Isometry and Automorphisms of Constant Dimension Codes

Thomas Feulner

University of Bayreuth

Fq10
July 12, 2011

Joint work with A.-L. Trautmann (University of Zurich)

Network Coding

Outline:

- Introduction to Network Coding
- Equivalence of constant dimension codes
- Rephrase Equivalence in terms of a finite group action
- Algorithm for the calculation of unique orbit representatives and the stabilizer for some given linear code
- Generalization for network codes

Network Coding

Outline:

- Introduction to Network Coding
- Equivalence of constant dimension codes
- Rephrase Equivalence in terms of a finite group action
- Algorithm for the calculation of unique orbit representatives and the stabilizer for some given linear code
- Generalization for network codes

Network Coding

Outline:

- Introduction to Network Coding
- Equivalence of constant dimension codes
- Rephrase Equivalence in terms of a finite group action
- Algorithm for the calculation of unique orbit representatives and the stabilizer for some given linear code
- Generalization for network codes

Network Coding

Outline:

- Introduction to Network Coding
- Equivalence of constant dimension codes
- Rephrase Equivalence in terms of a finite group action
- Algorithm for the calculation of unique orbit representatives and the stabilizer for some given linear code
- Generalization for network codes

Network Coding

Outline:

- Introduction to Network Coding
- Equivalence of constant dimension codes
- Rephrase Equivalence in terms of a finite group action
- Algorithm for the calculation of unique orbit representatives and the stabilizer for some given linear code
- Generalization for network codes

Network Coding

Projective Geometry

The *projective geometry* $\mathrm{PG}(\mathbb{F}_q^n)$ is the set of all subspaces of $\mathbb{F}_q^n$ ordered by inclusion.

Network Code

A *network code* C is a subset of the projective geometry $\mathrm{PG}(\mathbb{F}_q^n)$.

Constant Dimension Code

C is called *constant dimension code* if

- C is a network code and
- $\exists\, 1 \leq k \leq n$ such that $\mathcal{U} \in C \Longrightarrow \dim(\mathcal{U}) = k$

Network Coding

Projective Geometry

The *projective geometry* $\mathrm{PG}(\mathbb{F}_q^n)$ is the set of all subspaces of $\mathbb{F}_q^n$ ordered by inclusion.

Network Code

A *network code* C is a subset of the projective geometry $\mathrm{PG}(\mathbb{F}_q^n)$.

Constant Dimension Code

C is called *constant dimension code* if

- C is a network code and
- $\exists\, 1 \leq k \leq n$ such that $\mathcal{U} \in C \Longrightarrow \dim(\mathcal{U}) = k$

Network Coding

Projective Geometry

The *projective geometry* $\mathrm{PG}(\mathbb{F}_q^n)$ is the set of all subspaces of $\mathbb{F}_q^n$ ordered by inclusion.

Network Code

A *network code* C is a subset of the projective geometry $\mathrm{PG}(\mathbb{F}_q^n)$.

Constant Dimension Code

C is called *constant dimension code* if

- C is a network code and
- $\exists\, 1 \leq k \leq n$ such that $\mathcal{U} \in C \Longrightarrow \dim(\mathcal{U}) = k$

Network Coding

Projective Geometry

The *projective geometry* $\mathrm{PG}(\mathbb{F}_q^n)$ is the set of all subspaces of $\mathbb{F}_q^n$ ordered by inclusion.

Network Code

A *network code* C is a subset of the projective geometry $\mathrm{PG}(\mathbb{F}_q^n)$.

Constant Dimension Code

C is called *constant dimension code* if

- C is a network code and
- $\exists\, 1 \leq k \leq n$ such that $\mathcal{U} \in C \Longrightarrow \dim(\mathcal{U}) = k$

Network Coding

Projective Geometry

The *projective geometry* $\mathrm{PG}(\mathbb{F}_q^n)$ is the set of all subspaces of $\mathbb{F}_q^n$ ordered by inclusion.

Network Code

A *network code* C is a subset of the projective geometry $\mathrm{PG}(\mathbb{F}_q^n)$.

Constant Dimension Code

C is called *constant dimension code* if

- C is a network code and
- $\exists\ 1 \leq k \leq n$ such that $\mathcal{U} \in C \Longrightarrow \dim(\mathcal{U}) = k$

Metric

Subspace Distance in $\mathrm{PG}(\mathbb{F}_q^n)$

$$\mathrm{d}_S(\mathcal{U}, \mathcal{V}) := \dim(\mathcal{U}) + \dim(\mathcal{V}) - \dim(\mathcal{U} \cap \mathcal{V})$$

Isometry

Isometry

A map $\iota : \mathrm{PG}(\mathbb{F}_q^n) \to \mathrm{PG}(\mathbb{F}_q^n)$ preserving the subspace distance, i.e.

$$d_S(\mathcal{U}, \mathcal{V}) = d_S(\iota(\mathcal{U}), \iota(\mathcal{V})) \ \forall\, \mathcal{U}, \mathcal{V} \in \mathrm{PG}(\mathbb{F}_q^n)$$

is called an *isometry* on $\mathrm{PG}(\mathbb{F}_q^n)$.

