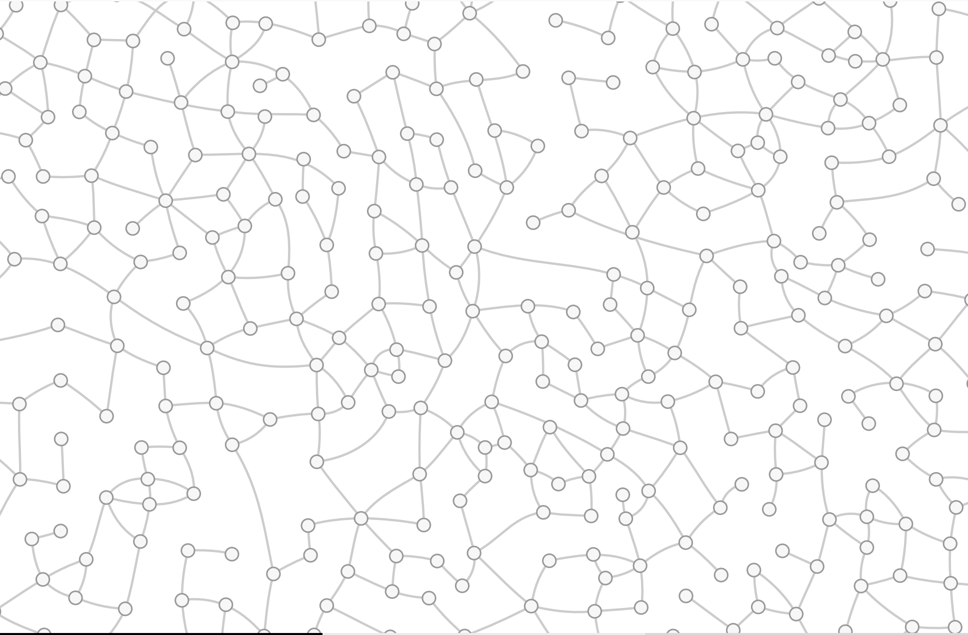
Lower Bounds *for* Local Approximation

Mika Göös, Juho Hirvonen & Jukka Suomela *(HIIT)*

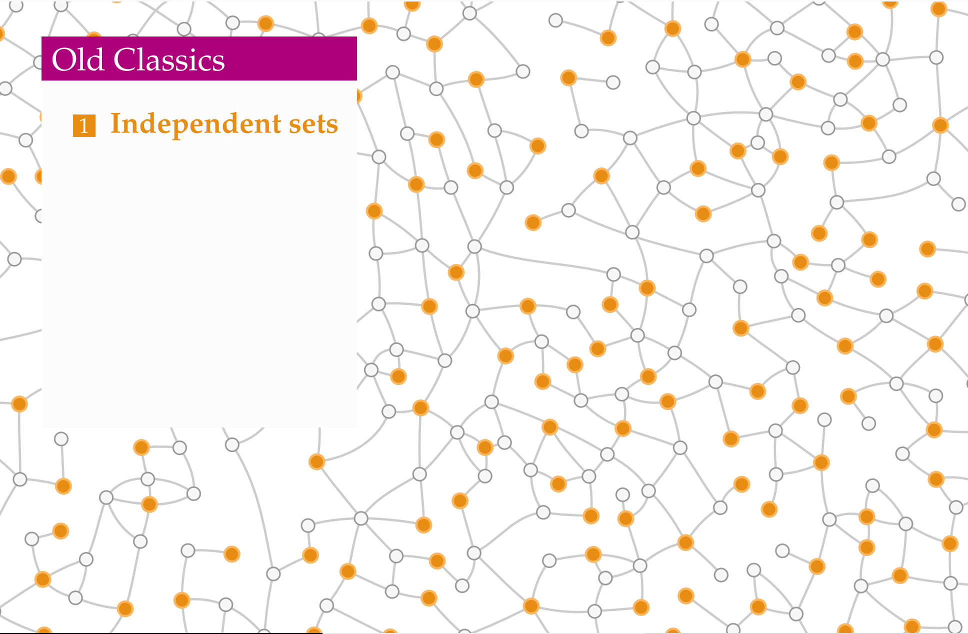
Lower Bounds *for* Local Approximation

We prove: **Local algorithms** do not need ***unique IDs*** when computing **approximations** to graph optimization problems

