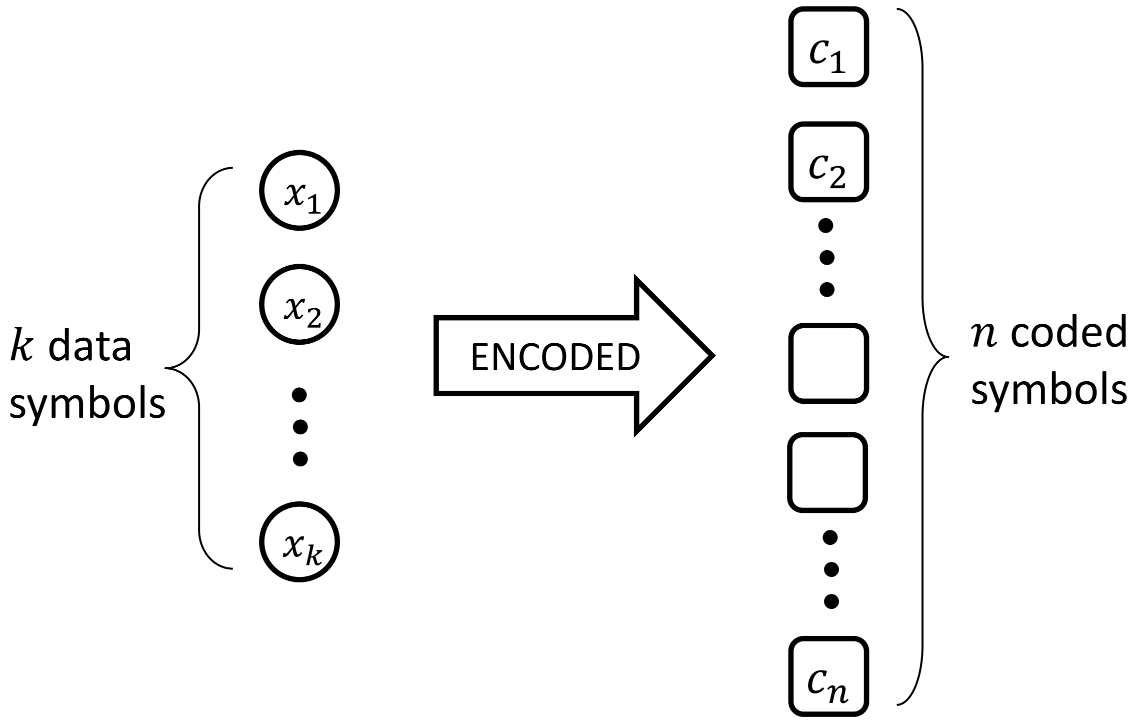
Coding with Constraints: Different Flavors

Hoang Dau
University of Illinois at Urbana-Champaign
Email: hoangdau@uiuc.edu

DIMACS Workshop on Network Coding: the Next 15 Years
Rutgers University, NJ, 2015

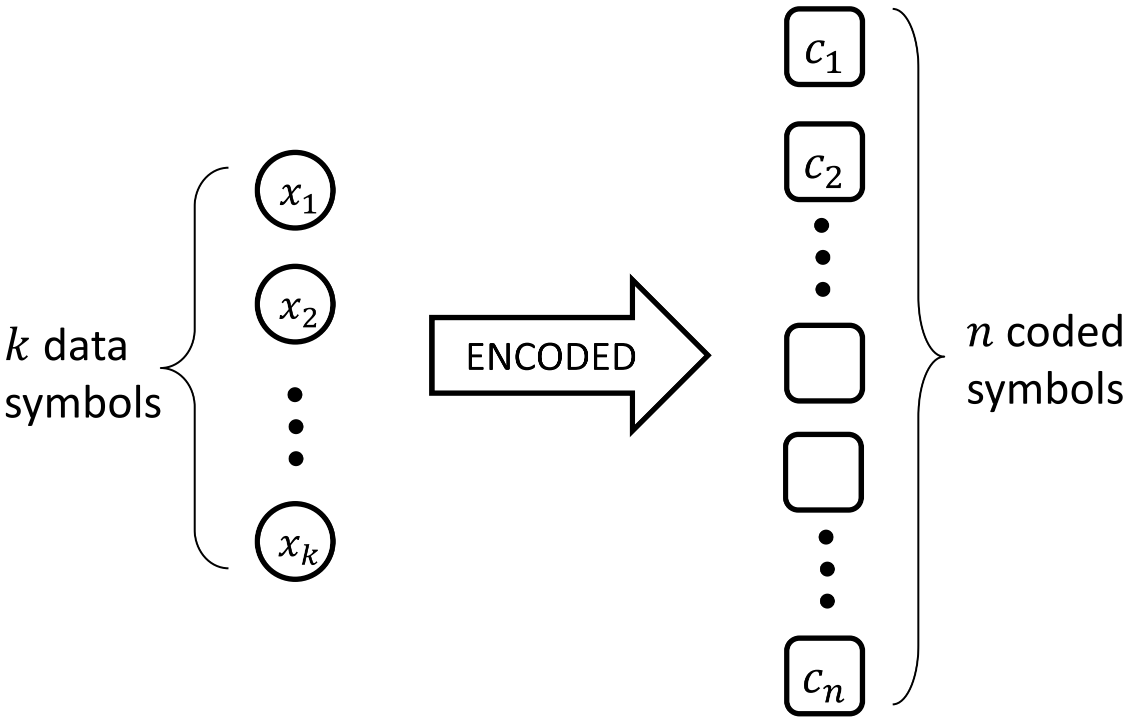
Part I
Coding with Constraints: A Quick Survey

