Introduction to rank metric codes

Maximilien Gadouleau
Durham University

Virtual Non-Lattice (!) Coding & Crypto Meeting
02/10/2020

Outline

Outline

Why rank metric codes?

Typical error-correcting codes are based on the Hamming metric: the codewords are vectors (typically $\mathbf{x} = (x_1, \ldots, x_n) \in \mathrm{GF}(q)^n$), and the distance between two codewords is their Hamming distance:

$$d_{\mathrm{H}}(\mathbf{x}, \mathbf{y}) = |\{i : x_i \neq y_i\}|$$

(the number of times they disagree).

But in some circumstances, we may want to use different kinds of codewords and a different metric.

Rank metric codes

For rank metric codes, the codewords are not vectors, but matrices.

Formally, a rank metric code is a subset $\mathscr{C}$ of $\mathrm{GF}(q)^{m\times n}$.

The rank distance is

$$d_{\mathrm{R}}(\mathbf{M},\mathbf{N}) = \mathrm{rank}(\mathbf{M}-\mathbf{N}).$$

(Yes, it is a metric.)

The minimum rank distance of $\mathscr{C}$ is then

$$d_{\mathrm{R}}(\mathscr{C}) = \min\{d_{\mathrm{R}}(\mathbf{M},\mathbf{N}) : \mathbf{M}\neq\mathbf{N}, \mathbf{M},\mathbf{N}\in\mathscr{C}\}.$$

Historical note. Rank metric codes were independently discovered in (Delsarte 78, Gabidulin 85, Roth 91).

The vector view of the rank metric

We can identify a vector in $\mathrm{GF}(q)^m$ as an element of of $\mathrm{GF}(q^m)$. Then the whole matrix $\mathbf{M}$ corresponds to a vector $\mathbf{x}$ in $\mathrm{GF}(q^m)^n$.

We still have $d_{\mathrm{R}}(\mathbf{x},\mathbf{y}) = \mathrm{rank}(\mathbf{x}-\mathbf{y})$, where $\mathrm{rank}(z)$ is the number of linearly independent coordinates of z.

This is similar to the Hamming metric: $d_{\mathrm{H}}(\mathbf{x},\mathbf{y}) = w_{\mathrm{H}}(\mathbf{x}-\mathbf{y})$, where $w_{\mathrm{H}}(z)$ is the number of nonzero coordinates of z.

Note that $\mathrm{rank}(\mathbf{z}) \leq w_{\mathrm{H}}(\mathbf{z})$, thus

$$d_{\mathrm{R}}(\mathbf{x},\mathbf{y}) \leq d_{\mathrm{H}}(\mathbf{x},\mathbf{y}).$$

The rank metric looks like Hamming

The similarity doesn't stop there!

- Hamming and rank metrics are (self-dual, translation) association schemes (Delsarte 78).
- There is a MacWilliams identity for rank metric codes (Delsarte 78, Gadouleau and Yan 08, Ravagnani 14).
- Number of vectors with (rank or Hamming) weight r:

$$N_{\mathrm{H}}(q^m, n, r) = \binom{n}{r} \prod_{i=0}^{r-1} (q^m - 1)$$

$$N_{\mathrm{R}}(q^m, n, r) = \begin{bmatrix} n \\ r \end{bmatrix} \prod_{i=0}^{r-1} (q^m - q^i),$$

where $\binom{n}{r}$ is the number of r-subsets of an n-set and $\begin{bmatrix} n \\ r \end{bmatrix}$ is the number of r-dimensional subspaces of an n-dimensional vector space over GF(q).

Outline

Singleton bound

Problem. What is the maximum cardinality $A_{\mathrm{R}}(q,m,n,d)$ of a rank metric code $\mathscr{C} \subseteq \mathrm{GF}(q)^{m\times n}$ with minimum distance $d_{\mathrm{R}}(\mathscr{C}) \geq d$?

Firstly, the rank distance is preserved by transposition (since $\mathrm{rank}(\mathbf{M}^\top) = \mathrm{rank}(\mathbf{M})$) so $A_{\mathrm{R}}(q,m,n,d) = A_{\mathrm{R}}(q,n,m,d)$. We only need to focus on $m \geq n$ ("tall and thin" matrices).