Theorem

Let $n \geq 3, \iota : \mathrm{PG}(\mathbb{F}_q^n) \to \mathrm{PG}(\mathbb{F}_q^n)$:

$$\iota \text{ isometry with } \iota(\{0\}) = \{0\} \iff \iota \text{ is an order preserving map}$$

Isometry

Isometry

A map $\iota : \mathrm{PG}(\mathbb{F}_q^n) \to \mathrm{PG}(\mathbb{F}_q^n)$ preserving the subspace distance, i.e.

$$d_S(\mathcal{U}, \mathcal{V}) = d_S(\iota(\mathcal{U}), \iota(\mathcal{V})) \ \forall\, \mathcal{U}, \mathcal{V} \in \mathrm{PG}(\mathbb{F}_q^n)$$

is called an *isometry* on $\mathrm{PG}(\mathbb{F}_q^n)$.

Theorem

Let $n \geq 3, \iota : \mathrm{PG}(\mathbb{F}_q^n) \to \mathrm{PG}(\mathbb{F}_q^n)$*:*

$$\iota \text{ isometry with } \iota(\{0\}) = \{0\} \iff \iota \text{ is an order preserving map}$$

Isometry

Fundamental Theorem of Projective Geometry

Let $n \geq 3$ and $\iota : \mathrm{PG}(\mathbb{F}_q^n) \to \mathrm{PG}(\mathbb{F}_q^n)$ isometry with $\iota(\{0\}) = \{0\}$, if and only if

$$\iota = (A, \alpha) \in \mathrm{P\Gamma L}_n = (\mathsf{GL}_n / \mathcal{Z}_n) \rtimes \mathrm{Aut}(\mathbb{F}_q)$$

(the projective semilinear group).

Equivalence

Definition

Two network codes C, C' are called *equivalent* if and only if there is an isometry $\iota : \mathrm{PG}(\mathbb{F}_q^n) \to \mathrm{PG}(\mathbb{F}_q^n)$ with $\iota(\{0\}) = \{0\}$ such that $\iota(C) = C'$.

Remark

Due to the transmission model applied in random network coding it is also reasonable to demand $\iota(\{0\}) = \{0\}$ for the definition of equivalence of network codes.

Equivalence

Definition

Two network codes C, C' are called *equivalent* if and only if there is an isometry $\iota : \mathrm{PG}(\mathbb{F}_q^n) \to \mathrm{PG}(\mathbb{F}_q^n)$ with $\iota(\{0\}) = \{0\}$ such that $\iota(C) = C'$.

Remark

Due to the transmission model applied in random network coding it is also reasonable to demand $\iota(\{0\}) = \{0\}$ for the definition of equivalence of network codes.

In terms of finite group actions

Goal

Let C be a constant dimension code. Calculate the

- stabilizer $Aut(C)$ under the action of $\mathrm{P\Gamma L}_n$,
- a *uniquely defined representative* of the orbit $\mathrm{P\Gamma L}_n \cdot C$

In terms of finite group actions

Goal

Let C be a constant dimension code. Calculate the

- stabilizer $Aut(C)$ under the action of $\mathrm{P\Gamma L}_n$,
- a *uniquely defined representative* of the orbit $\mathrm{P\Gamma L}_n \cdot C$

In terms of finite group actions

Goal

Let C be a constant dimension code. Calculate the

- stabilizer $Aut(C)$ under the action of $\mathrm{P\Gamma L}_n$,
- a *uniquely defined representative* of the orbit $\mathrm{P\Gamma L}_n \cdot C$

In terms of finite group actions

Goal

Let C be a constant dimension code. Calculate the

- stabilizer $Aut(C)$ under the action of $\mathrm{P\Gamma L}_n$,
- a *uniquely defined representative* of the orbit $\mathrm{P\Gamma L}_n \cdot C$

efficiently.

Matrix Representation

Matrix Representation

Let $C = \{\mathcal{U}_i \mid i = 0, \ldots, m-1\}$ be a constant dimension code, with $\mathcal{U}_i = \text{colspace}(U_i)$, $U_i \in \mathbb{F}_q^{n \times k}$.

- $\Gamma = (U_0 \ldots U_{m-1}) \in \mathbb{F}_q^{n \times km}$ is a *matrix representation* of C.
- The set of all matrix representations of C is equal to the orbit $(GL_k^m \rtimes \mathrm{S}_m) \cdot \Gamma$.

Remark

For „useful“ constant dimension codes it will allways be the case that the rank of Γ is equal to n.

Monomial Matrices

If $k = 1$ then $GL_k = \mathbb{F}_q^*$ and $GL_k^m \rtimes \mathrm{S}_m$ is equal to the set of monomial matrices.