Coding with Constraints: Definition



CONVENTIONAL CODE
$c_j = c_j(x_1, x_2, \ldots, x_k)$

Coding with Constraints: Definition



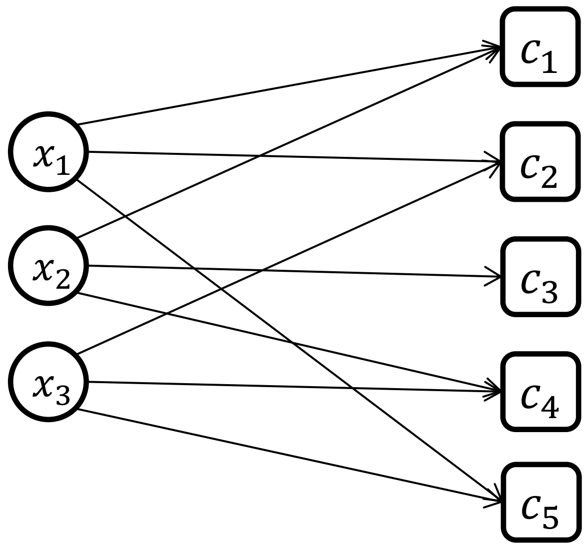
CONVENTIONAL CODE

$c_j = c_j(x_1, x_2, \ldots, x_k)$

CODE WITH CONSTRAINTS

$c_j = c_j(\{x_i : i \in C_j\}),\quad C_j \subseteq \{1,2,\ldots,k\}$

Coding with Constraints: Example



$$= c_1(x_1, x_2)$$

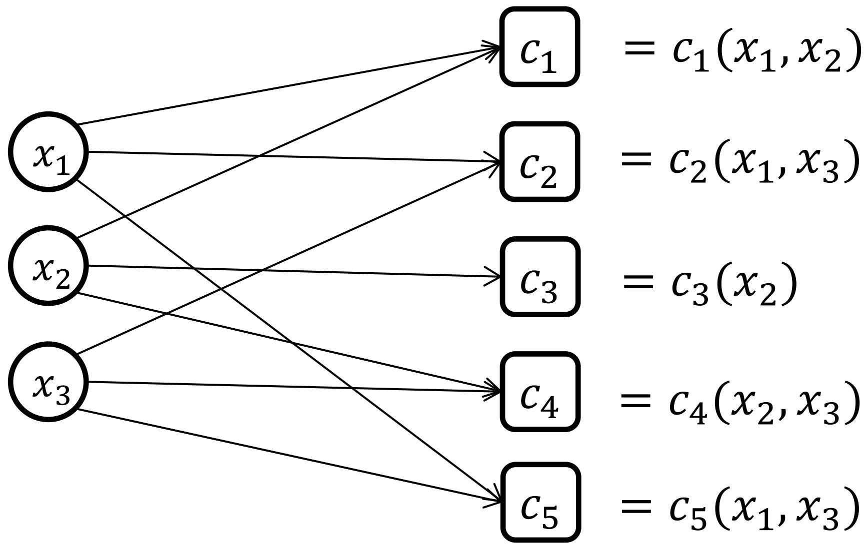
$$= c_2(x_1, x_3)$$

$$= c_3(x_2)$$

$$= c_4(x_2, x_3)$$

$$= c_5(x_1, x_3)$$

Coding with Constraints: Example



Linear code: $(c_1, c_2, c_3, c_4, c_5) = (x_1, x_2, x_3)\boldsymbol{G}$
where the generator matrix $\boldsymbol{G}$ is

$$\boldsymbol{G} = \begin{pmatrix} ? & ? & 0 & 0 & ? \\ ? & 0 & ? & ? & 0 \\ 0 & ? & 0 & ? & ? \end{pmatrix}$$

Coding with Constraints: Main Problem

Given the constraints

$$c_j = c_j(\{x_i : i \in C_j\}), \quad C_j \subseteq \{1,2,\dots,k\}$$

how to construct codes that

- achieve the optimal minimum distance
- over small field size $q \approx \mathrm{poly}(n)$

Coding with Constraints: Main Problem

Given the constraints

$$c_j = c_j(\{x_i : i \in C_j\}), \quad C_j \subseteq \{1,2,\ldots,k\}$$

how to construct codes that

- achieve the optimal minimum distance
- over small field size $q \approx \mathrm{poly}(n)$

Linear case: given

$$\boldsymbol{G} = \begin{pmatrix} ? & ? & 0 & 0 & ? \\ ? & 0 & ? & ? & 0 \\ 0 & ? & 0 & ? & ? \end{pmatrix}$$

how to replace "?"-entries by elements of F_q ($q \approx \mathrm{poly}(n)$) so that $\boldsymbol{G}$ generate a code with optimal distance

Coding with Constraints: Upper Bound

Upper Bound (Halbawi-Thill-Hassibi'15, Song-Dau-Yuen'15)

$$d \le d_{max} = 1 + \min_{\emptyset \neq I \subseteq \{1,\ldots,k\}} \left(\left| \cup_{i \in I} R_i \right| - |I| \right)$$

where $R_i = \{j : i \in C_j\}$

Coding with Constraints: Upper Bound

Upper Bound (Halbawi-Thill-Hassibi'15, Song-Dau-Yuen'15)

$$d \leq d_{max} = 1 + \min_{\emptyset \neq I \subseteq \{1,\ldots,k\}} \left(\left| \cup_{i \in I} R_i \right| - |I| \right)$$

where $R_i = \{j : i \in C_j\}$

Properties

- d_{max} can be found in time $\text{poly}(n)$
- codes with $d = d_{max}$ always exists over fields of size $\approx \binom{n}{d-1}$

Coding with Constraints: Upper Bound

Upper Bound (Halbawi-Thill-Hassibi'15, Song-Dau-Yuen'15)

$$d \le d_{max} = 1 + \min_{\emptyset \neq I \subseteq \{1,\ldots,k\}} \left(\left| \cup_{i \in I} R_i \right| - |I| \right)$$

where $R_i = \{j : i \in C_j\}$

Properties

- d_{max} can be found in time $\mathrm{poly}(n)$
- codes with $d = d_{max}$ always exists over fields of size $\approx \binom{n}{d-1}$

Question of interest: how about fields of size $\mathrm{poly}(n)$?

Coding with Constraints: Review

(Small field)

MDS Case: $d_{max} = n - k + 1$

- Optimal codes exist in a few special cases (Halbawi-Ho-Yao-Duursma*'14,* Dau-Song-Yuen*'14*, Yan-Sprintson-Zelenko'*14)*

Coding with Constraints: Review

(Small field)

MDS Case: $d_{max} = n - k + 1$

- Optimal codes exist in a few special cases (Halbawi-Ho-Yao-Duursma*'14,* Dau-Song-Yuen*'14*, Yan-Sprintson-Zelenko'*14)*
  - $k \leq 4$ (every n)

Coding with Constraints: Review

(Small field)

MDS Case: $d_{max} = n - k + 1$

- Optimal codes exist in a few special cases (Halbawi-Ho-Yao-Duursma*'14,* Dau-Song-Yuen*'14*, Yan-Sprintson-Zelenko'*14)*
  - $k \leq 4$ (every n)
  - rows of $\boldsymbol{G}$ partitioned into ≤ 3 groups, each has same "?"-pattern

Coding with Constraints: Review

(Small field)

MDS Case: $d_{max} = n - k + 1$

- Optimal codes exist in a few special cases (Halbawi-Ho-Yao-Duursma*'14,* Dau-Song-Yuen*'14*, Yan-Sprintson-Zelenko'*14)*
  - $k \leq 4$ (every n)
  - rows of $\boldsymbol{G}$ partitioned into ≤ 3 groups, each has same "?"-pattern
  - rows have $k - 1$ zeros & 2 different rows share ≤ 1 common zeros

Coding with Constraints: Review

(Small field)

MDS Case: $d_{max} = n - k + 1$

- Optimal codes exist in a few special cases (Halbawi-Ho-Yao-Duursma*'14,* Dau-Song-Yuen*'14*, Yan-Sprintson-Zelenko'*14)*
  - $k \leq 4$ (every n)
  - rows of $\boldsymbol{G}$ partitioned into ≤ 3 groups, each has same "?"-pattern
  - rows have $k - 1$ zeros & 2 different rows share ≤ 1 common zeros

General Case (Halbawi-Thill-Hassibi'15): $d_{max} \leq n - k + 1$

Coding with Constraints: Review

(Small field)

MDS Case: $d_{max} = n - k + 1$

- Optimal codes exist in a few special cases (Halbawi-Ho-Yao-Duursma*'14,* Dau-Song-Yuen*'14*, Yan-Sprintson-Zelenko'*14)*
  - $k \leq 4$ (every n)
  - rows of $\boldsymbol{G}$ partitioned into ≤ 3 groups, each has same "?"-pattern
  - rows have $k-1$ zeros & 2 different rows share ≤ 1 common zeros

General Case (Halbawi-Thill-Hassibi'15): $d_{max} \leq n - k + 1$

- Optimal codes exist if there are $\geq d_{max} - 1$ indices j's where $|C_j| = k$

Coding with Constraints: Review

(Small field)

MDS Case: $d_{max} = n - k + 1$

- Optimal codes exist in a few special cases (Halbawi-Ho-Yao-Duursma*'14,* Dau-Song-Yuen*'14*, Yan-Sprintson-Zelenko'*14)*
  - $k \leq 4$ (every n)
  - rows of $\boldsymbol{G}$ partitioned into ≤ 3 groups, each has same "?"-pattern
  - rows have $k - 1$ zeros & 2 different rows share ≤ 1 common zeros

General Case (Halbawi-Thill-Hassibi'15): $d_{max} \leq n - k + 1$

- Optimal codes exist if there are $\geq d_{max} - 1$ indices j's where $|C_j| = k$
- Optimal systematic codes always exists (smaller bound $d_{sys} \leq d_{max}$)

Coding with Constraints: Review

(Small field)