Input = Graph $\mathcal{G}$ = Communication Network



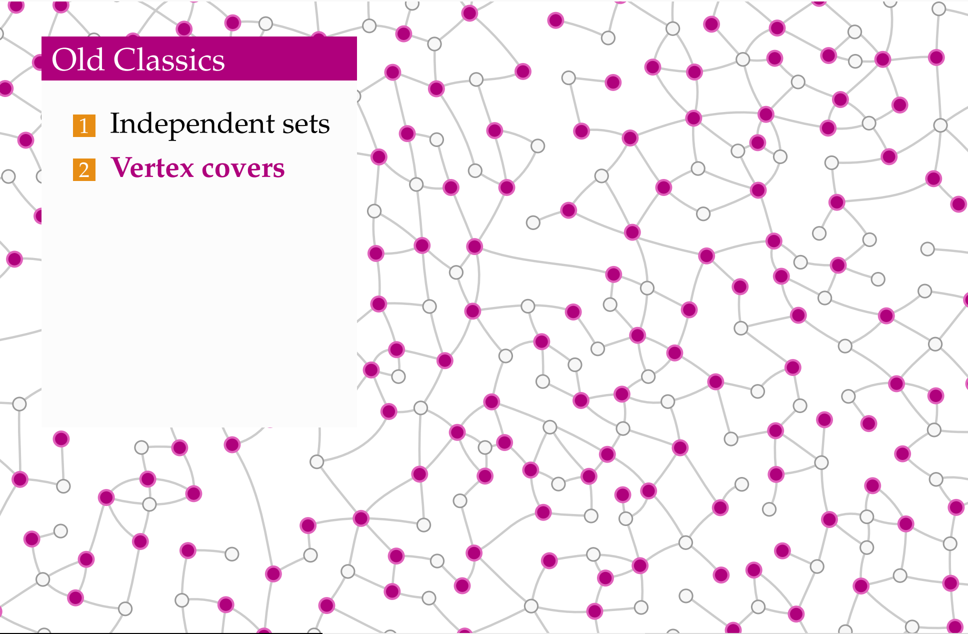
“Simple” Graph Problems



Old Classics

1. **Independent sets**

“Simple” Graph Problems



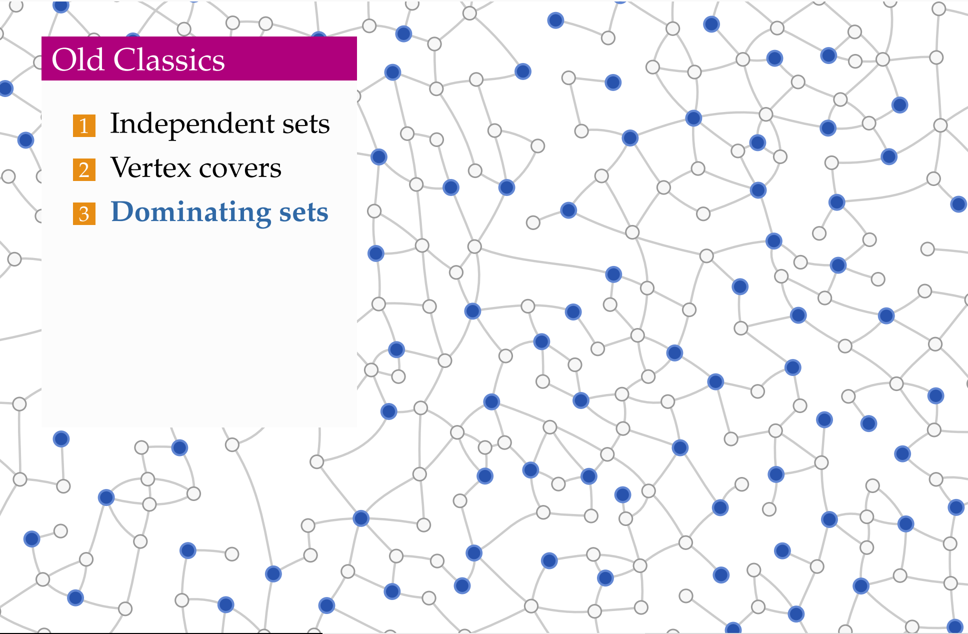
Old Classics

1. Independent sets
2. **Vertex covers**

“Simple” Graph Problems

Old Classics

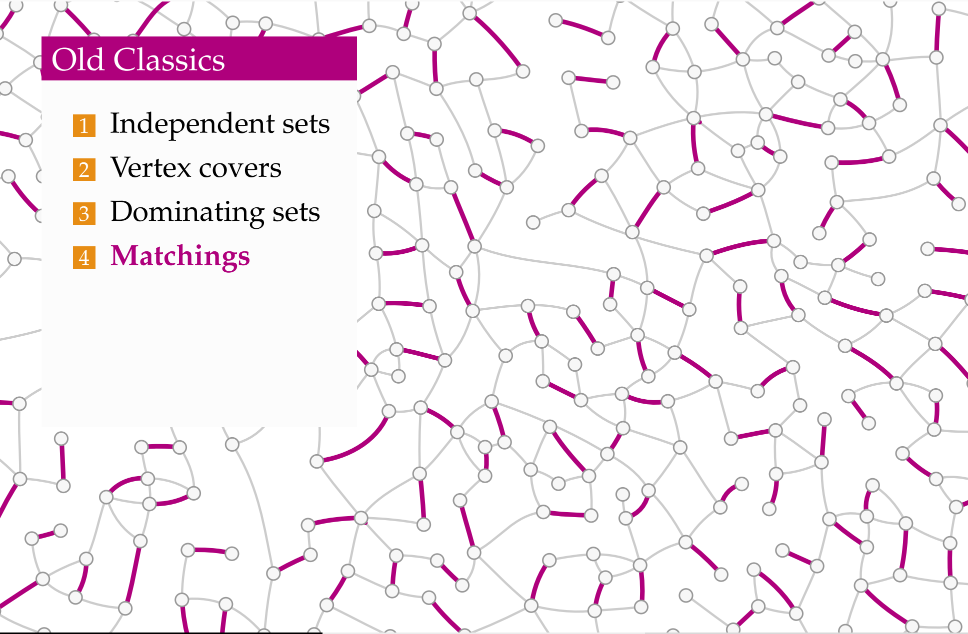
1. Independent sets
2. Vertex covers
3. **Dominating sets**