Matrix Representation

Matrix Representation

Let $C = \{\mathcal{U}_i \mid i = 0, \ldots, m-1\}$ be a constant dimension code, with $\mathcal{U}_i = \text{colspace}(U_i)$, $U_i \in \mathbb{F}_q^{n \times k}$.

- $\Gamma = (U_0 \ldots U_{m-1}) \in \mathbb{F}_q^{n \times km}$ is a *matrix representation* of C.
- The set of all matrix representations of C is equal to the orbit $(GL_k^m \rtimes \mathrm{S}_m) \cdot \Gamma$.

Remark

For „useful“ constant dimension codes it will allways be the case that the rank of Γ is equal to n.

Monomial Matrices

If $k = 1$ then $GL_k = \mathbb{F}_q^*$ and $GL_k^m \rtimes \mathrm{S}_m$ is equal to the set of monomial matrices.

Matrix Representation

Matrix Representation

Let $C = \{\mathcal{U}_i \mid i = 0, \ldots, m-1\}$ be a constant dimension code, with $\mathcal{U}_i = \text{colspace}(U_i)$, $U_i \in \mathbb{F}_q^{n \times k}$.

- $\Gamma = (U_0 \ldots U_{m-1}) \in \mathbb{F}_q^{n \times km}$ is a *matrix representation* of C.
- The set of all matrix representations of C is equal to the orbit $(GL_k^m \rtimes \mathrm{S}_m) \cdot \Gamma$.

Remark

For „useful“ constant dimension codes it will allways be the case that the rank of Γ is equal to n.

Monomial Matrices

If $k = 1$ then $GL_k = \mathbb{F}_q^*$ and $GL_k^m \rtimes \mathrm{S}_m$ is equal to the set of monomial matrices.

Matrix Representation

Matrix Representation

Let $C = \{\mathcal{U}_i \mid i = 0, \dots, m-1\}$ be a constant dimension code, with $\mathcal{U}_i = \text{colspace}(U_i)$, $U_i \in \mathbb{F}_q^{n \times k}$.

- $\Gamma = (U_0 \dots U_{m-1}) \in \mathbb{F}_q^{n \times km}$ is a *matrix representation* of C.
- The set of all matrix representations of C is equal to the orbit $(GL_k^m \rtimes \mathrm{S}_m) \cdot \Gamma$.

Remark

For „useful“ constant dimension codes it will allways be the case that the rank of Γ is equal to n.

Monomial Matrices

If $k = 1$ then $GL_k = \mathbb{F}_q^*$ and $GL_k^m \rtimes \mathrm{S}_m$ is equal to the set of monomial matrices.

Matrix Representation

Matrix Representation

Let $C = \{\mathcal{U}_i \mid i = 0, \ldots, m-1\}$ be a constant dimension code, with $\mathcal{U}_i = \operatorname{colspace}(U_i)$, $U_i \in \mathbb{F}_q^{n \times k}$.

- $\Gamma = (U_0 \ldots U_{m-1}) \in \mathbb{F}_q^{n \times km}$ is a *matrix representation* of C.
- The set of all matrix representations of C is equal to the orbit $(GL_k^m \rtimes \mathrm{S}_m) \cdot \Gamma$.

Remark

For „useful“ constant dimension codes it will allways be the case that the rank of Γ is equal to n.

Monomial Matrices

If $k = 1$ then $GL_k = \mathbb{F}_q^*$ and $GL_k^m \rtimes \mathrm{S}_m$ is equal to the set of monomial matrices.

Equivalence

Equivalence

- Two matrix representations $\Gamma, \Gamma' \in \mathbb{F}_q^{n \times km}$ are called equivalent if the corresponding constant dimension codes are equivalent.
- $\{\Gamma' \mid \Gamma' \text{ equivalent to } \Gamma\} = ((\mathrm{GL}_n \times \mathrm{GL}_k^m) \rtimes (\mathrm{S}_m \times \mathrm{Aut}(\mathbb{F}_q))) \cdot \Gamma$

Equivalence

Equivalence

- Two matrix representations $\Gamma, \Gamma' \in \mathbb{F}_q^{n \times km}$ are called equivalent if the corresponding constant dimension codes are equivalent.
- $\{\Gamma' \mid \Gamma' \text{ equivalent to } \Gamma\} =$ $((\mathrm{GL}_n \times \mathrm{GL}_k^m) \rtimes (\mathrm{S}_m \times \mathrm{Aut}(\mathbb{F}_q))) \cdot \Gamma$

Equivalence

Equivalence

- Two matrix representations $\Gamma, \Gamma' \in \mathbb{F}_q^{n \times km}$ are called equivalent if the corresponding constant dimension codes are equivalent.
- $\{\Gamma' \mid \Gamma' \text{ equivalent to } \Gamma\} =$
  $((\mathrm{GL}_n \times \mathrm{GL}_k^m) \rtimes (\mathrm{S}_m \times \mathrm{Aut}(\mathbb{F}_q))) \cdot \Gamma$

Linear and additive codes (another point of view)

The special case $k = 1$

- The rows of a matrix representation Γ of C generate a linear $[m, n]_q$-code L.
- $\{\Gamma' \mid \Gamma'$ generator matrix of L' equivalent to $L\} = ((\mathrm{GL}_n \times \mathrm{GL}_k^m) \rtimes (\mathrm{S}_m \times \mathrm{Aut}(\mathbb{F}_q))) \cdot \Gamma$

Corollary

We can calculate the automorphism group of C using the algorithms (Leon 1982, F. 2009) from classical coding theory.