Let us use the vector view ($\mathscr{C} \subseteq \mathrm{GF}(q^m)^n$). Since $d_{\mathrm{R}}(\mathscr{C}) \leq d_{\mathrm{H}}(\mathscr{C})$, the Singleton bound applies:

$$A_{\mathrm{R}}(q,m,n,d) \leq q^{m(n-d+1)}.$$

In fact, it is always reached:

$$A_{\mathrm{R}}(q,m,n,d) = q^{m(n-d+1)}.$$

Gabidulin codes

Reed-Solomon codes (extended or punctured) can be defined based on polynomial evaluations.

For $n \leq q^m$, choose n distinct elements $g_1, g_2, \ldots, g_n$ of $\mathrm{GF}(q^m)$. For any polynomial $f \in \mathrm{GF}(q^m)[x]$, let
$\mathbf{f} = (f(g_1), f(g_2), \ldots, f(g_n)) \in \mathrm{GF}(q^m)^n$, then

$$\mathscr{RS}(n,k) = \{\mathbf{f} : \deg(f) \leq k-1\}.$$

It reaches the Singleton bound for the Hamming metric:
$|\mathscr{RS}(n,k)| = q^{mk}$ and $d_{\mathrm{H}} = n-k+1$.

Gabidulin codes

Gabidulin codes are based on evaluations of linear maps (self-maps of $\mathrm{GF}(q^m)$ which are linear over $\mathrm{GF}(q)$).

A linearised polynomial is any of the form

$$f(x) = \sum_{i=0}^{m-1} f_i x^{q^i}, \quad f_i \in \mathrm{GF}(q^m).$$

The largest d such that $f_d \neq 0$ is called the q-degree of f.

For $n \leq m$, choose n linearly independent elements $g_1, g_2, \ldots, g_n$ of $\mathrm{GF}(q^m)$. For any linearised polynomial f, let $\mathbf{f} = (f(g_1), f(g_2), \ldots, f(g_n)) \in \mathrm{GF}(q^m)^n$, then

$$\mathscr{G}(n,k) = \{\mathbf{f} : q\text{-}\deg(f) \leq k-1\}.$$

It reaches the Singleton bound for the rank metric: $|\mathscr{G}(n,k)| = q^{mk}$ and $d_{\mathrm{R}} = n-k+1$.

Generator and parity-check matrices

Generator matrix: $g_1, g_2, \ldots, g_n \in \mathrm{GF}(q^m)$ linearly independent

$$\mathbf{G} = \begin{pmatrix} g_1^{q^0} & g_2^{q^0} & \cdots & g_n^{q^0} \\ g_1^{q^1} & g_2^{q^1} & \cdots & g_n^{q^1} \\ \vdots & \vdots & \vdots & \vdots \\ g_1^{q^{k-1}} & g_2^{q^{k-1}} & \cdots & g_n^{q^{k-1}} \end{pmatrix}$$

Parity-check matrix: the same form!

Decoding of Gabidulin codes

Because of their structure, many decoding algorithms for RS codes have been adapted to Gabidulin codes, e.g.

- Extended Euclidean Algorithm (Gabidulin 85)
- PGZ (Roth 91)
- Berlekamp-Massey (Richter and Plass 04)
- Welch-Berlekamp (Loidreau 05)
- Sudan 1-list (K otter and Kschischang 07)

The ring of linearised polynomials

The EEA decoding algorithm for Reed-Solomon codes works in the (commutative) ring of polynomials, with addition and multiplication.

For Gabidulin codes, we must work in the non-commutative ring of linearised polynomials, with addition and q-product:

$$f(x) = \sum_{i=0}^{m-1} f_i x^{q^i}$$

$$g(x) = \sum_{i=0}^{m-1} g_i x^{q^i}$$

$$f * g(x) = \sum_{i=0}^{m-1} \left(\sum_{j=0}^{i} f_j \cdot g_{i-j}^{q^j} \right) x^{q^i}.$$

The q-product in fact reflects the composition of linear maps, or equivalently the product of matrices.

