Flowering graphs

Interactive proximity test to codes on graphs by flowering

Hugo Delavenne, Louise Lallemand, Tanguy Medevielle, Élina Roussel

LIX, École Polytechnique, Institut Polytechnique de Paris
Inria

Monday 16th June 2025 @ Inria Paris

1. Interactive Oracle Proofs of Proximity
2. Flowering protocol
3. Flowering graphs

1. Interactive Oracle Proofs of Proximity
2. Flowering protocol
3. Flowering graphs

- SNARKs
- Interactive Oracle Proof of Proximity
- Fast Reed-Solomon IOPP

Why verifiable computing: a sad story

- $\mathcal{P}$ has executed a very long algorithm $A : x \mapsto y$
- very proud, it wants to share y to $\mathcal{V}$
- $\mathcal{V}$ only trusts what it sees
- it can't accept $y = A(x)$ and doesn't want to compute it itself
- $\mathcal{P}$ is very sad
- #emotion

SNARK means **S**uccinct **N**on-interactive **AR**gument of **K**nowledge.

It turns $A : x \mapsto y$ (that runs in $\tau(|x|)$) into

$$A' : x, \alpha \mapsto y, \pi$$

input randomness output SNARK that "$A(x) = y$"

satisfying:

- $|\pi| \ll \tau$
- A' runs in $\tilde{O}(\tau)$
- there is a verifier $\mathcal{V}$ in $\mathrm{poly}(|\pi|)$ such that
  - **Completeness:** if $A(x) = y$ then $\underset{\alpha}{\mathbb{P}}(\mathcal{V}(\pi)$ accepts$) = 1$
  - **Soundness:** if $A(x) \neq y$ then $\underset{\alpha}{\mathbb{P}}(\mathcal{V}(\pi)$ accepts$) \leqslant s$.

Code-based SNARK workflow

A, x

$\mathcal{P}$rover

$\mathcal{V}$erifier

Arithmetize A

Arithmetize A

Executes $A(x)$ $\xrightarrow[\text{trace}]{\text{execution}}$

Interactive **proximity test**

Notation **Relative Hamming distance**

Let $u, v \in \mathbb{F}^N$.

$$\Delta(u,v) := \frac{1}{N} |\{i \in [N] \mid u_i \neq v_i\}|$$

Notation **Reed-Solomon code**

Let $\mathcal{L} \subseteq \mathbb{F}$ and $k < |\mathcal{L}| = N$.

$$\mathsf{RS}[\mathcal{L}, k] := \{f : \mathcal{L} \to \mathbb{F} \mid f \in \mathbb{F}[X]_{\leqslant k-1}\}$$

Assume $\langle g \rangle$ has order K. **We want to compute u_{K-1}** where $\begin{cases} u_0 = u_1 = 1 \\ u_{i+2} = u_{i+1}^2 + u_i^2. \end{cases}$

$T(1) = 1$	$T(g) = 1$	$\cdots$	$T(g^i)$	$T(g^{i+1})$	$T(g^{i+2})$	$\cdots$	$T(g^{K-1})$

$$T(g^{i+2}) \leftarrow T(g^{i+1})^2 + T(g^i)^2$$

With $Q(X, Y, Z) := Z - Y^2 - X^2$, the trace is valid iff

$$\forall i \in \{0, \ldots, K-3\}, \qquad \underbrace{Q(T(g^i), T(g^{i+1}), T(g^{i+2}))}_{=: Q \circ T(g^i)} = 0.$$

With $Z(X) := \prod_{i=0}^{K-3} (X - g^i)$, the trace is valid iff

$$\exists f(X) \in \mathbb{F}[X]_{\leqslant K}, \qquad Q \circ T(X) = f(X) Z(X).$$

Arithmetization: Why testing proximity suffices?

Lemma **Reduction to proximity testing**

Let $\boldsymbol{T}, \boldsymbol{f} : \mathcal{L} \to \mathbb{F}$.

Assume that:

- $\Delta(\boldsymbol{T}, \mathrm{RS}[\mathcal{L}, K]) < \delta$ and $\Delta(\boldsymbol{f}, \mathrm{RS}[\mathcal{L}, K+1]) < \delta$ (proximity tests!)
- For any $T' \in \mathbb{F}[X]_{\leqslant K-1}, f' \in \mathbb{F}[X]_{\leqslant K}$,

$$Q \circ T'(X) \neq f'(X)Z(X). \qquad \text{(no valid trace!)}$$

Then

$$\mathbb{P}_{x \in \mathcal{L}}(Q \circ \boldsymbol{T}(x) = \boldsymbol{f}(x)Z(x)) \leqslant \frac{2K}{|\mathcal{L}|} + 2\delta.$$

[Bor22] Sarah Bordage. *Efficient protocols for testing proximity to algebraic codes*.
Theses, Institut Polytechnique de Paris, June 2022

Definition **Locally-testable code**

C is **(ℓ, δ, s)-locally-testable** if there is $\mathcal{V}$ with **only ℓ queries** to u such that

- **Completeness:** if $u \in C$ then $\mathbb{P}(\mathcal{V}^u \text{ accepts}) = 1$
- **Soundness:** if $\Delta(u, C) > \delta$ then $\mathbb{P}(\mathcal{V}^u \text{ accepts}) \leqslant s$.

C has **locality ℓ** if C is (ℓ, δ, s)-locally-testable for $\delta, s < 1$.

Example **Reed-Solomon codes are not locally-testable**

RS$[\mathcal{L}, K]$ does **not** have locality $\ell \leqslant K$.

Proof.
Any K values correspond to a degree $\leqslant K - 1$ polynomial by interpolation.

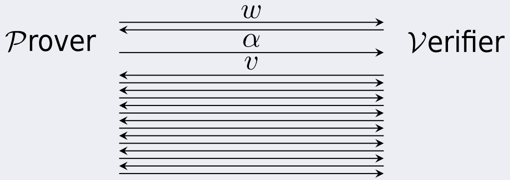
Definition **Non-interactive proof**

$$\mathcal{P}\text{rover} \xrightarrow{\quad w \quad} \mathcal{V}\text{erifier}$$

Definition **Sigma protocol**

$$\mathcal{P}\text{rover} \quad \begin{array}{c} \xrightarrow{\quad w \quad} \\ \xleftarrow{\quad \alpha \quad} \\ \xrightarrow[\quad v \quad]{} \end{array} \quad \mathcal{V}\text{erifier}$$

Definition **Interactive Proof**



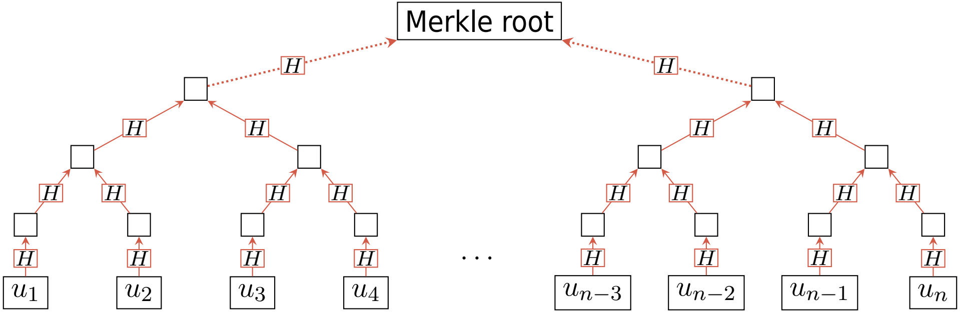
Remember: we want **S**uccinct **N**on-interactive **AR**guments of **K**nowledge.

- made non-interactive with **Fiat-Shamir**
- made succinct with **oracle accesses**

Definition **Oracle access**

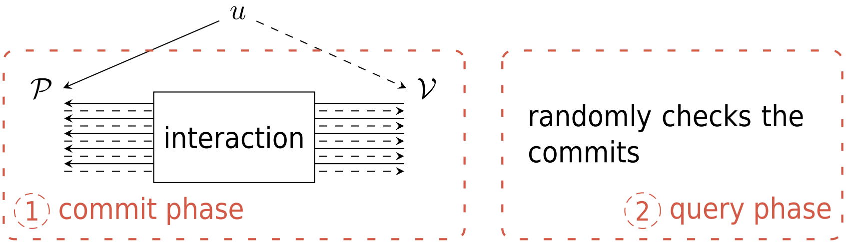
$\mathcal{P}$ provides to $\mathcal{V}$ an **oracle access** to $u \in \mathbb{F}^n$ by giving **black-box** access to u.