Linear and additive codes (another point of view)

The special case $k = 1$

- The rows of a matrix representation Γ of C generate a linear $[m, n]_q$-code L.
- $\{\Gamma' \mid \Gamma' \text{ generator matrix of } L' \text{ equivalent to } L\} = ((\mathrm{GL}_n \times \mathrm{GL}_k^m) \rtimes (\mathrm{S}_m \times \mathrm{Aut}(\mathbb{F}_q))) \cdot \Gamma$

Corollary

We can calculate the automorphism group of C using the algorithms (Leon 1982, F. 2009) from classical coding theory.

Linear and additive codes (another point of view)

The special case $k = 1$

- The rows of a matrix representation Γ of C generate a linear $[m, n]_q$-code L.
- $\{\Gamma' \mid \Gamma' \text{ generator matrix of } L' \text{ equivalent to } L\} = ((\mathrm{GL}_n \times \mathrm{GL}_k^m) \rtimes (\mathrm{S}_m \times \mathrm{Aut}(\mathbb{F}_q))) \cdot \Gamma$

Corollary

We can calculate the automorphism group of C using the algorithms (Leon 1982, F. 2009) from classical coding theory.

Linear and additive codes (another point of view)

The special case $k = 1$

- The rows of a matrix representation Γ of C generate a linear $[m, n]_q$-code L.
- $\{\Gamma' \mid \Gamma' \text{ generator matrix of } L' \text{ equivalent to } L\} = ((\mathrm{GL}_n \times \mathrm{GL}_k^m) \rtimes (\mathrm{S}_m \times \mathrm{Aut}(\mathbb{F}_q))) \cdot \Gamma$

Corollary

We can calculate the automorphism group of C using the algorithms (Leon 1982, F. 2009) from classical coding theory.

Linear and additive codes (another point of view)

The special case $k = 1$

- The rows of a matrix representation Γ of C generate a linear $[m, n]_q$-code L.
- $\{\Gamma' \mid \Gamma'$ generator matrix of L' equivalent to $L\} = ((\mathrm{GL}_n \times \mathrm{GL}_k^m) \rtimes (\mathrm{S}_m \times \mathrm{Aut}(\mathbb{F}_q))) \cdot \Gamma$

Corollary

We can calculate the automorphism group of C using the algorithms (Leon 1982, F. 2009) from classical coding theory. But, $k = 1$ does not yield applicable constant dimension codes. Hence, we will suppose $k > 1$ in the following.

Linear and additive codes (another point of view)

The case $k \neq 1$ (Y. Edel)

- The rows of a matrix representation Γ of C generate an additive (m, q^n)-code A over $\mathbb{F}_{q^k}$.
- $\{\Gamma' \mid \Gamma' \text{ represents } A' \text{ equivalent to } A\} = ((\mathrm{GL}_n \times \mathrm{GL}_k^m) \rtimes (\mathrm{S}_m \times \mathrm{Aut}(\mathbb{F}_q))) \cdot \Gamma$

Corollary

Our new algorithm solves the similar problem for additive codes.

Linear and additive codes (another point of view)

The case $k \neq 1$ (Y. Edel)

- The rows of a matrix representation Γ of C generate an additive (m, q^n)-code A over $\mathbb{F}_{q^k}$.
- $\{\Gamma' \mid \Gamma' \text{ represents } A' \text{ equivalent to } A\} = ((\mathrm{GL}_n \times \mathrm{GL}_k^m) \rtimes (\mathrm{S}_m \times \mathrm{Aut}(\mathbb{F}_q))) \cdot \Gamma$

Corollary

Our new algorithm solves the similar problem for additive codes.

Linear and additive codes (another point of view)

The case $k \neq 1$ (Y. Edel)

- The rows of a matrix representation Γ of C generate an additive (m, q^n)-code A over $\mathbb{F}_{q^k}$.
- $\{\Gamma' \mid \Gamma' \text{ represents } A' \text{ equivalent to } A\} = ((\mathrm{GL}_n \times \mathrm{GL}_k^m) \rtimes (\mathrm{S}_m \times \mathrm{Aut}(\mathbb{F}_q))) \cdot \Gamma$

Corollary

Our new algorithm solves the similar problem for additive codes.

