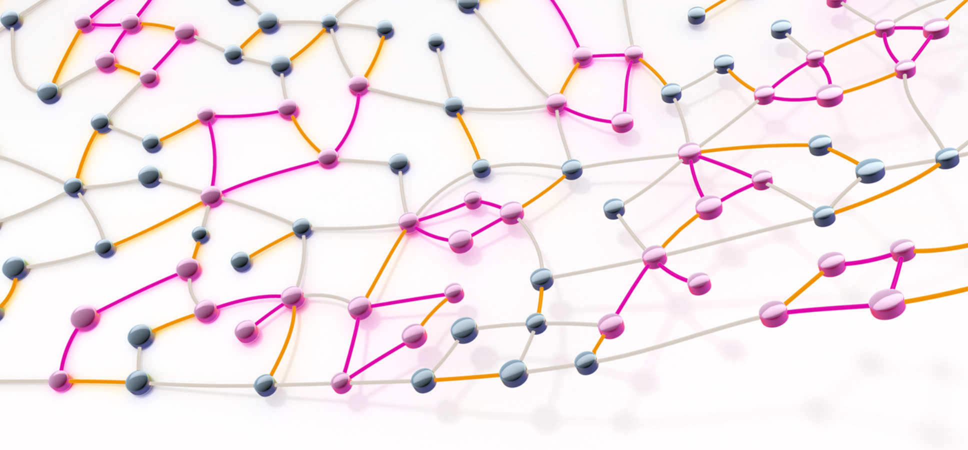


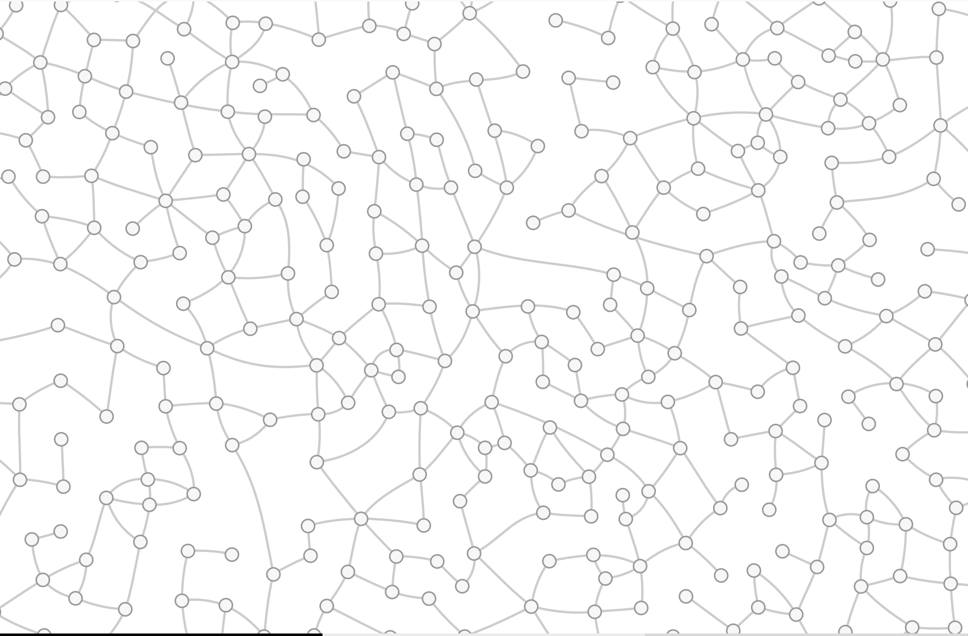
Lower Bounds *for* Local Approximation

Mika Göös, Juho Hirvonen & Jukka Suomela
University of Helsinki

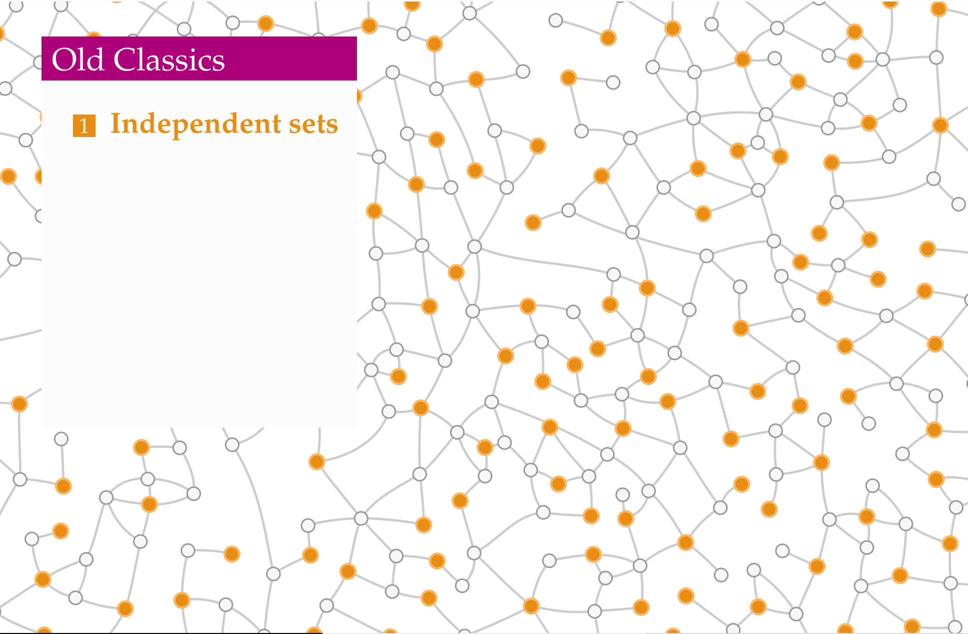
Lower Bounds *for* Local Approximation

Main Result: **Local algorithms** do not need ***unique IDs*** when solving graph optimization problems