In practice, $\mathcal{P}$ provides the root of a **Merkle tree**.



Merkle root created in $O(n)$, Merkle branch opened in $O(\log n)$

Summary: IOPPs



- **Completeness:** if $u \in C$ then $\mathbb{P}(\mathcal{V}^{u,\leftrightarrow\mathcal{P}}$ accepts$) = 1$
- **Soundness:** if $\Delta(u, C) > \delta$ then for any $\mathcal{P}$, $\mathbb{P}(\mathcal{V}^{u,\leftrightarrow\mathcal{P}}$ accepts$) \leqslant s$

[BCS16] Eli Ben-Sasson, Alessandro Chiesa, and Nicholas Spooner. Interactive Oracle Proofs.
In *Theory of Cryptography: 14th International Conference, TCC 2016-B, Beijing, China, October 31-November 3, 2016, Proceedings, Part II 14*, pages 31–60. Springer, 2016

Idea: Test even and odd parts $f(X) =: f_{\text{even}}(X^2) + X f_{\text{odd}}(X^2)$.

Definition **Folding**

Let $f : \mathcal{L}_0 \to \mathbb{F}$ and $\alpha \in \mathbb{F}$.

$$\text{Fold}[f, \alpha](X^2) := f_{\text{even}}(X^2) + \alpha f_{\text{odd}}(X^2) = \frac{f(X) + f(-X)}{2} + \alpha \frac{f(X) - f(-X)}{2X}$$

- **Field restriction:** $\mathcal{L}_1 := \{x^2 \mid x, -x \in \mathcal{L}_0\}$ so $\mathbb{F}$ must have lots of roots of unity
- **Validity preservation:** $f \in \text{RS}[\mathcal{L}_0, K] \iff \mathbb{P}_{\alpha}(\text{Fold}[f, \alpha] \in \text{RS}[\mathcal{L}_1, K/2]) > \frac{1}{|\mathbb{F}|}$
- **Local check:** $\mathcal{V}$ computes $\text{Fold}[f, \alpha](x^2)$ with 2 queries to f

[BBHR18] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Fast Reed-Solomon Interactive Oracle Proofs of Proximity. **In *45th international colloquium on automata, languages, and programming (ICALP 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018**

$$f_0 : \mathcal{L}_0 \to \mathbb{F}$$

① commit phase

$\mathcal{P}$ $\quad$ $\mathcal{V}$

$$\alpha_0 \xleftarrow{\$} \mathbb{F}$$

$$f_1 : \mathcal{L}_1 \to \mathbb{F}$$

$$f_1 = \mathsf{Fold}[f_0, \alpha_0]$$

$$\vdots$$

$$\alpha_{r-1} \xleftarrow{\$} \mathbb{F}$$

$$f_r : \mathcal{L}_r \to \mathbb{F}$$

$$f_r = \mathsf{Fold}[f_{r-1}, \alpha_{r-1}]$$

② query phase

$$x_0 \xleftarrow{\$} \mathcal{L}_0,\ x_1 := x_0^2$$

$$f_1(x_1) \stackrel{?}{=} \mathsf{Fold}[f_0, \alpha_0](x_1)$$

$$\vdots$$

$$x_r := x_{r-1}^2$$

$$f_r(x_r) \stackrel{?}{=} \mathsf{Fold}[f_{r-1}, \alpha_{r-1}](x_r)$$

repeat m times

$$f_r \stackrel{?}{\in} \mathsf{RS}[\mathcal{L}_r, K/2^r]$$

[BBHR18] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Fast Reed-Solomon Interactive Oracle Proofs of Proximity. **In *45th international colloquium on automata, languages, and programming (ICALP 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018**

Let $N := |\mathcal{L}_0|$.

Proposition **FRI complexities [BBHR18]**

FRI protocol for RS$[\mathcal{L}_0, K]$ with m repetitions has following complexity:

- Prover complexity: $< 8N$ (after encoding)
- Verifier complexity: $< 2m \log K$
- Number of queries: $2m \log K$
- Number of rounds: $\log K$

[BBHR18] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Fast Reed-Solomon Interactive Oracle Proofs of Proximity. **In *45th international colloquium on automata, languages, and programming (ICALP 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018**

$$f_0 : \mathcal{L}_0 \to \mathbb{F}$$

$\mathcal{P}$ $\mathcal{V}$

$$\alpha_0 \xleftarrow{\$} \mathbb{F}$$

$$f_1 : \mathcal{L}_1 \to \mathbb{F}$$

$$f_1 = \mathsf{Fold}[f_0, \alpha_0]$$

$$\vdots$$

$$\alpha_{r-1} \xleftarrow{\$} \mathbb{F}$$

$$f_r : \mathcal{L}_r \to \mathbb{F}$$

$$f_r = \mathsf{Fold}[f_{r-1}, \alpha_{r-1}]$$

1 commit phase

A **commit error** is when

$$\Delta(\mathsf{Fold}[f_i, \alpha_i], C_{i+1}) < \Delta(f_i, C_i).$$

Proposition [BCI+23]

It happens w.p. over α_i

$$\leqslant \frac{10^7 N^{3.5}}{K^{1.5}|\mathbb{F}|}.$$

[BCI+23] Eli Ben-Sasson, Dan Carmon, Yuval Ishai, Swastik Kopparty, and Shubhangi Saraf. Proximity Gaps for Reed–Solomon Codes. ***J. ACM*, 70(5), October 2023**

A **query error** is when for some i

$$f_{i+1} \neq \mathsf{Fold}[f_i, \alpha_i]$$

but

$$f_{i+1}(x_{i+1}) = \mathsf{Fold}[f_i, \alpha_i](x_{i+1}).$$

Proposition [BCI+23]

If no commit error, it happens w.p.

$$\leqslant \left(1 - \min\left(\delta, 1 - \frac{21}{20}\sqrt{K/N}\right)\right)^m.$$

$$x_0 \xleftarrow{\$} \mathcal{L}_0,\ x_1 := x_0^2$$

$$f_1(x_1) \stackrel{?}{=} \mathsf{Fold}[f_0, \alpha_0](x_1)$$

$$\vdots$$

$$x_r := x_{r-1}^2$$

$$f_r(x_r) \stackrel{?}{=} \mathsf{Fold}[f_{r-1}, \alpha_{r-1}](x_r)$$

$$f_r \stackrel{?}{\in} \mathsf{RS}[\mathcal{L}_r, K/2^r]$$

repeat m times

2 query phase

[BCI+23] Eli Ben-Sasson, Dan Carmon, Yuval Ishai, Swastik Kopparty, and Shubhangi Saraf. Proximity Gaps for Reed–Solomon Codes. ***J. ACM*, 70(5), October 2023**

Proposition FRI completeness

If $f_0 \in \mathsf{RS}[\mathcal{L}_0, K]$ then $\mathcal{V}$ accepts with probability 1.