MDS Case: $d_{max} = n - k + 1$

- Optimal codes exist in a few special cases (Halbawi-Ho-Yao-Duursma*'14,* Dau-Song-Yuen*'14*, Yan-Sprintson-Zelenko'*14)*
  - $k \leq 4$ (every n)
  - rows of $\boldsymbol{G}$ partitioned into ≤ 3 groups, each has same "?"-pattern
  - rows have $k - 1$ zeros & 2 different rows share ≤ 1 common zeros

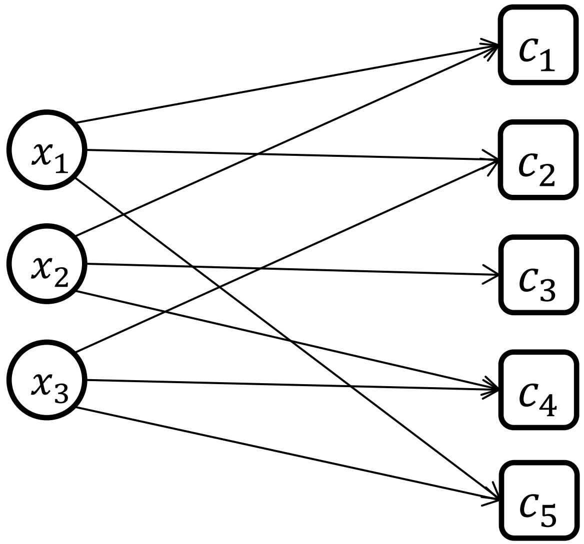
General Case (Halbawi-Thill-Hassibi'15): $d_{max} \leq n - k + 1$

- Optimal codes exist if there are $\geq d_{max} - 1$ indices j's where $|C_j| = k$
- Optimal systematic codes always exists (smaller bound $d_{sys} \leq d_{max}$)

Common Technique: Reed-Solomon (sub-) code

Coding with Constraints: Review

Common Technique: Reed-Solomon (sub-) code



$= c_1(x_1, x_2)$

$= c_2(x_1, x_3)$

$= c_3(x_2)$

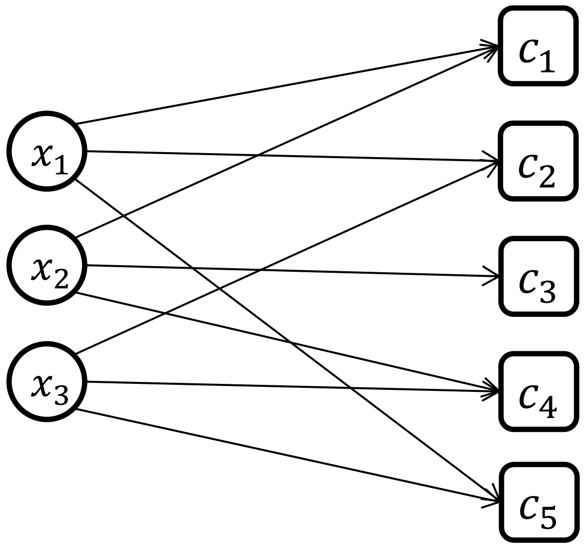
$= c_4(x_2, x_3)$

$= c_5(x_1, x_3)$

$$\boldsymbol{G} = \begin{pmatrix} ? & ? & 0 & 0 & ? \\ ? & 0 & ? & ? & 0 \\ 0 & ? & 0 & ? & ? \end{pmatrix}$$

Coding with Constraints: Review

Common Technique: Reed-Solomon (sub-) code



$= c_1(x_1, x_2)$

$= c_2(x_1, x_3)$

$= c_3(x_2)$

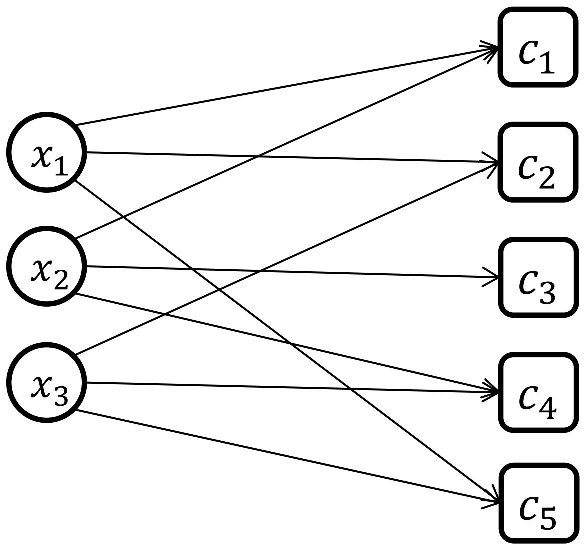
$= c_4(x_2, x_3)$

$= c_5(x_1, x_3)$

$$\begin{array}{ccccccc} & & \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 & \alpha_5 \\ \boldsymbol{G} & = & \multicolumn{5}{c}{\begin{pmatrix} ? & ? & 0 & 0 & ? \\ ? & 0 & ? & ? & 0 \\ 0 & ? & 0 & ? & ? \end{pmatrix}} \end{array}$$

Coding with Constraints: Review

Common Technique: Reed-Solomon (sub-) code



$= c_1(x_1, x_2)$

$= c_2(x_1, x_3)$

$= c_3(x_2)$

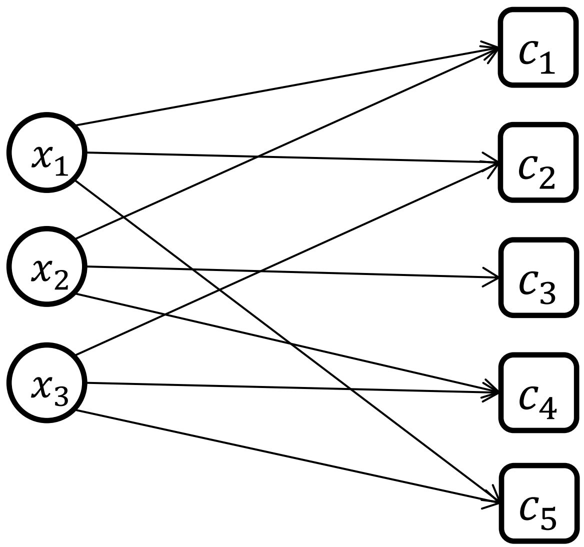
$= c_4(x_2, x_3)$

$= c_5(x_1, x_3)$

$$\boldsymbol{G} = \begin{array}{c} \alpha_1 \;\; \alpha_2 \;\; \alpha_3 \;\; \alpha_4 \;\; \alpha_5 \\ \begin{pmatrix} ? & ? & 0 & 0 & ? \\ ? & 0 & ? & ? & 0 \\ 0 & ? & 0 & ? & ? \end{pmatrix} \end{array} = \begin{pmatrix} f_1(\alpha_1) & f_1(\alpha_2) & 0 & 0 & f_1(\alpha_5) \\ f_2(\alpha_1) & 0 & f_2(\alpha_3) & f_2(\alpha_4) & 0 \\ 0 & f_3(\alpha_2) & 0 & f_3(\alpha_4) & f_3(\alpha_5) \end{pmatrix}$$

Coding with Constraints: Review

Common Technique: Reed-Solomon (sub-) code



$= c_1(x_1, x_2)$

$= c_2(x_1, x_3)$

$= c_3(x_2)$

$= c_4(x_2, x_3)$

$= c_5(x_1, x_3)$

$$\begin{matrix} \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 & \alpha_5 \end{matrix}$$

