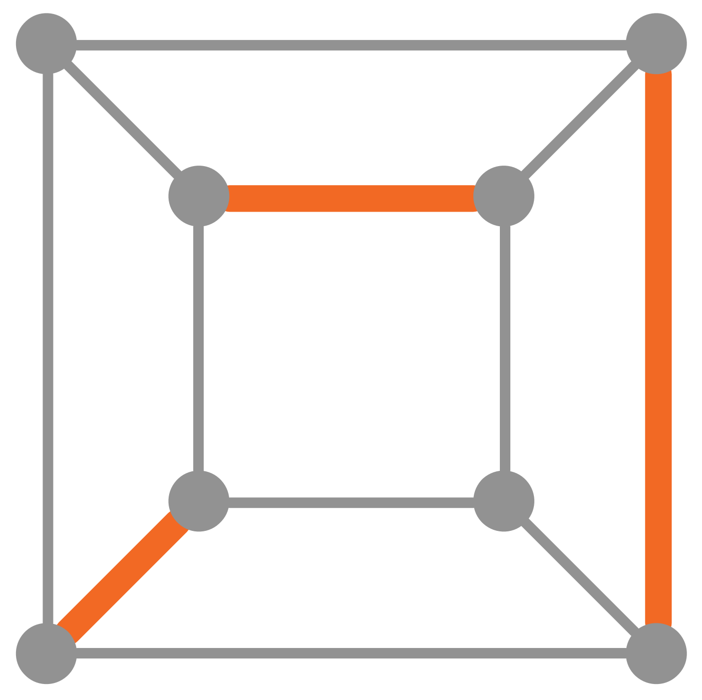
Lower Bounds for Maximal Matchings and Maximal Independent Sets

Alkida Balliu, **Sebastian Brandt**, Juho Hirvonen,
Dennis Olivetti, Mikaël Rabie, Jukka Suomela

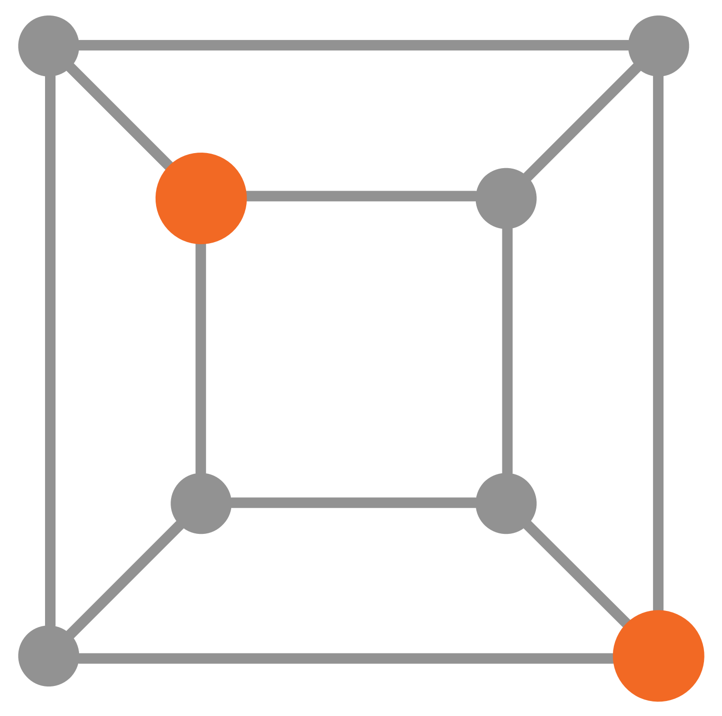
Aalto University, Finland
ETH Zurich, Switzerland
LIP6 - Sorbonne, France