Theorem FRI soundness

If $\Delta(f_0, \mathsf{RS}[\mathcal{L}_0, K]) > \delta$ then for any $\tilde{\mathcal{P}}$ and $\varepsilon > 0$, $\mathcal{V}$ accepts with probability

$$\leqslant \frac{10^7 N^{3.5} \log K}{K^{1.5}|\mathbb{F}|} + \left(1 - \min\left(\delta, 1 - \frac{21}{20}\sqrt{K/N}\right)\right)^m \tag{1}$$

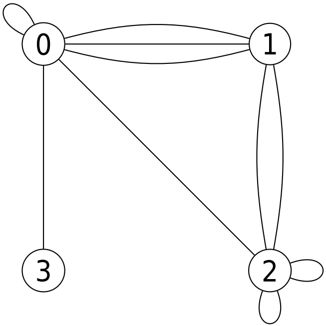
- $|\mathbb{F}| > 2^\lambda \cdot \frac{10^7 N^{3.5} \log K}{K^{1.5}}$, for security λ. Ex: extension of $\mathbb{F}_{2^{64}-2^{32}+1}$
- (1) does not take advantage of $\delta \in [1 - \sqrt{K/N}, 1 - K/N]$

[BCI+23] Eli Ben-Sasson, Dan Carmon, Yuval Ishai, Swastik Kopparty, and Shubhangi Saraf. Proximity Gaps for Reed–Solomon Codes. *J. ACM*, 70(5), October 2023

- Codes on graphs
- Folding graphs
- Flowering

$\Gamma = (V, E)$ is a n-RIM:

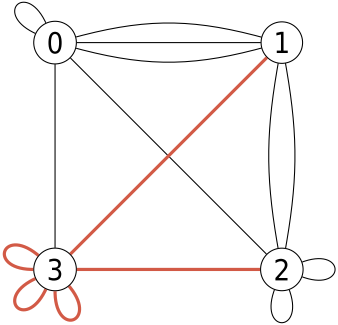
- **Multigraph:** multiple edges and loops



$\Gamma = (V, E)$ is a n-RIM:

▷ **Multigraph:** multiple edges and loops

▶ **Regular:** same number n of edges

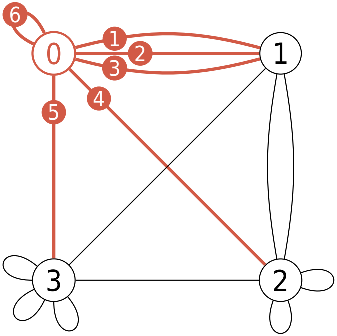


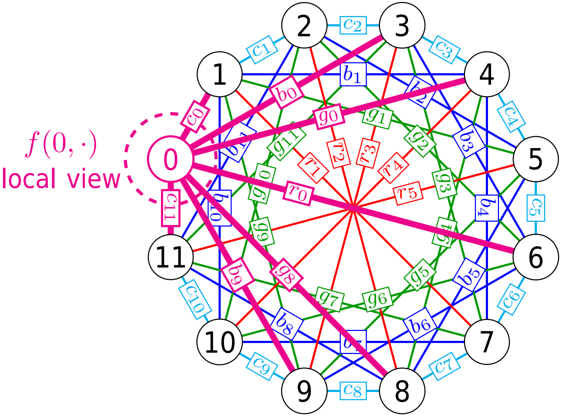
$\Gamma = (V, E)$ is a n-RIM:

▷ **Multigraph:** multiple edges and loops

▷ **Regular:** same number n of edges

▶ **Indexed:** edge is $(v, \ell) \in V \times [n]$
Write $\boldsymbol{E(v, \ell)} \in V$ the neighbor of v by ℓ





Word $f : V \times [n] \to \mathbb{F}$ on a graph Γ

Definition **Code $\mathcal{C}[\Gamma, C_0]$**

Given Γ a n-RIM and $C_0 \subseteq \mathbb{F}^n$,

$$f \in \mathcal{C}[\Gamma, C_0] \iff \forall v,\ f(v, \cdot) \in C_0.$$

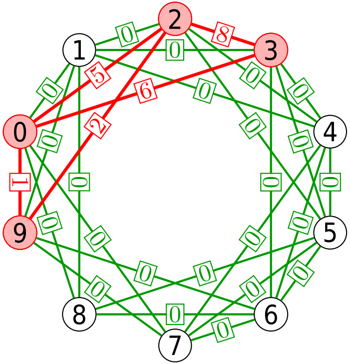
We'll only use $C_0 = \mathrm{RS}[n, k]$.

Example

For $v = 0$, being a codeword requires $(c_0, b_0, g_0, r_0, g_8, b_9, c_{11}) \in \mathrm{RS}[7, k]$.

Hamming: $\Delta(f, f') := \dfrac{\text{\#diff edges}}{\text{\#edges}} = \dfrac{5}{30} \approx 0.167$

Vertex: $\Delta_V(f, f') := \dfrac{\text{\#diff vertices}}{\text{\#vertices}} = \dfrac{4}{10} = 0.4$



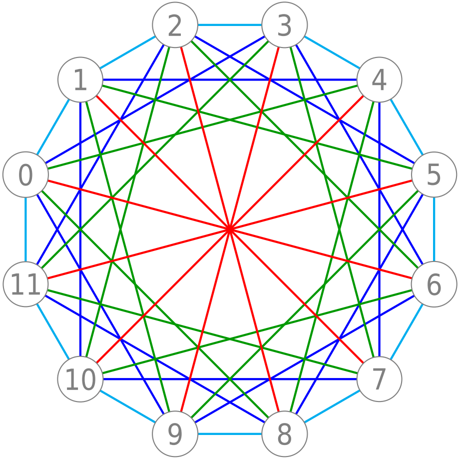
$f - f'$

Proposition **Hamming is more fine grain**

For any f, f', $\Delta(f, f') \leqslant \Delta_V(f, f')$.

[DMR25] Hugo Delavenne, Tanguy Medevielle, and Élina Roussel. Interactive Oracle Proofs of Proximity to Codes on Graphs, 2025. **Accepted to ISIT 2025**

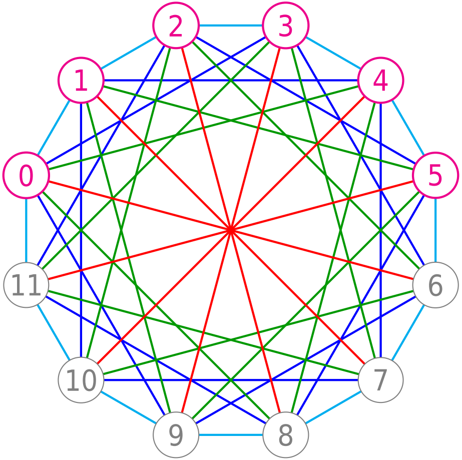
Cut-graph $\mathbf{\Gamma' = Cut[\Gamma, \mathit{V'}]}$:



[DMR25] Hugo Delavenne, Tanguy Medevielle, and Élina Roussel. Interactive Oracle Proofs of Proximity to Codes on Graphs, 2025. **Accepted to ISIT 2025**

Cut-graph $\mathbf{\Gamma' = Cut[\Gamma, V']}$:

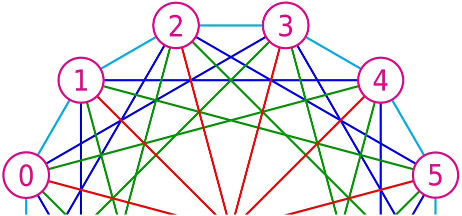
- Choose vertices $V' \subseteq V$



[DMR25] Hugo Delavenne, Tanguy Medevielle, and Élina Roussel. Interactive Oracle Proofs of Proximity to Codes on Graphs, 2025.
Accepted to ISIT 2025

Cut-graph $\boldsymbol{\Gamma' = \mathbf{Cut}[\Gamma, V']}$:

- Choose vertices $V' \subseteq V$
- Cut the rest

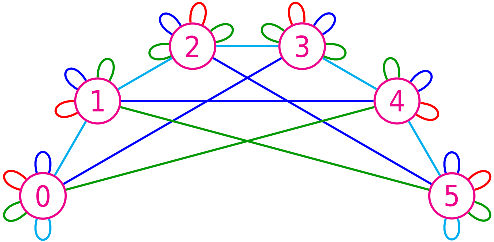


[DMR25] Hugo Delavenne, Tanguy Medevielle, and Élina Roussel. Interactive Oracle Proofs of Proximity to Codes on Graphs, 2025. **Accepted to ISIT 2025**

Cut-graph $\boldsymbol{\Gamma' = \mathbf{Cut}[\Gamma, V']}$:

- Choose vertices $V' \subseteq V$
- Cut the rest
- Enjoy your new graph

$$E_{\Gamma'}(v,\ell) = \begin{cases} E_\Gamma(v,\ell) & \text{if } E_\Gamma(v,\ell) \in V' \\ v & \text{otherwise} \end{cases}$$

