# Linear-in-Δ lower bounds in the LOCAL model

Mika Göös, University of Toronto
**Juho Hirvonen**, Aalto University & HIIT
Jukka Suomela, Aalto Univesity & HIIT

PODC 16.7.2014

# This work

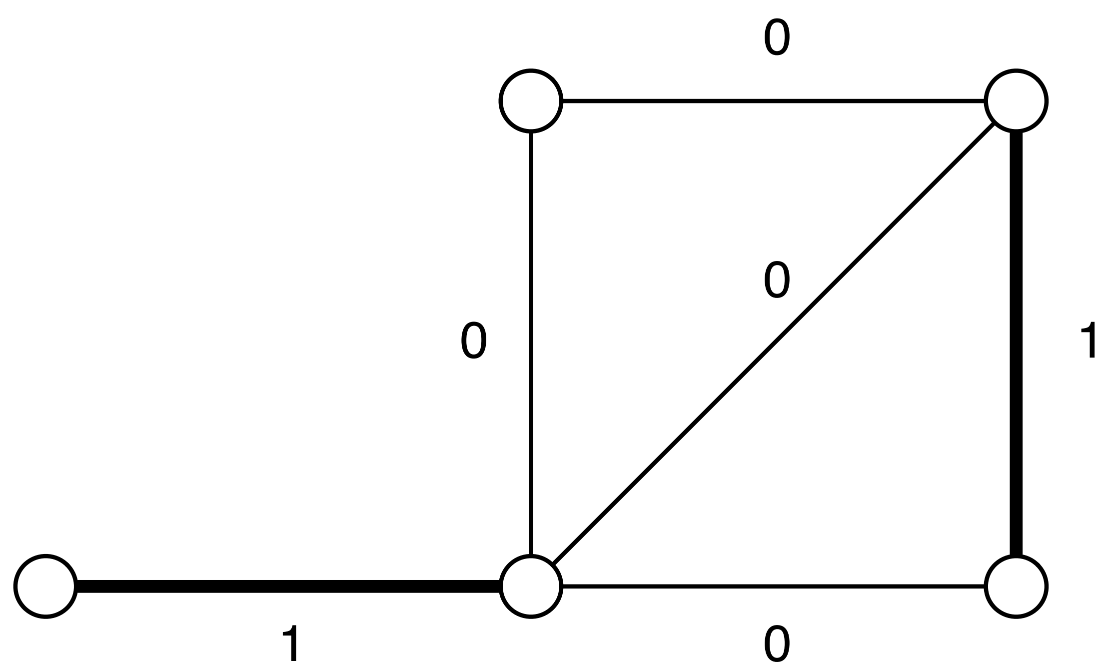
The first linear-in-Δ lower bound for a natural graph problem in the LOCAL model

**Fractional maximal matching:**

- There is no $o(\Delta)$-algorithm, independent of $n$
- There is an $O(\Delta)$-algorithm, independent of $n$

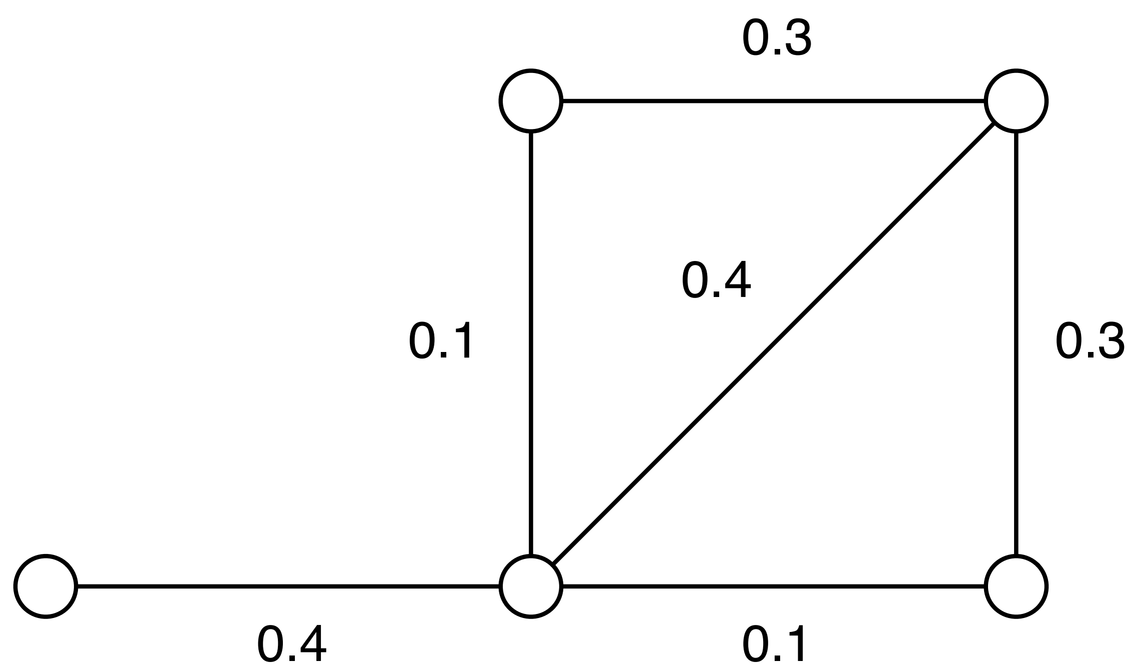
($\Delta$ = maximum degree, $n$ = number of vertices)