Two classical graph problems

Maximal matching

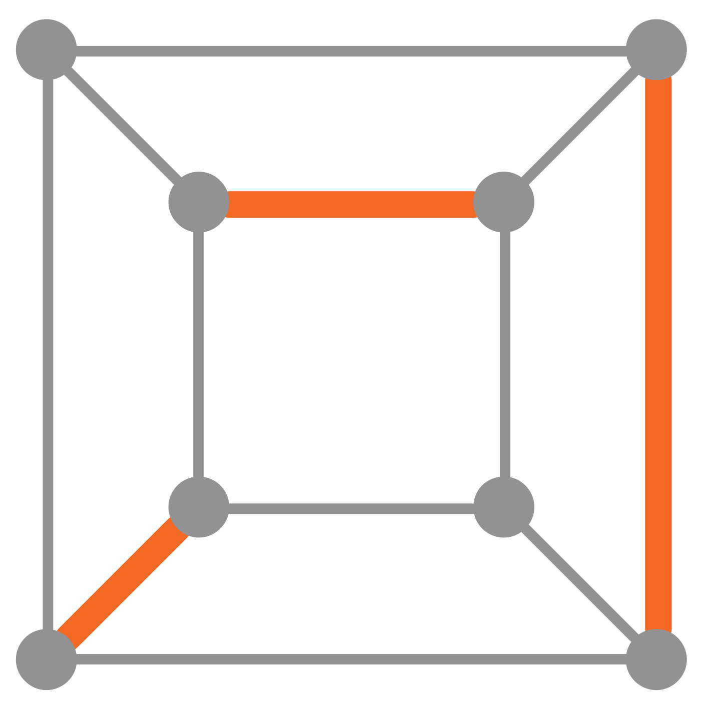


Maximal independent set

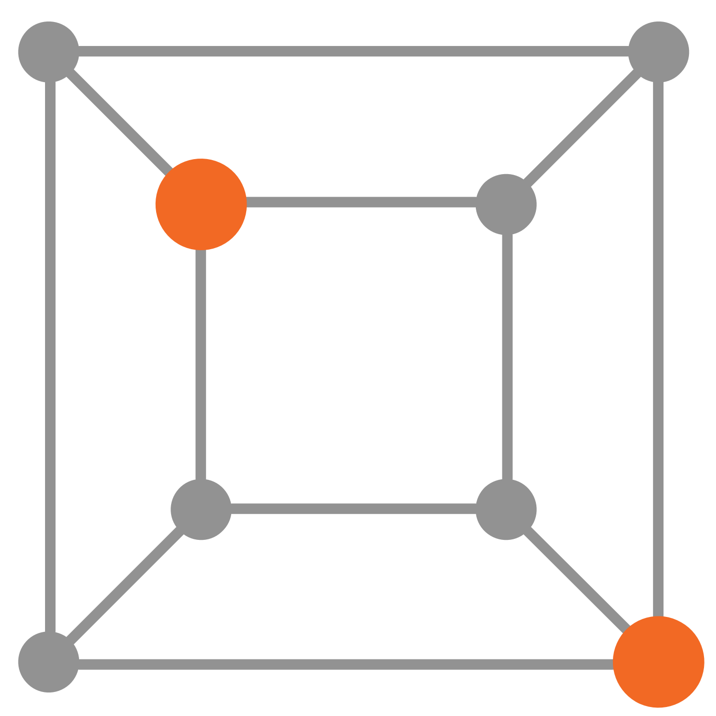


Two classical graph problems

Maximal matching



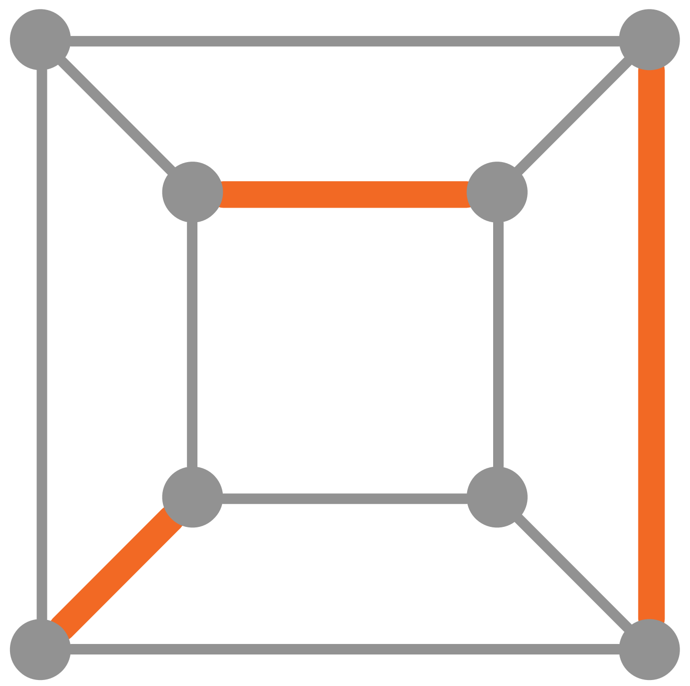
Maximal independent set



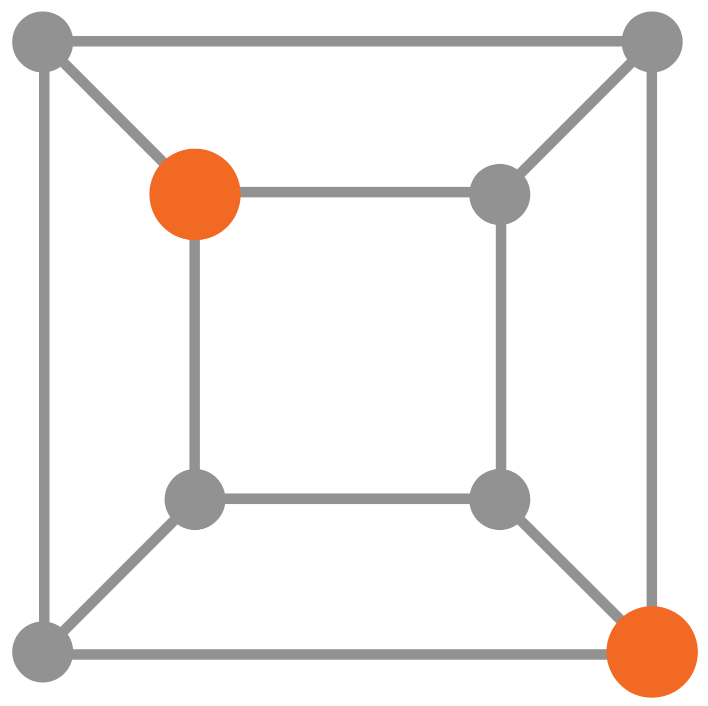
- Very ***easy*** to solve in the ***centralized*** setting: greedily add edges/nodes until not possible

Two classical graph problems

Maximal matching



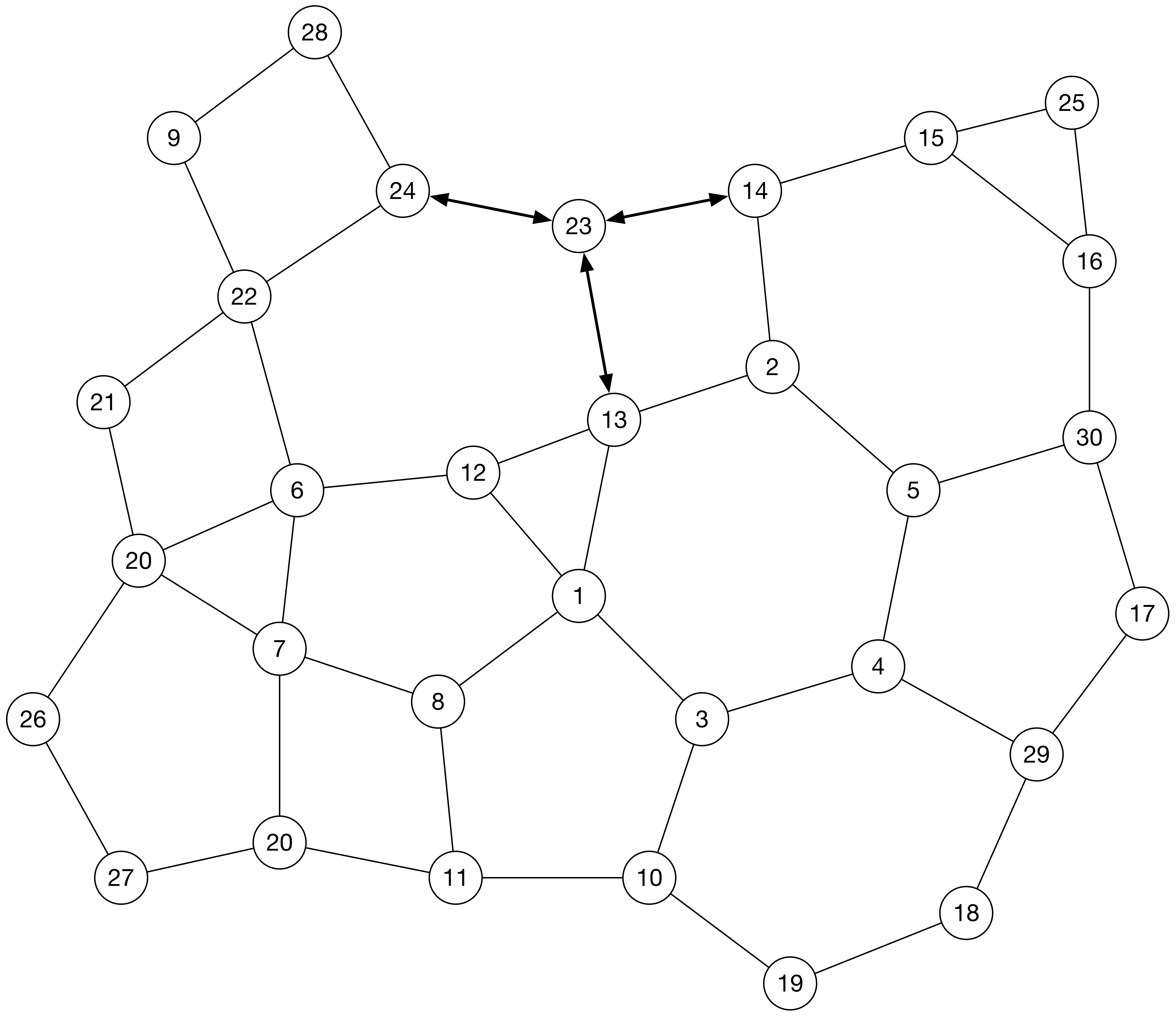
Maximal independent set



- Very ***easy*** to solve in the ***centralized*** setting: greedily add edges/nodes until not possible
- Can these problems be ***solved efficiently*** in a ***distributed*** setting?

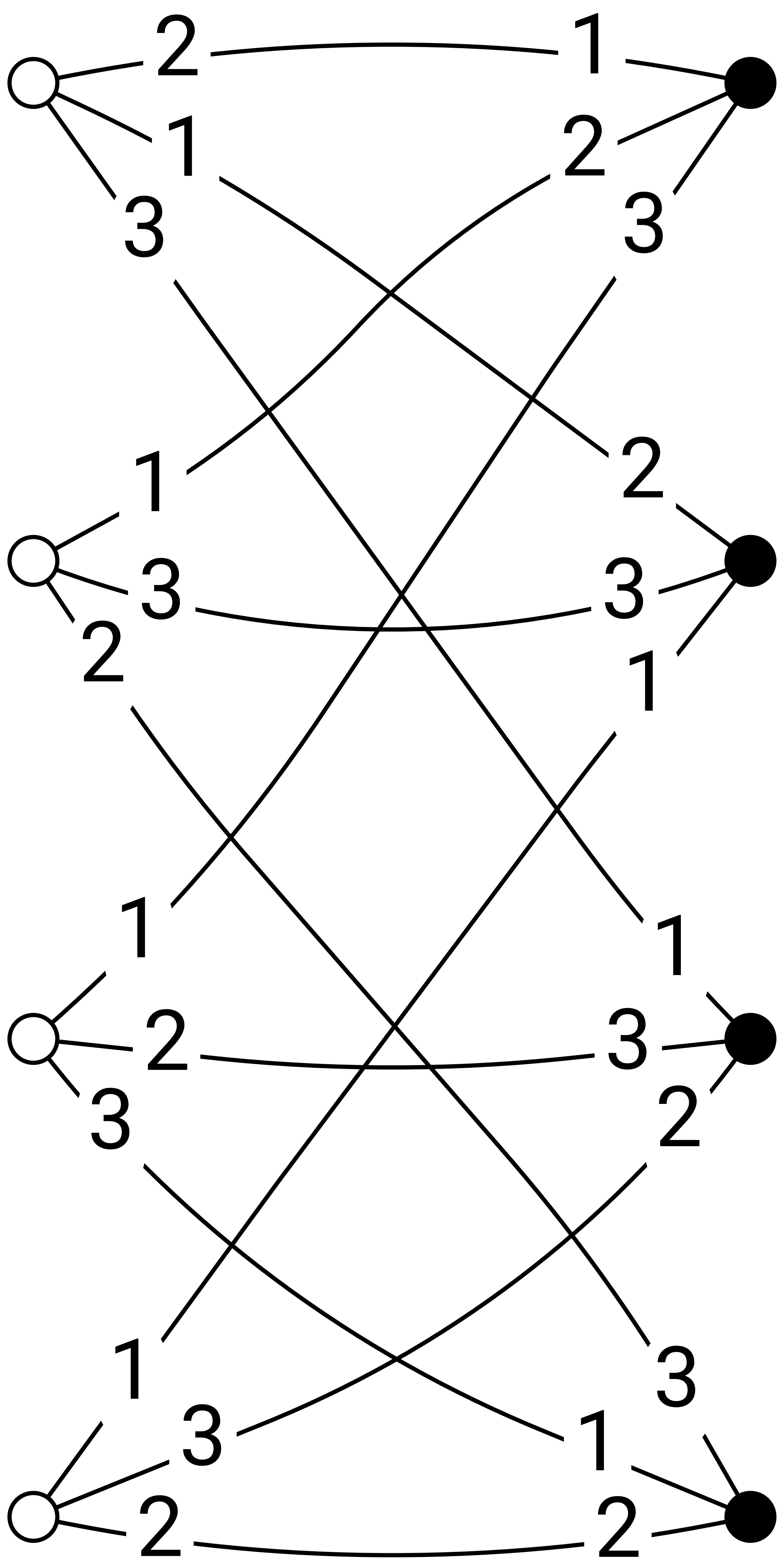
Distributed setting (LOCAL model)