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



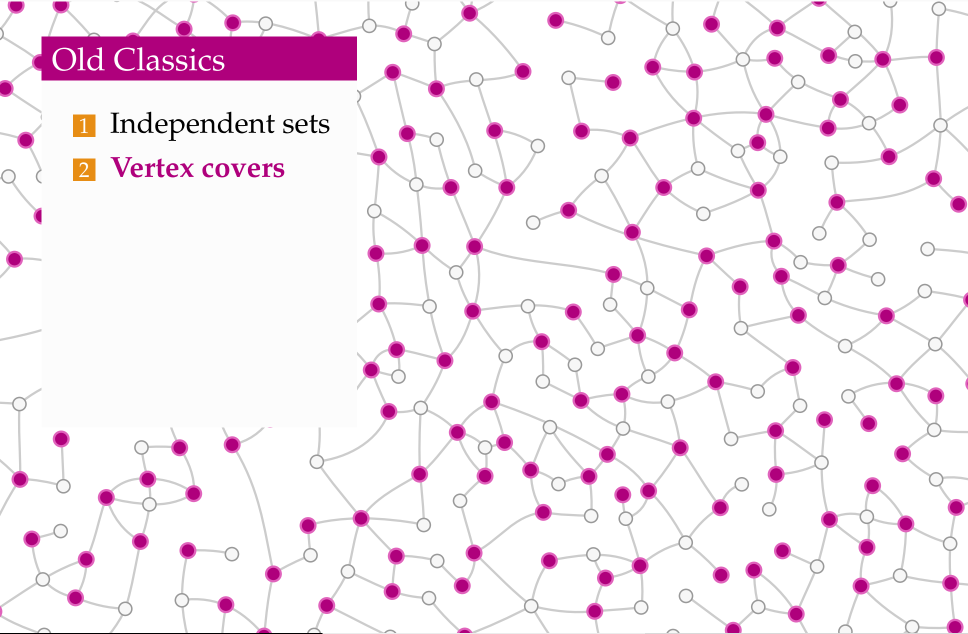
“Simple” Graph Problems



Old Classics

1. **Independent sets**

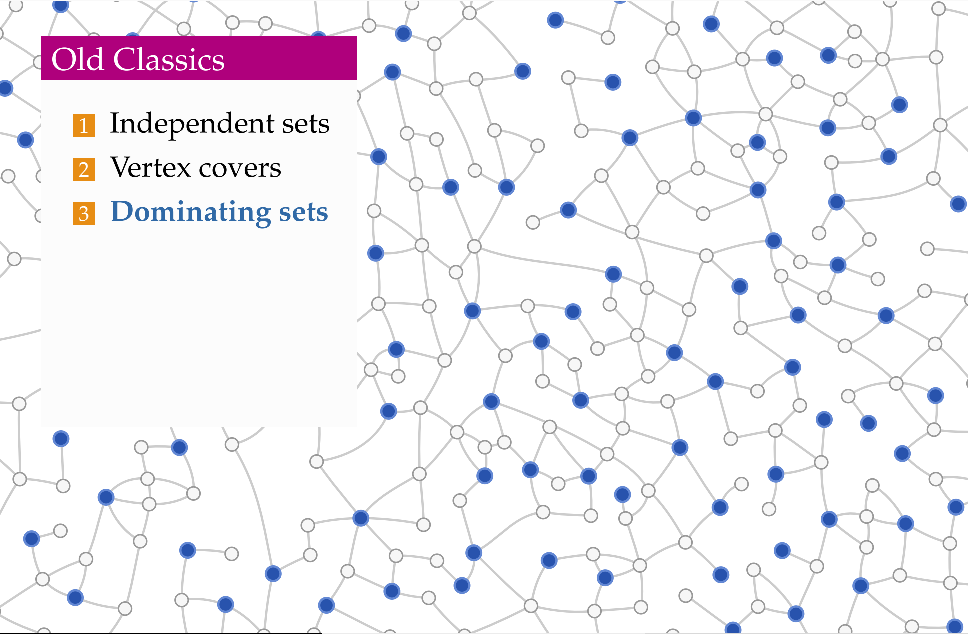
“Simple” Graph Problems



Old Classics

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

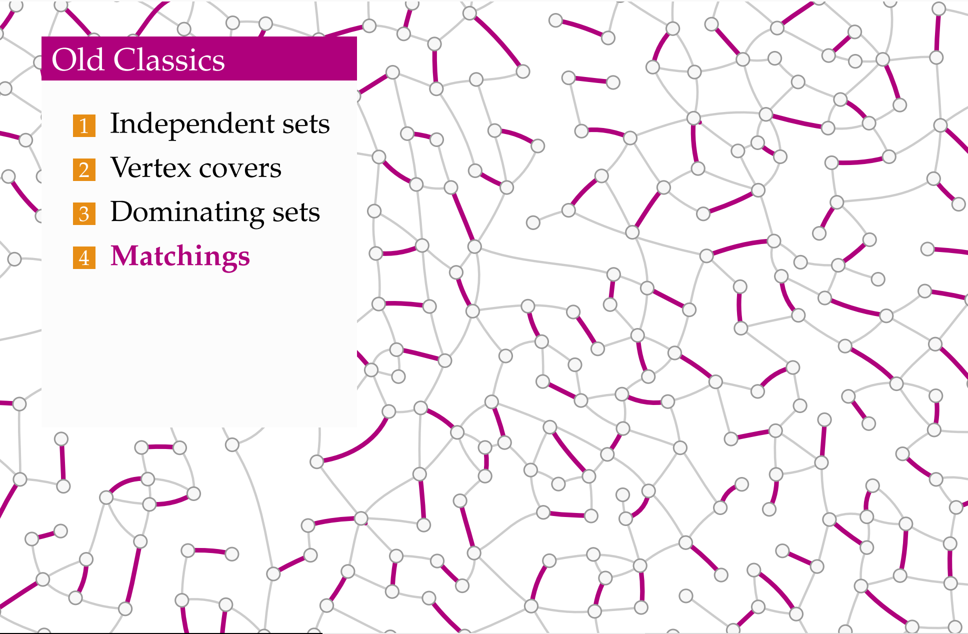
"Simple" Graph Problems



Old Classics

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

“Simple” Graph Problems