# Matching



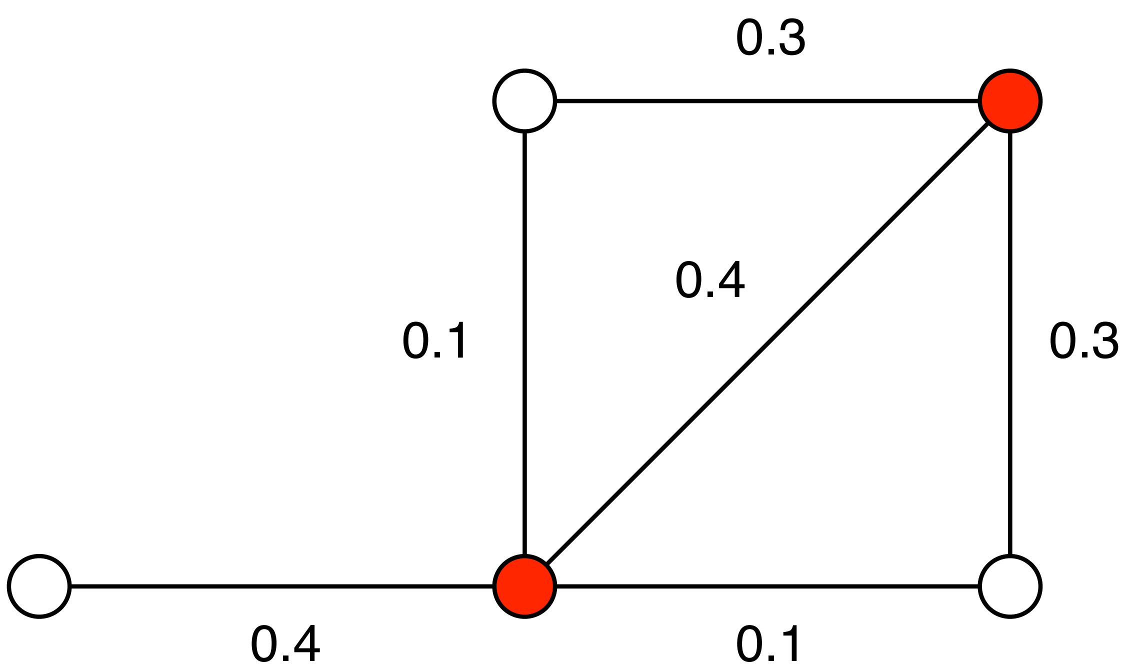

Matching assigns weight 1 to matched edges and weight 0 to the rest

# Fractional matching



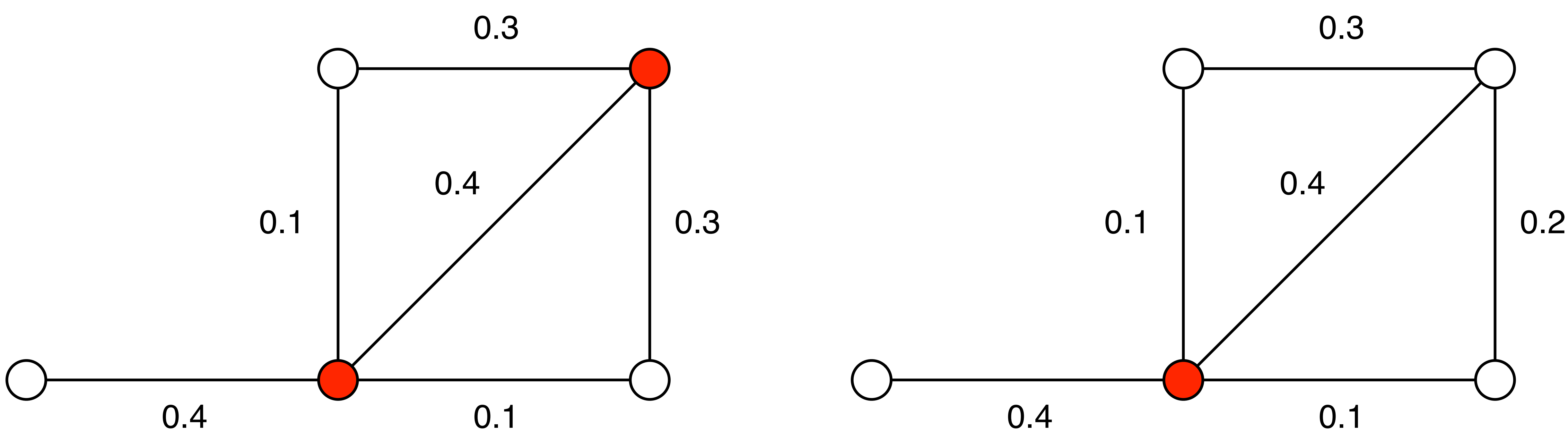

FM is a linear relaxation of matching: weights of the incident edges sum up to at most 1

# Maximal fractional matching




A node is *saturated*, if the sum of the weights of the incident edges is equal to one

# Maximal fractional matching




The fractional matching is *maximal*,
if no two unsaturated nodes are adjacent

# Standard LOCAL model

- Synchronous communication
- No bandwidth restrictions
- Running time = number of communication rounds
- Both deterministic and randomized algorithms

# This work

The first linear-in-Δ lower bound for
a natural graph problem in the LOCAL model

**Fractional maximal matching:**

- There is no $o(\Delta)$-algorithm, independent of $n$
- There is an $O(\Delta)$-algorithm, independent of $n$

($\Delta$ = maximum degree, $n$ = number of vertices)

# Prior work

Coordination problems:

- Maximal matching
- Maximal independent set
- $(\Delta+1)$-coloring

Algorithms $O(\Delta + \log^* n)$ also $O(\text{polylog}(n))$

Lower bounds $\Omega(\log^* n)$ and $\Omega(\log \Delta)$

[Linial '92] [Kuhn et al. '05]

# Prior work

Coordination problems:

- Maximal matching
- Maximal independent set
- $(\Delta+1)$-coloring

Algorithms $O(\Delta + \log^* n)$ also $O(\text{polylog}(n))$

Lower bounds $\Omega(\log^* n)$ and $\Omega(\log \Delta)$

[Linial '92] [Kuhn et al. '05]

# The Proof

# The Proof

**A short guide**

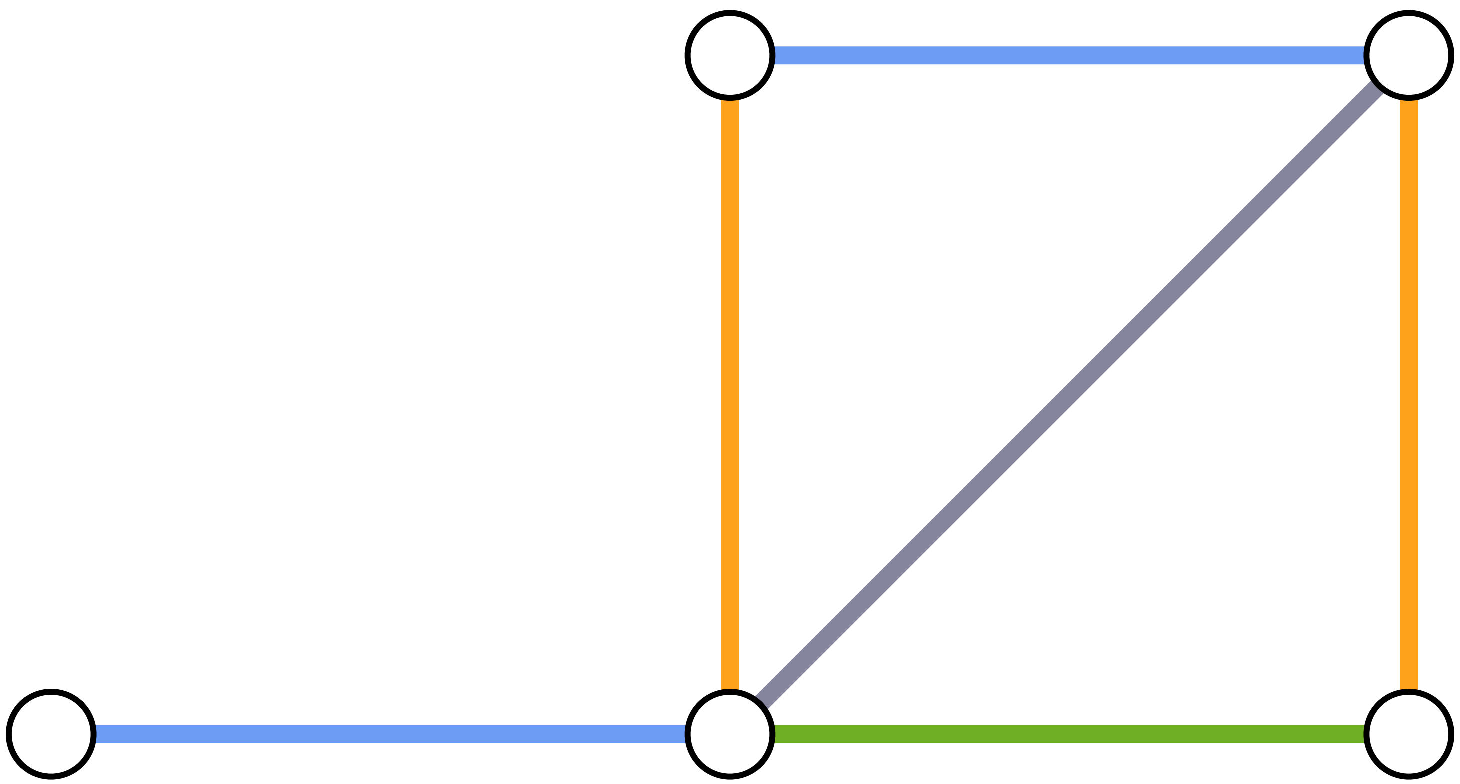
- Step 0: Introduce models EC, PO, OI and ID
- Step 1: $\Omega(\Delta)$-lower bound in the EC-model
- Step 2: Simulation result EC→PO→OI→ID
- Step 3: ID → Randomized algorithms