Linear and additive codes (another point of view)

The case $k \neq 1$ (Y. Edel)

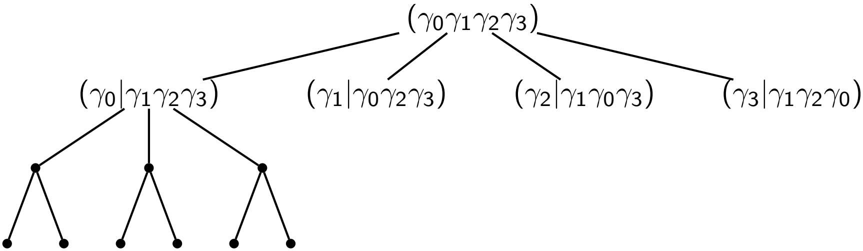
- The rows of a matrix representation Γ of C generate an additive (m, q^n)-code A over $\mathbb{F}_{q^k}$.
- $\{\Gamma' \mid \Gamma'$ represents A' equivalent to $A\} =$ $((\mathrm{GL}_n \times \mathrm{GL}_k^m) \rtimes (\mathrm{S}_m \times \mathrm{Aut}(\mathbb{F}_q))) \cdot \Gamma$

Corollary

Our new algorithm solves the similar problem for additive codes.

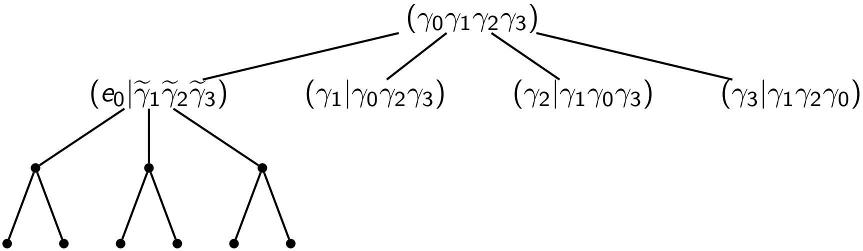
The algorithm for linear codes

Backtracking over S_m permutes columns



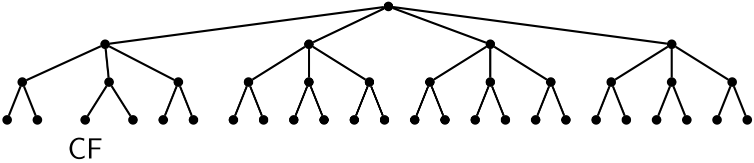
The algorithm for linear codes

Minimize fixed columns, not changing the previously fixed using the remaining part of the group $(\mathsf{GL}_n \times \mathsf{GL}_k^m) \rtimes \mathrm{Aut}(\mathbb{F}_q)$



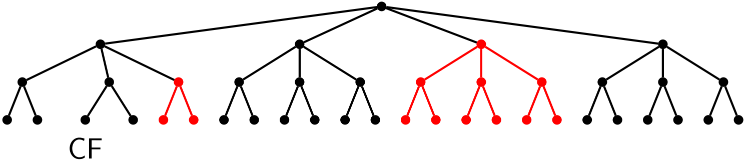
The algorithm for linear codes

Canonical Form equals minimum over all leaf nodes



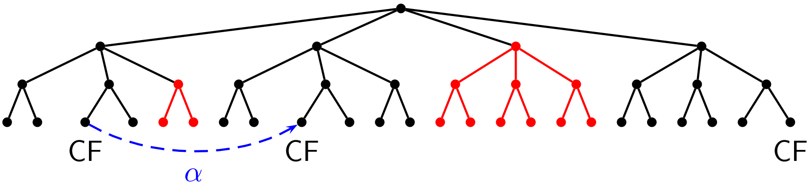
The algorithm for linear codes

Prune subtrees rooted in nonoptimal initial parts



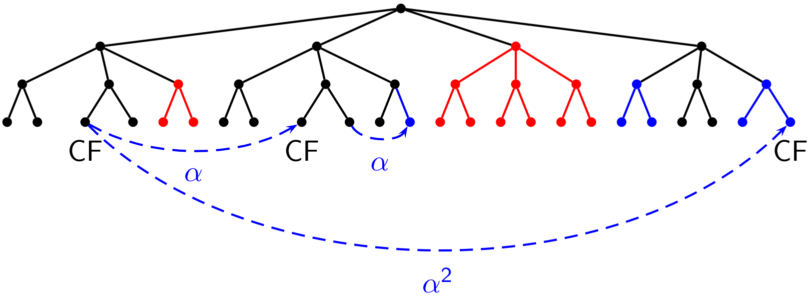
The algorithm for linear codes

Automorphisms can be found by paths connecting equal leafs



The algorithm for linear codes

Automorphisms can be used for further pruning the tree



Improvement: Refinements

Try to distinguish columns by "properties" (invariant under the remaining group action)