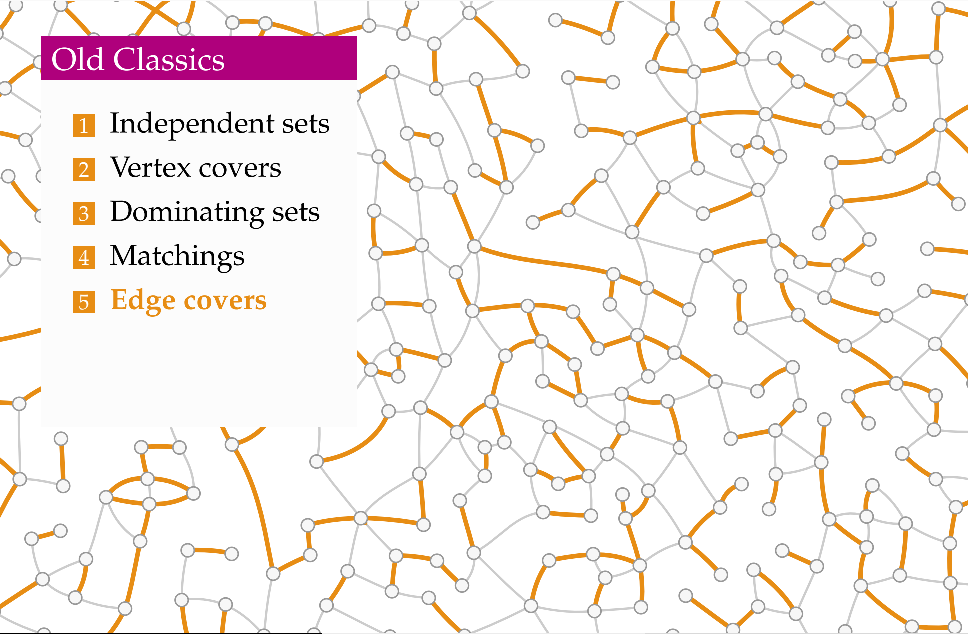
Old Classics

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

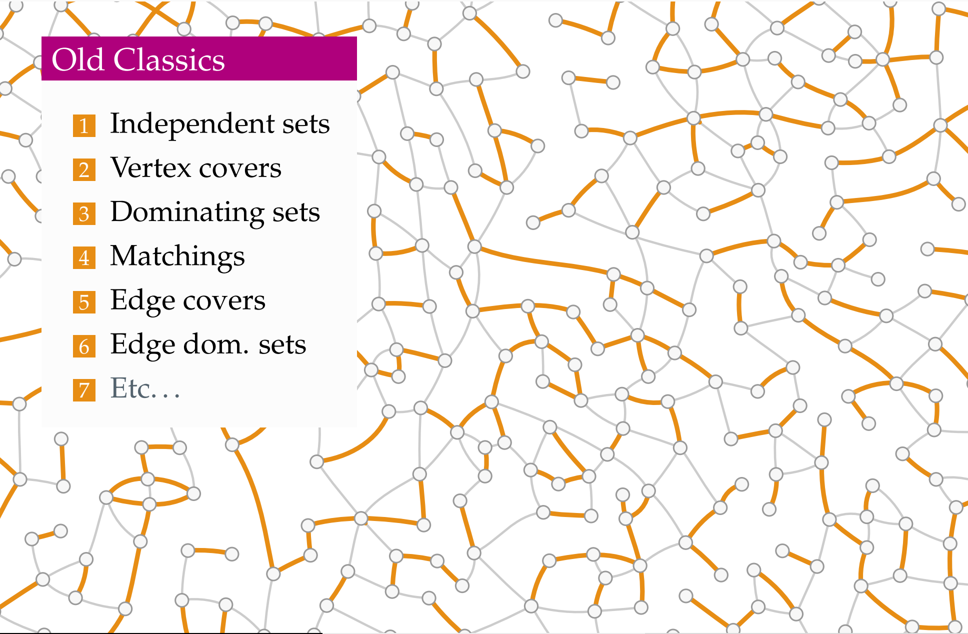
"Simple" Graph Problems

Old Classics

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



“Simple” Graph Problems



Old Classics

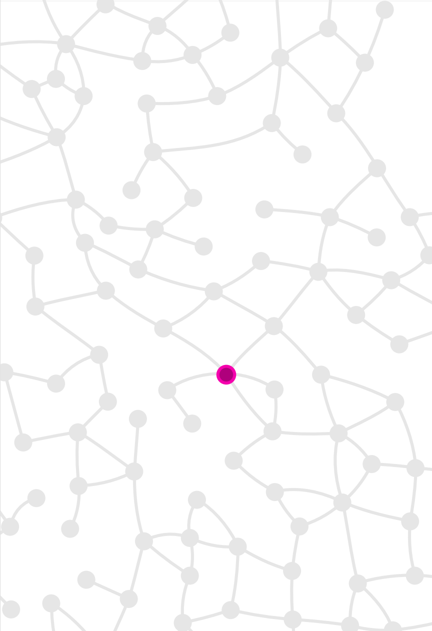
1. Independent sets
2. Vertex covers
3. Dominating sets
4. Matchings
5. Edge covers
6. Edge dom. sets
7. Etc. . .