- ***Graph*** = communication network
- ***Synchronous*** rounds
- Time complexity = ***number of rounds*** required to solve the problem
- Nodes have IDs



Simple scenario

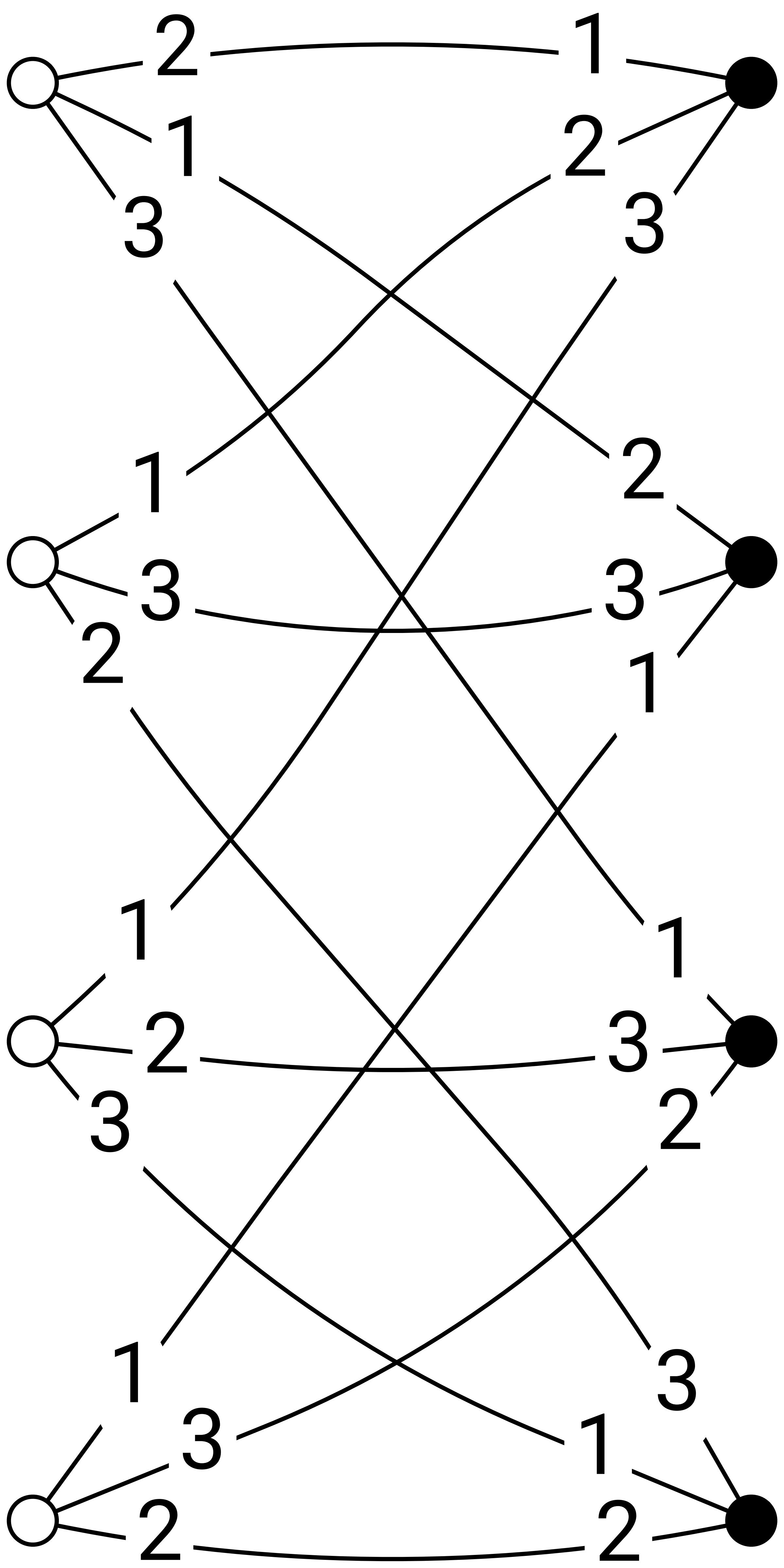
- Nodes are 2 colored
- The communication graph is Δ-regular