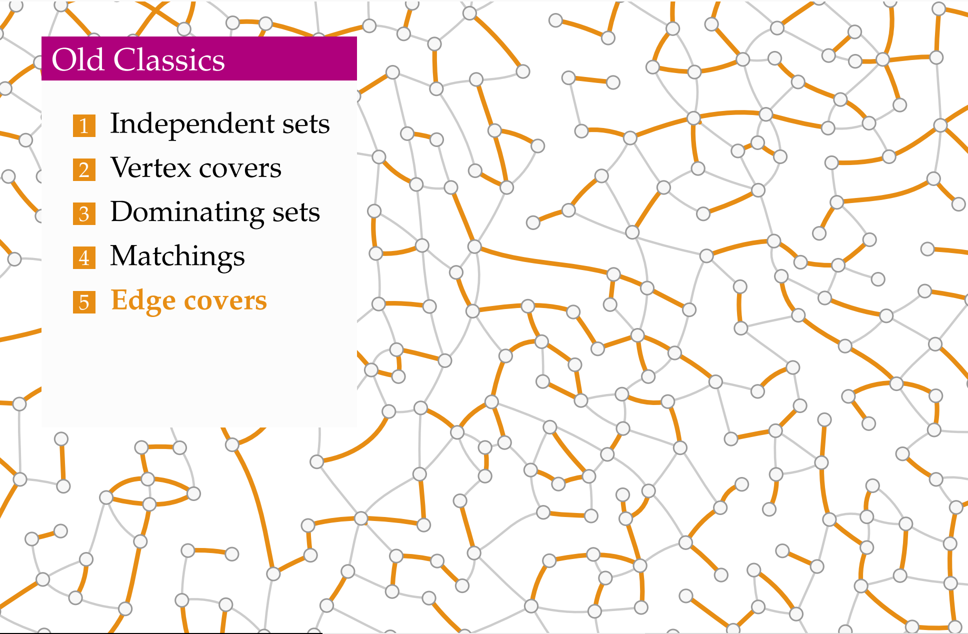
“Simple” Graph Problems

Old Classics

1. Independent sets
2. Vertex covers
3. Dominating sets
4. **Matchings**



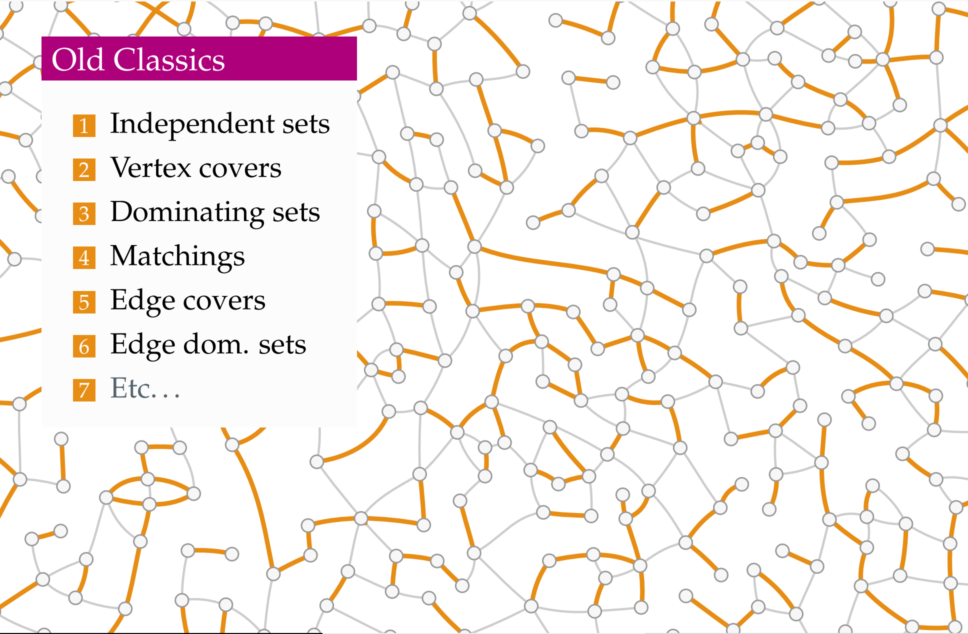
"Simple" Graph Problems



Old Classics

1. Independent sets
2. Vertex covers
3. Dominating sets
4. Matchings
5. **Edge covers**

“Simple” Graph Problems



Old Classics

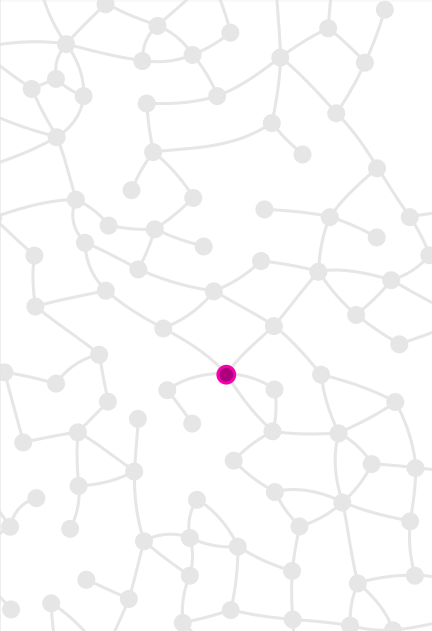
1. Independent sets
2. Vertex covers
3. Dominating sets
4. Matchings
5. Edge covers
6. Edge dom. sets
7. Etc. . .

