $i = 1, \ldots, n$, with $r_i \in A$, A abelian group, $|A| = n$, $\sum_{i=1}^{n} r_i = 0$, $\exists$ permutations $(a_i)_1^n$ and $(b_i)_1^n$ of A such that $r_i = b_i - a_i \ \forall \ i = 1, \ldots, n$.

Proof by quadratic-time algorithm: assign a_i, b_i to all r_i in growing order of i, at each step possibly changing the assignments for $j < i$.

Partial proofs of Conjecture: information/coding theory

- **Wang, Kiah, Cassuto 2015** $\boldsymbol{G}_k$ is 2^{k-1}-batch

  computer proof for $k \leq 8$, then induction algorithm
- **Hollmann, Khathuria, R., Skachek 2023**: $\boldsymbol{G}_k$ is 2^{k-1}-odd-batch $\rightarrow$ 2^{k-1}-batch, (vectors outside hyperplane)
- **Yohananov, Yaakobi 2022**: $\boldsymbol{G}_k$ is t-functional-batch for $t = \lfloor (5/6)2^{k-1} \rfloor - k$, involved proof
- **Lember, R. 2024**: $[\boldsymbol{G}_k \boldsymbol{G}_k]$ is 2^k-functional-batch

Partial proofs of Conjecture: combinatorics

- **Balister, Győri, Schelp 2011** Conjecture, proved simple case
- **Preissmann, Mischler 2009** "Seating couples problem": true in $\mathbb{F}_p$ (instead of $\mathbb{F}_2^k$). Independently proved in **Hollmann, Khathuria, R., Skachek 2023**
- **Kovács 2023** "Finding a perfect matching of $\mathbb{F}_2^n$ with prescribed differences", arXiv, see for references True for:
  - few distinct requests;
  - most requests equal
- **Correia, Pokrovskiy, Sudakov 2023** "Short proofs of rainbow matching results" t-functional-batch for $t = 2^{k-1} - O(2^{15k/16})$ for k large. General result for "full rainbow matchings" . Probabilistic $\exists$-proof, *weak* $\approx\Rightarrow$ *strong*
- see: **Bowtell, Freschi, Kronenberg, Yan 2025+** "A note on improved bounds for hypergraph rainbow matching problems"

Conjecture in rainbow matchings

- Conjecture. n matchings (not necessarily disjoint) of size $n+2$ have a rainbow matching of size n.
- Formulated by **Gao, Ramadurai, Wanless, Wormald 2021**, noticed by **Aharoni, Berger, Chudnovsky, Howard 2019**.
- Question. Do n (not necessarily disjoint) matchings of size $n+1$ have a rainbow matching of size n?
- Known counterexamples from $\sum_{i=2}^{2^k} \boldsymbol{r}_i = 0 \Rightarrow$ since $\sum_{i=1}^{2^k} \boldsymbol{r}_i = 0$, we have $\boldsymbol{r}_1 = 0$. $\lightning$
- No different counterexamples $\Rightarrow$ Functional Batch Code Conjecture: For $\forall$ request $\boldsymbol{r}_i$ take new copies of $\forall$ edges $\boldsymbol{a}_i\boldsymbol{b}_i$ with $\boldsymbol{r}_i = \boldsymbol{a}_i + \boldsymbol{b}_i$ to form i^{th} matching.

Ryser-Brualdi-Stein and generalizations

↓ Stronger

- **Ryser-Brualdi-Stein** Conj: $\forall$ partition of edge-set of $K_{n,n}$ into n matchings, $\exists$ rainbow matching size $n-1$

  Montgomery 2023+: true for large even n
- Matchings may intersect, not on same vertex set:

  Aharoni, Berger 2009 Conj: $\forall$ n matchings size n in bipartite graph, $\exists$ rainbow matching size $n-1$
- Graph need not be bipartite: **Aharoni, Berger 2009** *(weak)* Conj: $\forall$ n matchings size n, $\exists$ rainbow matching size $n-1$
- Need full rainbow matching: **Aharoni, Berger, Chudnovsky, Howard, Seymour 2019; Gao, Ramadurai, Wanless, Wormald 2021** Conj: $\forall$ n matchings size $n+2$, $\exists$ rainbow matching size n *(strong)*

  Stronger if size $n+1$: add disjoint edge e to all matchings

Conclusion

- The Functional Batch Code Conjecture or the conjecture that $\boldsymbol{G}_k$ is (optimal) 2^{k-1}-functional-batch is open and very interesting, and needs everyone's attention!