Local Algorithms

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

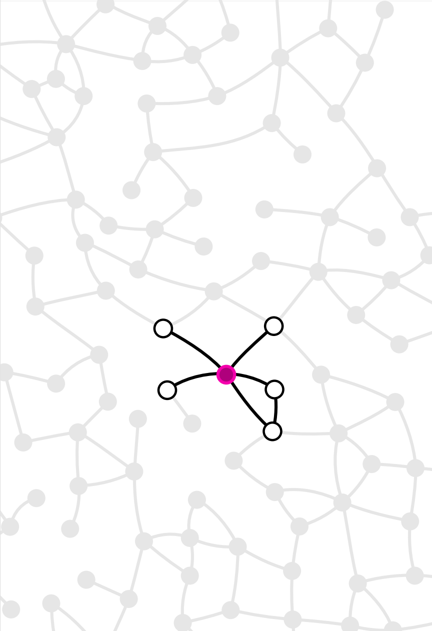
Local Algorithms

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



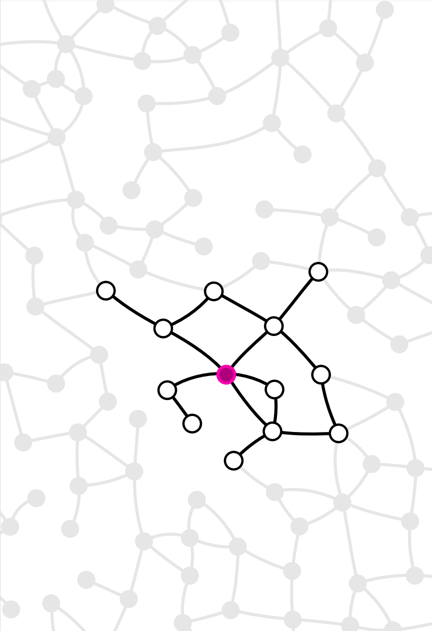
Local Algorithms

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



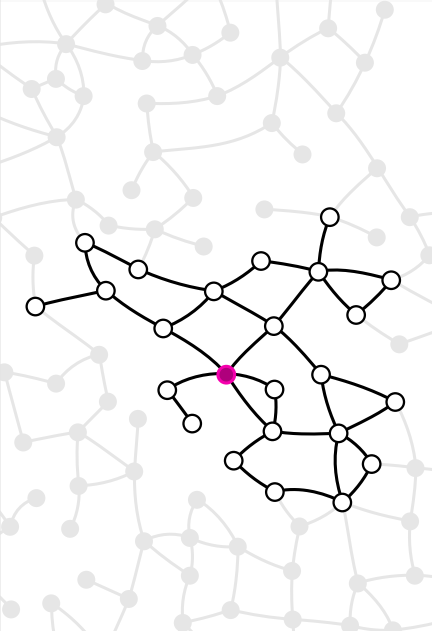
Local Algorithms

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



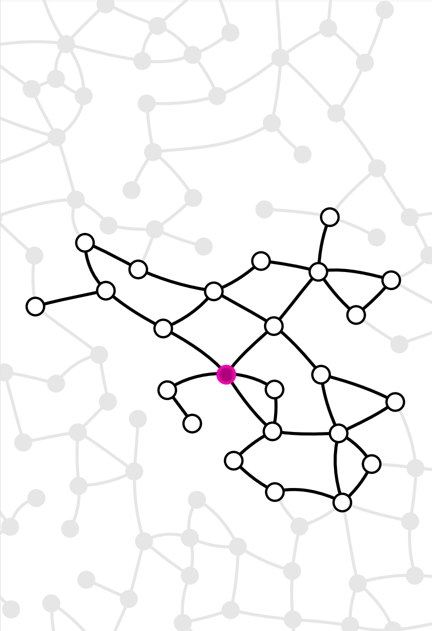
Local Algorithms

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



Local Algorithms

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

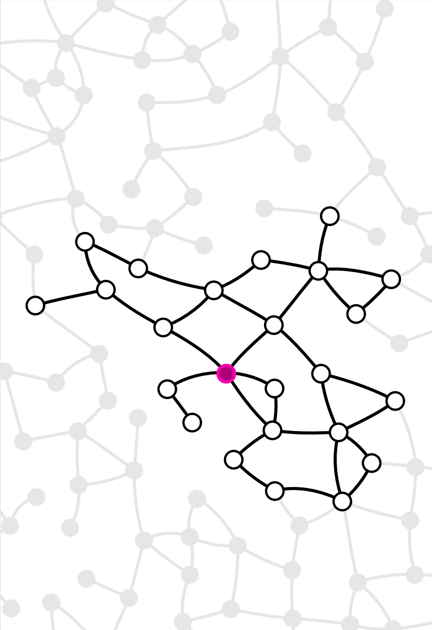


Local Algorithms

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

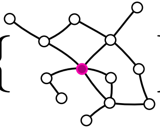
On ***bounded degree graphs*** ($\Delta = O(1)$) running time is a constant:

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



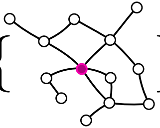
Local Algorithms

Definition:

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

Local Algorithms

Definition:

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