Local Algorithms

1. Distributed algorithm **A**
2. Deterministic, synchronous

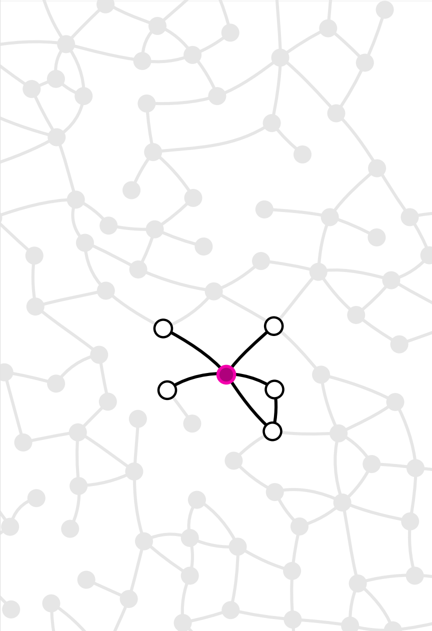
Local Algorithms

1. Distributed algorithm **A**
2. Deterministic, synchronous



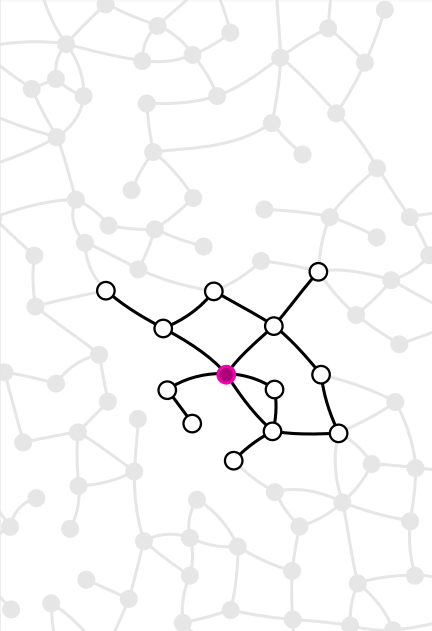
Local Algorithms

1. Distributed algorithm **A**
2. Deterministic, synchronous



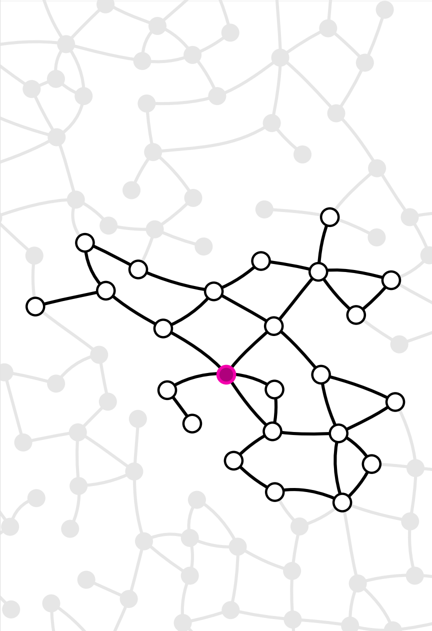
Local Algorithms

1. Distributed algorithm **A**
2. Deterministic, synchronous



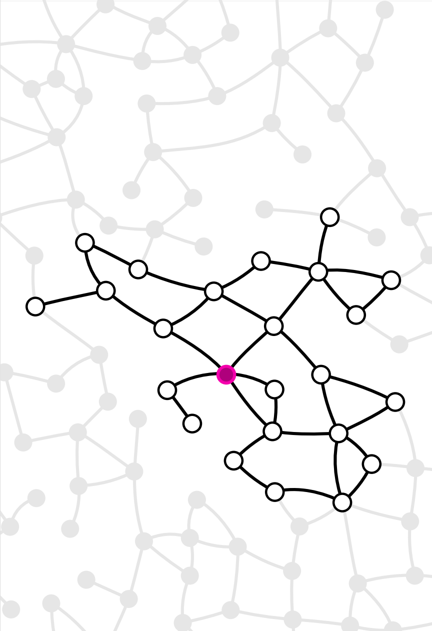
Local Algorithms

1. Distributed algorithm **A**
2. Deterministic, synchronous



Local Algorithms

1. Distributed algorithm **A**
2. Deterministic, synchronous
3. **Locality:** running time r of **A** is
   - **independent** of $n = |\mathcal{G}|$
   - may depend on maximum degree Δ of $\mathcal{G}$

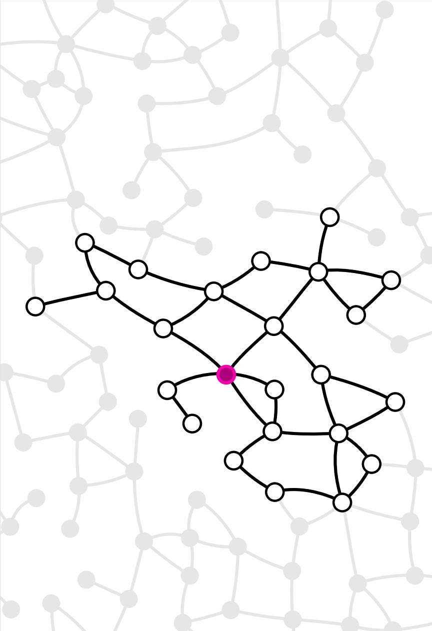


Local Algorithms

1. Distributed algorithm **A**
2. Deterministic, synchronous
3. **Locality:** running time r of **A** is
   - **independent** of $n = |\mathcal{G}|$
   - may depend on maximum degree Δ of $\mathcal{G}$

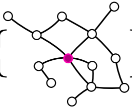
On ***bounded degree graphs*** ($\Delta = O(1)$) running time is a constant:

$$r \in \mathbb{N} \qquad \text{(e.g., } r = \mathbf{3}\text{)}$$



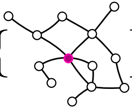
Local Algorithms

Definition:

$\mathbf{A} : \{$  $\} \to \{0, 1\}$

Local Algorithms

Definition:

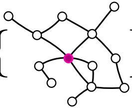
$\mathbf{A} : \{$  $\} \to \{0, 1\}$

Output:

Vertex set: Set of vertices v with $\mathbf{A}(\mathcal{G}, v) = 1$

Local Algorithms

Definition:

$\mathbf{A} : \{$  $\} \to \{0,1\}^{\Delta}$

Output:

Vertex set: Set of vertices v with $\mathbf{A}(\mathcal{G}, v) = 1$

Edge set: $\mathbf{A}(\mathcal{G}, v)$ is a vector of length Δ indicating which edges incident to v are included

Two Network Models

Unique Identifiers

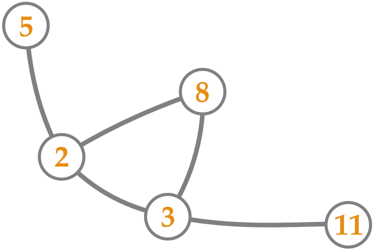
Anonymous Networks
with Port Numbering

Two Network Models

Unique Identifiers

- Each node has a unique $O(\log n)$-bit **label**:

$$V(\mathcal{G}) \subseteq \{1, 2, \ldots, \mathrm{poly}(n)\}$$



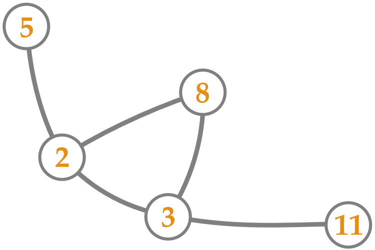
Anonymous Networks
with Port Numbering

Two Network Models

Unique Identifiers

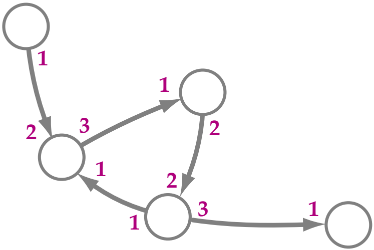
- Each node has a unique $O(\log n)$-bit **label**:

$$V(\mathcal{G}) \subseteq \{1, 2, \dots, \mathrm{poly}(n)\}$$



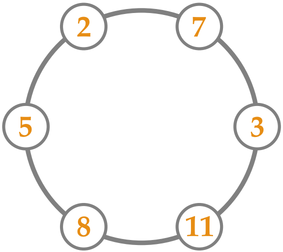
Anonymous Networks
with Port Numbering

- Node v can refer to its neighbours via **ports** $1, 2, \dots, \deg(v)$
- Edges are **oriented**