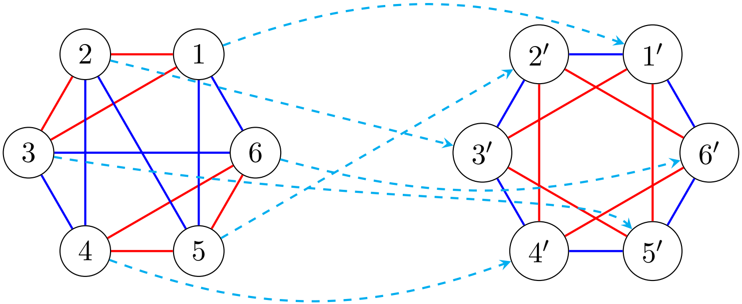


[DMR25] Hugo Delavenne, Tanguy Medevielle, and Élina Roussel. Interactive Oracle Proofs of Proximity to Codes on Graphs, 2025.
Accepted to ISIT 2025

Definition **Graph isomorphism**

A bijection $\varphi : V \to V'$ is an **isomorphism** $\Gamma \to \Gamma'$ if

$$\forall (v, \ell) \in V \times [n], \quad \varphi(E(v, \ell)) = E'(\varphi(v), \ell).$$

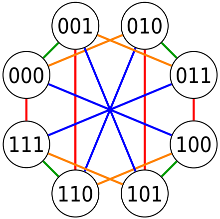


[DMR25] Hugo Delavenne, Tanguy Medevielle, and Élina Roussel. Interactive Oracle Proofs of Proximity to Codes on Graphs, 2025.
Accepted to ISIT 2025

Definition **Flowering cut collection**

Let $V_0, \dots, V_{m-1} \subseteq V$, $\boldsymbol{V = \bigcup_{i=0}^{m} V_i}$.

The cuts are **flowering** if $(\mathrm{Cut}[\Gamma, V_i])_{i=0,\dots,m-1}$ are **isomorphic**.

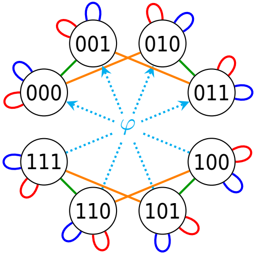


The **cut-word** $\mathrm{Cut}[f, V_i]$ is the restriction $f_{|V_i \times [n]}$.

Definition **Folding**

For $\alpha \in \mathbb{F}$, $(v, \ell) \in V_0 \times [n]$, and $\varphi_i : \mathrm{Cut}[\Gamma, V_i] \xrightarrow{\sim} \mathrm{Cut}[\Gamma, V_0]$

$$\mathrm{Fold}[f, \alpha](v, \ell) := \sum_{i=0}^{m-1} \alpha^i \mathrm{Cut}[f, V_i](\varphi_i(v), \ell)$$



[DMR25] Hugo Delavenne, Tanguy Medevielle, and Élina Roussel. Interactive Oracle Proofs of Proximity to Codes on Graphs, 2025. **Accepted to ISIT 2025**

Illustration: flowering cut collection

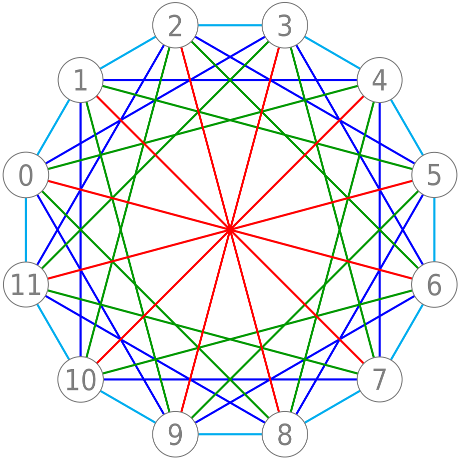
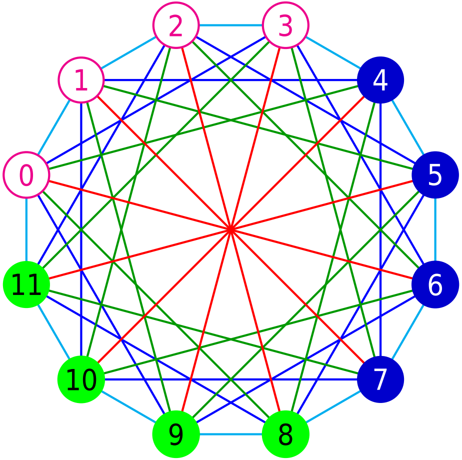
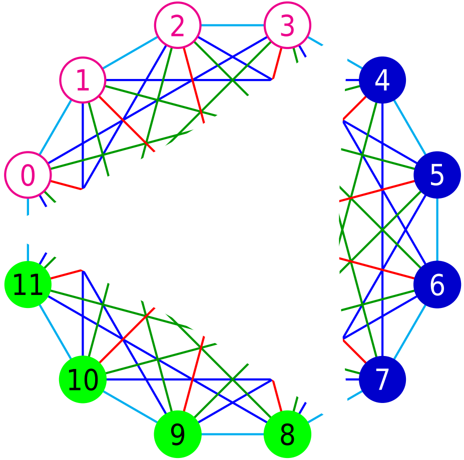


Illustration: flowering cut collection

- Choose 3 sets of vertices $V_0, V_1, V_2 \subseteq V$



- Choose 3 sets of vertices $V_0, V_1, V_2 \subseteq V$
- Cut outgoing edges



- Choose 3 sets of vertices $V_0, V_1, V_2 \subseteq V$
- Cut outgoing edges
- Get new graphs with petals

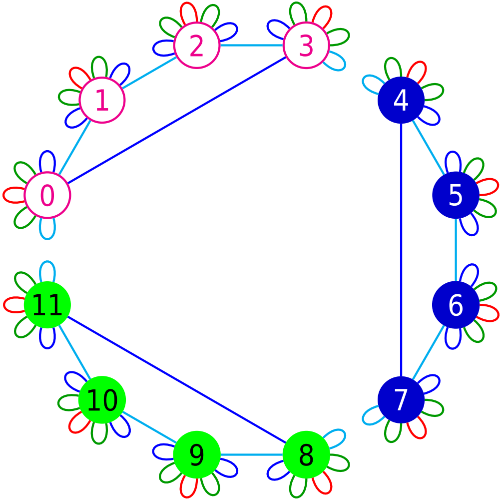
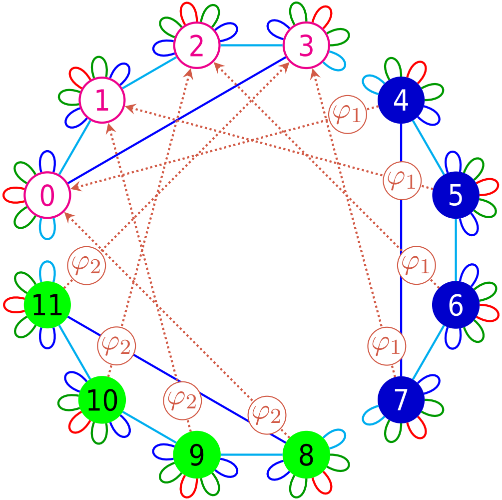
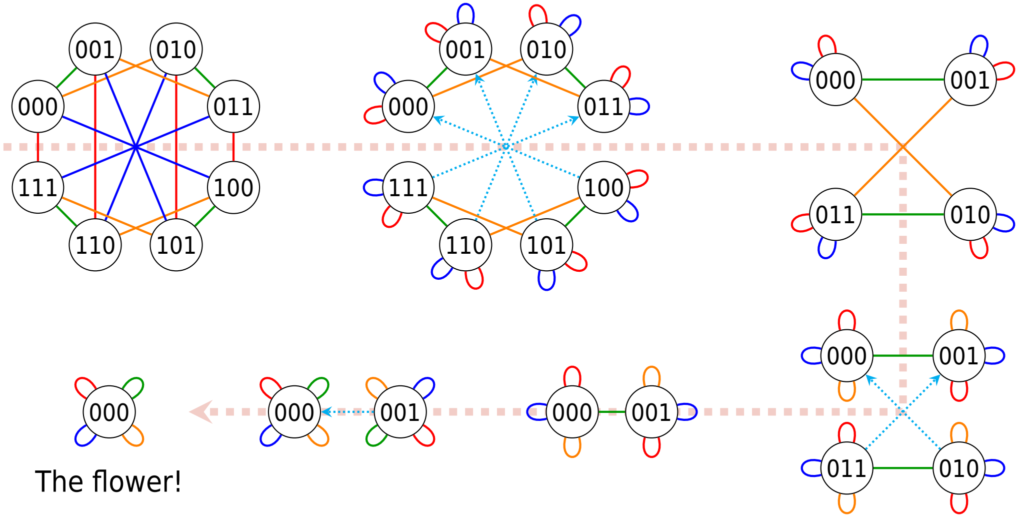