$(\gamma_0\gamma_1\gamma_2\gamma_3\gamma_4)$ ⟵ S_5 acting

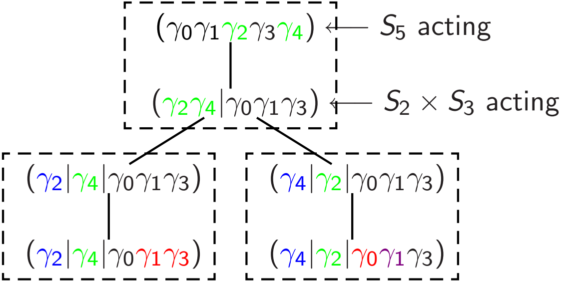
$(\gamma_2\gamma_4|\gamma_0\gamma_1\gamma_3)$ ⟵ $S_2 \times S_3$ acting

$(\gamma_2|\gamma_4|\gamma_0\gamma_1\gamma_3)$ $(\gamma_4|\gamma_2|\gamma_0\gamma_1\gamma_3)$

$(\gamma_2|\gamma_4|\gamma_0\gamma_1\gamma_3)$ $(\gamma_4|\gamma_2|\gamma_0\gamma_1\gamma_3)$

Improvement: Refinements

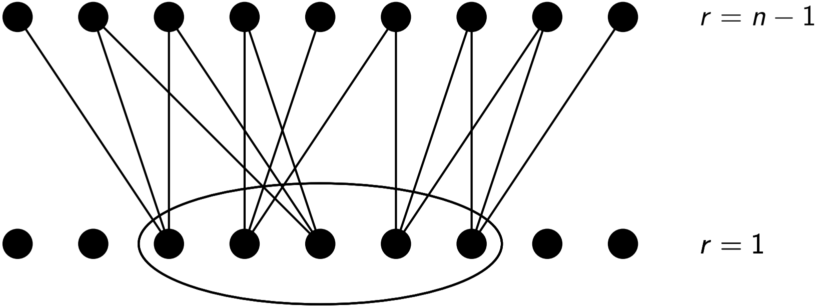
Try to distinguish columns by "properties" (invariant under the remaining group action)



$\Longrightarrow$ Ordering on "properties" allows further pruning.

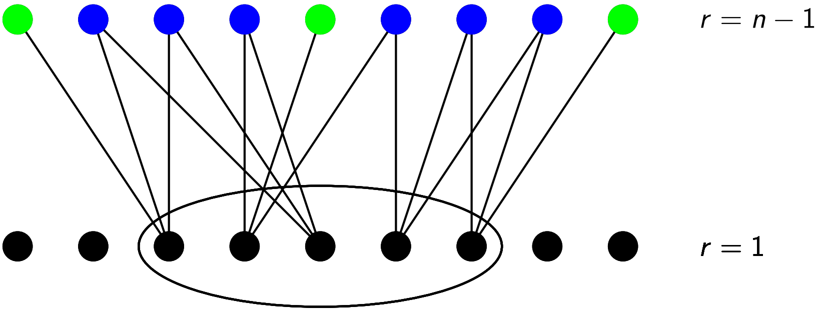
Refinement in the case of linear codes

Example: Incidence with hyperplanes



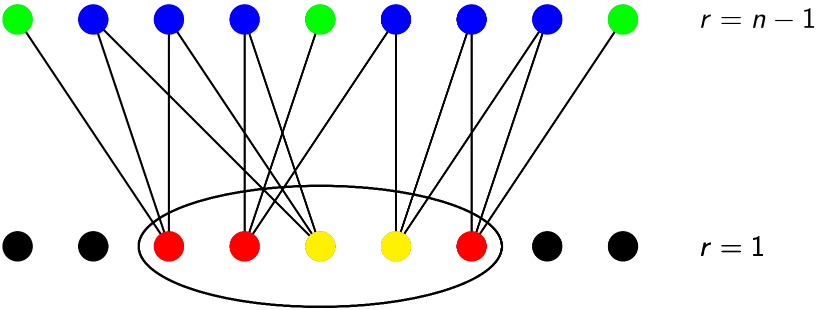
Refinement in the case of linear codes

Example: Incidence with hyper-planes



Refinement in the case of linear codes

Example: Repeat process until stable

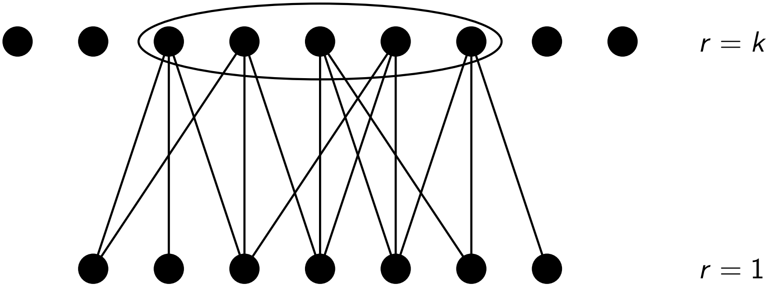


Work in progress: Canonization of network codes

Idea: Adapt the algorithm for linear codes to be used for constant dimension codes.