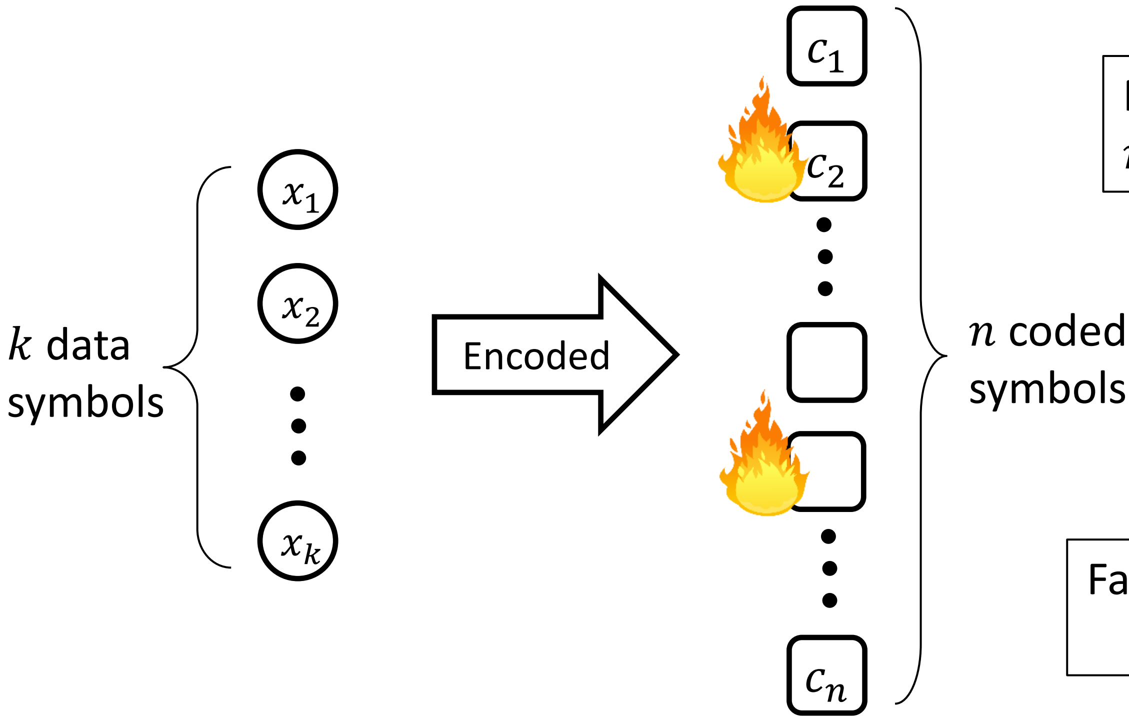
$$\boldsymbol{G} = \begin{pmatrix} ? & ? & 0 & 0 & ? \\ ? & 0 & ? & ? & 0 \\ 0 & ? & 0 & ? & ? \end{pmatrix} = \begin{pmatrix} f_1(\alpha_1) & f_1(\alpha_2) & 0 & 0 & f_1(\alpha_5) \\ f_2(\alpha_1) & 0 & f_2(\alpha_3) & f_2(\alpha_4) & 0 \\ 0 & f_3(\alpha_2) & 0 & f_3(\alpha_4) & f_3(\alpha_5) \end{pmatrix}$$

Difficulty: G may not be full rank

Part II:
Joint Design of Different MDS Codes

(joint work with H. Kiah, W. Song, and C. Yuen)

MDS Codes for Distributed Storage



MDS: tolerate any $n - k$ node failures

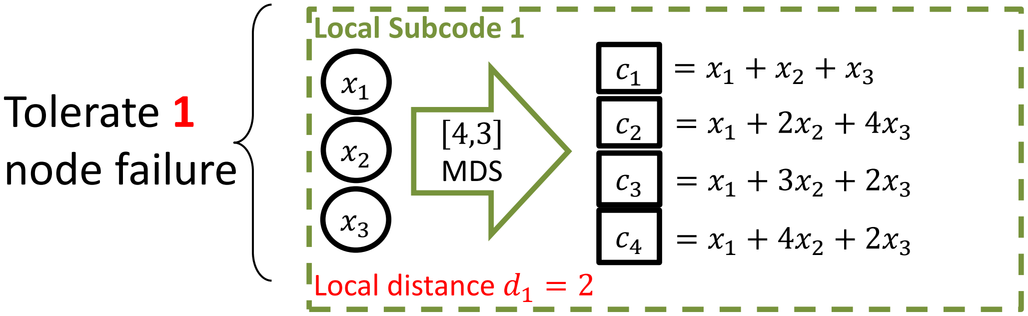
Facebook: Reed-Solomon $n = 14, k = 10$

Question of Interest

- If two (or more) independent DSS share some common data, can we jointly design the corresponding MDS codes to get a better overall failure protection?

Question of Interest

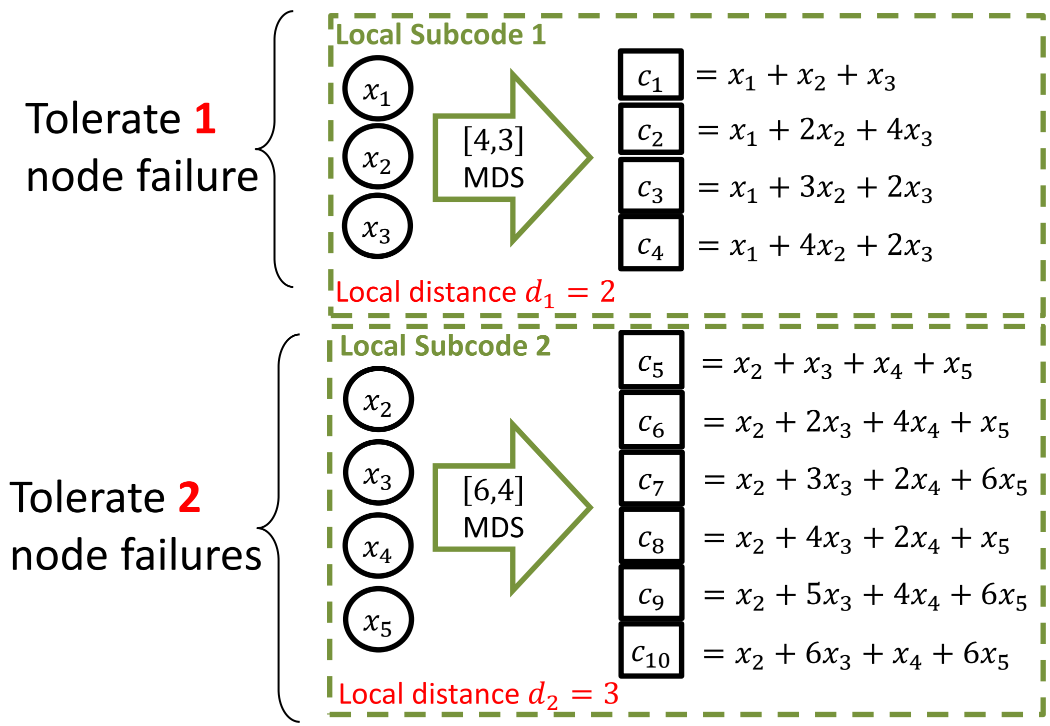
- If two (or more) independent DSS share some common data, can we jointly design the corresponding MDS codes to get a better overall failure protection?