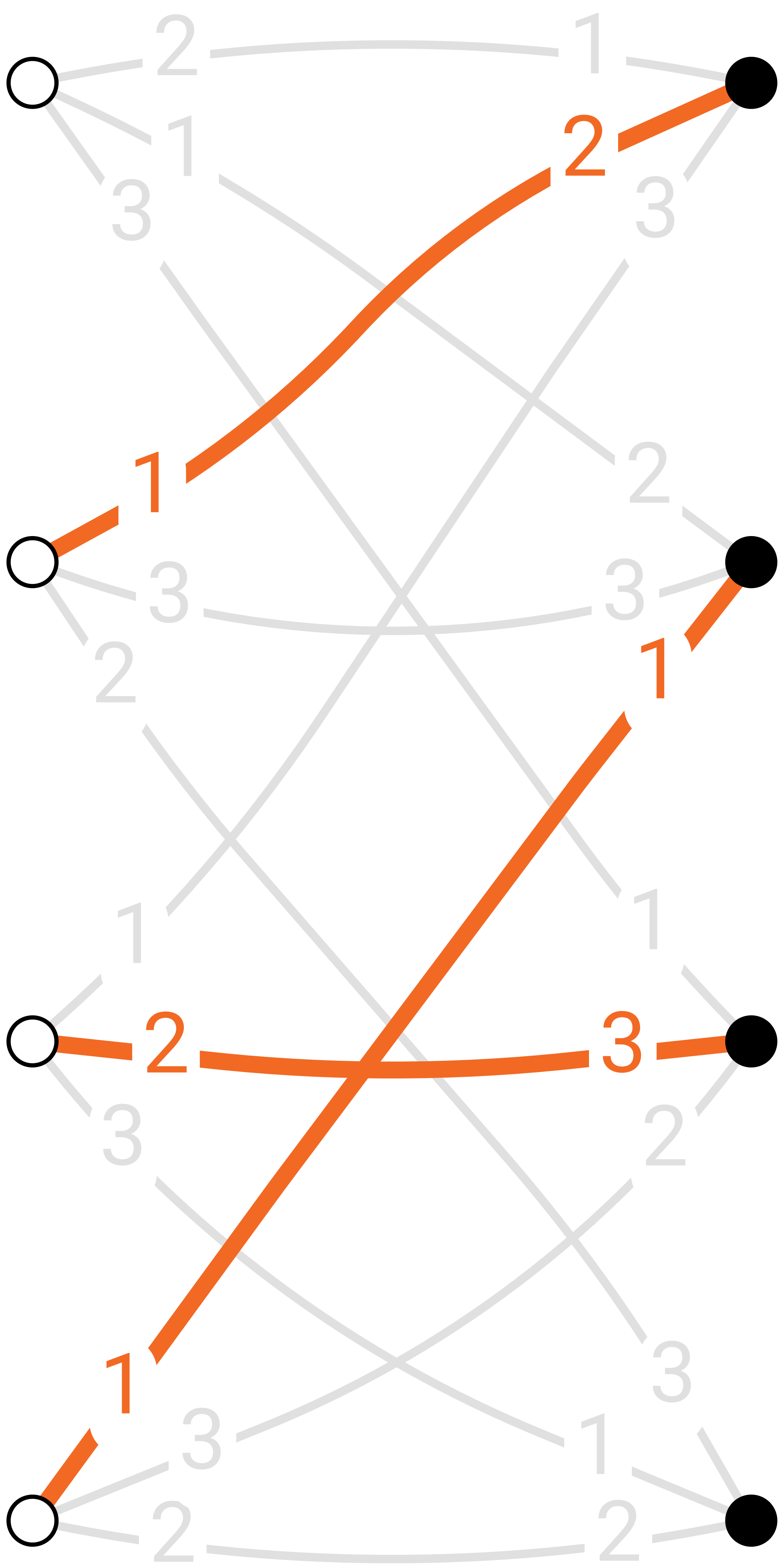
computer network with port numbering

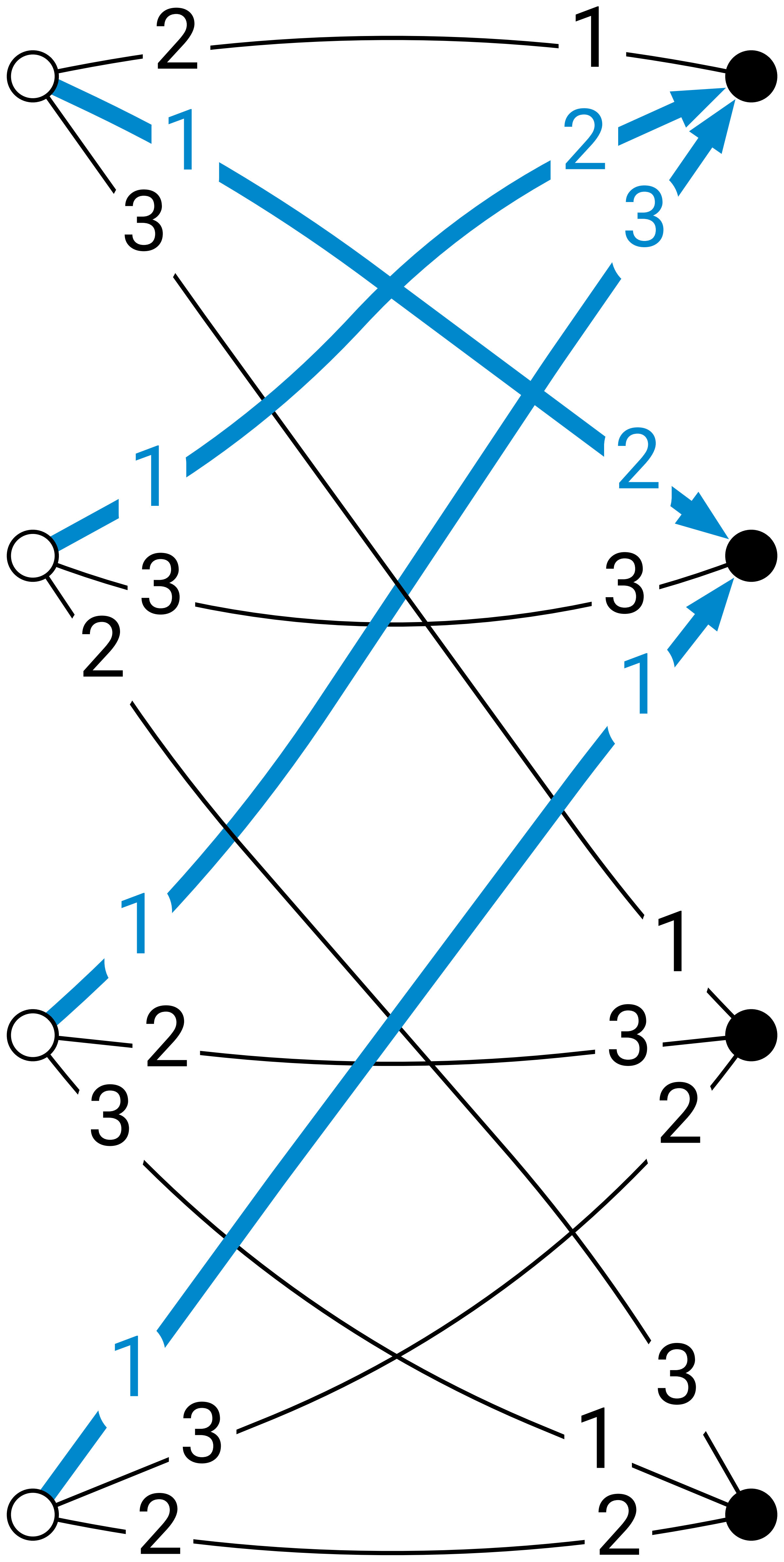
bipartite, 2-colored graph

Δ-regular (here Δ = 3)