# The Proof

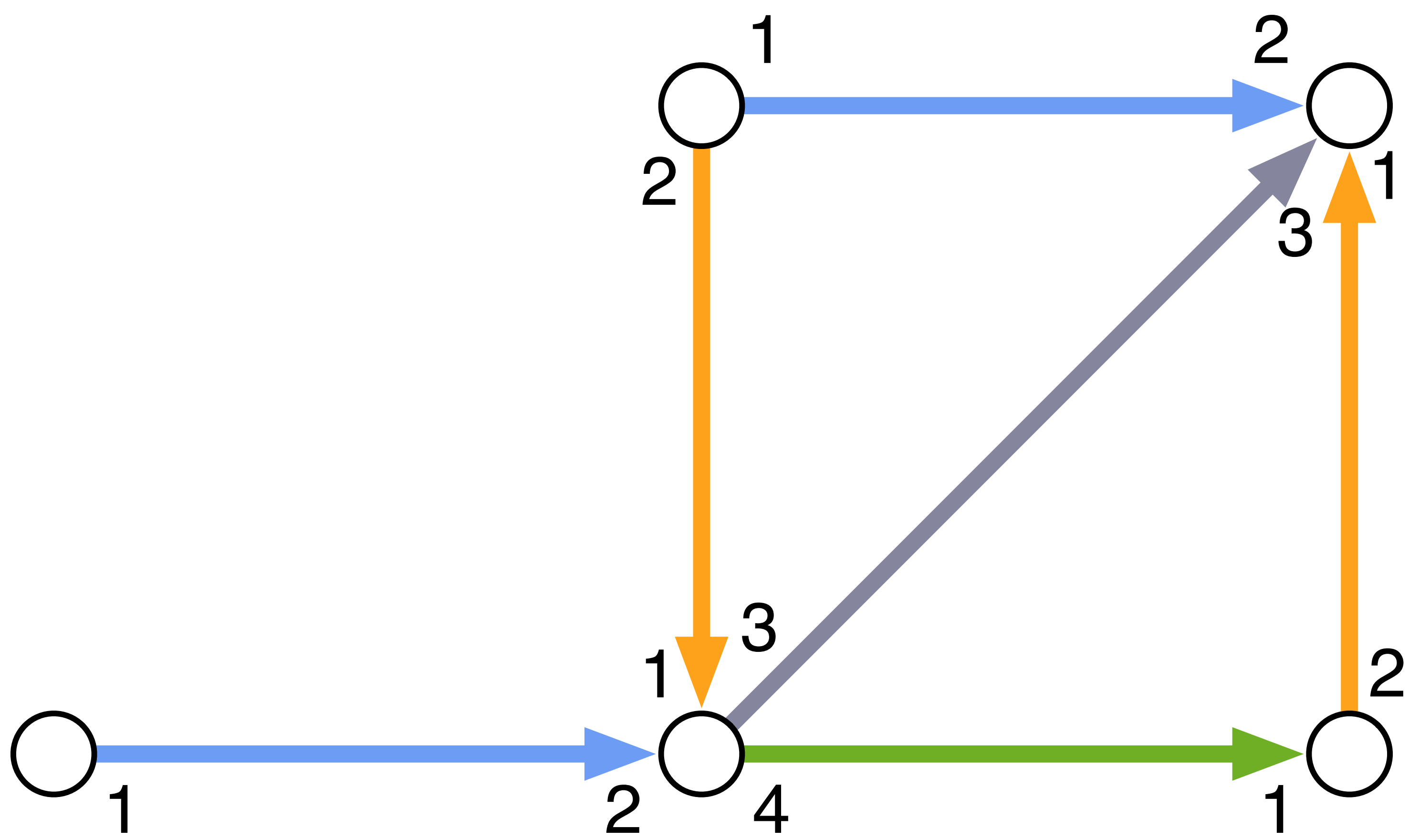
**A short guide**

- **Step 0: Introduce models EC, PO, OI and ID**
- Step 1: Ω(Δ)-lower bound in the EC-model
- Step 2: Simulation result EC→PO→OI→ID
- Step 3: ID → Randomized algorithms