Outline

Rank metric codes for storage

Suppose you store data in a two-dimensional array (say m rows and n columns), each cell containing an element of GF(q). So your data can be viewed as a matrix $\mathbf{M} \in \mathrm{GF}(q)^{m \times n}$.

The data can be corrupted in two main ways: an entire row or column can be corrupted, e.g.

$$\mathbf{M} = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \\ j & k & l \end{pmatrix} \to \mathbf{M}_1 = \begin{pmatrix} a & b & c \\ * & * & * \\ g & h & i \\ j & k & l \end{pmatrix},$$

or

$$\mathbf{M} = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \\ j & k & l \end{pmatrix} \to \mathbf{M}_2 = \begin{pmatrix} * & b & c \\ * & e & f \\ * & h & i \\ * & k & l \end{pmatrix}.$$

Rank metric codes for storage

Corrupting a line (row or column) can be viewed as adding an error matrix of rank one, e.g. $\mathbf{M}_1 = \mathbf{M} + \mathbf{E}_1$, where $\mathbf{E}_1$ has nonzero entries only on the second row.

If multiple errors occur (say t lines are corrupted), then we obtain the matrix $\mathbf{N}$, where

$$\mathbf{N} = \mathbf{M} + \mathbf{E}, \quad \operatorname{rank}(\mathbf{E}) \leq t.$$

This is called crisscross errors (Roth 91).

Then clearly, a rank metric code with $d_{\mathrm{R}}(\mathscr{C}) \geq 2t+1$ can correct t crisscross errors.

Outline

McEliece's original system (1978)

Secret key:

- $\bar{\mathbf{G}}$: generator matrix of a binary irreducible Goppa code.
- $\mathbf{S}$: $k \times k$ nonsingular matrix.
- $\mathbf{P}$: $n \times n$ permutation matrix.

Public key: $\mathbf{G} = \mathbf{S}\bar{\mathbf{G}}\mathbf{P}$.

Encryption: $\mathbf{v} = \mathbf{m}\mathbf{G} + \mathbf{e}$.

Decryption:

1. Compute $\mathbf{z} = \mathbf{v}\mathbf{P}^{-1}$.
2. Decode $\mathbf{z}$ to obtain a codeword $\mathbf{c}$ and the corresponding original message $\bar{\mathbf{m}}$ (for $\bar{\mathbf{G}}$).
3. Compute $\mathbf{m} = \bar{\mathbf{m}}\mathbf{S}^{-1}$.

Original parameters:

- plaintext size: $k = 524$ bits
- ciphertext size: $n = 1024$ bits
- error weight: 50

The trouble with the Hamming metric

Suppose that the encoding did not involve adding an error: $\mathbf{v} = \mathbf{mG}$. Then the attacker could easily find the message:

1. Find a $k \times k$ nonsingular matrix $\mathbf{G}_I$ of $\mathbf{G}$ (I is an information set)
2. Then $\mathbf{m} = \mathbf{v}_I \mathbf{G}_I^{-1}$.

In general, information set decoding (Prange 62) then keeps choosing information sets until it finds one that is not corrupted by the error vector.

A long list of improvements of the original idea, see (Becker et al. 12) and (May and Ozerov 15).

The GPT cryptosystem

But the rank metric allows to scramble all positions! For instance, the vector $\mathbf{e} = (a, a, \ldots, a)$ for any $a \in \mathrm{GF}(q^m)^*$ has rank one.

(Gabidulin, Paramonov and Tretjakov 91) proposed the GPT cryptosystem, based on Gabidulin codes.

Unfortunately, Gabidulin codes are too well structured, and that system was broken by Overbeck.