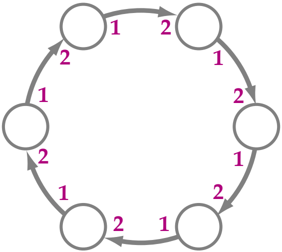


Example: Independent Sets on a Cycle

ID-model

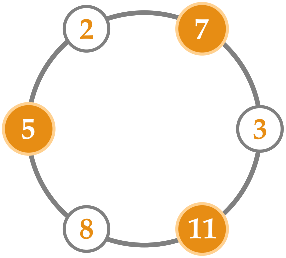


PO-model



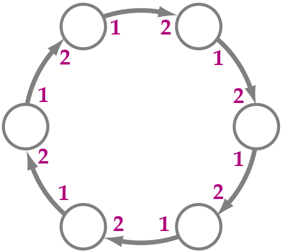
Example: Independent Sets on a Cycle

ID-model



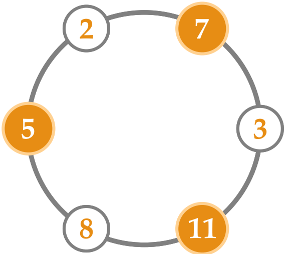
- [Cole–Vishkin 86, Linial 92]:
  Maximal independent set
  can be computed in $\Theta(\log^* n)$
  rounds

PO-model



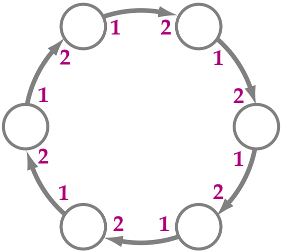
Example: Independent Sets on a Cycle

ID-model



- [Cole–Vishkin 86, Linial 92]: **Maximal independent set** can be computed in $\Theta(\log^* n)$ rounds

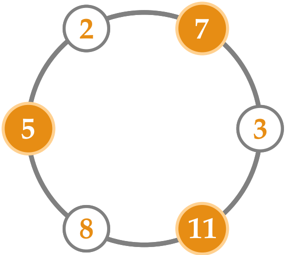
PO-model



- Above **PO**-network is **fully symmetric**

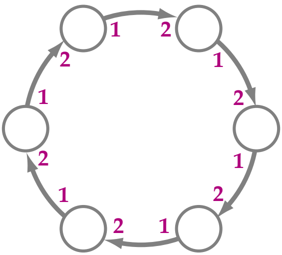
Example: Independent Sets on a Cycle

ID-model



- [Cole–Vishkin 86, Linial 92]: **Maximal independent set** can be computed in $\Theta(\log^* n)$ rounds

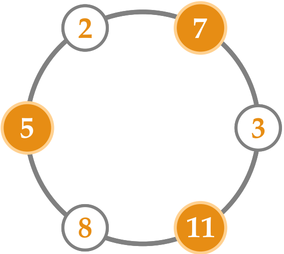
PO-model



- Above **PO**-network is **fully symmetric**
- $\Rightarrow$ All nodes give same output

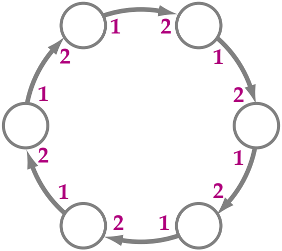
Example: Independent Sets on a Cycle

ID-model



- [Cole–Vishkin 86, Linial 92]: **Maximal independent set** can be computed in $\Theta(\log^* n)$ rounds

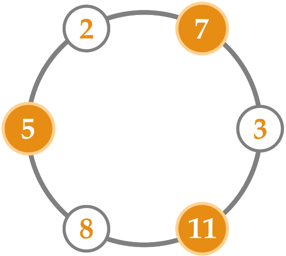
PO-model

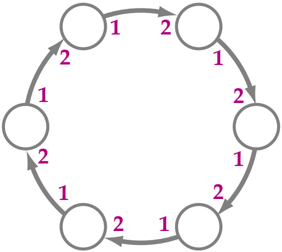


- Above **PO**-network is **fully symmetric**
- $\Rightarrow$ All nodes give same output
- $\Rightarrow$ Empty set is computed!

Example: Independent Sets on a Cycle

ID-model $\gg$ **in general** **PO-model**



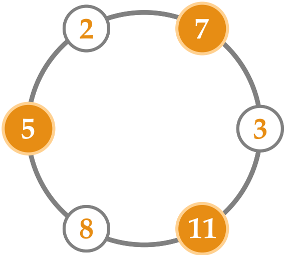


- [Cole–Vishkin 86, Linial 92]: **Maximal independent set** can be computed in $\Theta(\log^* n)$ rounds

- Above **PO**-network is **fully symmetric**
- $\Rightarrow$ All nodes give same output
- $\Rightarrow$ Empty set is computed!

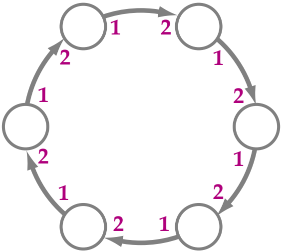
Example: Independent Sets on a Cycle

ID-model



- [Lenzen–Wattenhofer 08, Czygrinow et al. 08]: MIS **cannot** be approximated to within a **constant factor** in $O(1)$ rounds!

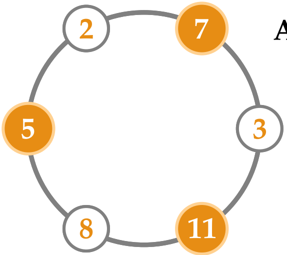
PO-model



- Above **PO**-network is **fully symmetric**
- $\Rightarrow$ All nodes give same output
- $\Rightarrow$ Empty set is computed!

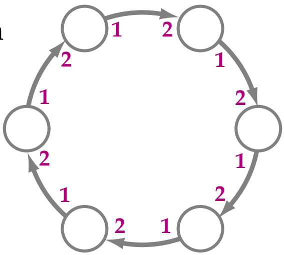
Example: Independent Sets on a Cycle

ID-model



$\approx$ **for Local Approximation**

PO-model



- [Lenzen–Wattenhofer 08, Czygrinow et al. 08]: MIS **cannot** be approximated to within a **constant factor** in $O(1)$ rounds!