See also our DCC paper, or also on arXiv: **Hollmann, Khathuria, R., Skachek 2023** "On some batch code properties of the simplex code" Conjectures on abelian groups; application of Alon's Nullstellensatz.

Lember, R. 2024 "Equal requests are asymptotically hardest for data recovery", e.g. arXiv, probabilistic results, $\frac{1}{2}$-fractional functional batch code conjecture proof: $[\boldsymbol{G}_k \boldsymbol{G}_k]$ serves 2^k requests

THANK YOU! *Ago-Erik Riet, Univ. of Tartu, ago-erik.riet @ ut.ee*

Odd-batch and $\frac{1}{2}$-fractional-batch as corollaries

Odd-batch (outside hyperplane) unpublished proof: For hyperplane $H \subseteq \mathbb{F}_2^k$, i.e. $(k-1)$-dim. linear subspace, i.e. $\mathbb{F}_2^k = H \cup (H + \boldsymbol{a})$, given sequence $\boldsymbol{r}_1, \ldots, \boldsymbol{r}_{2^{k-1}} \in H + \boldsymbol{a}$, let $\boldsymbol{r}'_i = \boldsymbol{r}_i - \boldsymbol{a}$, $i < 2^{k-1}$, $\boldsymbol{r}'_{2^{k-1}} = -\sum_{i<2^{k-1}} \boldsymbol{r}'_i$. Get permutations $(\boldsymbol{y}'_i)$, $(\boldsymbol{x}'_i)$ of H with $\boldsymbol{r}'_i = \boldsymbol{y}'_i - \boldsymbol{x}'_i$. Then $\forall i : \boldsymbol{x}_i = \boldsymbol{x}'_i \in H$, $\forall i : \boldsymbol{y}_i = \boldsymbol{y}'_i + \boldsymbol{a} \in H + \boldsymbol{a}$ are all distinct, and $\boldsymbol{r}_i = \boldsymbol{y}_i - \boldsymbol{x}_i$, $i < 2^{k-1}$. Adding fixed $\boldsymbol{b}$ to each $\boldsymbol{x}_i, \boldsymbol{y}_i$, w.m.a. $\boldsymbol{x}_{2^{k-1}} = 0$, $\boldsymbol{r}_{2^{k-1}} = \boldsymbol{y}_{2^{k-1}}$. Serve from $\boldsymbol{G}_k$.

$\frac{1}{2}$-fractional-batch (Lember, R. 2024) Given request sequence $\boldsymbol{r}_1, \ldots, \boldsymbol{r}_{2^k} \in \mathbb{F}_2^k$, let $\boldsymbol{r}'_i = \boldsymbol{r}_i$, $i < 2^{k-1}$, $\boldsymbol{r}'_{2^{k-1}} = -\sum_{i<2^{k-1}} \boldsymbol{r}'_i$. Get permutations $(\boldsymbol{y}'_i)$, $(\boldsymbol{x}'_i)$ of $\mathbb{F}_2^k$ with $\boldsymbol{r}'_i = \boldsymbol{y}'_i - \boldsymbol{x}'_i$. Then letting $\boldsymbol{x}_i = \boldsymbol{x}'_i + \boldsymbol{b}$, $\boldsymbol{y}_i = \boldsymbol{y}'_i + \boldsymbol{b}$ for such fixed $\boldsymbol{b}$ that $\boldsymbol{y}_{2^k} = \boldsymbol{r}_{2^k}$, we have $\boldsymbol{r}_i = \boldsymbol{y}_i - \boldsymbol{x}_i$, $\forall i < 2^k$ and $\boldsymbol{r}_{2^k} = \boldsymbol{y}_{2^k}$. Serve from two copies of $\boldsymbol{G}_k$.