Two Network Models

Unique Identifiers

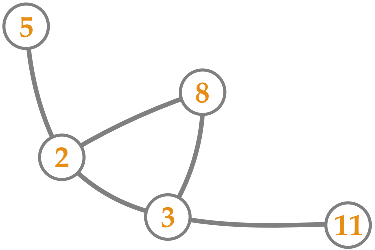
Anonymous Networks
with Port Numbering

Two Network Models

Unique Identifiers

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

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



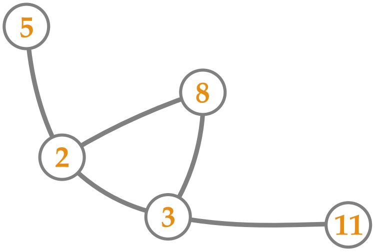
Anonymous Networks
with Port Numbering

Two Network Models

Unique Identifiers

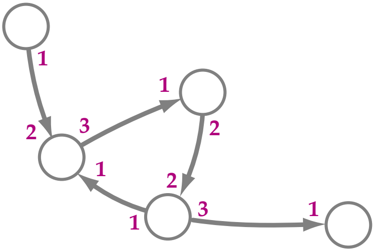
- Each node has a unique $O(\log n)$-bit **label**:

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



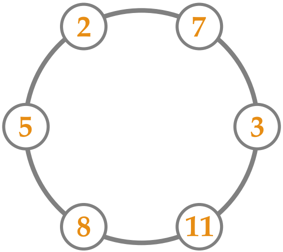
Anonymous Networks
with Port Numbering

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

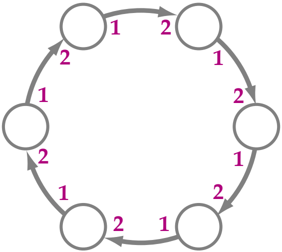


Example: Independent Sets on a Cycle

ID-model

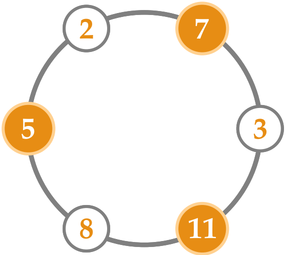


PO-model



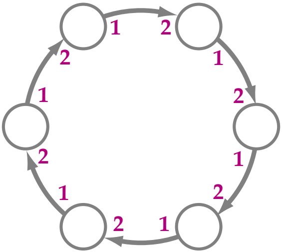
Example: Independent Sets on a Cycle

ID-model



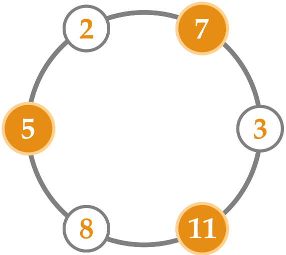
- [Cole–Vishkin 86, Linial 92]: **Maximal independent set** can be computed in $\Theta(\log^* n)$ rounds