- Above **PO**-network is **fully symmetric**
- $\Rightarrow$ All nodes give same output
- $\Rightarrow$ Empty set is computed!

Known Approximation Ratios

	ID	PO
Max **Independent Set**	∞	∞
Max **Matching**	∞	∞
Min **Vertex Cover**	2	2
Min **Edge Cover**	2	2
Min **Dominating Set**	$\Delta' + 1$	$\Delta' + 1$
	$(\Delta' := 2\lfloor \Delta/2 \rfloor)$	

Known Approximation Ratios

	ID	PO
Max **Independent Set**	∞	∞
Max **Matching**	∞	∞
Min **Vertex Cover**	2	2
Min **Edge Cover**	2	2
Min **Dominating Set**	$\Delta' + 1$	$\Delta' + 1$
	$(\Delta' := 2\lfloor \Delta/2 \rfloor)$	
Min **Edge Dominating Set**	α	$4 - 2/\Delta'$

$$3 \le \alpha \le 4 - 2/\Delta' \quad ???$$

Our Result (informally)

Main Thm: When **Local Algorithms** compute **constant factor** approximations,

$$\mathbf{ID} = \mathbf{PO}$$

for a general class of graph problems

Proof!

OI: Order Invariant Algorithms

[Naor–Stockmeyer 95]:

Ramsey's theorem implies that **local ID-algorithms** can only **compare** identifiers

$$\mathbf{ID} = \mathbf{OI} \stackrel{?}{=} \mathbf{PO}$$

OI: Order Invariant Algorithms

[Naor–Stockmeyer 95]:

Ramsey's theorem implies that **local ID-algorithms** can only **compare** identifiers

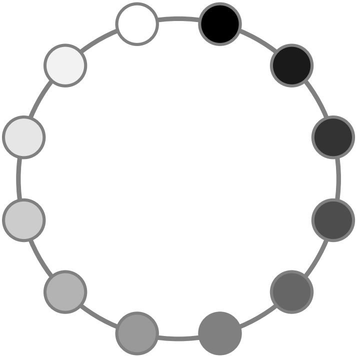
$$\mathbf{ID} = \mathbf{OI} \stackrel{?}{=} \mathbf{PO}$$

Order invariant algorithm A:

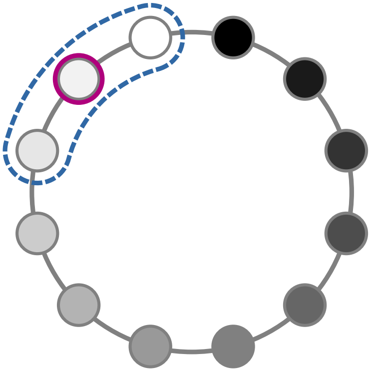
Input: Ordered graph $(\mathcal{G}, \leq)$

Output: $\mathbf{A}(\mathcal{G}, \leq, v)$ depends only on **order type** of the radius-r neighbourhood of v

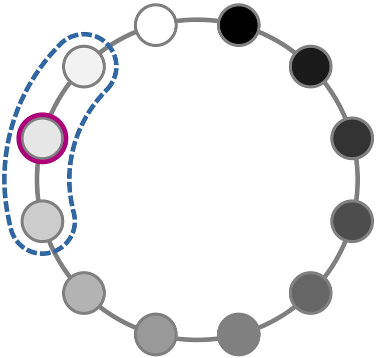
Example: Ordered Cycle



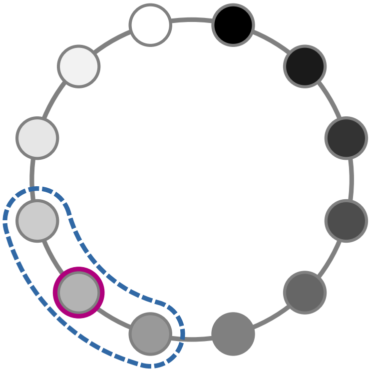
Example: Ordered Cycle



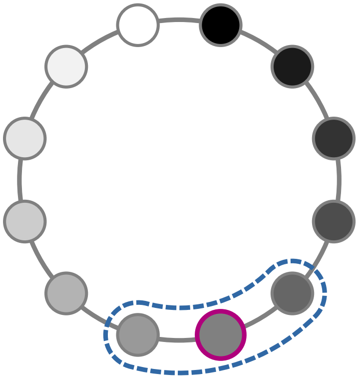
Example: Ordered Cycle



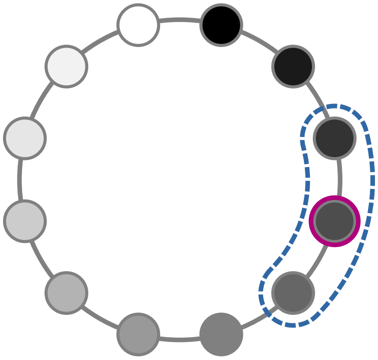
Example: Ordered Cycle



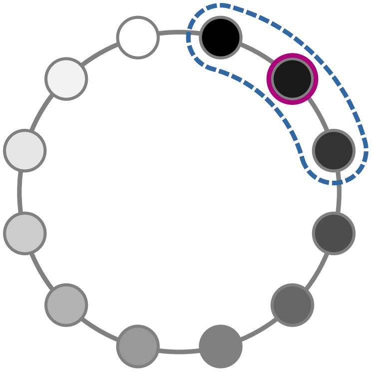
Example: Ordered Cycle



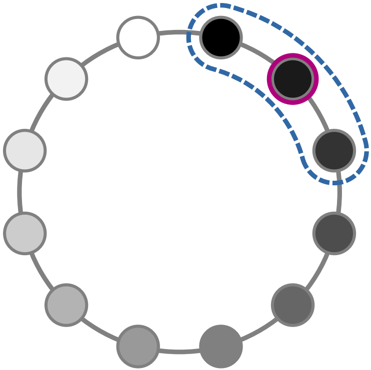
Example: Ordered Cycle



Example: Ordered Cycle



Example: Ordered Cycle

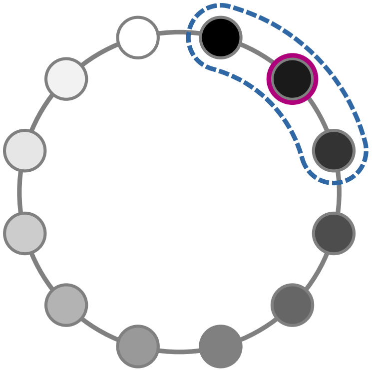


- On a large enough cycle

  $(1-\epsilon)$**-fraction**

  of neighbourhoods are isomorphic

Example: Ordered Cycle



- On a large enough cycle

  $(1-\epsilon)$**-fraction**

  of neighbourhoods are isomorphic

- **OI-algorithm** outputs the same almost everywhere!