# Edge coloring (EC)

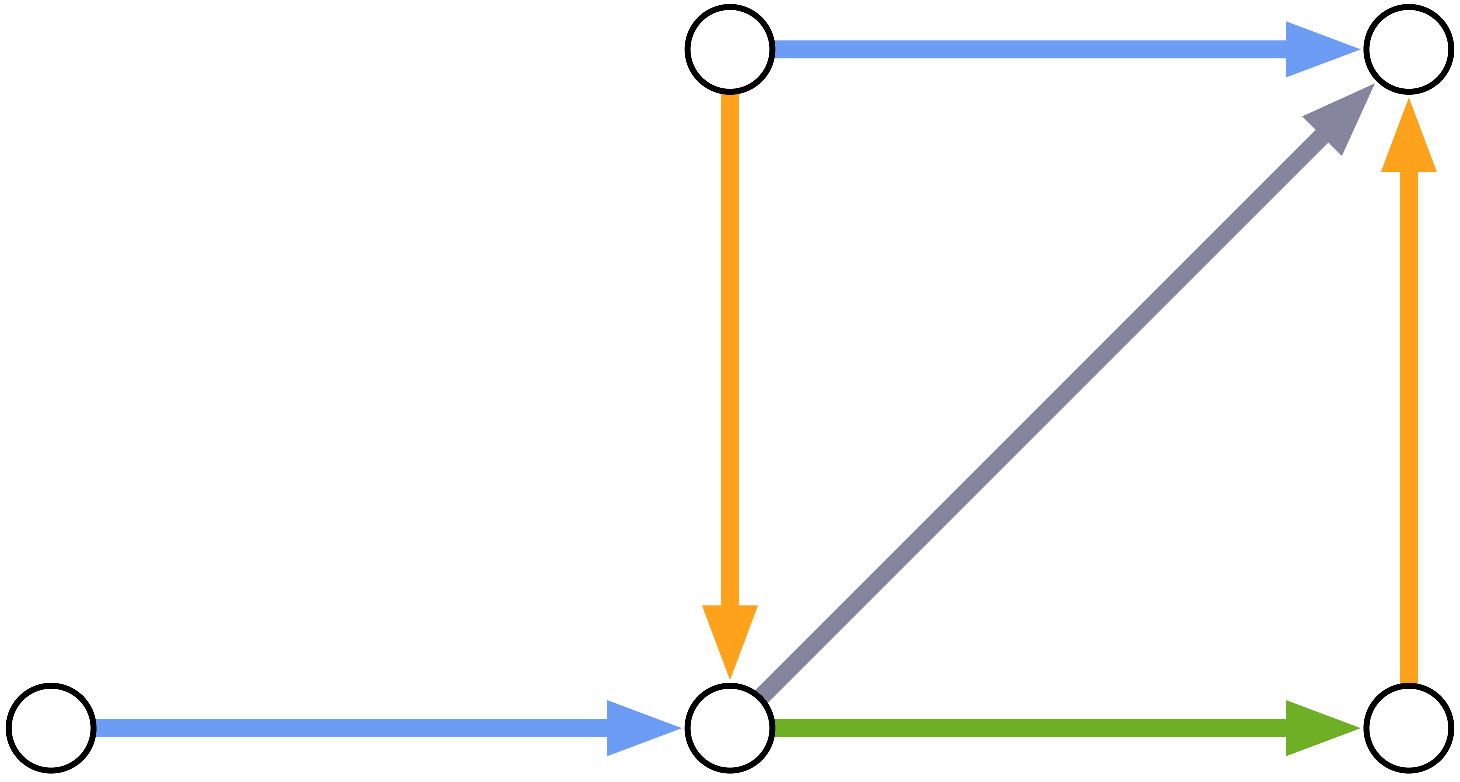


# Port-numbering and orientation (PO)



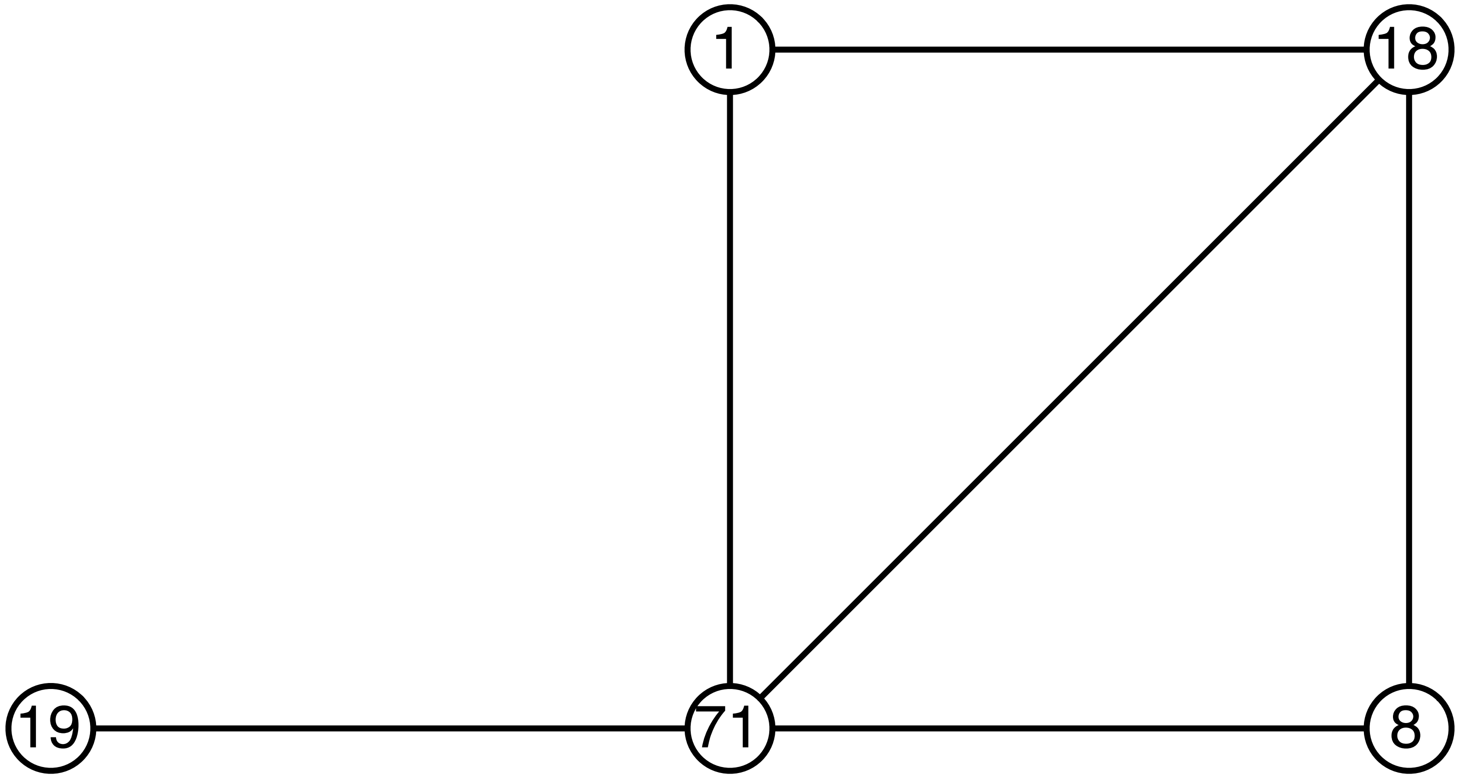

EC → **PO** → OI → ID → R

# Port-numbering and orientation (PO)



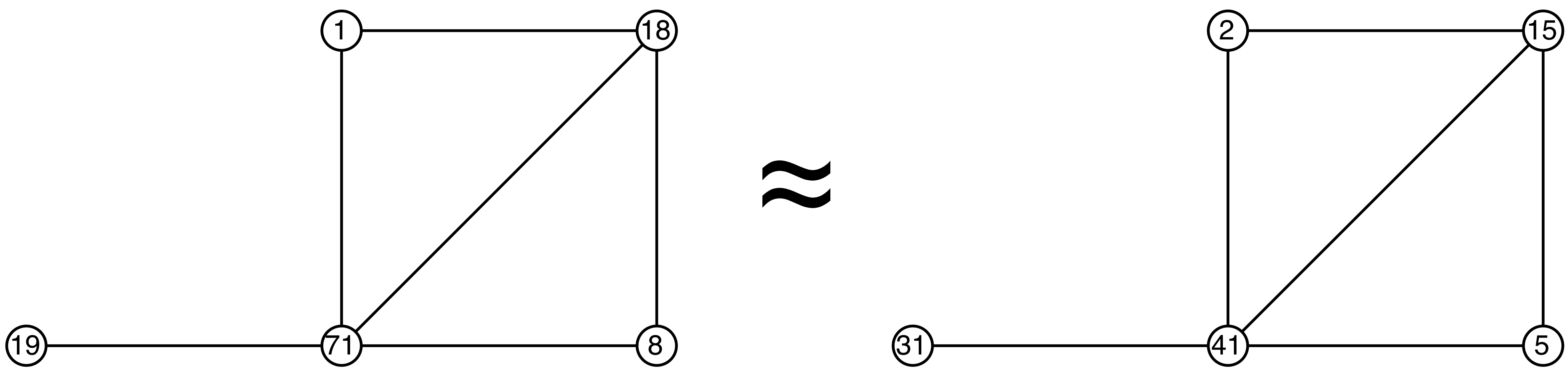
EC ⇝ **PO** ⇝ OI ⇝ ID ⇝ R

# Unique Identifiers (ID)




EC ⇝ PO ⇝ OI ⇝ **ID** ⇝ R

# Order Invariant (OI)



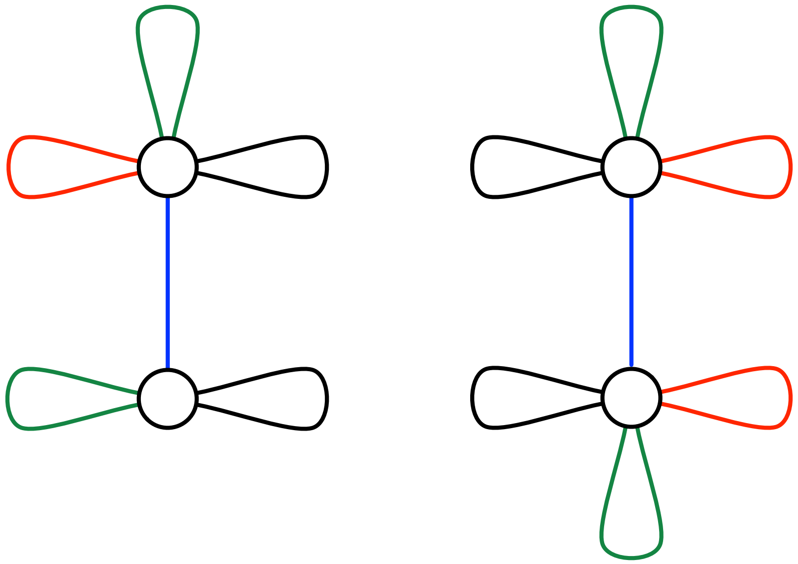

EC ⇝ PO ⇝ **OI** ⇝ ID ⇝ R

# The Proof

**A short guide**

- Step 0: Introduce models EC, PO, OI and ID
- **Step 1: Ω(Δ)-lower bound in the EC-model**
- Step 2: Simulation result EC→PO→OI→ID
- Step 3: ID → Randomized algorithms

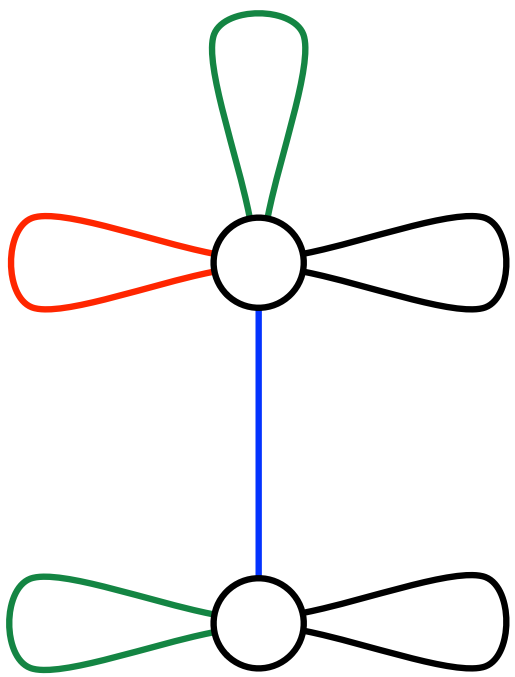
# Loopy graphs



*k*=2 *k*=3

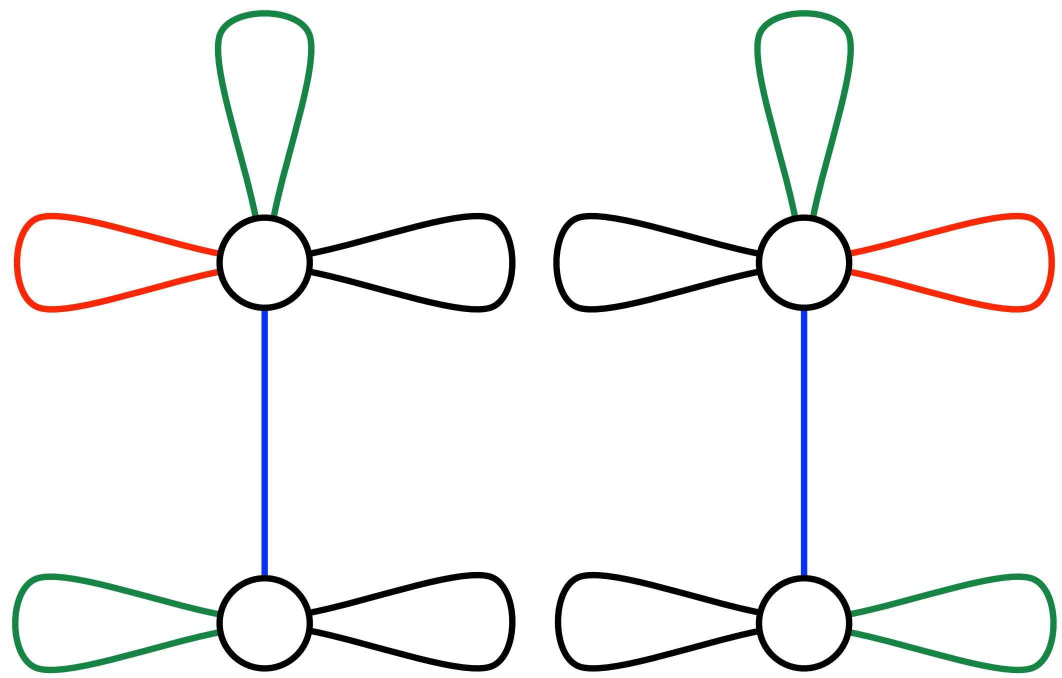
A graph is *k-loopy*, if it has at least *k* self-loops at each node