Code has minimum distance d: tolerate $d - 1$ node failures

Question of Interest

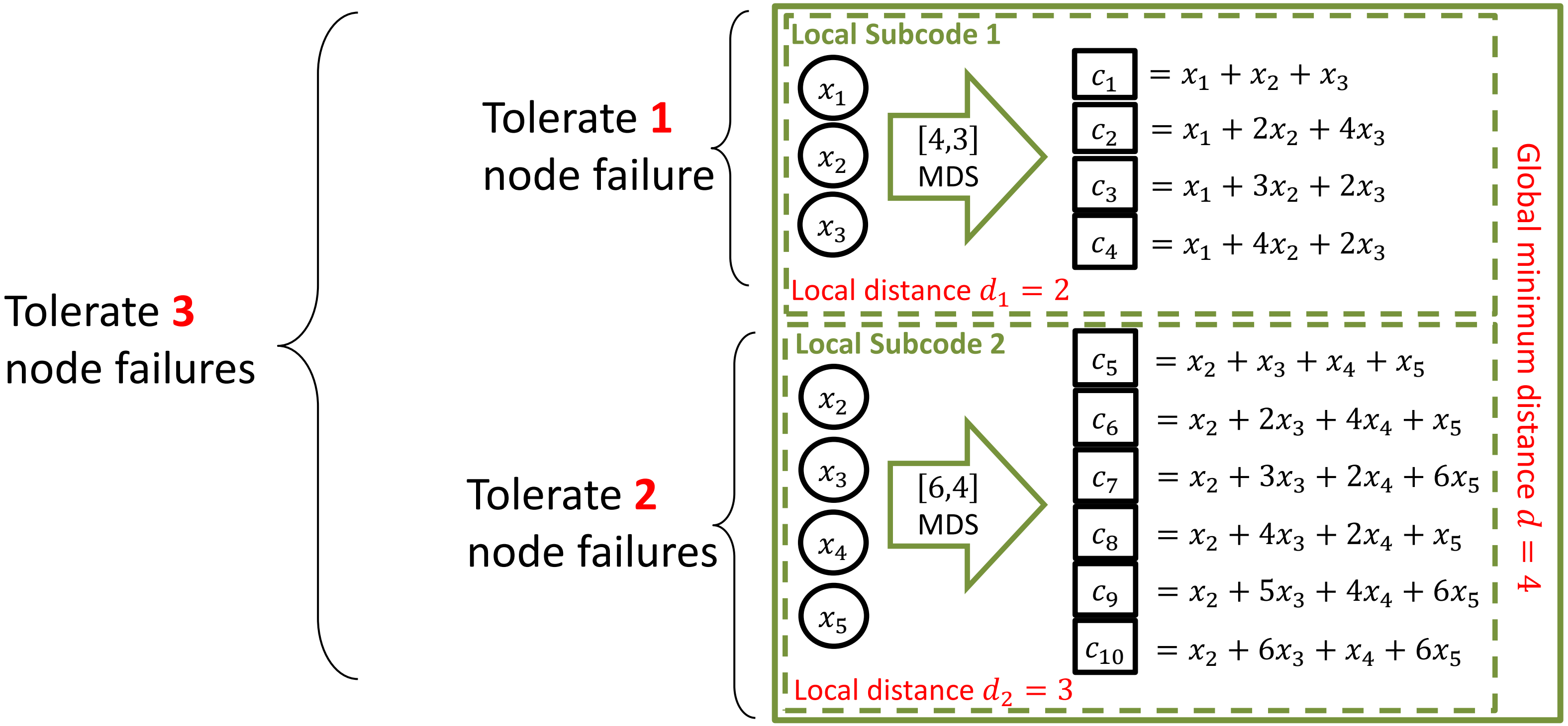
- If two (or more) independent DSS share some common data, can we jointly design the corresponding MDS codes to get a better overall failure protection?



Code has minimum distance d: tolerate $d - 1$ node failures

Question of Interest

- If two (or more) independent DSS share some common data, can we jointly design the corresponding MDS codes to get a better overall failure protection?



Code has minimum distance d: tolerate $d - 1$ node failures

Upper Bound for Global Minimum Distance

For linear code

$$(c_1, c_2, \ldots, c_{10}) = (x_1, x_2, \ldots, x_6)\boldsymbol{G}$$

where

$$\boldsymbol{G} = \left[\begin{array}{cccc|cccccc} ? & ? & ? & ? & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline ? & ? & ? & ? & ? & ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? & ? & ? & ? & ? & ? \\ \hline 0 & 0 & 0 & 0 & ? & ? & ? & ? & ? & ? \\ 0 & 0 & 0 & 0 & ? & ? & ? & ? & ? & ? \end{array}\right]$$

Upper Bound for Global Minimum Distance

For linear code

$$(c_1, c_2, \dots, c_{10}) = (x_1, x_2, \dots, x_6)\boldsymbol{G}$$

where

$$\boldsymbol{G} = \left[\begin{array}{cccc|cccccc} ? & ? & ? & ? & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline ? & ? & ? & ? & ? & ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? & ? & ? & ? & ? & ? \\ \hline 0 & 0 & 0 & 0 & ? & ? & ? & ? & ? & ? \\ 0 & 0 & 0 & 0 & ? & ? & ? & ? & ? & ? \end{array}\right]$$

Goal: replace "?"-entries with F_q-elements so that

Upper Bound for Global Minimum Distance

For linear code

$$(c_1, c_2, \ldots, c_{10}) = (x_1, x_2, \ldots, x_6)\boldsymbol{G}$$

where

[4,3]-MDS

$$\boldsymbol{G} = \left[\begin{array}{cccc|cccccc} ? & ? & ? & ? & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline ? & ? & ? & ? & ? & ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? & ? & ? & ? & ? & ? \\ \hline 0 & 0 & 0 & 0 & ? & ? & ? & ? & ? & ? \\ 0 & 0 & 0 & 0 & ? & ? & ? & ? & ? & ? \end{array}\right]$$

[6,4]-MDS

Goal: replace "?"-entries with F_q-elements so that

- two local subcodes are MDS (additional requirement)

Upper Bound for Global Minimum Distance

For linear code

$$(c_1, c_2, \ldots, c_{10}) = (x_1, x_2, \ldots, x_6)\boldsymbol{G}$$

where

[4,3]-MDS

$$\boldsymbol{G} = \left[\begin{array}{cccc|cccccc} ? & ? & ? & ? & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline ? & ? & ? & ? & ? & ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? & ? & ? & ? & ? & ? \\ \hline 0 & 0 & 0 & 0 & ? & ? & ? & ? & ? & ? \\ 0 & 0 & 0 & 0 & ? & ? & ? & ? & ? & ? \end{array}\right]$$

[6,4]-MDS

Goal: replace "?"-entries with F_q-elements so that

- two local subcodes are MDS (additional requirement)
- the global code has optimal distance (same as coding with constraints)

Upper Bound for Global Minimum Distance

For linear code

$$(c_1, c_2, \ldots, c_{10}) = (x_1, x_2, \ldots, x_6)\boldsymbol{G}$$

where

[4,3]-MDS

$$\boldsymbol{G} = \left[\begin{array}{cccc|cccccc} ? & ? & ? & ? & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline ? & ? & ? & ? & ? & ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? & ? & ? & ? & ? & ? \\ \hline 0 & 0 & 0 & 0 & ? & ? & ? & ? & ? & ? \\ 0 & 0 & 0 & 0 & ? & ? & ? & ? & ? & ? \end{array}\right]$$

[6,4]-MDS

Goal: replace "?"-entries with F_q-elements so that

- two local subcodes are MDS (additional requirement)
- the global code has optimal distance (same as coding with constraints)

Same cut-set bound apply & codes over large fields achieve this bound

$$d \le d_{max} = 1 + \min_{\emptyset \ne I \subseteq \{1,\ldots,k\}} \left(\left| \cup_{i \in I} R_i \right| - |I| \right)$$