Illustration: flowering cut collection

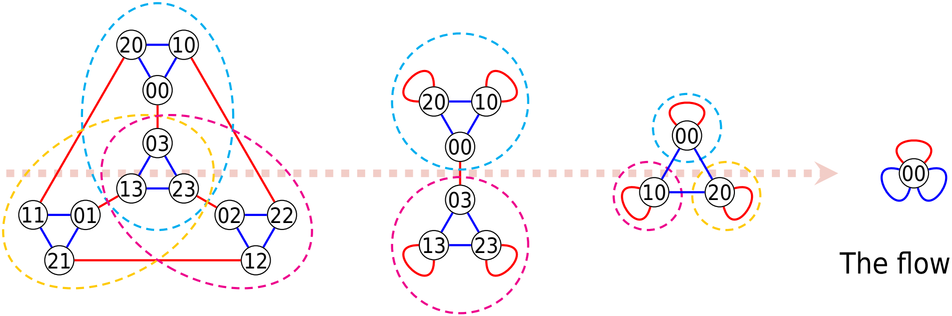
- ▷ Choose 3 sets of vertices $V_0, V_1, V_2 \subseteq V$
- ▷ Cut outgoing edges
- ▷ Get new graphs with petals
- ▶ Define the the isomorphisms

$$\varphi_1 : i \mapsto i - 4 \qquad \varphi_2 : i \mapsto i - 8$$




001
010
000
011
111
100
110
101
001
010
000
011
111
100
110
101
000
001
011
010
000
001
011
010
000
001
000
001
000

The flower!



The flower!

$$f_0 : V_0 \times [n] \to \mathbb{F}$$

$\mathcal{P}$ $\mathcal{V}$

$$\alpha_0 \xleftarrow{\$} \mathbb{F}$$

$$f_1 : V_1 \times [n] \to \mathbb{F}$$

$$f_1 = \mathsf{Fold}[f_0, \alpha_0]$$

$$\vdots$$

$$\alpha_{r-1} \xleftarrow{\$} \mathbb{F}$$

$$f_r : V_r \times [n] \to \mathbb{F}$$

$$f_r = \mathsf{Fold}[f_{r-1}, \alpha_{r-1}]$$

1 commit phase

$$v_0 \xleftarrow{\$} V_0,\ v_1 := \varphi_{1,i}(v_0)\ (v_0 \in V_{1,i})$$

$$f_1(v_1, \cdot) \stackrel{?}{=} \mathsf{Fold}[f_0, \alpha_0](v_1, \cdot)$$

$$\vdots$$

$$v_r := \varphi_{r,i}(v_{r-1})\ (v_{r-1} \in V_{r,i})$$

$$f_r(v_r, \cdot) \stackrel{?}{=} \mathsf{Fold}[f_{r-1}, \alpha_{r-1}](v_r, \cdot)$$

repeat m times

$$f_r \stackrel{?}{\in} \mathsf{RS}[n, k]$$

2 query phase

[DMR25] Hugo Delavenne, Tanguy Medevielle, and Élina Roussel. Interactive Oracle Proofs of Proximity to Codes on Graphs, 2025. **Accepted to ISIT 2025**

Flowering commit soundness

$f_0 : V_0 \times [n] \to \mathbb{F}$

$\mathcal{P}$ $\mathcal{V}$

$\alpha_0 \xleftarrow{\$} \mathbb{F}$

$f_1 : V_1 \times [n] \to \mathbb{F}$

$f_1 = \mathsf{Fold}[f_0, \alpha_0]$

$\vdots$

$\alpha_{r-1} \xleftarrow{\$} \mathbb{F}$

$f_r : V_r \times [n] \to \mathbb{F}$

$f_r = \mathsf{Fold}[f_{r-1}, \alpha_{r-1}]$

1 commit phase

A **commit error** is when

$$\Delta_V(\mathsf{Fold}[f_i, \alpha_i], C_{i+1}) < \Delta_V(f_i, C_i).$$

Proposition [DMR25]

It happens w.p. over α_i

$$\leqslant \frac{N}{|\mathbb{F}|}$$

[DMR25] Hugo Delavenne, Tanguy Medevielle, and Élina Roussel. Interactive Oracle Proofs of Proximity to Codes on Graphs, 2025. **Accepted to ISIT 2025**

A **query error** is when for some i

$$f_{i+1} \neq \mathsf{Fold}[f_i, \alpha_i]$$

but

$$f_{i+1}(v_{i+1}, \cdot) = \mathsf{Fold}[f_i, \alpha_i](v_{i+1}, \cdot)$$

Proposition [DMR25]

If no commit error, it happens w.p.

$$\leqslant (1-\delta)^m$$

$$v_0 \xleftarrow{\$} V_0,\ v_1 := \varphi_{1,i}(v_0)\ (v_0 \in V_{1,i})$$

$$f_1(v_1, \cdot) \stackrel{?}{=} \mathsf{Fold}[f_0, \alpha_0](v_1, \cdot)$$

$$\vdots$$

$$v_r := \varphi_{r,i}(v_{r-1})\ (v_{r-1} \in V_{r,i})$$

$$f_r(v_r, \cdot) \stackrel{?}{=} \mathsf{Fold}[f_{r-1}, \alpha_{r-1}](v_r, \cdot)$$

$$f_r \stackrel{?}{\in} \mathsf{RS}[n, k]$$

repeat m times

2 query phase

[DMR25] Hugo Delavenne, Tanguy Medevielle, and Élina Roussel. Interactive Oracle Proofs of Proximity to Codes on Graphs, 2025. **Accepted to ISIT 2025**

Proposition **Flowering completeness**

If $f \in \mathcal{C}[\Gamma, \mathsf{RS}[n,k]]$ then $\mathcal{V}$ accepts with probability 1.

Theorem **Flowering soundness**

If $\Delta(f, \mathcal{C}[\Gamma, \mathsf{RS}[n,k]]) > \delta$ then for any $\tilde{\mathcal{P}}$ and $\varepsilon > 0$, $\mathcal{V}$ accepts with probability

$$\leqslant \frac{Nr}{|\mathbb{F}|} + (1-\delta)^m$$

Recall FRI:

$$\frac{10^7 N^{3.5} \log K}{K^{1.5}|\mathbb{F}|} + \left(1 - \min\left(\delta, 1 - \frac{21}{20}\sqrt{K/N}\right)\right)^m$$

[DMR25] Hugo Delavenne, Tanguy Medevielle, and Élina Roussel. Interactive Oracle Proofs of Proximity to Codes on Graphs, 2025.
Accepted to ISIT 2025

Complexity comparison

For codes $[N, K]$: RS$[\mathcal{L}, K]$ or $\mathcal{C}[\Gamma, \text{RS}[n, k]]$

Protocol	FRI	[DMR25]	[DL25]
Prover	$< 8N$	$< 3N$	$< 5\kappa N \log N$
Verifier	$< 2m \log K$	$< 4mn \log N$	$< 8\kappa mn \log N$
Query	$< 2m \log K$	$2mn \log N$	$3\kappa mn \log N$
Rounds	$\log K$	$< \log N$	$< \kappa \log N$

Complexity depends on the **graphs used** and the **cutting strategy**.