GPT Cryptosystem

Private key:

- $\mathbf{G}$: $(k \times n)$ generator matrix of a Gabidulin code over GF(q^m) with error correction capability t
- $\mathbf{S}$: $(k \times k)$ invertible matrix (scrambler)
- $\mathbf{X}$: $(k \times n)$ distortion matrix with distortion parameter t_1. For any plaintext $\mathbf{c}$, rank$(\mathbf{cX}) = t_1$

Public key: $\mathbf{G}' = \mathbf{SG} + \mathbf{X}$ and $t - t_1$

GPT Cryptosystem

- Encryption: $\mathbf{y} = \mathbf{cG}' + \mathbf{e}$ with $\text{rank}(\mathbf{e}) = t - t_1$
- Decryption: $\mathbf{y} = \mathbf{cSG} + (\mathbf{cX} + \mathbf{e})$
  - The decryption algorithm gives $\mathbf{c}' = \mathbf{cS}$
  - $\mathbf{c} = \mathbf{c}'\mathbf{S}^{-1}$
- Parameters: $q = 2, m = n = 32$

Outline

Network coding
(Yeung and Zhang 99, Ahlswede *et al.* 00)

- Routing does not reach maximum throughput for mulitcast
- Network coding: intermediate nodes combine packets
- Examples
  - Packet selection: routing
  - Linear combinations: Linear network coding
- Advantages
  - Higher throughput for multicast
  - Greater adaptability to topology changes
  - Robustness to packet losses

Network coding: the butterfly network

Each node sends the same message; $x_1, x_2 \in \mathrm{GF}(2)$.