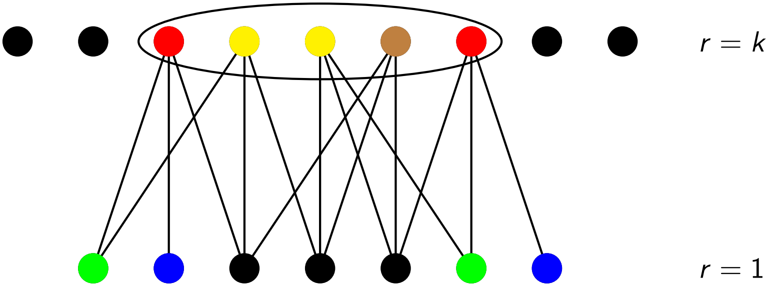
Canonization of network codes

Step 1: Choose appropriate linear code



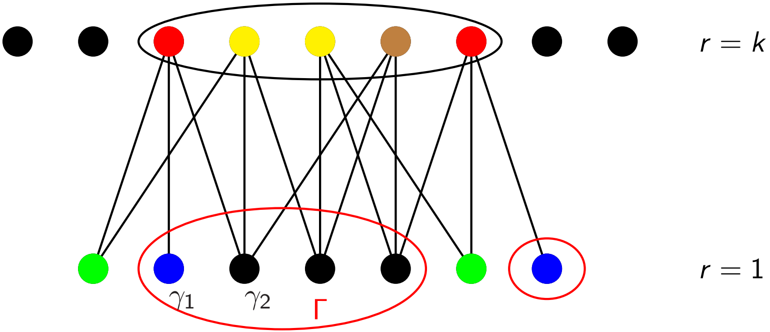
Canonization of network codes

Step 1: Choose appropriate linear code using the refinement like above



Canonization of network codes

Step 1: Choose appropriate linear code, i.e. some union of equally colored points



Canonization of network codes

Adaption of the algorithm

- Root node is identified by
  $\left(\begin{array}{ccc|ccc} \gamma_0 & \dots & \gamma_{x-1} & U_0 & \dots & U_{m-1} \end{array} \right)$
- Backtracking over $S_x \times S_m$
- Refinement of $S_x \times S_m$ via incidence between $\gamma_0, \dots, \gamma_{x-1}$ and $U_0, \dots, U_{m-1}$

Like before

- Restriction of $\mathrm{GL}_n \rtimes \mathrm{Aut}(\mathbb{F}_q)$ by minimization of fixed columns γ_i
- Pruning whenever some node $\left(\begin{array}{ccc|ccc} \widetilde{\gamma}_{\pi(0)} & \dots & \widetilde{\gamma}_{\pi(x-1)} & \widetilde{U}_{\sigma(0)} & \dots & \widetilde{U}_{\sigma(m-1)} \end{array} \right)$ is identified to be nonoptimal
- Use known automorphisms for pruning.

Canonization of network codes

Adaption of the algorithm

- Root node is identified by $\left(\begin{array}{ccc|ccc} \gamma_0 & \ldots & \gamma_{x-1} & U_0 & \ldots & U_{m-1} \end{array} \right)$
- Backtracking over $S_x \times S_m$
- Refinement of $S_x \times S_m$ via incidence between $\gamma_0, \ldots, \gamma_{x-1}$ and $U_0, \ldots, U_{m-1}$

Like before

- Restriction of $\mathrm{GL}_n \rtimes \mathrm{Aut}(\mathbb{F}_q)$ by minimization of fixed columns γ_i
- Pruning whenever some node $\left(\begin{array}{ccc|ccc} \widetilde{\gamma}_{\pi(0)} & \ldots & \widetilde{\gamma}_{\pi(x-1)} & \widetilde{U}_{\sigma(0)} & \ldots & \widetilde{U}_{\sigma(m-1)} \end{array} \right)$ is identified to be nonoptimal
- Use known automorphisms for pruning.

Canonization of network codes

Adaption of the algorithm

- Root node is identified by
  $\left(\begin{array}{ccc|ccc} \gamma_0 & \dots & \gamma_{x-1} & U_0 & \dots & U_{m-1} \end{array} \right)$
- Backtracking over $S_x \times S_m$
- Refinement of $S_x \times S_m$ via incidence between $\gamma_0, \dots, \gamma_{x-1}$ and $U_0, \dots, U_{m-1}$

Like before

- Restriction of $\mathrm{GL}_n \rtimes \mathrm{Aut}(\mathbb{F}_q)$ by minimization of fixed columns γ_i
- Pruning whenever some node $\left(\begin{array}{ccc|ccc} \widetilde{\gamma}_{\pi(0)} & \dots & \widetilde{\gamma}_{\pi(x-1)} & \widetilde{U}_{\sigma(0)} & \dots & \widetilde{U}_{\sigma(m-1)} \end{array} \right)$ is identified to be nonoptimal
- Use known automorphisms for pruning.