[DMR25] Hugo Delavenne, Tanguy Medevielle, and Élina Roussel. Interactive Oracle Proofs of Proximity to Codes on Graphs, 2025.
Accepted to ISIT 2025

[DL25] Hugo Delavenne and Louise Lallemand. Codes on any Cayley graph have an Interactive Oracle Proof of Proximity, 2025.
Submitted

1 Interactive Oracle Proofs of Proximity

2 Flowering protocol

3 Flowering graphs

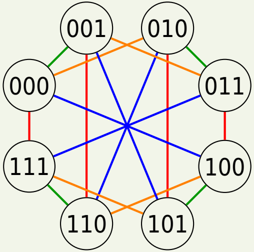
- First graphs [DMR25]
- Expander graphs [DL25]
- Current work

Definition **Cayley graph Cay(G, S)**

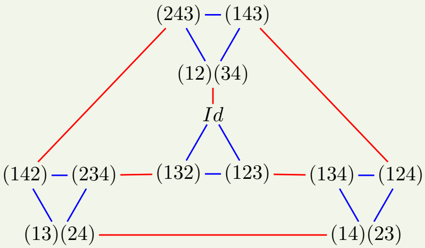
Let $(G, \cdot)$ and $S \subseteq G$ such that $\langle S \rangle = G$ and $S^{-1} = S$.
Define **Cay(G, S)** $= (G, E)$ by $E(g, s) = g \cdot s$.

Example

$G = (\mathbb{F}_2^3, +)$, $S = \{100, 010, 001, 111\}$

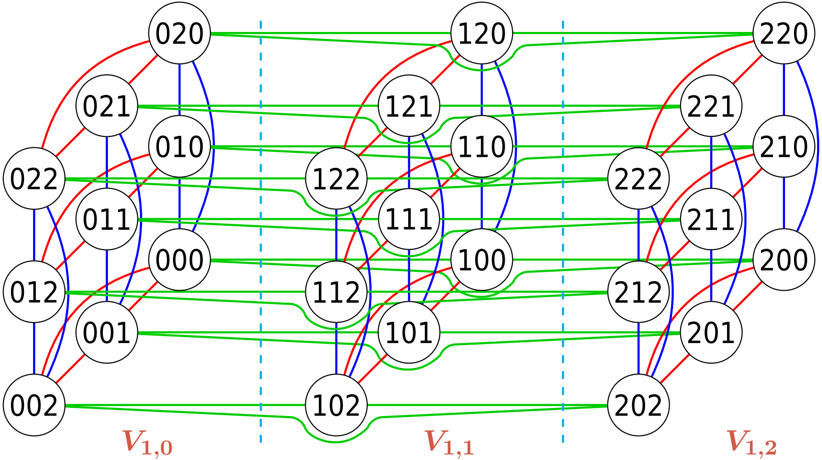


$G = \mathcal{A}_4$, $S = \{(123), (132), (12)(34)\}$



[Cay78] Arthur Cayley. Desiderata and Suggestions: No. 2. The Theory of Groups: Graphical Representation. ***American Journal of Mathematics*, 1(2):174–176, 1878**

We take $G = (\mathbb{F}_m^r, +)$ and $S \subseteq G$ of size n. The cuts are $V_{i,j} = \{j\}^i \times \mathbb{F}_m^{r-i}$



Let $G = (\mathbb{F}_2^r, +)$.

Using S the columns of a parity check matrix of a $[n, n-r, d]_2$ binary code

Proposition **Parameters of the code**

- $N = n2^{r-1}$
- $\text{rate}\, C \geqslant \frac{2k}{n} - 1$
- $\frac{1}{2}\delta \leqslant \Delta(C) \leqslant \delta$, with $\delta = \frac{1}{2^{r-d+1}}\left(1 - \frac{k-1}{n}\right) = \frac{n2^{d-2}}{N}\left(1 - \frac{k-1}{n}\right)$

If $d \ll r$ (i.e. far from MDS), δ is **terrible** ($O(1/N)$).

[DMR25] Hugo Delavenne, Tanguy Medevielle, and Élina Roussel. Interactive Oracle Proofs of Proximity to Codes on Graphs, 2025. **Accepted to ISIT 2025**

Definition **Graph expansion**

Let $\Gamma = (V, E)$ a n-regular graph and $A \in \{0,1\}^{|V|\times|V|}$ its adjacency matrix.
Let $\Lambda_1 \geqslant \Lambda_2 \geqslant \ldots \geqslant \Lambda_n \in \mathbb{R}$ be the eigenvalues of A.

Then Γ is **λ-expander** if $|\Lambda_i| \leqslant n\lambda$ for $i \geqslant 2$.

Lemma **Minimal distance expansion lower bound [AC88]**

If Γ is λ-expander, with $\delta = 1 - \frac{k+1}{n}$, $\mathcal{C}[\Gamma, \mathsf{RS}[n,k]]$ has minimal distance $\geqslant \delta(\delta - \lambda)$.

Thus if $(\Gamma_i)_{i\in\mathbb{N}}$ has **constant expansion**,

then $(\mathcal{C}[\Gamma_i, \mathsf{RS}[n_i, \rho n_i]])_{i\in\mathbb{N}}$ has **constant minimal distance**.

[AC88] Noga Alon and Fan Chung. Explicit construction of linear sized tolerant networks.
***Discrete Mathematics*, 72(1-3):15–19, 1988**

Definition **Graph expansion**

Let $\Gamma = (V, E)$ a n-regular graph and $A \in \{0,1\}^{|V|\times|V|}$ its adjacency matrix.
Let $\Lambda_1 \geqslant \Lambda_2 \geqslant \ldots \geqslant \Lambda_n \in \mathbb{R}$ be the eigenvalues of A.

Then Γ is **λ-expander** if $|\Lambda_i| \leqslant n\lambda$ for $i \geqslant 2$.

$\lambda \in [0, 1]$ characterizes random walk propagation in Γ.

Small λ means good expansion.

Lemma **Minimal distance expansion lower bound [AC88]**

If Γ is λ-expander, with $\delta = 1 - \frac{k+1}{n}$, $\mathcal{C}[\Gamma, \mathsf{RS}[n, k]]$ has minimal distance $\geqslant \delta(\delta - \lambda)$.

Thus if $(\Gamma_i)_{i\in\mathbb{N}}$ has **constant expansion**,

then $(\mathcal{C}[\Gamma_i, \mathsf{RS}[n_i, \rho n_i]])_{i\in\mathbb{N}}$ has **constant minimal distance**.

[AC88] Noga Alon and Fan Chung. Explicit construction of linear sized tolerant networks.
***Discrete Mathematics*, 72(1-3):15–19, 1988**

Example of expander graphs

Let $p \neq q$ be primes such that $p \equiv 1 \mod 4$ and $q \equiv 1 \mod 4$. Let

$$G_{p,q} := \begin{cases} \mathrm{PSL}_2(\mathbb{F}_q) & \text{if } p \text{ is a quadratic residue mod } q \\ \mathrm{PGL}_2(\mathbb{F}_q) & \text{otherwise} \end{cases}$$

There is a generating set $S_{p,q}$ of size $p+1$ such that $(\text{Cay}(G_{p,q}, S_{p,q}))_q$ is Ramanujan (optimal expander).

Let $S = \{s_0, \ldots, s_{n-1}\}$ and $G = \langle S \rangle$. Write $\tilde{s}_i := s_{i \bmod \tilde{n}}$.