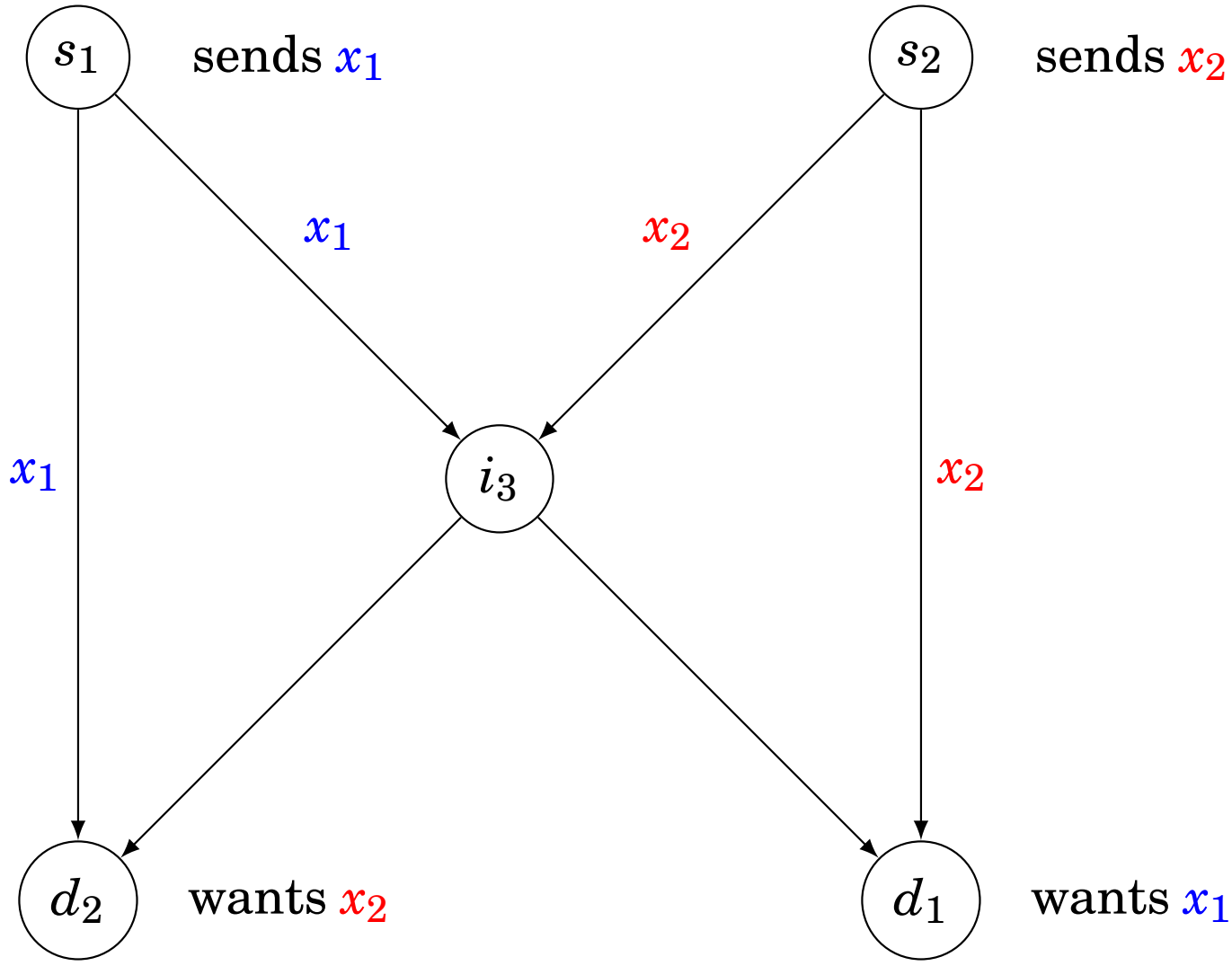

Figure: The butterfly

Network coding: the butterfly network

Each node sends the same message; $x_1, x_2 \in \mathrm{GF}(2)$.