# Loopy graphs



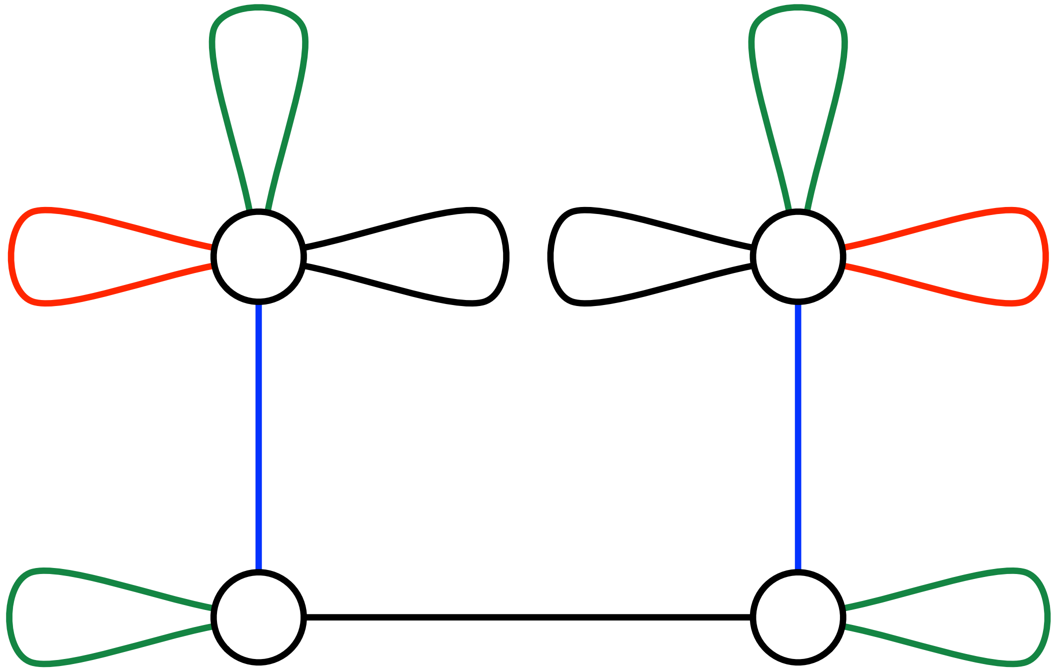
Loopy graphs are a compact representation of simple graphs with lots of symmetry

# Loopy graphs



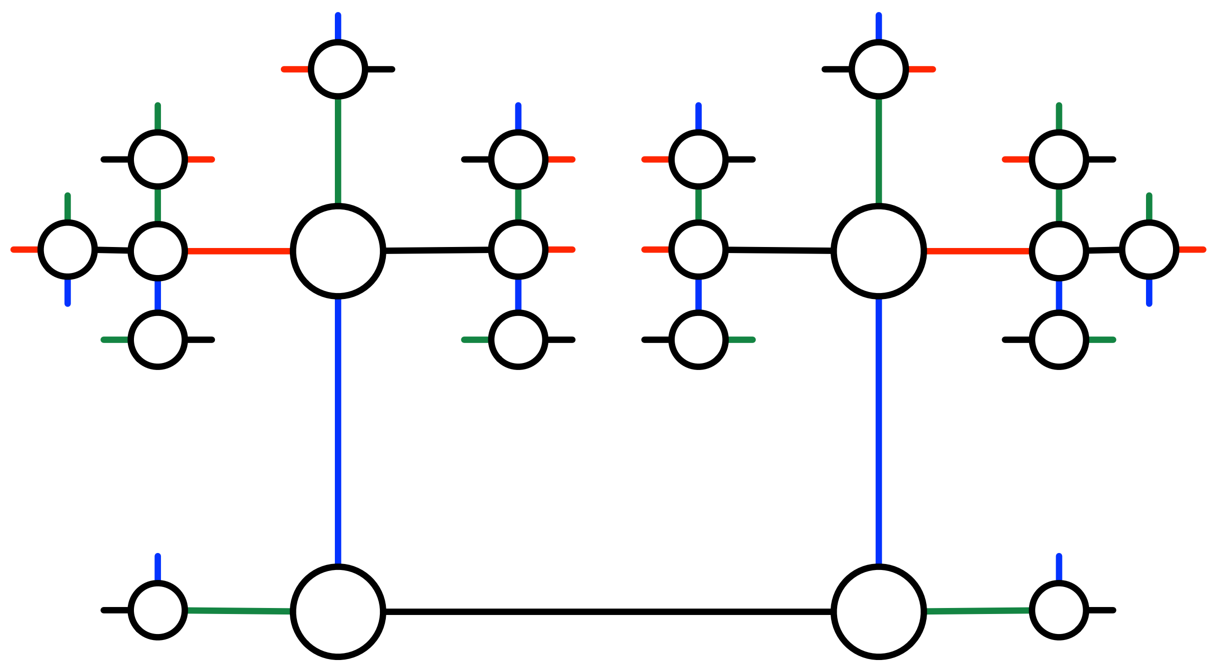
A loopy graph can be *unfolded* to get a simple graph

# Loopy graphs



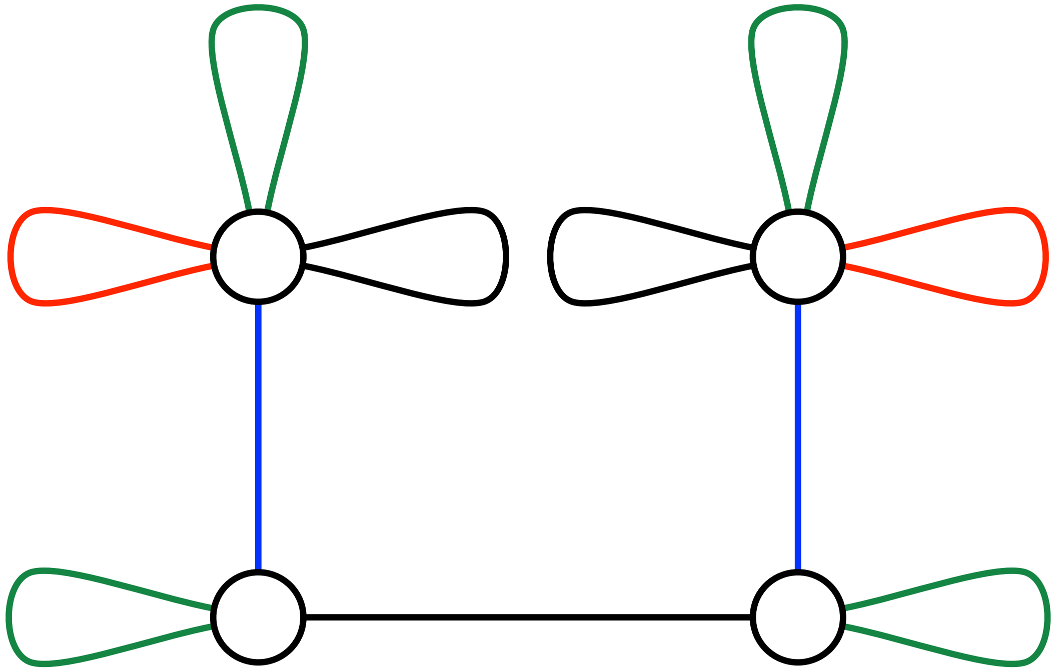
A loopy graph can be *unfolded* to get a simple graph