Two Advantages of OI over PO

1. **Global** linear order vs. **Local** port numbering

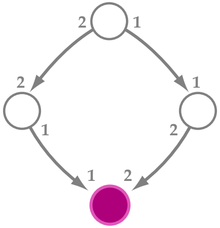
- Linear order introduces **symmetry breaking**
- **Main challenge:** How to control this?

Two Advantages of OI over PO

1. **Global** linear order vs. **Local** port numbering

- Linear order introduces **symmetry breaking**
- **Main challenge:** How to control this?

2. **PO**-algorithms cannot detect **small cycles**

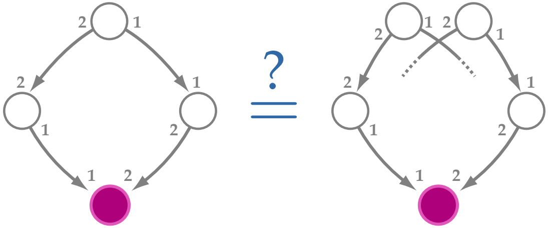


Two Advantages of OI over PO

1. **Global** linear order vs. **Local** port numbering

- Linear order introduces **symmetry breaking**
- **Main challenge:** How to control this?

2. **PO**-algorithms cannot detect **small cycles**



Two Advantages of OI over PO

1. **Global** linear order vs. **Local** port numbering

- Linear order introduces **symmetry breaking**
- **Main challenge:** How to control this?

2. **PO**-algorithms cannot detect **small cycles**

- This disadvantage disappears on **high girth graphs**

Two Advantages of OI over PO

1. **Global** linear order vs. **Local** port numbering

- Linear order introduces **symmetry breaking**
- **Main challenge:** How to control this?

2. **PO**-algorithms cannot detect **small cycles**

- This disadvantage disappears on **high girth graphs**
- **We assume:** Feasibility of a solution can be checked by a **PO**-algorithm

Two Advantages of OI over PO

$$\mathbf{1 + 2 =}$$

Need to understand

linear orders on **high girth graphs**

Order Homogeneity

Definition: We say that an **ordered** graph $(\mathcal{G}, \leq)$ is

(α, r)**-homogeneous**

if α**-fraction** of nodes have isomorphic radius-r neighbourhoods

Order Homogeneity

Definition: We say that an **ordered** graph $(\mathcal{G}, \leq)$ is

(α, r)**-homogeneous**

if α**-fraction** of nodes have isomorphic radius-r neighbourhoods

Example: **Large cycles** are $(1 - \epsilon, r)$-homogeneous ($\alpha = 1$ not possible)

Homogeneous High Girth Graphs

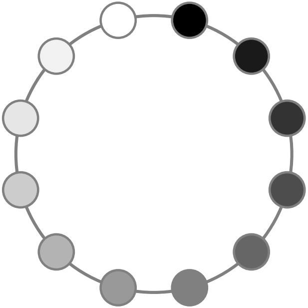
1. $(1 - \epsilon, r)$-homogeneous
2. $2k$-regular
3. Large girth
4. Finite graph

Homogeneous High Girth Graphs

1. $(1-\epsilon, r)$-homogeneous
2. $2k$-regular
3. Large girth
4. Finite graph

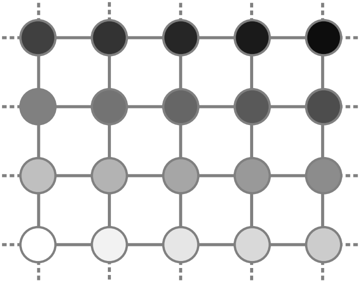
Homogeneous High Girth Graphs

1. $(1-\epsilon, r)$-homogeneous
2. $2k$-regular
3. Large girth
4. Finite graph



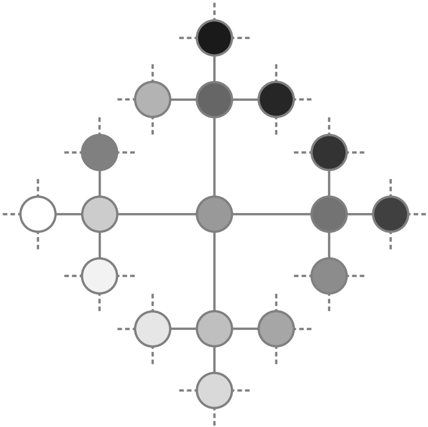
Homogeneous High Girth Graphs

1. $(1-\epsilon, r)$-homogeneous
2. $2k$-regular
3. Large girth
4. Finite graph



Homogeneous High Girth Graphs

1. $(1-\epsilon, r)$-homogeneous
2. $2k$-regular
3. Large girth
4. Finite graph



Homogeneous High Girth Graphs

1. $(1-\epsilon, r)$-homogeneous
2. $2k$-regular
3. Large girth
4. Finite graph

Main Technical Result:

Graphs $(\mathcal{H}_\epsilon, \leq_\epsilon)$ with properties 1–4 exist

Homogeneous High Girth Graphs

1. $(1-\epsilon, r)$-homogeneous
2. $2k$-regular
3. Large girth
4. Finite graph

Main Technical Result:

Graphs $(\mathcal{H}_\epsilon, \leq_\epsilon)$ with properties 1–4 exist

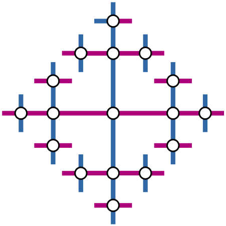
Proof. We use **Cayley graphs** of **soluble groups**

Homogeneous High Girth Graphs

1. $(1-\epsilon, r)$-homogeneous
2. $2k$-regular
3. Large girth
4. Finite graph

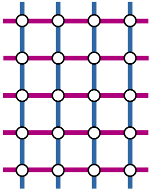
Main Technical Result:

Graphs $(\mathcal{H}_\epsilon, \leq_\epsilon)$ with properties 1–4 exist



$F(\alpha, \beta)$

?