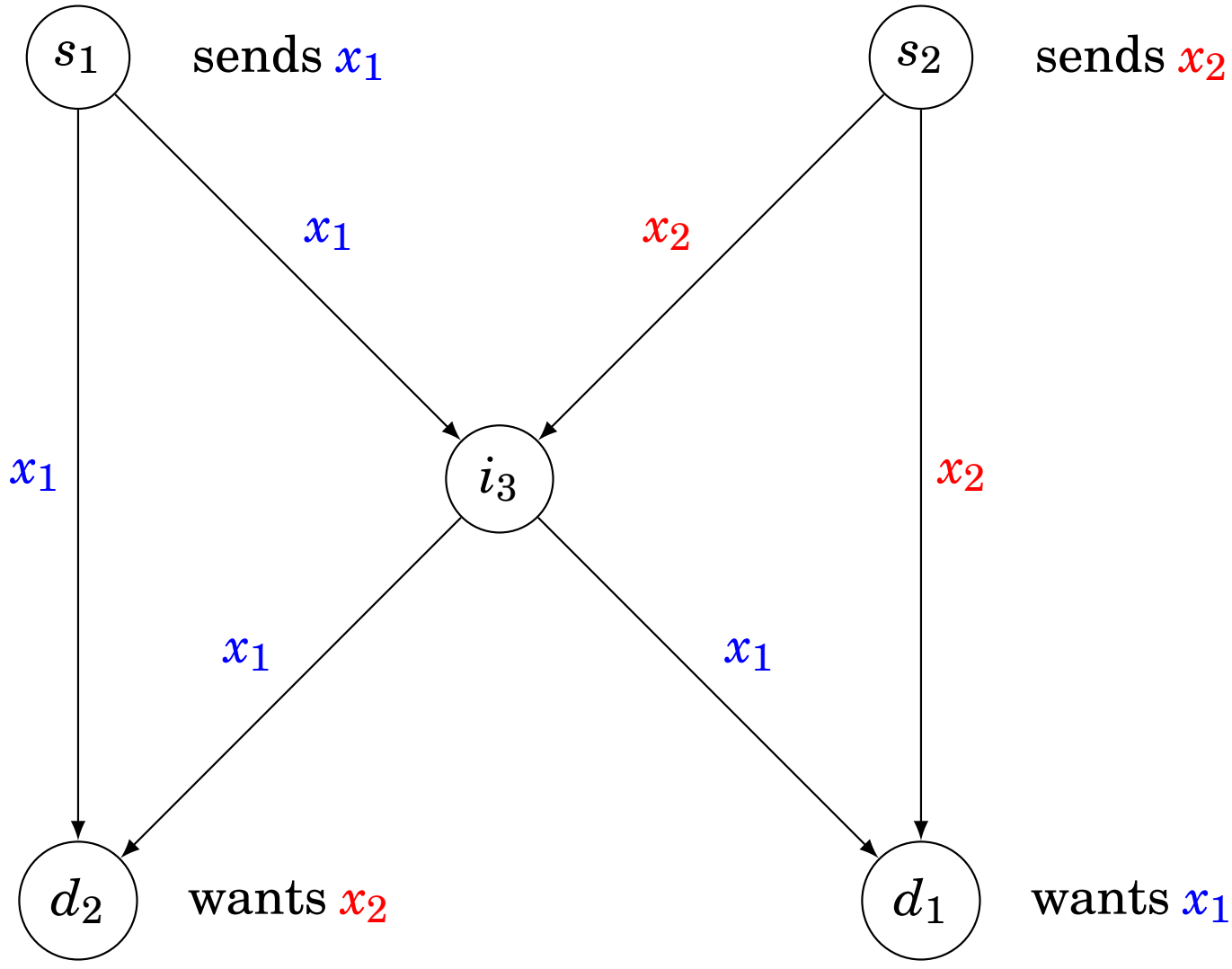

Figure: The butterfly: Routing

Network coding: the butterfly network

Each node sends the same message; $x_1, x_2 \in \mathrm{GF}(2)$.