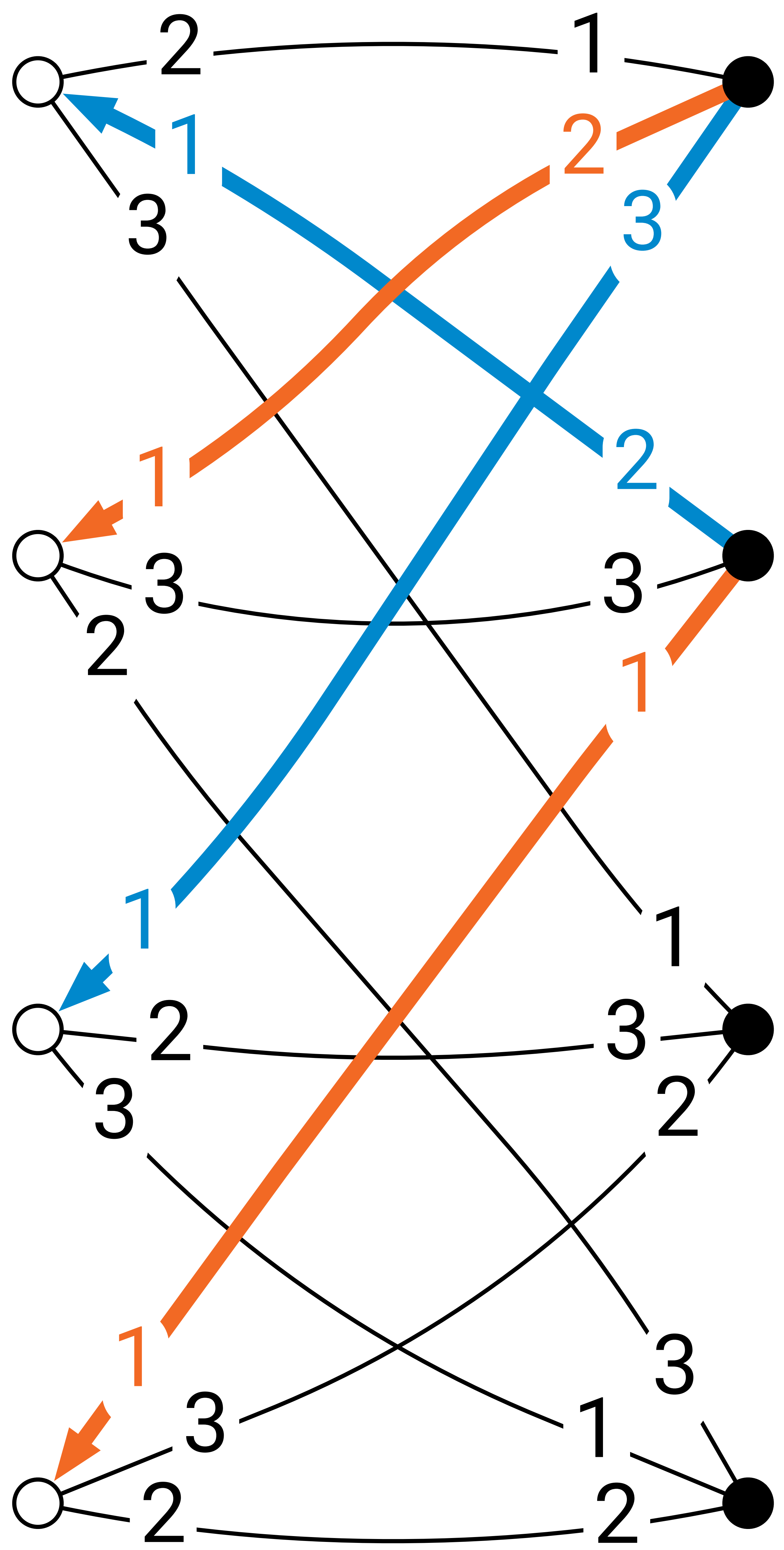
output: ***maximal matching***





Very simple algorithm

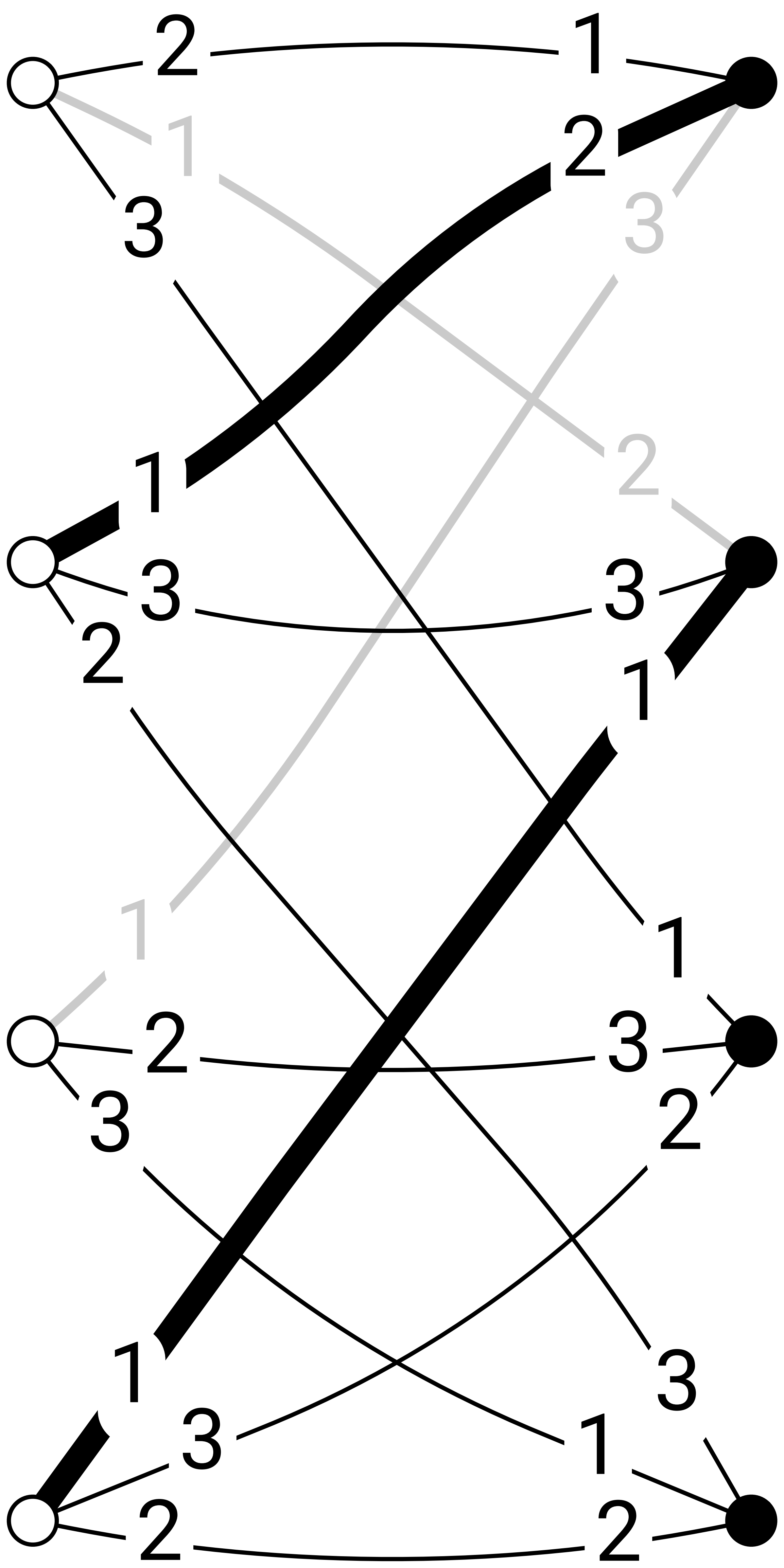
unmatched white nodes:
send ***proposal*** to port 1



Very simple algorithm

unmatched white nodes:
send ***proposal*** to port 1

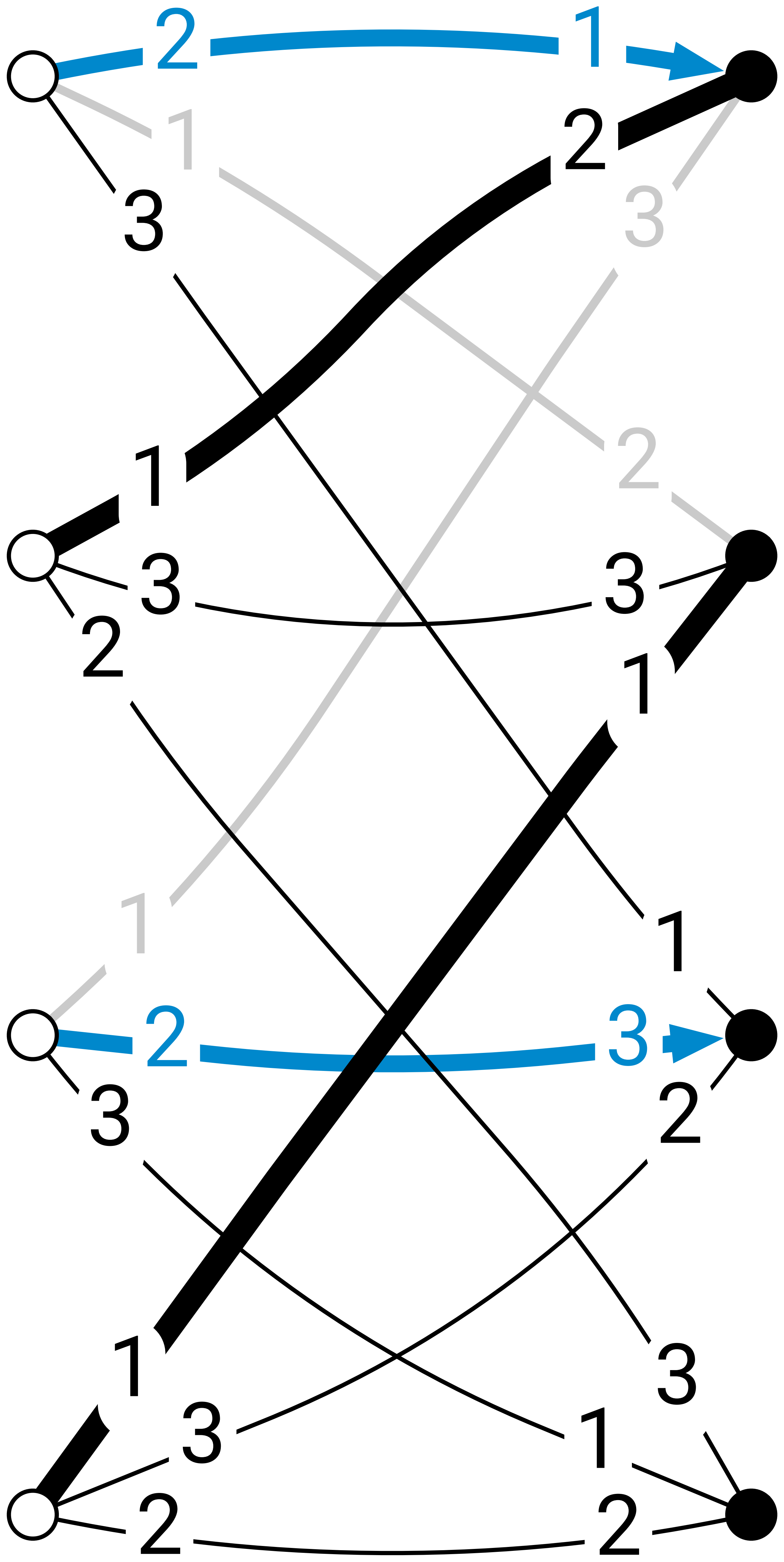
black nodes:
accept the first proposal you get, ***reject*** everything else (break ties with port numbers)