Upper Bound for Global Minimum Distance: Two Local Subcodes

- $d \leq d_{max} = 1 + t + \min\{n_1 - k_1, n_2 - k_2\}$
- t = #{common x_i}

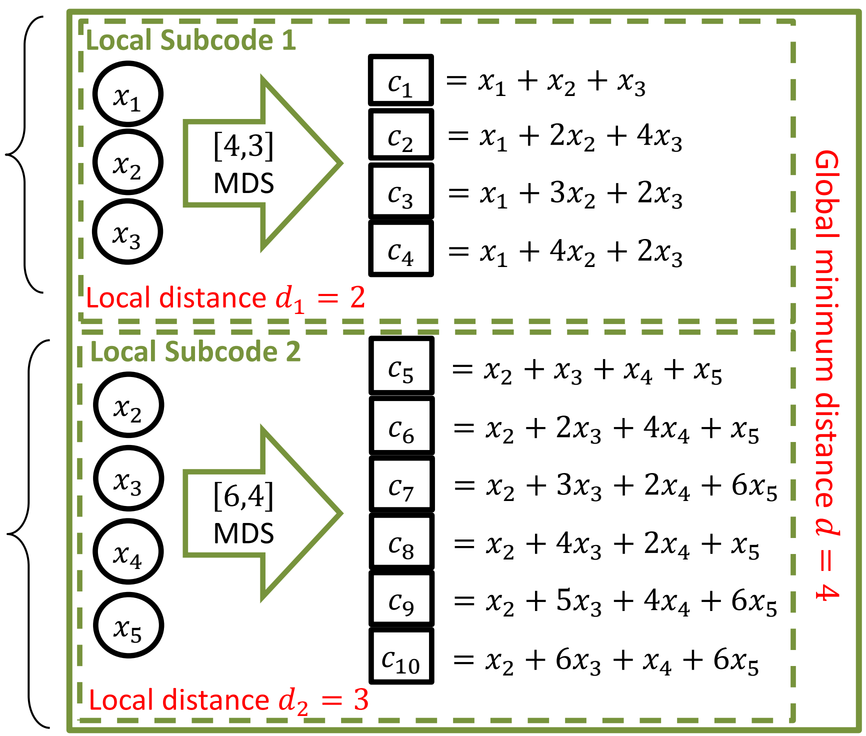
Upper Bound for Global Minimum Distance: Two Local Subcodes

- $d \le d_{max} = 1 + t + \min\{n_1 - k_1, n_2 - k_2\}$
- t = #{common x_i}

$n_1 = 4,\ k_1 = 3$

$t = |\{x_2, x_3\}| = 2$

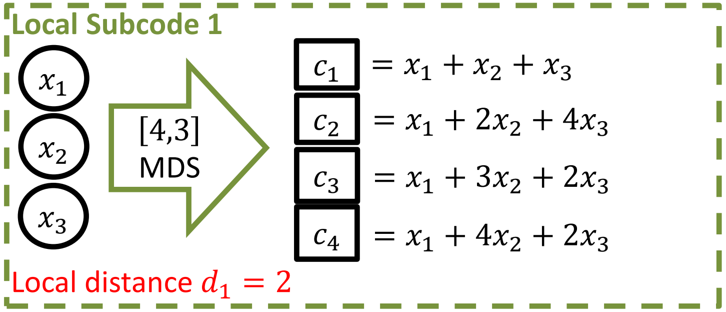
$n_2 = 6,\ k_2 = 4$



In this example: $d \le 1 + 2 + \min\{4 - 3, 6 - 4\} = 4$ -> optimal code here

Generator Matrix Representation

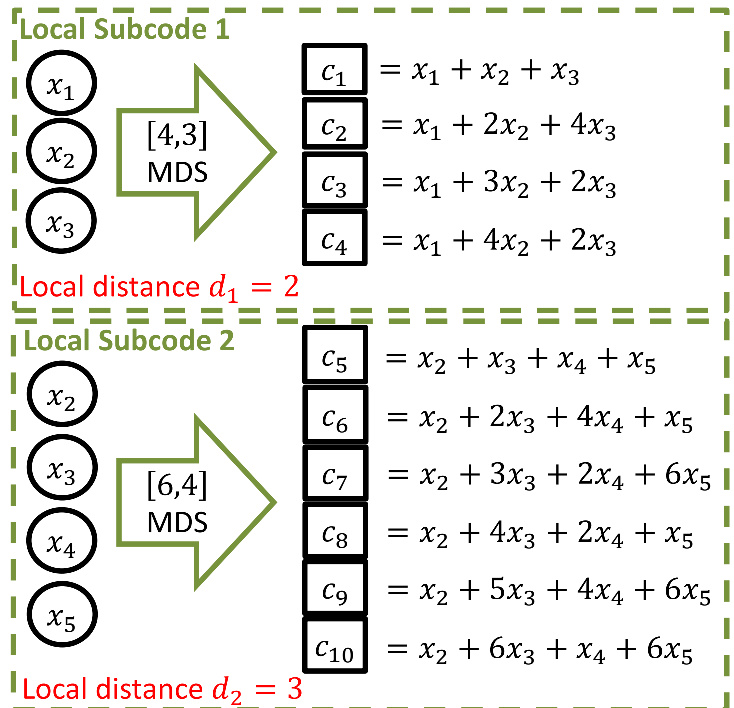
1st code: $\boldsymbol{G_1} = \left[\begin{array}{cccc} 1 & 1 & 1 & 1 \\ \hline 1 & 2 & 3 & 4 \\ 1 & 4 & 2 & 2 \end{array}\right] = \left[\frac{\boldsymbol{U}}{\boldsymbol{A}}\right]$



Generator Matrix Representation

1st code: $\boldsymbol{G_1} = \left[\begin{array}{cccc} 1 & 1 & 1 & 1 \\ \hline 1 & 2 & 3 & 4 \\ 1 & 4 & 2 & 3 \end{array}\right] = \left[\begin{array}{c} \boldsymbol{U} \\ \hline \boldsymbol{A} \end{array}\right]$

2nd code: $\boldsymbol{G_2} = \left[\begin{array}{cccccc} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 4 & 2 & 2 & 4 & 1 \\ 1 & 1 & 6 & 1 & 6 & 6 \end{array}\right] = \left[\begin{array}{c} \boldsymbol{B} \\ \hline \boldsymbol{V} \end{array}\right]$

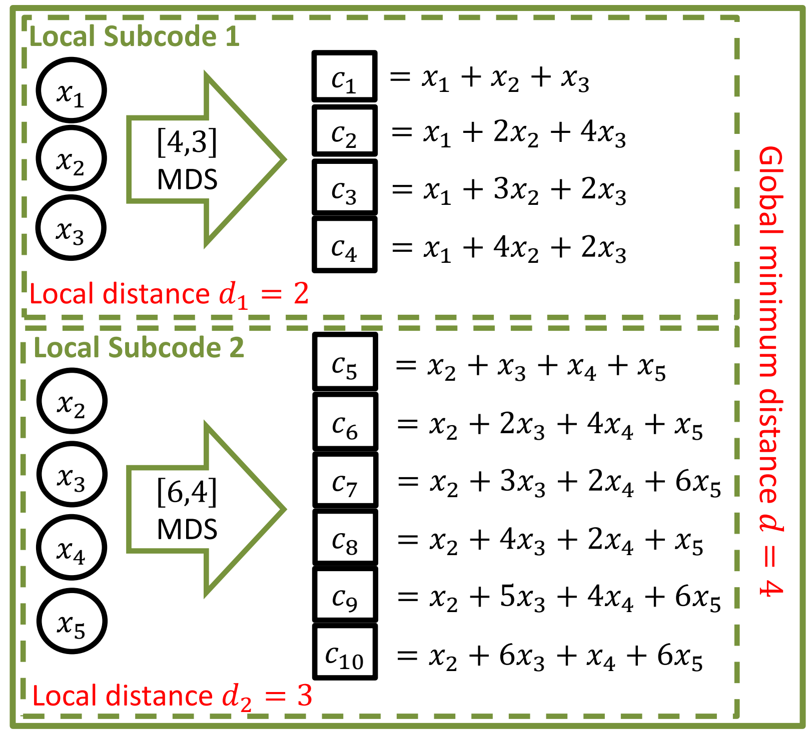


Generator Matrix Representation

1st code: $\boldsymbol{G_1} = \left[\begin{array}{cccc} 1 & 1 & 1 & 1 \\ \hline 1 & 2 & 3 & 4 \\ 1 & 4 & 2 & 3 \end{array}\right] = \left[\begin{array}{c} \boldsymbol{U} \\ \hline \boldsymbol{A} \end{array}\right]$

2nd code: $\boldsymbol{G_2} = \left[\begin{array}{cccccc} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 4 & 2 & 2 & 4 & 1 \\ 1 & 1 & 6 & 1 & 6 & 6 \end{array}\right] = \left[\begin{array}{c} \boldsymbol{B} \\ \hline \boldsymbol{V} \end{array}\right]$