# Loopy graphs



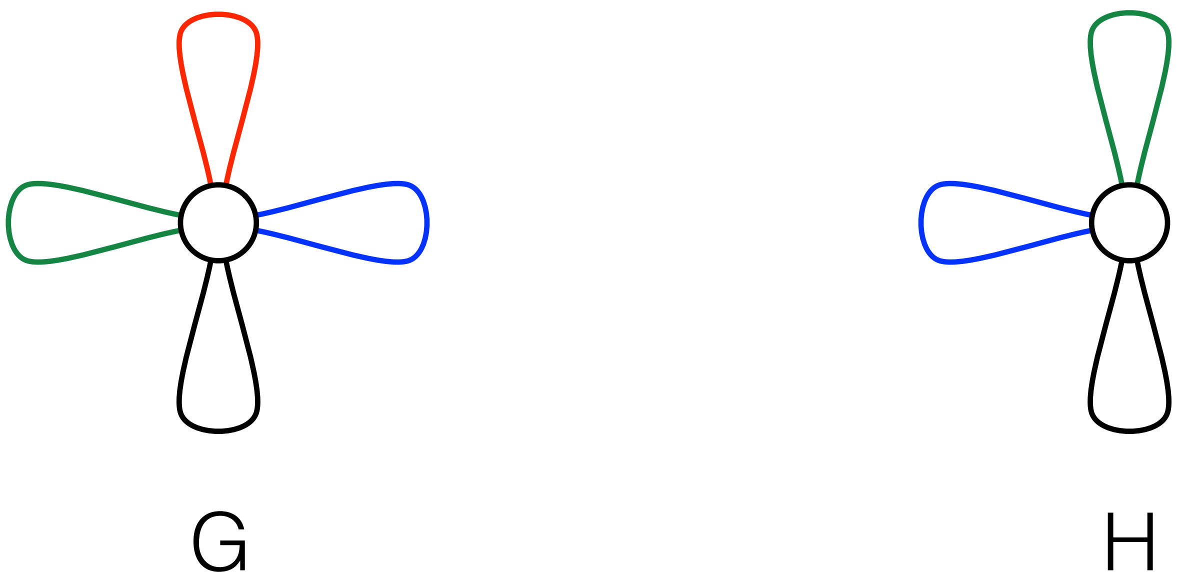
loopy graphs ≈ infinite trees

# Loopy graphs

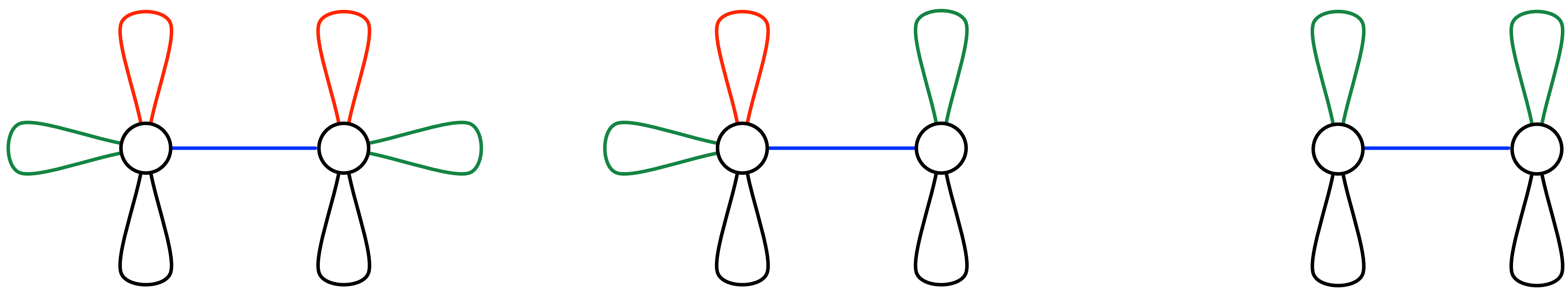


Key observation: a maximal fractional matching must saturate all nodes of a loopy graph!

# EC lower bound

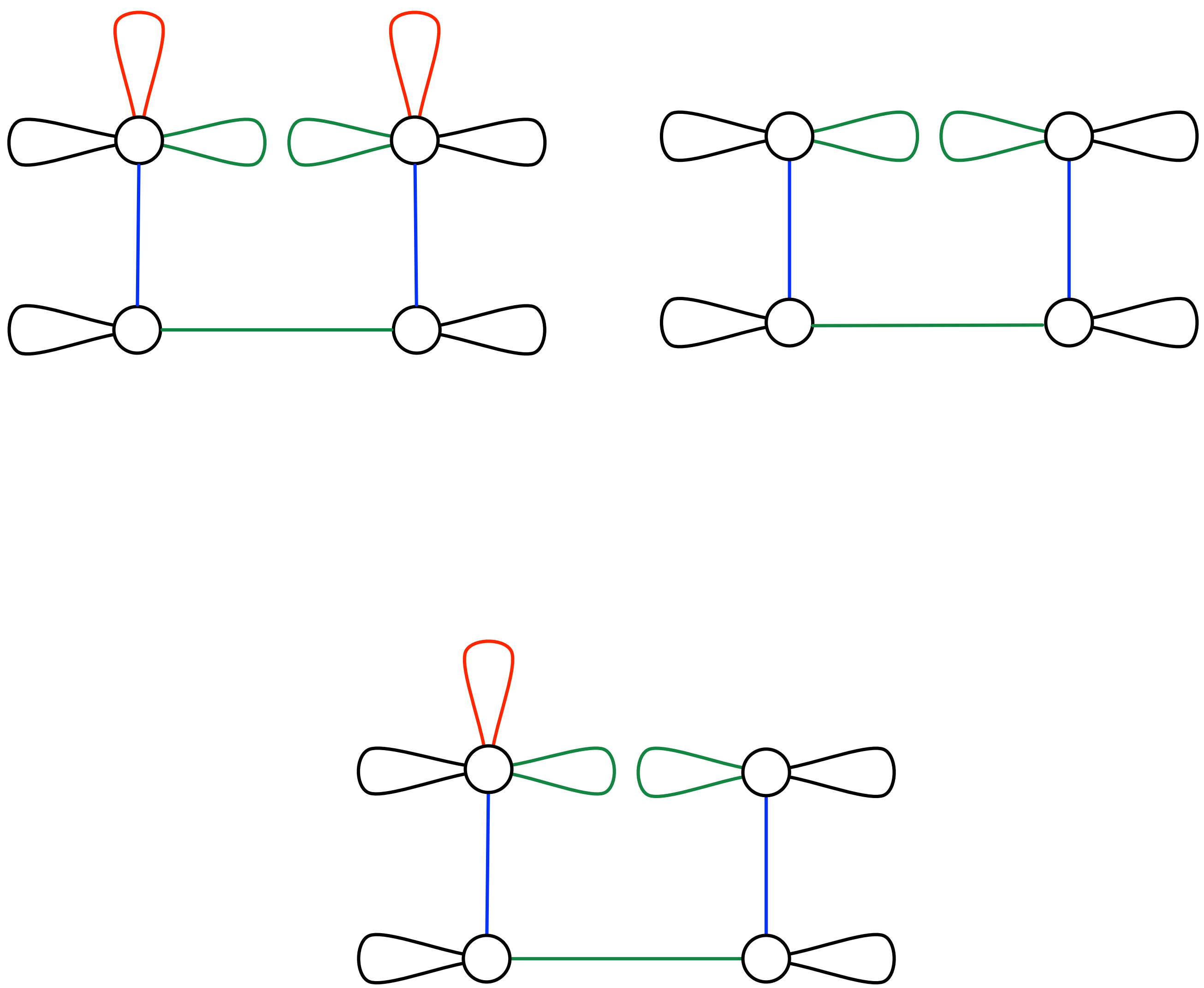


# EC lower bound



GG GH HH

# EC lower bound



# The Proof

A short guide to the proof

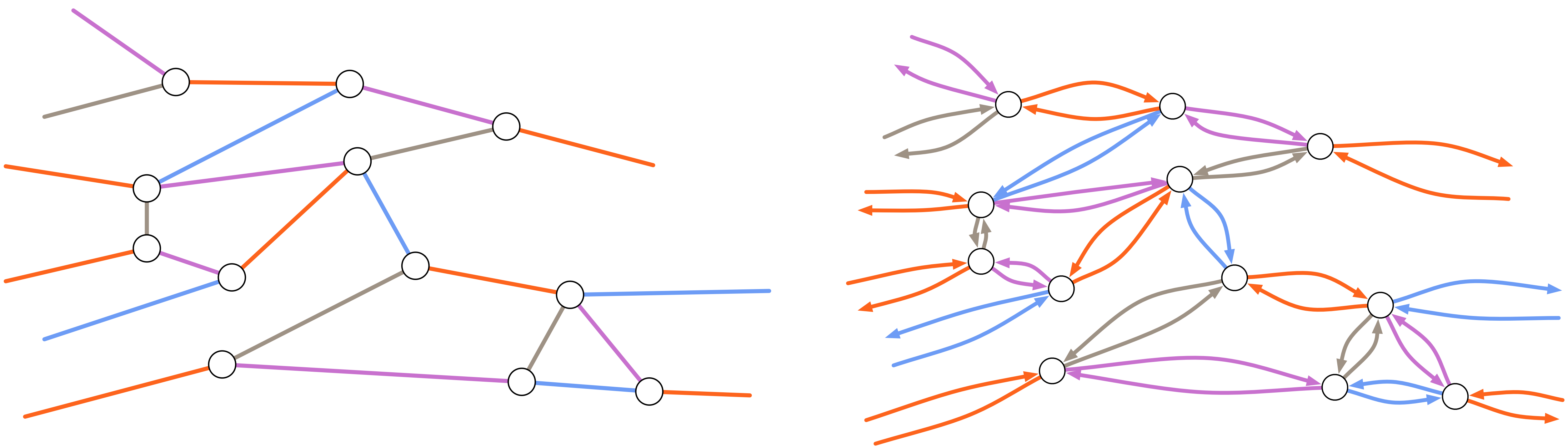
- Step 0: Introduce models EC, PO, OI and ID
- Step 1: $\Omega(\Delta)$-lower bound in the EC-model
- **Step 2: Simulation result EC⇝PO⇝OI⇝ID**
- Step 3: ID ⇝ Randomized algorithms