PO-model



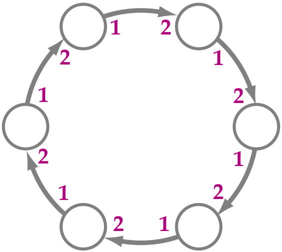
Example: Independent Sets on a Cycle

ID-model



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

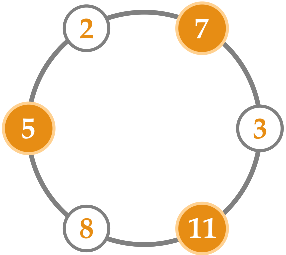
PO-model



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

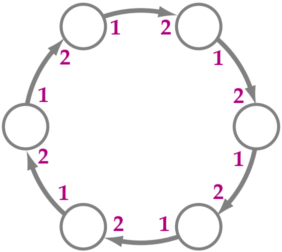
Example: Independent Sets on a Cycle

ID-model



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

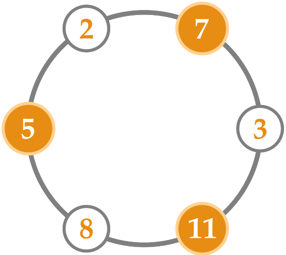
PO-model



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

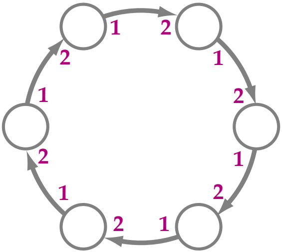
Example: Independent Sets on a Cycle

ID-model



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

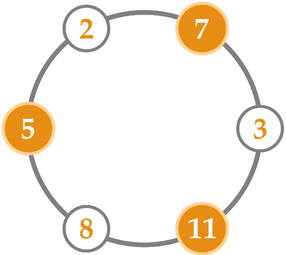
PO-model

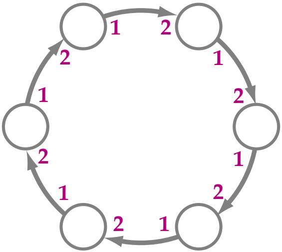


- Above **PO**-network is **fully symmetric**
- $\Rightarrow$ All nodes give same output
- $\Rightarrow$ Empty set is computed!

Example: Independent Sets on a Cycle

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



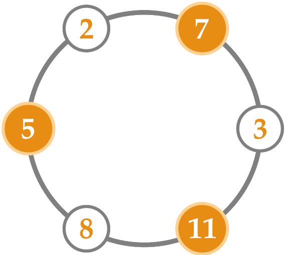


- [Cole–Vishkin 86, Linial 92]: **Maximal independent set** can be computed in $\Theta(\log^* n)$ rounds

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

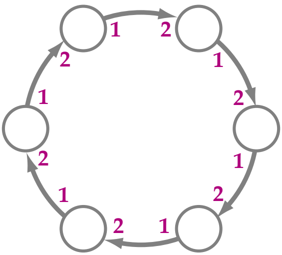
Example: Independent Sets on a Cycle

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!

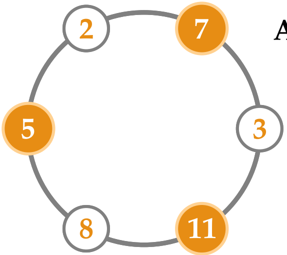
Example: Independent Sets on a Cycle

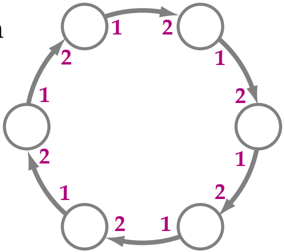
ID-model

$\approx$

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**	∞	∞	[LW **DISC'08**]
Max **Matching**	∞	∞	[CHW **DISC'08**]
Min **Vertex Cover**	2	2	[ÅS **SPAA'10**]
Min **Edge Cover**	2	2	
Min **Dominating Set**	$\Delta + 1$	$\Delta + 1$	
Min **Edge Dominating Set**	???	$4 - \frac{2}{\Delta}$	[Suomela **PODC'10**]

Known Approximation Ratios

	ID	PO	
Max **Independent Set**	∞	∞	[LW **DISC'08**]
Max **Matching**	∞	∞	[CHW **DISC'08**]
Min **Vertex Cover**	2	2	[ÅS **SPAA'10**]
Min **Edge Cover**	2	2	
Min **Dominating Set**	$\Delta + 1$	$\Delta + 1$	
Min **Edge Dominating Set**	$4 - \frac{2}{\Delta}$	$4 - \frac{2}{\Delta}$	[Suomela **PODC'10**]

Our Result (informally)

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

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

for a general class of graph problems

Our Result (informally)

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

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

for a general class of graph problems

next: *proof idea...*

What is known?