Global code: $\boldsymbol{G} = \left[\begin{array}{c|c} \boldsymbol{U} & \boldsymbol{O} \\ \hline \boldsymbol{A} & \boldsymbol{B} \\ \hline \boldsymbol{O} & \boldsymbol{V} \end{array}\right]$



Easy Case: Two Codes Have Few Common Data

- If few common data, i.e.

$$t \leq 1 + \max\{n_2 - k_2, n_1 - k_1\}$$

using two Vandermonde matrices as G_1, G_2 : optimal minimum distance

- Finite field size required: $|F_q| \geq \max\{n_1, n_2\}$

Easy Case: Two Codes Have Few Common Data

- If few common data, i.e.

$$t \leq 1 + \max\{n_2 - k_2, n_1 - k_1\}$$

using two Vandermonde matrices as G_1, G_2 : optimal minimum distance

- Finite field size required: $|F_q| \geq \max\{n_1, n_2\}$

More specifically, in this case, if

- $\boldsymbol{G_1} = \begin{bmatrix} \boldsymbol{U} \\ \boldsymbol{A} \end{bmatrix}$ is a nested MDS code: $\boldsymbol{G_1}, \boldsymbol{U}$ are generator matrices of MDS codes
- $\boldsymbol{G_2} = \begin{bmatrix} \boldsymbol{B} \\ \boldsymbol{V} \end{bmatrix}$ is a nested MDS code: $\boldsymbol{G_2}, \boldsymbol{V}$ are generator matrices of MDS codes

then the global code achieves the optimal minimum distance (attains the upper bound)

Harder Case: Two Codes Have the Same Redundancy

- If same redundancy, i.e.

$$n_1 - k_1 = n_2 - k_2$$

we construct codes that have optimal global minimum distance

- Finite field size required: $q > n = n_1 + n_2$

The construction uses the BCH bound

Two Codes Have the Same Redundancy: BCH Bound

- F_q: finite field of q elements
- ω: primitive element of F_q, i.e. $F_q = \{0,1,\omega,\omega^2,\omega^3,\ldots\}$
- Identify a vector $c = (c_1, \ldots, c_n) \in F_q^n$ with the polynomial

$$c(\mathrm{x}) = c_1 + c_2\mathrm{x} + c_3\mathrm{x}^2 + \cdots + c_n\mathrm{x}^{n-1}$$

BCH Bound: If every coded vector $\boldsymbol{c}$ satisfies

$$c(\omega^i) = 0, \text{ for every } i = 0,1,\ldots,\delta - 1$$

i.e, they all have δ consecutive powers of ω as roots, then the code has minimum distance $d \geq \delta + 1$

Two Codes Have the Same Redundancy: Construction

We construct the generator matrix of the optimal code. Example: F_{13}

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{U} & \boldsymbol{O} \\ \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{O} & \boldsymbol{V} \end{bmatrix} =$$

Two Codes Have the Same Redundancy: Construction

We construct the generator matrix of the optimal code. Example: F_{13}

$$G = \begin{bmatrix} U & O \\ A & B \\ O & V \end{bmatrix} =$$

$n_1 = 5$

$k_1 = 4$

$$\left(\begin{array}{ccccc|cccccc}
5 & 1 & 6 & 1 & & & & & & & \\
 & 5 & 1 & 6 & 1 & & & & & & \\
\hline
 & & 5 & 11 & 10 & & & & & 12 & 1 \\
 & & & 9 & 4 & & & & 7 & 5 & 1 \\
\hline
 & & & & & & & 5 & 1 & 6 & 1 \\
 & & & & & & 5 & 1 & 6 & 1 & \\
 & & & & & 5 & 1 & 6 & 1 & &
\end{array}\right)$$