EC $\rightsquigarrow$ PO

# EC ⇝ PO

Assume we have an $o(\Delta)$-time algorithm **A** for maximal edge packing in the PO model

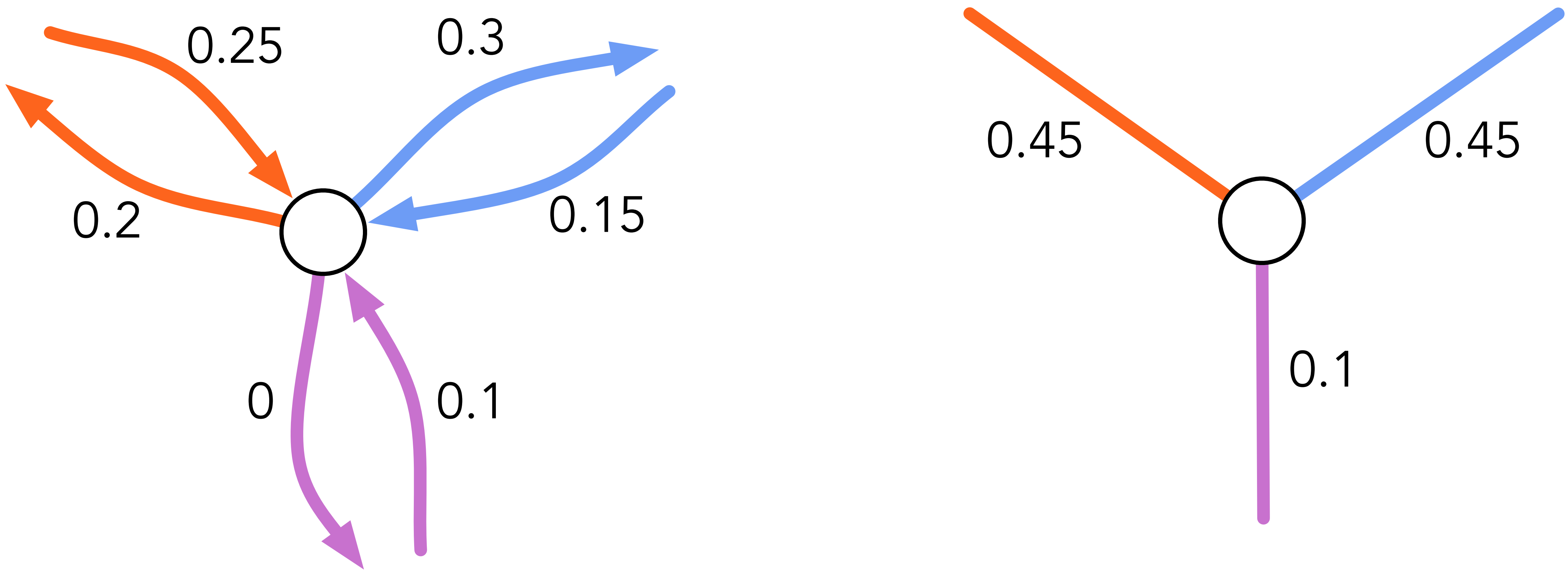
# EC → PO



Transform EC graph into PO graph by replacing each edge with two oriented edges

**EC → PO** → OI → ID → R

# EC → PO




Simulate the PO-algorithm **A** and combine the weights of the corresponding edges

# EC $\rightsquigarrow$ PO

We get an $o(\Delta)$-algorithm in the EC-model, which is a contradiction

PO $\rightsquigarrow$ OI

# PO ⇝ OI

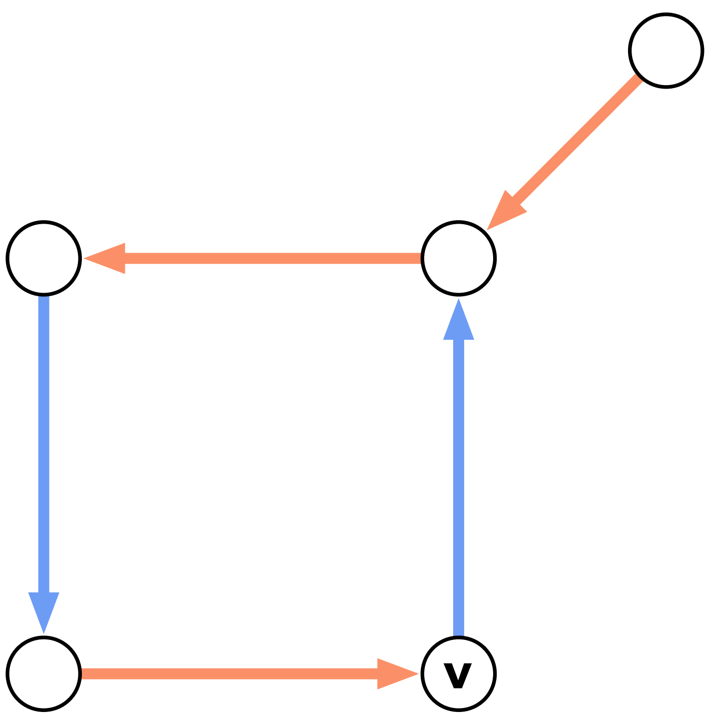
- Similar technology as Göös et al. (2012)
- Now we do not need any approximation guarantees