ID **PO**

What is known?

ID **PO**

[Naor–Stockmeyer 95]:

Local ID-algorithms can only **compare** identifiers

What is known?

ID = OI PO

[Naor–Stockmeyer 95]:

Local ID-algorithms can only **compare** identifiers

What is known?

ID = OI   PO

[Naor–Stockmeyer 95]:

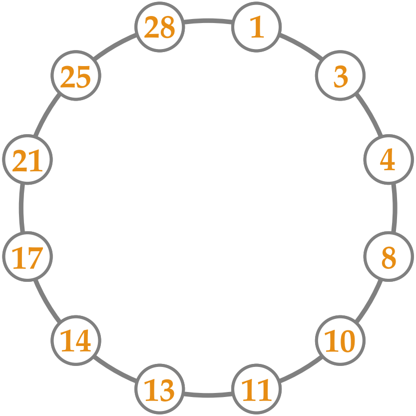
Local ID-algorithms can only **compare** identifiers

Order invariant (OI) 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

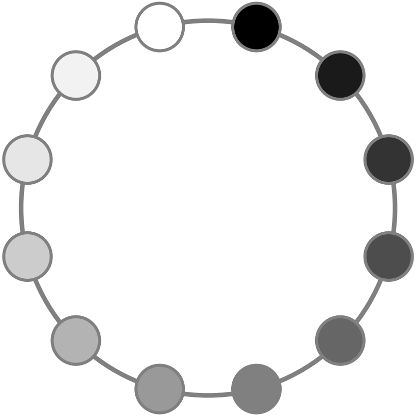
What is known?

ID = OI PO



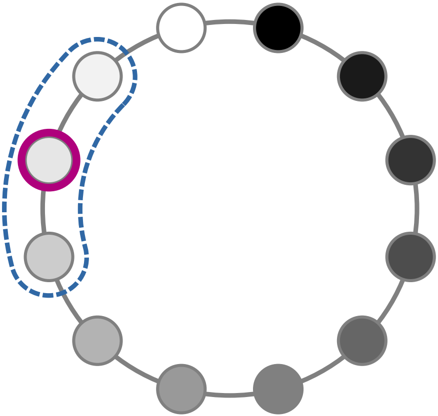
What is known?

ID = **OI**   PO



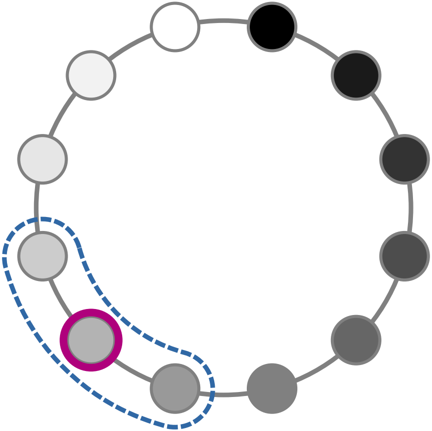
What is known?

ID = **OI** PO



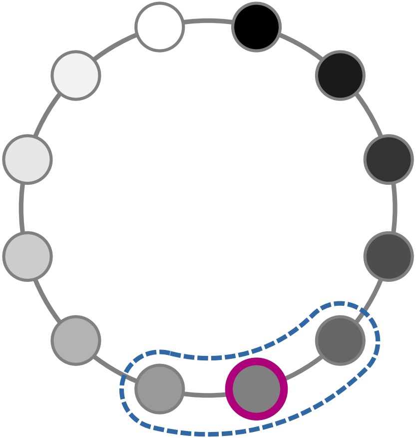
What is known?

ID = **OI** PO



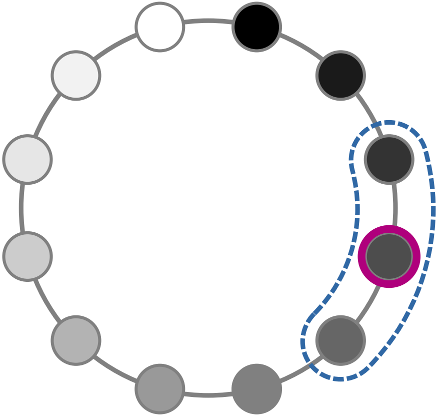
What is known?

ID = **OI** PO



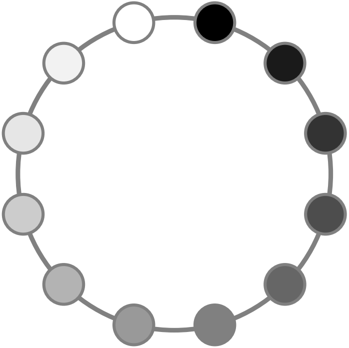
What is known?