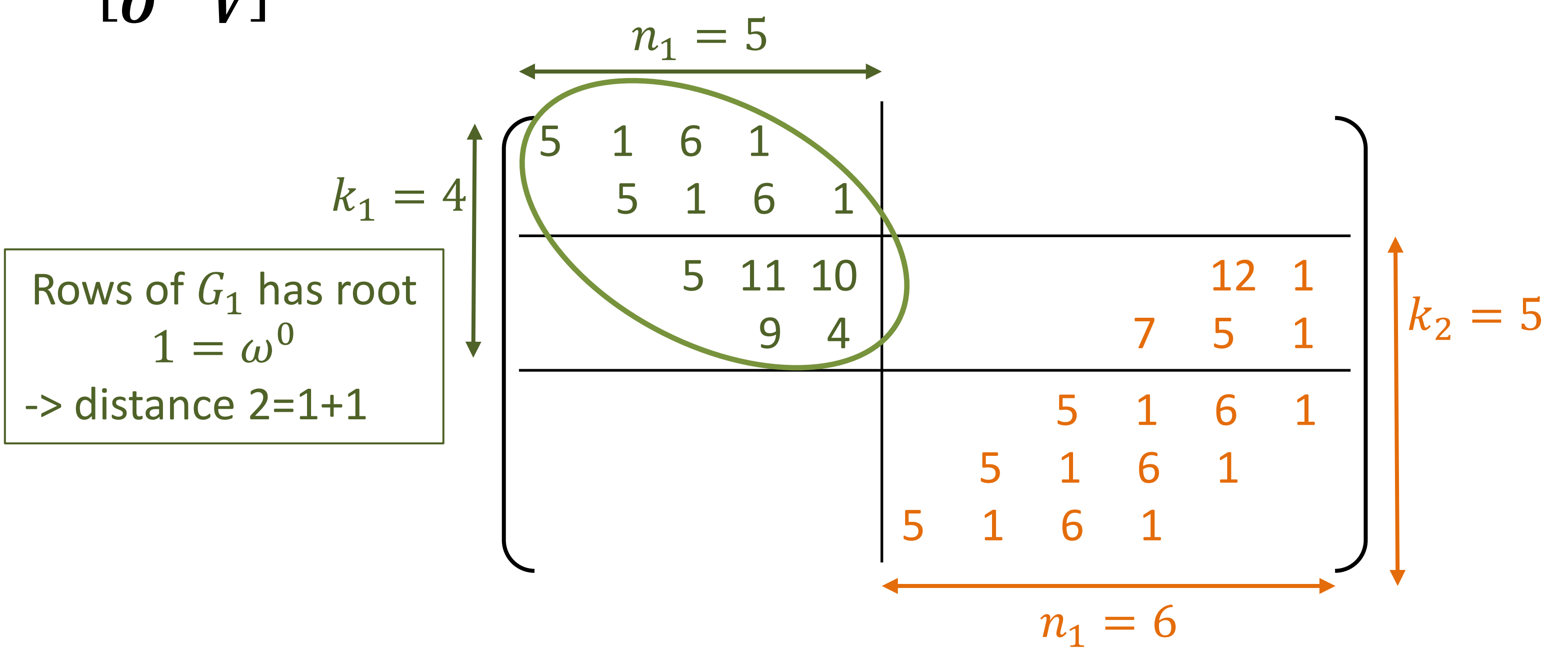
$k_2 = 5$

$n_1 = 6$

Two Codes Have the Same Redundancy: Construction

We construct the generator matrix of the optimal code. Example: F_{13}

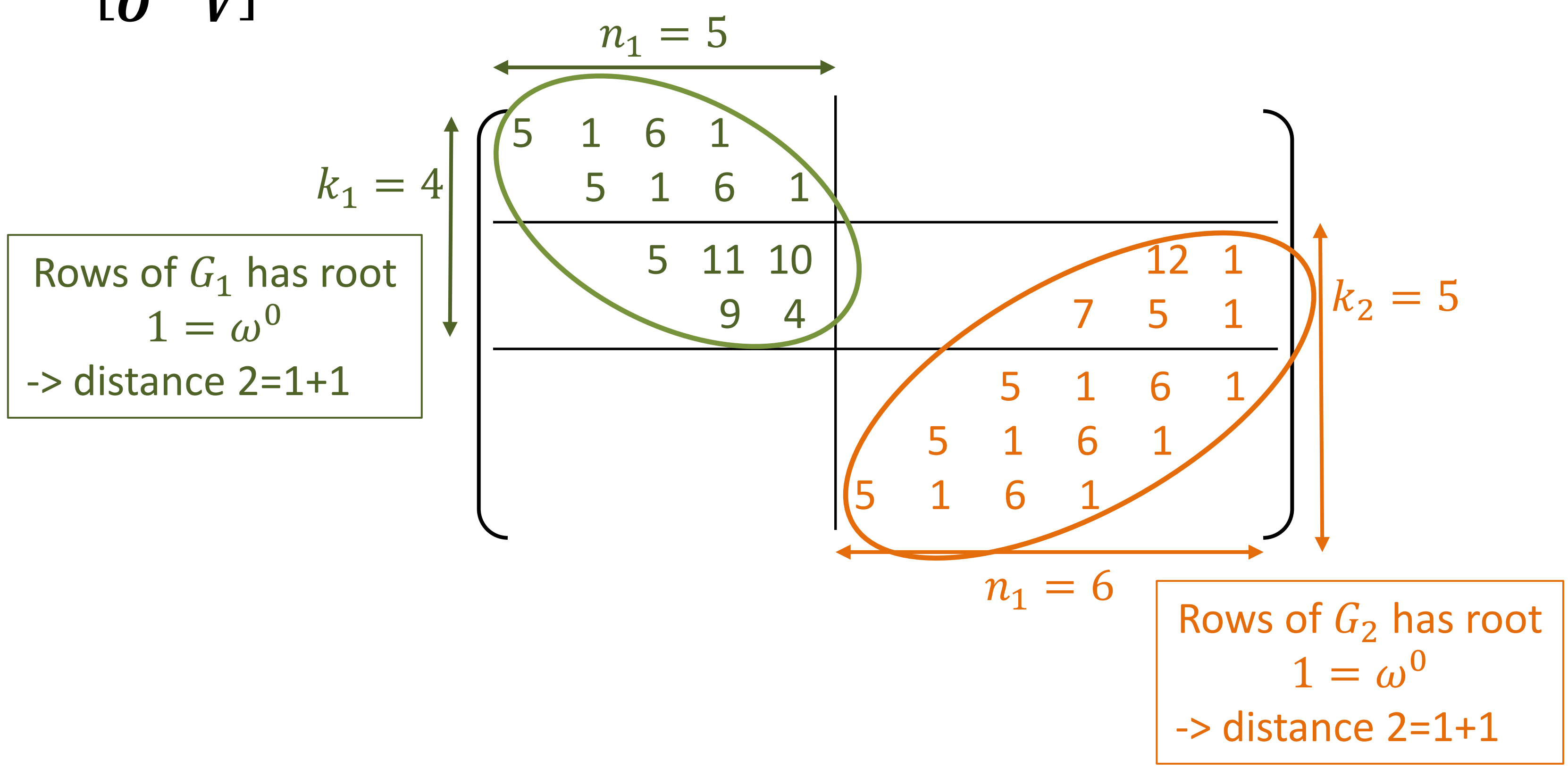
$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{U} & \boldsymbol{O} \\ \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{O} & \boldsymbol{V} \end{bmatrix} =$$



Two Codes Have the Same Redundancy: Construction

We construct the generator matrix of the optimal code. Example: F_{13}

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{U} & \boldsymbol{O} \\ \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{O} & \boldsymbol{V} \end{bmatrix} =$$

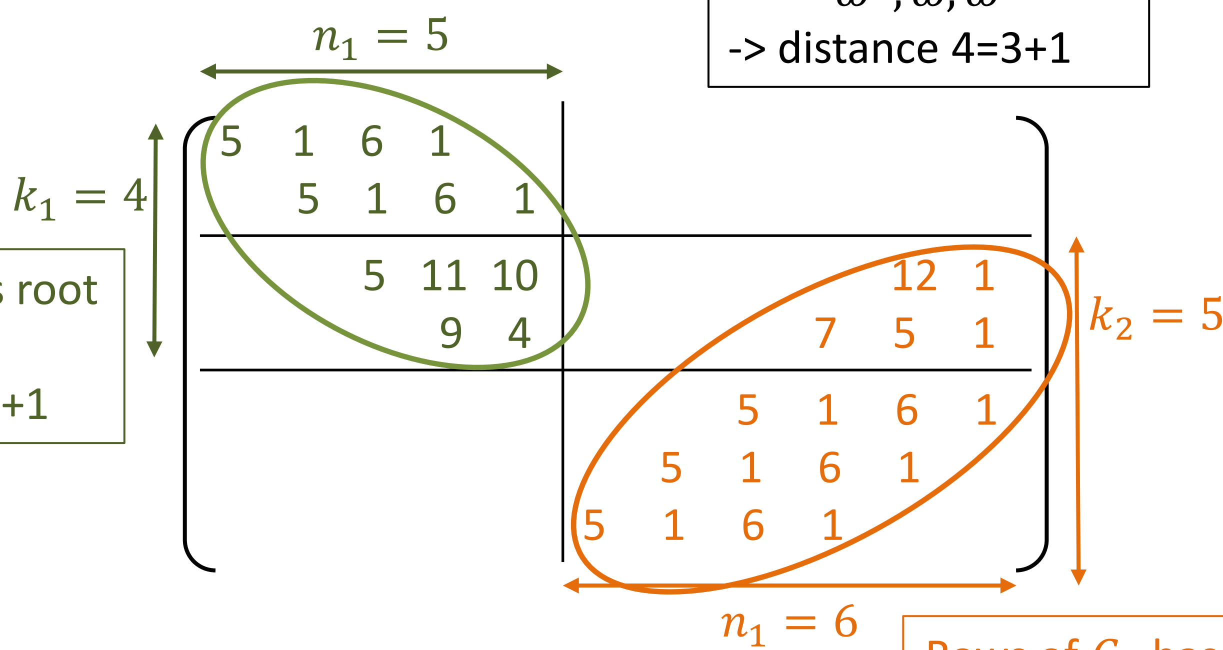


Two Codes Have the Same Redundancy: Construction

We construct the generator matrix of the optimal code. Example: F_{13}

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{U} & \boldsymbol{O} \\ \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{O} & \boldsymbol{V} \end{bmatrix} =$$

Rows of $\boldsymbol{G}$ has roots $\omega^0, \omega, \omega^2$
-> distance 4=3+1



Rows of G_1 has root $1 = \omega^0$
-> distance 2=1+1

Rows of G_2 has root $1 = \omega^0$
-> distance 2=1+1

Two Codes Have the Same Redundancy: Construction

Summary of this construction

- Rows: treated as polynomial having certain roots
- Solving systems of linear equations to determine rows
- BCH bound → global code & local codes have desired distances

$$
\left(\begin{array}{ccccc|cccccc}
5 & 1 & 6 & 1 & & & & & & & \\
 & 5 & 1 & 6 & 1 & & & & & & \\
\hline
 & & 5 & 11 & 10 & & & & & 12 & 1 \\
 & & & 9 & 4 & & & & 7 & 5 & 1 \\
\hline
 & & & & & & & 5 & 1 & 6 & 1 \\
 & & & & & & 5 & 1 & 6 & 1 & \\
 & & & & & 5 & 1 & 6 & 1 & &
\end{array}\right)
$$

Conclusions

What we have done

- Introduce a new coding problem: how to jointly design 2 (or more) MDS codes to have better overall failure tolerance
- **Construct optimal codes for two cases**
  - There are few common data
  - Two codes have the same amount of redundancy

Open Questions

  - Codes over small field size for 2 local codes: $n_1 - k_1 \neq n_2 - k_2$
  - Codes over small field size for more than two local codes