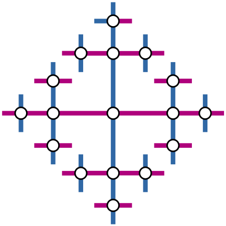
$\mathbb{Z} \times \mathbb{Z}$

Homogeneous High Girth Graphs

1. $(1 - \epsilon, r)$-homogeneous
2. $2k$-regular
3. Large girth
4. Finite graph

Main Technical Result:

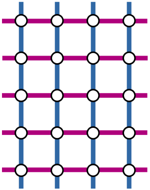
Graphs $(\mathcal{H}_\epsilon, \leq_\epsilon)$ with properties 1–4 exist



$F(\alpha, \beta)$

Want:

- **Invariant order:**
  $x < y \iff gx < gy$



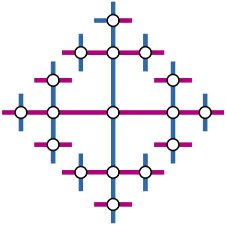
$\mathbb{Z} \times \mathbb{Z}$

Homogeneous High Girth Graphs

1. $(1-\epsilon, r)$-homogeneous
2. $2k$-regular
3. Large girth
4. Finite graph

Main Technical Result:

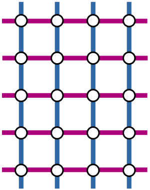
Graphs $(\mathcal{H}_\epsilon, \leq_\epsilon)$ with properties 1–4 exist



$F(\alpha, \beta)$

Want:

- Invariant order
- **Polynomial growth** [Gromov's theorem]



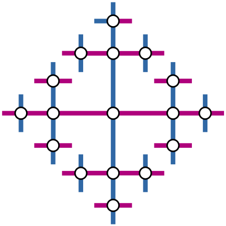
$\mathbb{Z} \times \mathbb{Z}$

Homogeneous High Girth Graphs

1. $(1-\epsilon, r)$-homogeneous
2. $2k$-regular
3. Large girth
4. Finite graph

Main Technical Result:

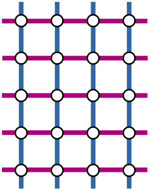
Graphs $(\mathcal{H}_\epsilon, \leq_\epsilon)$ with properties 1–4 exist



$F(\alpha, \beta)$

Want:

- Invariant order
- Polynomial growth
- **Large girth** [Gamburd et al. 09]



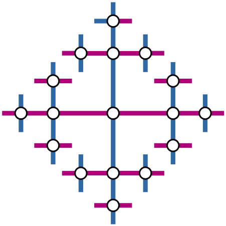
$\mathbb{Z} \times \mathbb{Z}$

Homogeneous High Girth Graphs

1. $(1-\epsilon, r)$-homogeneous
2. $2k$-regular
3. Large girth
4. Finite graph

Main Technical Result:

Graphs $(\mathcal{H}_\epsilon, \leq_\epsilon)$ with properties 1–4 exist

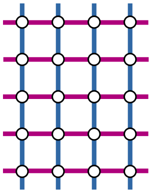


$F(\alpha, \beta)$

Want:

- Invariant order
- Polynomial growth
- Large girth

$$G_1 = \mathbb{Z},$$
$$G_{i+1} = (G_i \times G_i) \rtimes \mathbb{Z}$$



$\mathbb{Z} \times \mathbb{Z}$

Homogeneous High Girth Graphs

1. $(1-\epsilon, r)$-homogeneous
2. $2k$-regular
3. Large girth
4. Finite graph

Main Technical Result:

Graphs $(\mathcal{H}_\epsilon, \leq_\epsilon)$ with properties 1–4 exist

Proof of Main Thm:

Form graph products $(\mathcal{H}_\epsilon, \leq_\epsilon) \times \mathcal{G}$

Conclusion

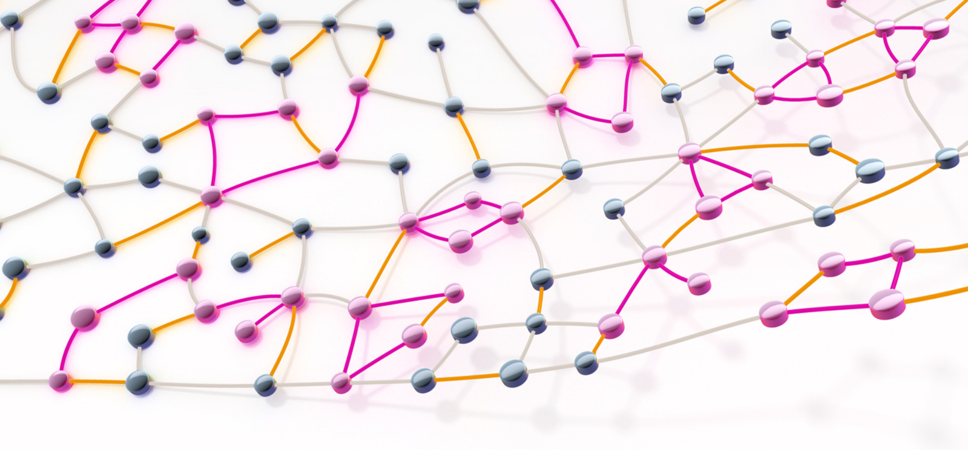
Our result:

For Local Approximation,

$$\mathbf{ID} = \mathbf{OI} = \mathbf{PO}$$

Open problems:

- Planar graphs?
- Applications elsewhere—descriptive complexity?



Cheers!