ID = OI PO



What is known?

ID = **OI** PO

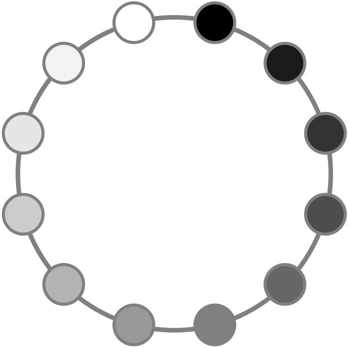


Ordered cycle:

$(1-\epsilon)$-**fraction** of neighbourhoods are isomorphic

What is known?

ID = **OI** PO



Ordered cycle:

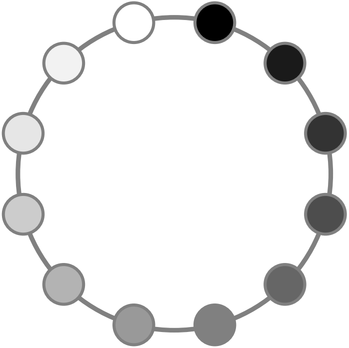
$(1-\epsilon)$**-fraction** of neighbourhoods are isomorphic

$\Updownarrow$

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

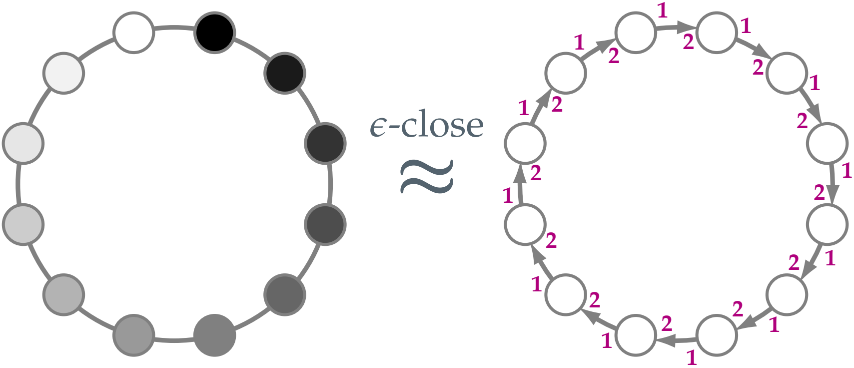
What is known?

ID = **OI** PO



What is known?

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



Key Construction

High-degree homogeneous graphs:

(analogue to *homogeneously ordered cycles*)

Key Construction

High-degree homogeneous graphs:

(analogue to *homogeneously ordered cycles*)

1. $(1-\epsilon)$-homogeneous
2. Large girth
3. Finite graph

Key Construction

High-degree homogeneous graphs:

(analogue to *homogeneously ordered cycles*)

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

Key Construction

High-degree homogeneous graphs:

(analogue to *homogeneously ordered cycles*)

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

Main Technical Result:

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

Key Construction

High-degree homogeneous graphs:

(analogue to *homogeneously ordered cycles*)

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

Main Technical Result:

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

Algebraic construction:

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

Key Construction

High-degree homogeneous graphs:

(analogue to *homogeneously ordered cycles*)

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

Main Technical Result:

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

Proof of Main Thm:

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

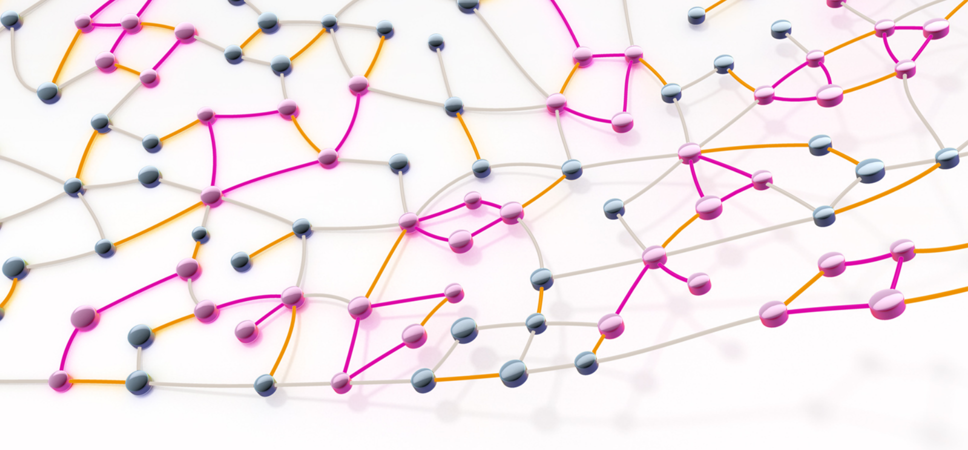
Our result:

For Local Approximation,

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

Open problems:

- Planar graphs?
- More applications for **homogeneous graphs**?
- Randomness!



Cheers!