# PO ⇝ OI

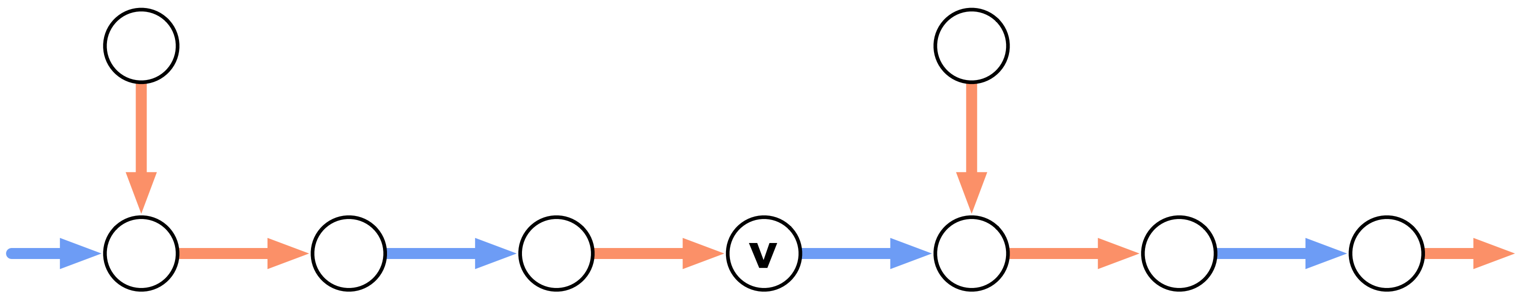
Assume we have a PO-algorithm **A**

We use port numbers and orientation to get a *local* ordering

# PO ⇝ OI




G

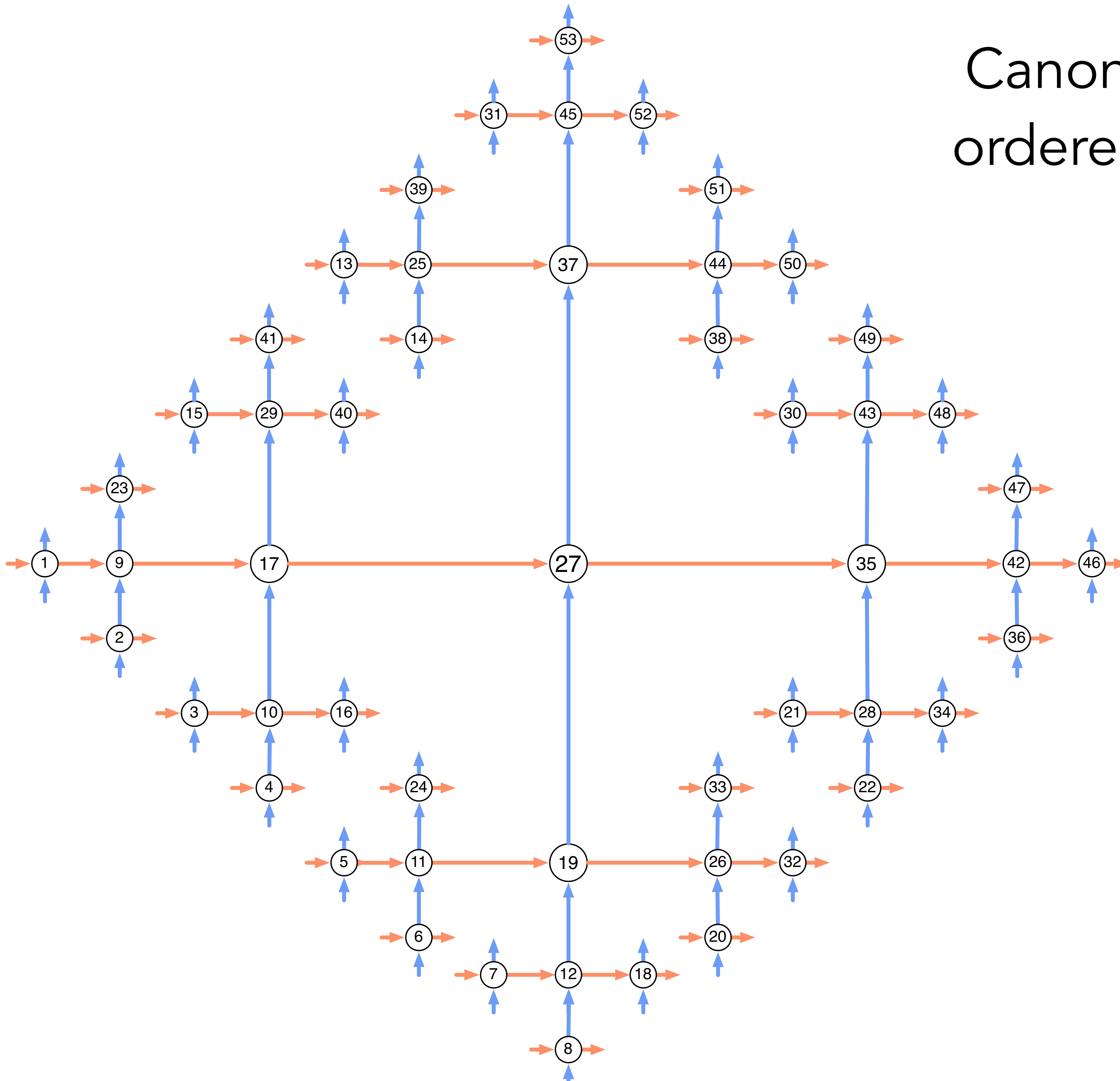



**U**(G)

Take the universal cover of G

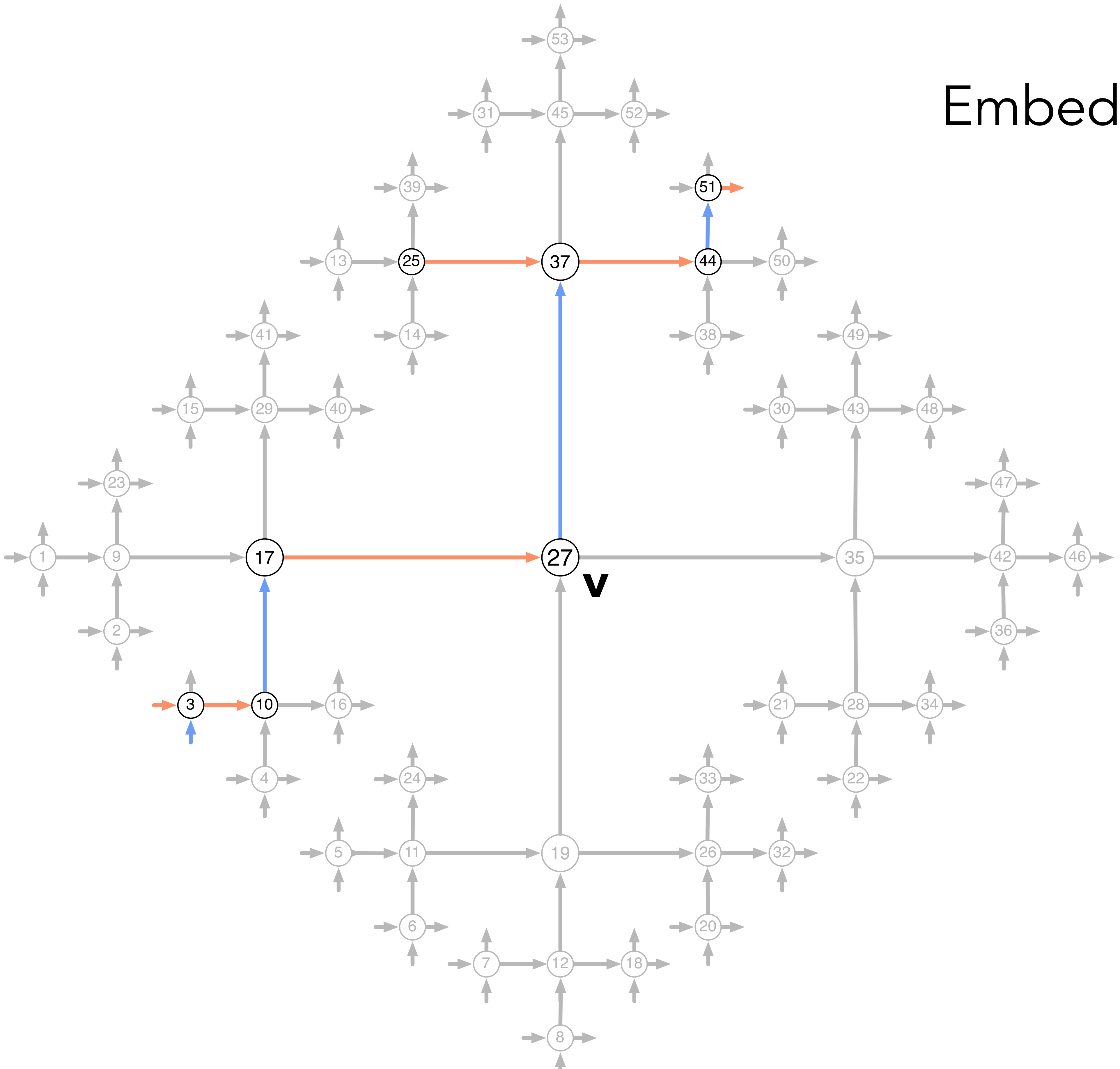
EC ⇝ **PO ⇝ OI** ⇝ ID ⇝ R

# Canonically ordered tree




EC → **PO** → **OI** → ID → R

# Embed **U**(G)




EC → **PO** → **OI** → ID → R

# PO ⇝ OI

It is possible to make a PO-graph an OI-graph locally

Use this to simulate **A**

OI $\rightsquigarrow$ ID

# OI ⇝ ID

Use the OI ⇝ ID lemma of
Naor and Stockmeyer (1995)
(essentially Ramsey's Theorem)

The idea is to force any ID-algorithm **A** to behave like an OI-algorithm on *some* inputs

# OI → ID

Trick: consider an algorithm **A*** that simulates **A** and outputs 1 at saturated nodes and 0 elsewhere to apply the Lemma

This forces *all nodes to be saturated* in **A** in loopy neighborhoods

Any change must propagate outside **A**'s run time

# The Proof

**A short guide**

- Step 0: Introduce models EC, PO, OI and ID
- Step 1: $\Omega(\Delta)$-lower bound in the EC-model
- Step 2: Simulation result EC→PO→OI→ID
- **Step 3: ID → Randomized algorithms**

# Randomized algorithms

Idea: Reduce random algorithms back to deterministic ones

Again use a lemma of Naor and Stockmeyer (1995)

EC → PO → OI → **ID → R**

# Summary

**This work**

Fractional maximal matching has complexity $\Theta(\Delta)$

**Open questions**

What is the complexity of *maximal matching*?

What is the complexity of *2-colored maximal matching*?