Very simple algorithm

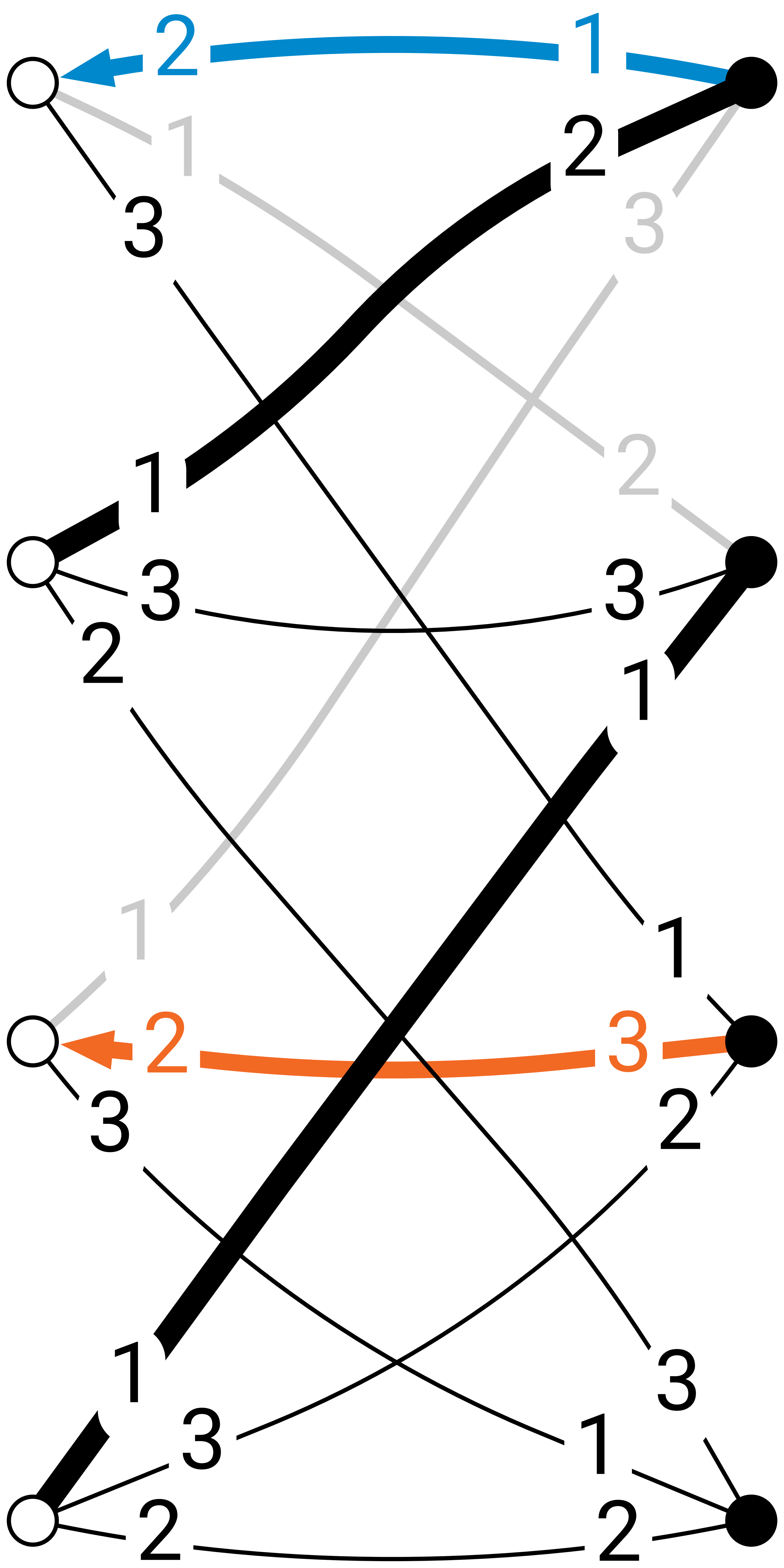
unmatched white nodes:
send ***proposal*** to port 1

black nodes:
accept the first proposal you get, ***reject*** everything else (break ties with port numbers)



Very simple algorithm

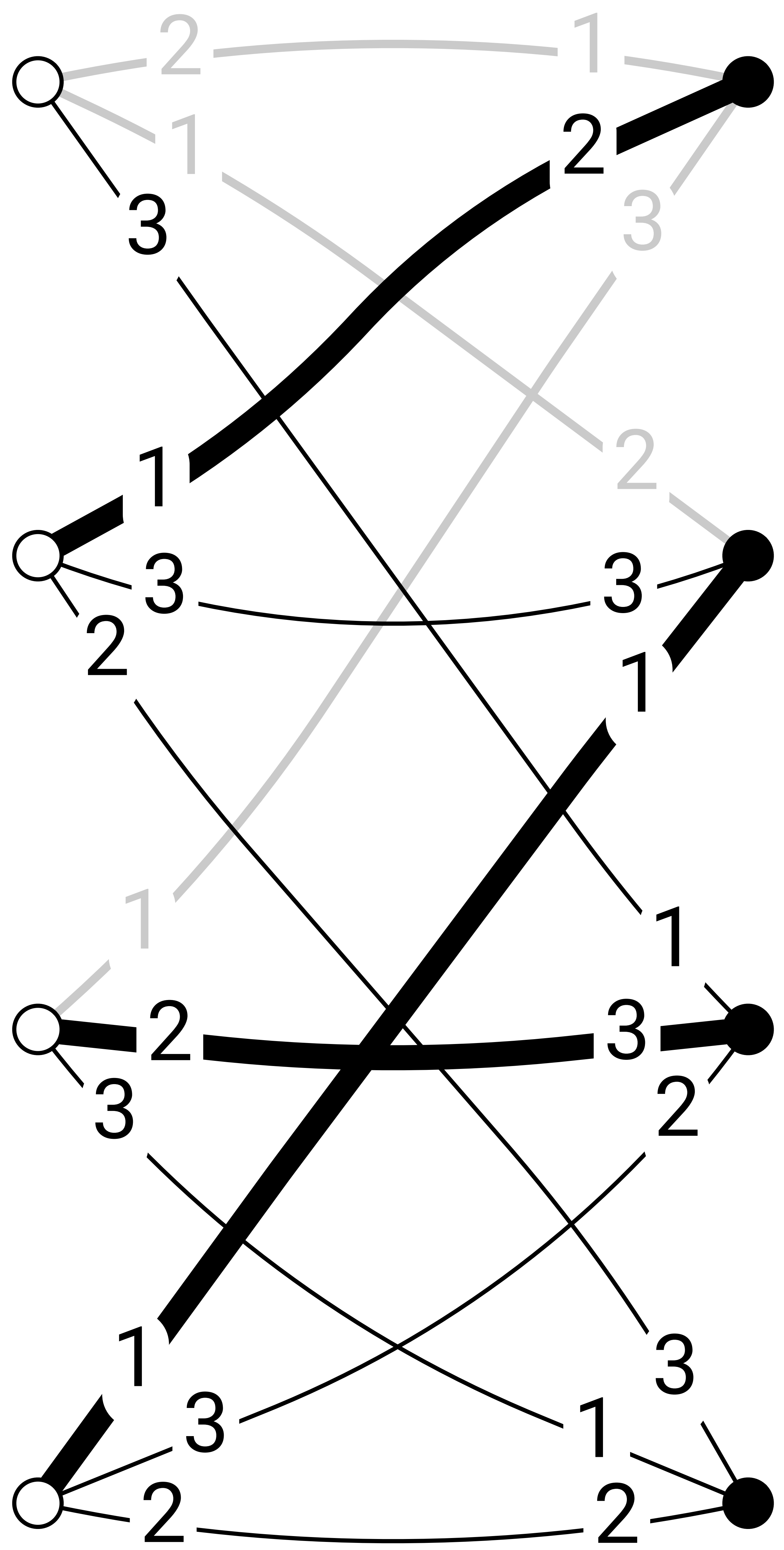
unmatched white nodes:
send ***proposal*** to port 2