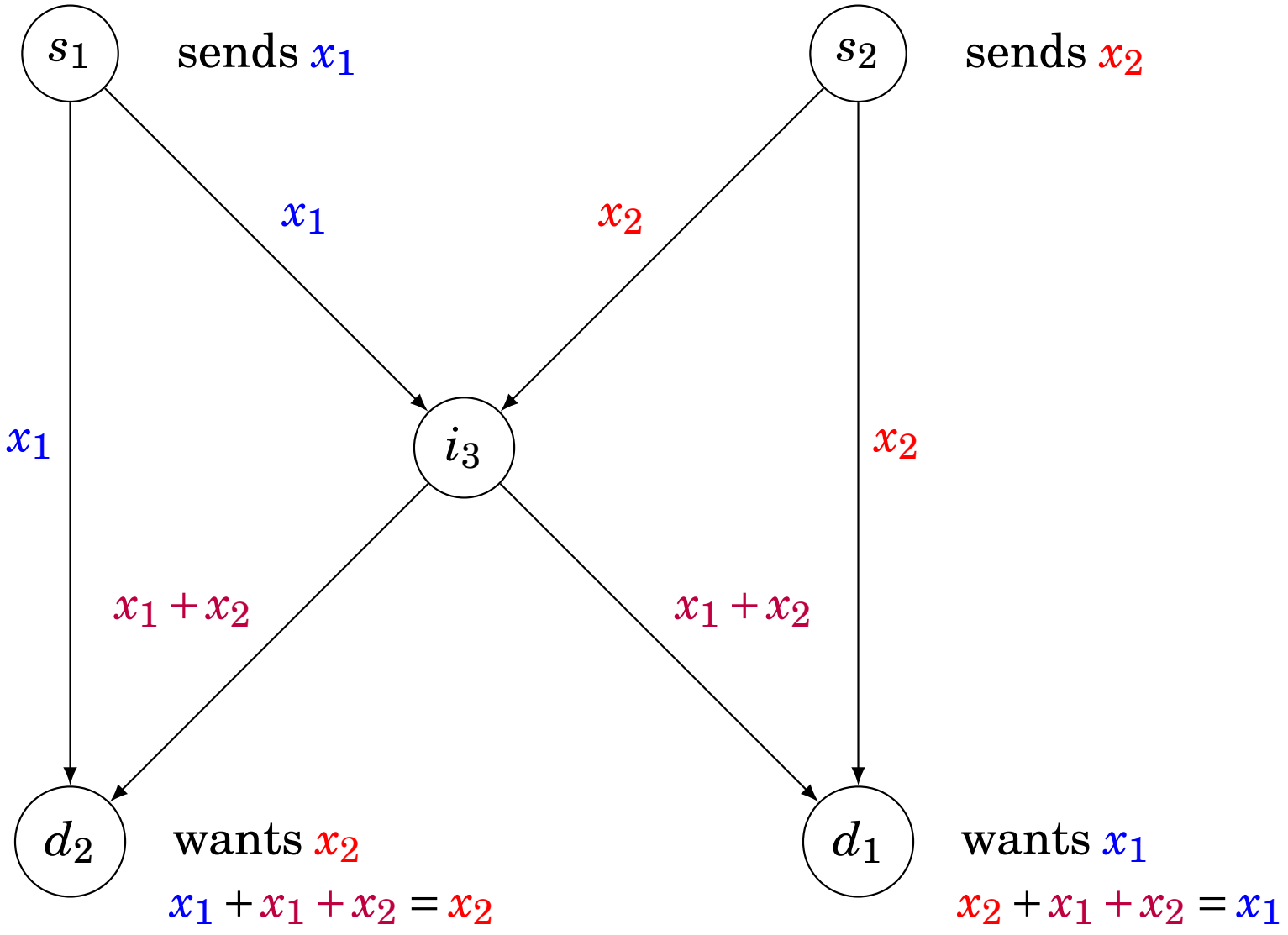

Figure: The butterfly: Network Coding

Random linear network coding

- Fixed linear combinations: too complex, too rigid
  $\Rightarrow$ Solution: choose the linear combinations randomly
- Success probability tends to 1 with field size (Kötter and Médard 02)
- Header to make the rows linearly independent and record combinations ('lifting')

$$a,b,c,d \rightarrow \left(\begin{array}{cccc|c} 1 & 0 & 0 & 0 & a \\ 0 & 1 & 0 & 0 & b \\ 0 & 0 & 1 & 0 & c \\ 0 & 0 & 0 & 1 & d \end{array}\right) = (\mathbf{I}_4|\mathbf{M})$$

- Easy decoding: we receive $(\mathbf{L}|\mathbf{LM})$, we compute $\mathbf{L}^{-1}(\mathbf{LM})$

Error control for RLNC

- RLNC is sensitive to errors for two main reasons.
- 1. There are different types of errors:
  - Faulty links
  - Insufficient field size
  - Faulty or malevolent nodes
  - Adversary on the network, etc.
- 2. Error propagation: a packet in error can corrupt all packets after linear combination.
- Hamming metric codes are unadapted to a noncoherent approach.

Operator channel (Kötter and Kschischang 08)

In RLNC without errors, the input is $(\mathbf{I}_k|\mathbf{M})$ and the output is $\mathbf{L}(\mathbf{I}_k|\mathbf{M})$. So RLNC preserves the row space of the transmitted matrix.

Then RLNC is modelled as the transmission of a linear subspace: send U, receive V with $d_S(U,V) \leq t$ where d_S is the subspace distance:

$$d_S(U,V) = 2\dim(U+V) - \dim(U) - \dim(V).$$

Constant-dimension code (CDC): set of linear subspaces of $\mathrm{GF}(q)^n$ with equal dimension k (Delsarte 76, Schwartz and Etzion 02, K. and K. 08).

Liftings of rank metric codes

- Lifting. $\mathbf{M} \in \mathrm{GF}(q)^{k\times(n-k)}$, $I(\mathbf{M})$: row space of $(\mathbf{I}_k|\mathbf{M})$
- Lifting preserves distance

$$d_{\mathrm{S}}(I(\mathbf{M}),I(\mathbf{N})) = 2d_{\mathrm{R}}(\mathbf{M},\mathbf{N})$$

$\Rightarrow I(\mathscr{C})$ CDC, $d_{\mathrm{S}}(I(\mathscr{C})) = 2d_{\mathrm{R}}(\mathscr{C})$

- Error control for RLNC with liftings is a rank metric problem. Lifting of a Gabidulin code is then a nearly optimal CDC with an efficient decoding algorithm.

Further work in that area

Some extensions:

- further work on subspace codes: bounds and constructions see (Terra Bastos et al. 18) and (Heinlein 19)
- other correspondence using constant-rank codes (Gadouleau and Yan 09)

But in the end, Network Coding is unlikely to be deployed in large scale...

Current/Future work on rank metric codes

For cryptography: come up with more classes of codes, that can be efficient while possible to hide.

List decoding of Gabidulin or more general classes of rank metric codes.

Combinatorics/geometry of the rank metric:

- work on covering codes,
- formula for intersection of spheres in (Claridge 16),
- we should dive deeper in the "q-combinatorics of finite sets:" Erdős-Ko-Rado, LYM inequality, Steiner theorem etc.

Look into related association schemes, e.g. that of alternating or Hermitian linear forms.