Proposition

Let $\Gamma = \text{Cay}[G, S]$. There exists $r \leqslant n \operatorname{diam}(\Gamma)$ such that

$$G = \{\tilde{s}_0^{\alpha_0} \cdot \cdots \cdot \tilde{s}_{r-1}^{\alpha_{r-1}} \mid \alpha_0, \ldots, \alpha_{r-1} \in \{0, 1\}\}$$

At round i there are two cuts for $j \in \{0, 1\}$

$$V_{i,j} := \left\{\tilde{s}_i^j \cdot \tilde{s}_{i+1}^{\alpha_{i+1}} \cdots \tilde{s}_{r-1}^{\alpha_{r-1}} \mid \alpha_{i+1}, \ldots, \alpha_k \in \{0, 1\}\right\}$$

- $\boldsymbol{\Gamma_{i,1} \cong \Gamma_{i,0}}$ with $\varphi_i(g) = \tilde{s}_i^{-1} \cdot g$
- $\Gamma_{r,0}$ is a flower so there are $\boldsymbol{r}$ **rounds**

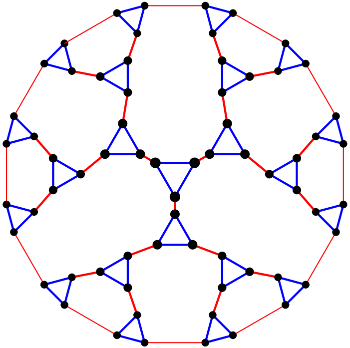
[DL25] Hugo Delavenne and Louise Lallemand. Codes on any Cayley graph have an Interactive Oracle Proof of Proximity, 2025. **Submitted**

Example

Let $S = \{b, b^{-1}, r\}$.

Assume that $G = \langle S \rangle$ is

$$G = \{b^{-1,0,1} r^{0,1} b^{-1,0,1} r^{0,1} b^{-1,0,1} r^{0,1} b^{-1,0,1}\}$$



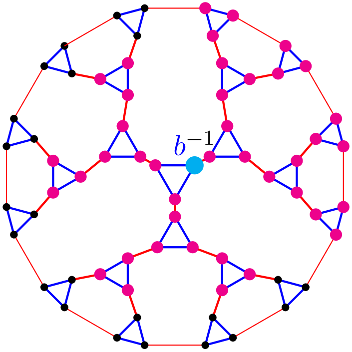
Let $S = \{b, b^{-1}, r\}$.

Assume that $G = \langle S \rangle$ is

$$G = \{b^{-1,0,1} r^{0,1} b^{-1,0,1} r^{0,1} b^{-1,0,1} r^{0,1} b^{-1,0,1}\}$$

The first cuts are

$$\boldsymbol{V_{0,-1}} = \{b^{-1} \cdot r^{0,1} b^{-1,0,1} r^{0,1} b^{-1,0,1} r^{0,1} b^{-1,0,1}\}$$



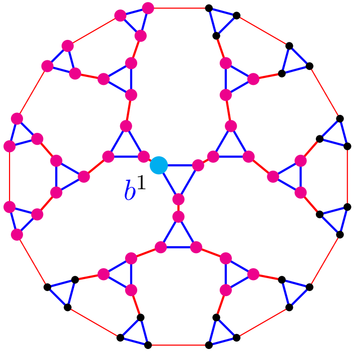
Let $S = \{b, b^{-1}, r\}$.

Assume that $G = \langle S \rangle$ is

$$G = \{b^{-1,0,1} r^{0,1} b^{-1,0,1} r^{0,1} b^{-1,0,1} r^{0,1} b^{-1,0,1}\}$$

The first cuts are

$$\boldsymbol{V_{0,1}} = \{b^1 \cdot r^{0,1} b^{-1,0,1} r^{0,1} b^{-1,0,1} r^{0,1} b^{-1,0,1}\}$$



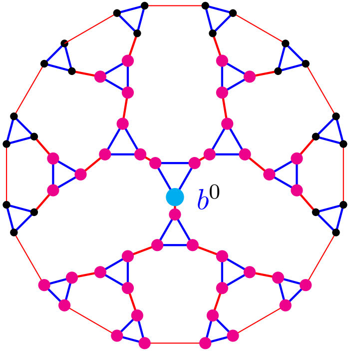
Let $S = \{b, b^{-1}, r\}$.

Assume that $G = \langle S \rangle$ is

$$G = \{b^{-1,0,1} r^{0,1} b^{-1,0,1} r^{0,1} b^{-1,0,1} r^{0,1} b^{-1,0,1}\}$$

The first cuts are

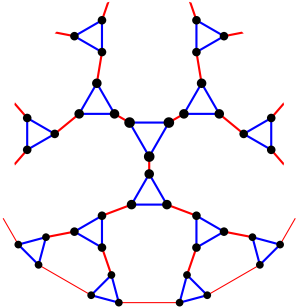
$$\boldsymbol{V_{0,0}} = \{b^0 \cdot r^{0,1} b^{-1,0,1} r^{0,1} b^{-1,0,1} r^{0,1} b^{-1,0,1}\}$$



Let $S = \{b, b^{-1}, r\}$.

Assume that $G = \langle S \rangle$ is

$$G = \{b^{-1,0,1} r^{0,1} b^{-1,0,1} r^{0,1} b^{-1,0,1} r^{0,1} b^{-1,0,1}\}$$

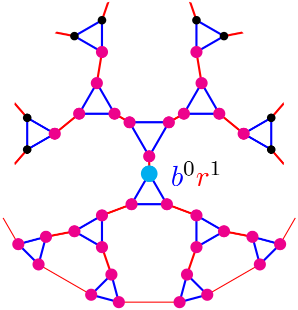


Let $S = \{b, b^{-1}, r\}$.
Assume that $G = \langle S \rangle$ is

$$G = \{b^{-1,0,1} r^{0,1} b^{-1,0,1} r^{0,1} b^{-1,0,1} r^{0,1} b^{-1,0,1}\}$$

The second cuts are

$$\boldsymbol{V_{1,1}} = \{b^0 r^1 \cdot b^{-1,0,1} r^{0,1} b^{-1,0,1} r^{0,1} b^{-1,0,1}\}$$



Let $S = \{b, b^{-1}, r\}$.

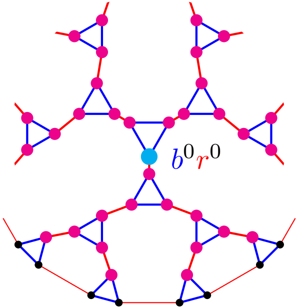
Assume that $G = \langle S \rangle$ is

$$G = \{b^{-1,0,1} r^{0,1} b^{-1,0,1} r^{0,1} b^{-1,0,1} r^{0,1} b^{-1,0,1}\}$$

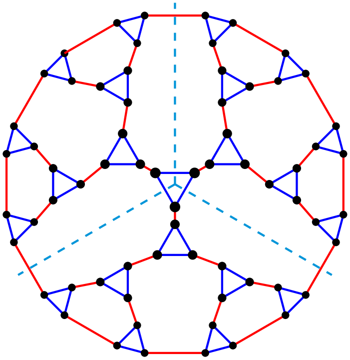
The second cuts are

$$V_{1,0} = \{b^0 r^0 \cdot b^{-1,0,1} r^{0,1} b^{-1,0,1} r^{0,1} b^{-1,0,1}\}$$

The cuts **overlap** a lot!

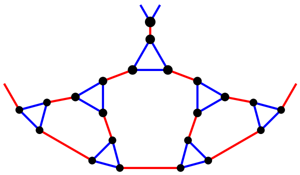


- This cut is defined with shortest paths.



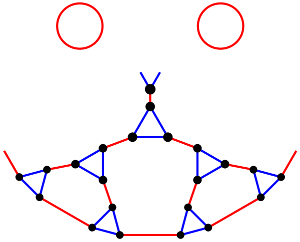
Why can’t we do the obvious cut?

- This cut is defined with shortest paths.
- Not suitable because
  - We lose **symmetries** and **algebraic properties**.
  - We still **can’t bound** the size of cuts.



Why can't we do the obvious cut?

- This cut is defined with shortest paths.
- Not suitable because
  - We lose **symmetries** and **algebraic properties**.
  - We still **can't bound** the size of cuts.
- Here we can still define new cuts :-)



Let $\Gamma = (V, E)$ and $v \in V$. Define the **ball** $B(v,d) := \{v' \in V \mid d_\Gamma(v',v) \leqslant d\}$.

Definition **Distance-d dominating set**

$M \subseteq V$ is **distance-d dominating** if $V = \bigcup_{v \in M} B(v,d)$.

Finding a small such set is $\mathbf{NP}$-hard... but expansion gives an **existence bound**!