Very simple algorithm

unmatched white nodes:
send ***proposal*** to port 2

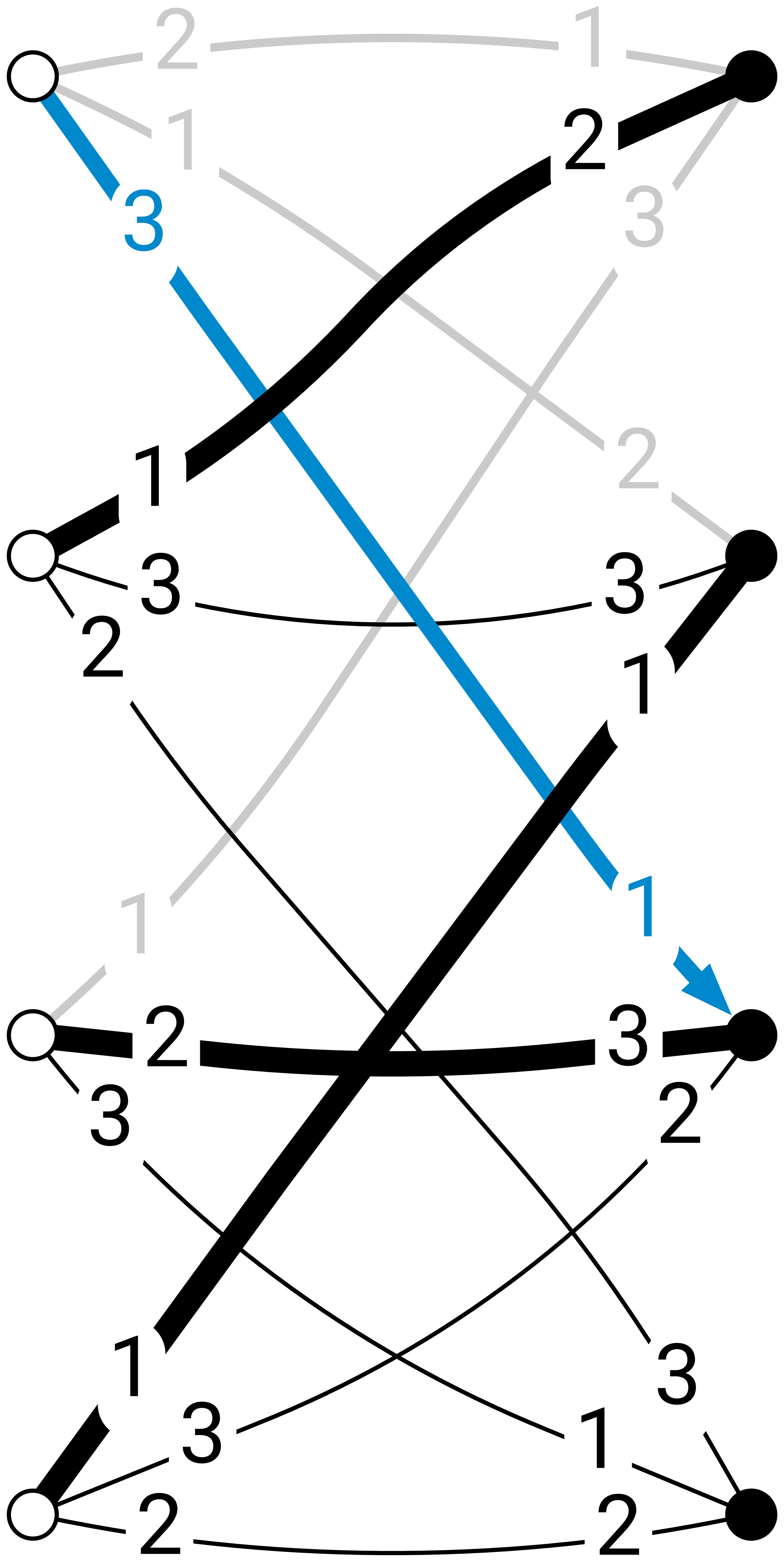
black nodes:
accept the first proposal you get, ***reject*** everything else (break ties with port numbers)



Very simple algorithm

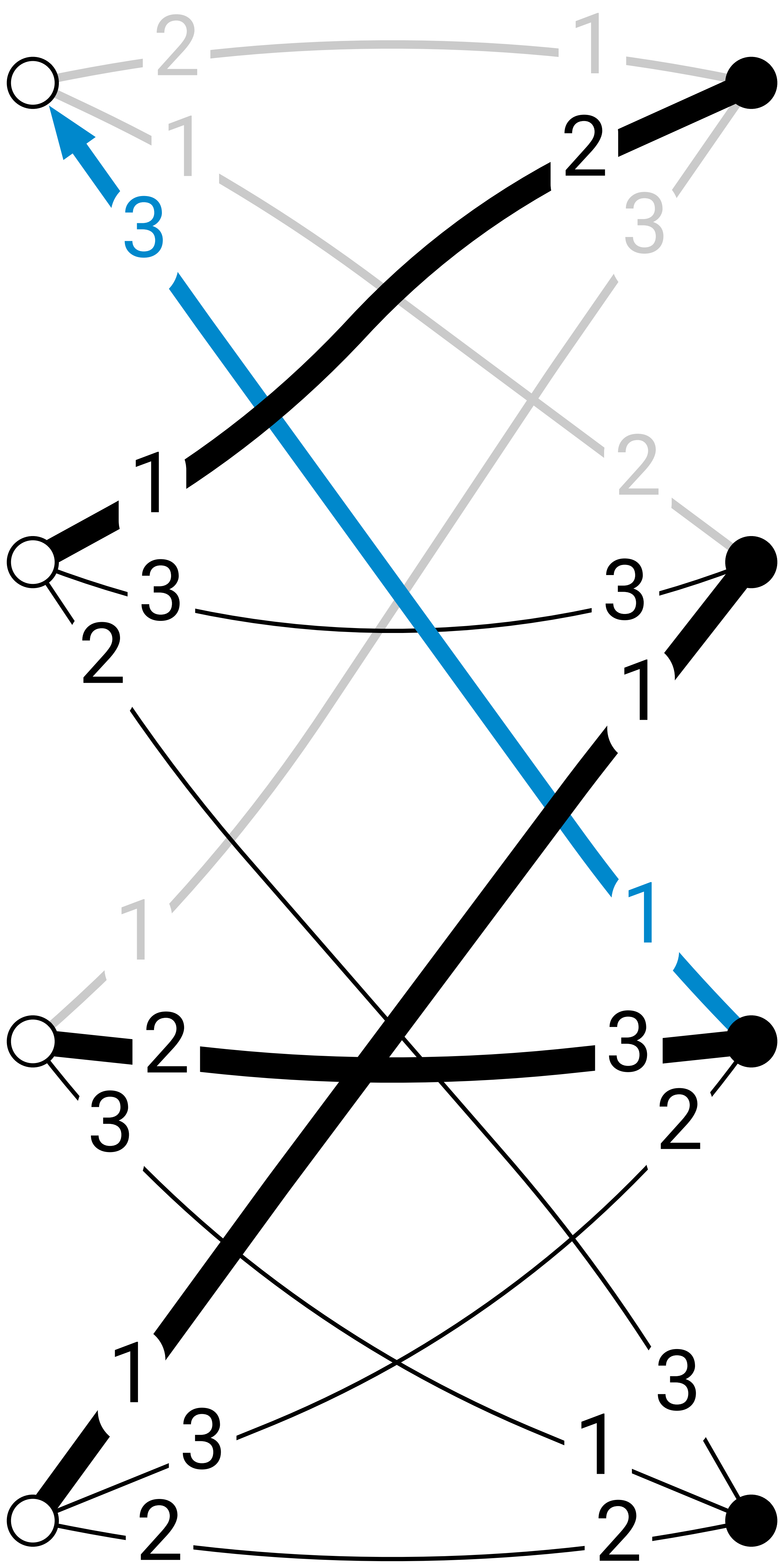
unmatched white nodes:
send ***proposal*** to port 2

black nodes:
accept the first proposal you get, ***reject*** everything else (break ties with port numbers)