Canonization of network codes

Adaption of the algorithm

- Root node is identified by
  $\left(\begin{array}{ccc|ccc} \gamma_0 & \dots & \gamma_{x-1} & U_0 & \dots & U_{m-1} \end{array} \right)$
- Backtracking over $S_x \times S_m$
- Refinement of $S_x \times S_m$ via incidence between $\gamma_0, \dots, \gamma_{x-1}$ and $U_0, \dots, U_{m-1}$

Like before

- Restriction of $\mathrm{GL}_n \rtimes \mathrm{Aut}(\mathbb{F}_q)$ by minimization of fixed columns γ_i
- Pruning whenever some node $\left(\begin{array}{ccc|ccc} \widetilde{\gamma}_{\pi(0)} & \dots & \widetilde{\gamma}_{\pi(x-1)} & \widetilde{U}_{\sigma(0)} & \dots & \widetilde{U}_{\sigma(m-1)} \end{array} \right)$ is identified to be nonoptimal
- Use known automorphisms for pruning.

Canonization of network codes

Adaption of the algorithm

- Root node is identified by
  $\left(\begin{array}{ccc|ccc} \gamma_0 & \dots & \gamma_{x-1} & U_0 & \dots & U_{m-1} \end{array}\right)$
- Backtracking over $S_x \times S_m$
- Refinement of $S_x \times S_m$ via incidence between $\gamma_0, \dots, \gamma_{x-1}$ and $U_0, \dots, U_{m-1}$

Like before

- Restriction of $\mathrm{GL}_n \rtimes \mathrm{Aut}(\mathbb{F}_q)$ by minimization of fixed columns γ_i
- Pruning whenever some node $\left(\begin{array}{ccc|ccc} \widetilde{\gamma}_{\pi(0)} & \dots & \widetilde{\gamma}_{\pi(x-1)} & \widetilde{U}_{\sigma(0)} & \dots & \widetilde{U}_{\sigma(m-1)} \end{array}\right)$ is identified to be nonoptimal
- Use known automorphisms for pruning.

Canonization of network codes

Adaption of the algorithm

- Root node is identified by
  $\left(\begin{array}{ccc|ccc} \gamma_0 & \dots & \gamma_{x-1} & U_0 & \dots & U_{m-1} \end{array}\right)$
- Backtracking over $S_x \times S_m$
- Refinement of $S_x \times S_m$ via incidence between $\gamma_0, \dots, \gamma_{x-1}$ and $U_0, \dots, U_{m-1}$

Like before

- Restriction of $\mathrm{GL}_n \rtimes \mathrm{Aut}(\mathbb{F}_q)$ by minimization of fixed columns γ_i
- Pruning whenever some node $\left(\begin{array}{ccc|ccc} \widetilde{\gamma}_{\pi(0)} & \dots & \widetilde{\gamma}_{\pi(x-1)} & \widetilde{U}_{\sigma(0)} & \dots & \widetilde{U}_{\sigma(m-1)} \end{array}\right)$ is identified to be nonoptimal
- Use known automorphisms for pruning.

Canonization of network codes

Adaption of the algorithm

- Root node is identified by
  $\left(\begin{array}{ccc|ccc} \gamma_0 & \ldots & \gamma_{x-1} & U_0 & \ldots & U_{m-1} \end{array} \right)$
- Backtracking over $S_x \times S_m$
- Refinement of $S_x \times S_m$ via incidence between $\gamma_0, \ldots, \gamma_{x-1}$ and $U_0, \ldots, U_{m-1}$

Like before

- Restriction of $\mathrm{GL}_n \rtimes \mathrm{Aut}(\mathbb{F}_q)$ by minimization of fixed columns γ_i
- Pruning whenever some node
  $\left(\begin{array}{ccc|ccc} \widetilde{\gamma}_{\pi(0)} & \ldots & \widetilde{\gamma}_{\pi(x-1)} & \widetilde{U}_{\sigma(0)} & \ldots & \widetilde{U}_{\sigma(m-1)} \end{array} \right)$ is identified to be nonoptimal
- Use known automorphisms for pruning.

Canonization of network codes

Adaption of the algorithm

- Root node is identified by
  $\begin{pmatrix} \gamma_0 & \dots & \gamma_{x-1} & \vert & U_0 & \dots & U_{m-1} \end{pmatrix}$
- Backtracking over $S_x \times S_m$
- Refinement of $S_x \times S_m$ via incidence between $\gamma_0, \dots, \gamma_{x-1}$ and $U_0, \dots, U_{m-1}$

Like before

- Restriction of $\mathrm{GL}_n \rtimes \mathrm{Aut}(\mathbb{F}_q)$ by minimization of fixed columns γ_i
- Pruning whenever some node $\begin{pmatrix} \widetilde{\gamma}_{\pi(0)} & \dots & \widetilde{\gamma}_{\pi(x-1)} & \vert & \widetilde{U}_{\sigma(0)} & \dots & \widetilde{U}_{\sigma(m-1)} \end{pmatrix}$ is identified to be nonoptimal
- Use known automorphisms for pruning.