Proposition

Assume Γ is Ramanujan of regularity n.
Then there is a a **distance-d dominating set** of size

$$\leqslant \frac{|V| \log |V|}{\left(\frac{3}{2} - \frac{1}{\sqrt{n}}\right)^d}$$

Let $\Gamma = \text{Cay}[G, S]$. Then

$$B(g,d) = \{g \cdot s_0 \cdots s_{d-1} \mid s_0, \ldots, s_{d-1} \in S \cup \{1_G\}\}$$

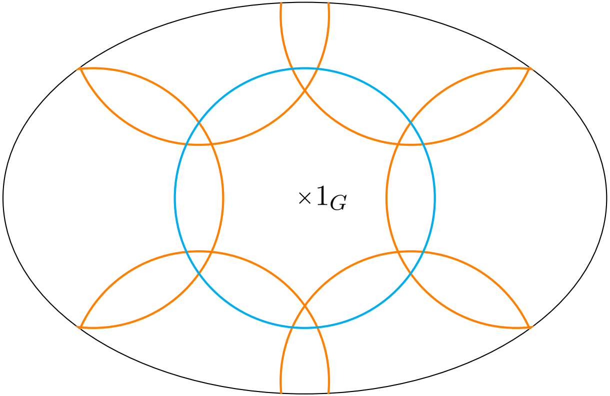
So $\boldsymbol{B(g_0, d) \cong B(g_1, d)}$ with $\varphi : g \mapsto g_1 g_0^{-1} \cdot g$.

Let $\Gamma = \mathsf{Cay}[G, S]$. Then

$$B(g, d) = \{g \cdot s_0 \cdots s_{d-1} \mid s_0, \ldots, s_{d-1} \in S \cup \{1_G\}\}$$

So $\boldsymbol{B(g_0, d) \cong B(g_1, d)}$ with $\varphi : g \mapsto g_1 g_0^{-1} \cdot g$.

- Balls still lose properties

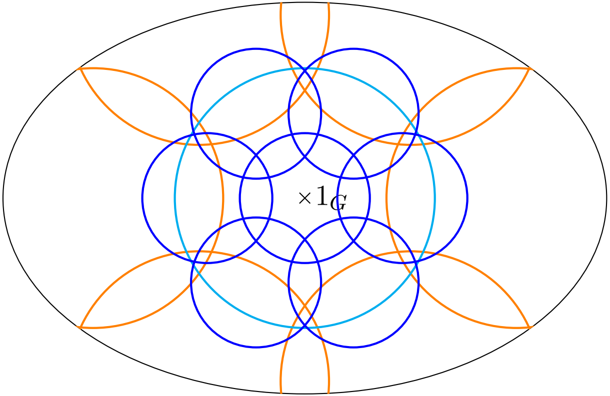


Let $\Gamma = \mathsf{Cay}[G, S]$. Then

$$B(g, d) = \{g \cdot s_0 \cdots s_{d-1} \mid s_0, ..., s_{d-1} \in S \cup \{1_G\}\}$$

So $\boldsymbol{B(g_0, d) \cong B(g_1, d)}$ with $\varphi : g \mapsto g_1 g_0^{-1} \cdot g$.

- Balls still lose properties
- Cover balls with smaller balls

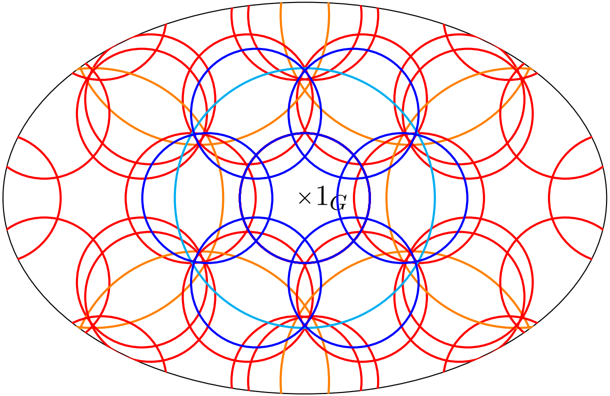


Let $\Gamma = \mathrm{Cay}[G, S]$. Then

$$B(g, d) = \{g \cdot s_0 \cdots s_{d-1} \mid s_0, \ldots, s_{d-1} \in S \cup \{1_G\}\}$$

So $\boldsymbol{B(g_0, d) \cong B(g_1, d)}$ with $\varphi : g \mapsto g_1 g_0^{-1} \cdot g$.

- Balls still lose properties
- Cover balls with smaller balls
- Looks like a mess



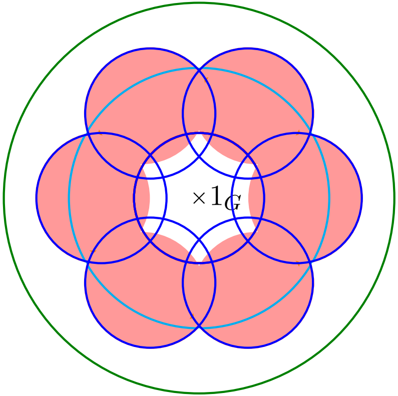
Let $\Gamma = \text{Cay}[G, S]$. Then

$$B(g, d) = \{g \cdot s_0 \cdots s_{d-1} \mid s_0, ..., s_{d-1} \in S \cup \{1_G\}\}$$

So $B(g_0, d) \cong B(g_1, d)$ with $\varphi : g \mapsto g_1 g_0^{-1} \cdot g$.

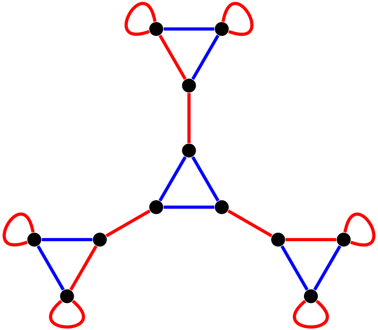
- Balls still lose properties
- Cover balls with smaller balls
- Looks like a mess
- ...but we can control overlap:

$$B(g, d_1) \subseteq \bigcup_i B(g_i, d_2) \subseteq B(g, d_1 + d_2)$$



Modify graphs to **get rid of good minimal distance** and obtain **linear encoding**.

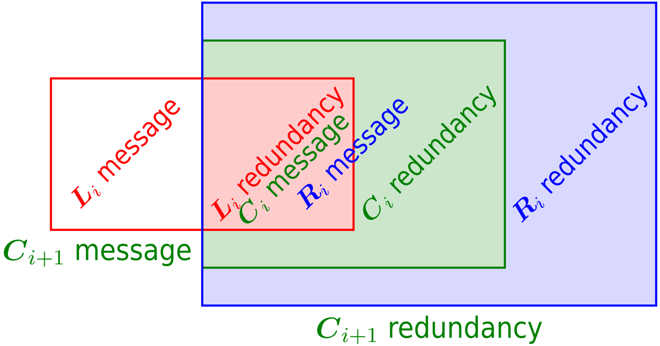
Here $\mathcal{C}[\Gamma, \mathbf{RS}[3, 2]]$ is linearly encodable



Message is in **blue**. Redundancy is in **red**.

[LM09] Jin Lu and José MF Moura. Linear time encoding of LDPC codes.
IEEE Transactions on Information Theory**, 56(1):233–249, 2009**

Let $(\boldsymbol{L}_i)_{i\in\mathbb{N}}$ and $(\boldsymbol{R}_i)_{i\in\mathbb{N}}$ be **systematic linear-time encodable** codes.
Build $\boldsymbol{C}_{i+1}$ to be **linear-time encodable** with good minimal distance:



Question: Can we test proximity on $\boldsymbol{L}_i$'s and $\boldsymbol{R}_i$'s to get proximity on $\boldsymbol{C}_i$'s?

[Spi95] Daniel A Spielman. Linear-time encodable and decodable error-correcting codes.
In *Proceedings of the twenty-seventh annual ACM symposium on Theory of computing*, pages 388–397, 1995

Competing parameters with FRI

- Better soundness
- Less constraint on the field
- Complexity to be improved

Make this actually useful

- Arithmetize circuits to graphs: colored De Bruijn graphs
- Adapt Spielman's construction to have a linear time encoding