Very simple algorithm

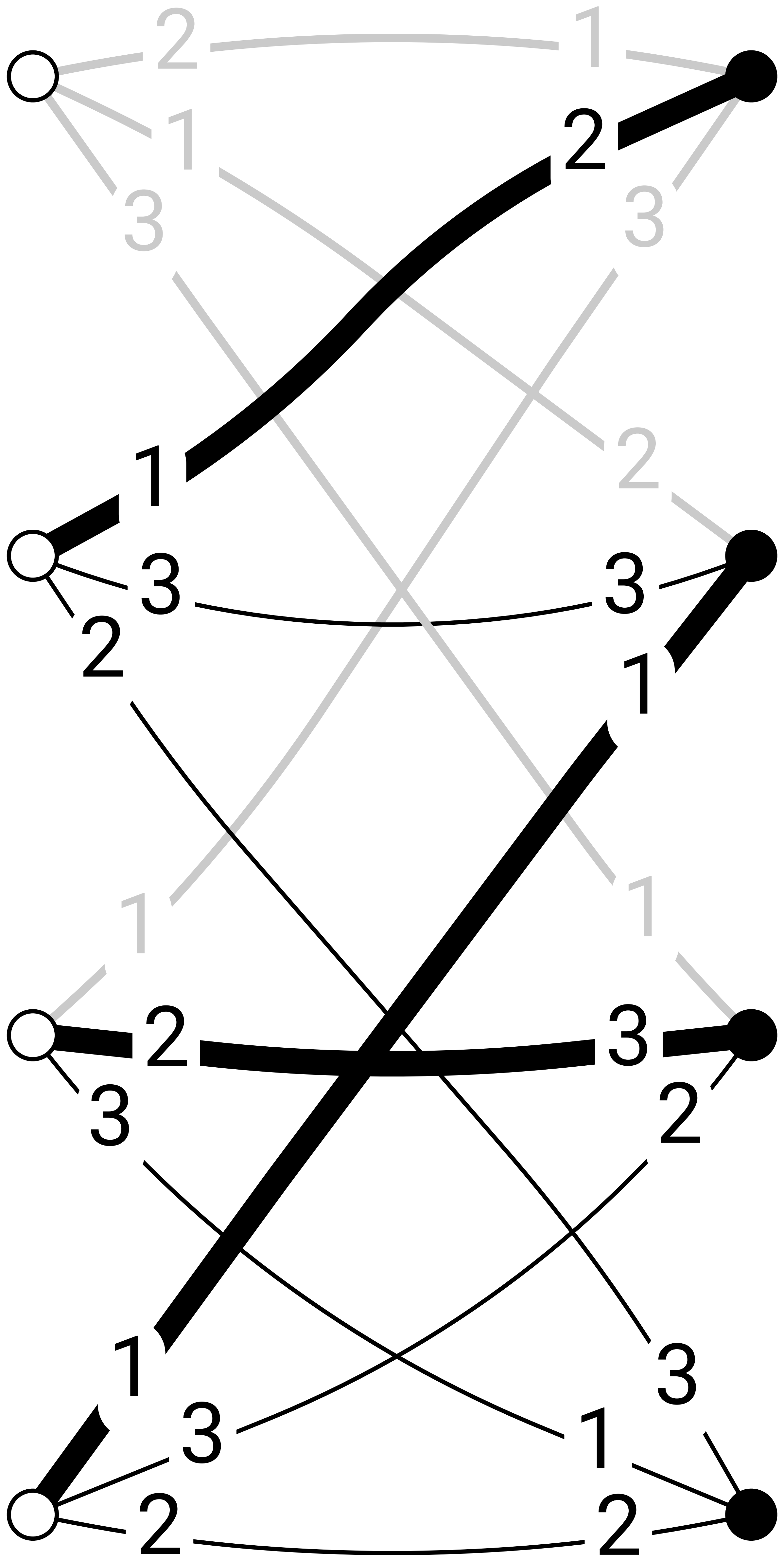
unmatched white nodes:
send ***proposal*** to port 3



Very simple algorithm

unmatched white nodes:
send ***proposal*** to port 3

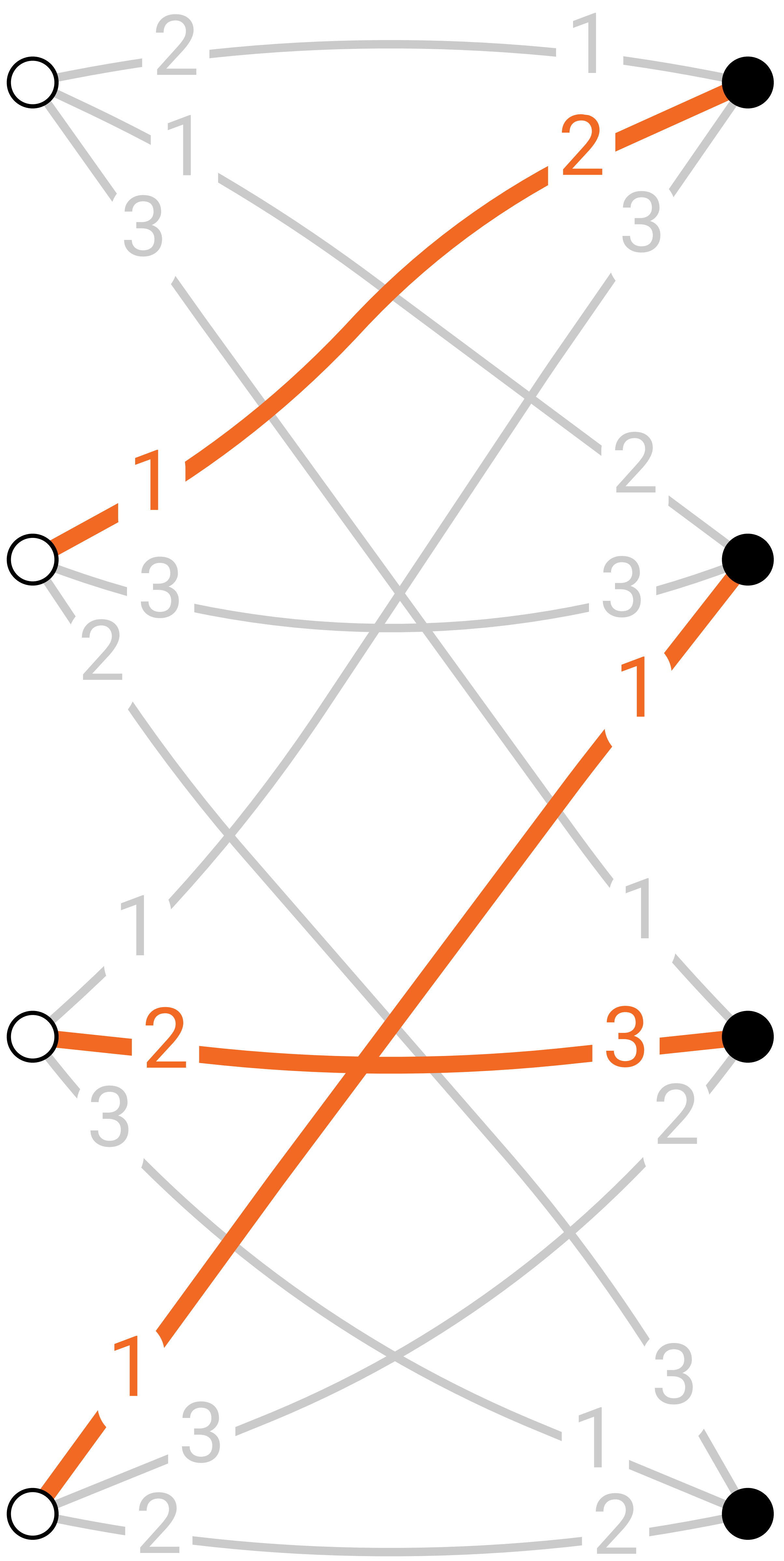
black nodes:
accept the first proposal you get, ***reject*** everything else
(break ties with port numbers)



Very simple algorithm

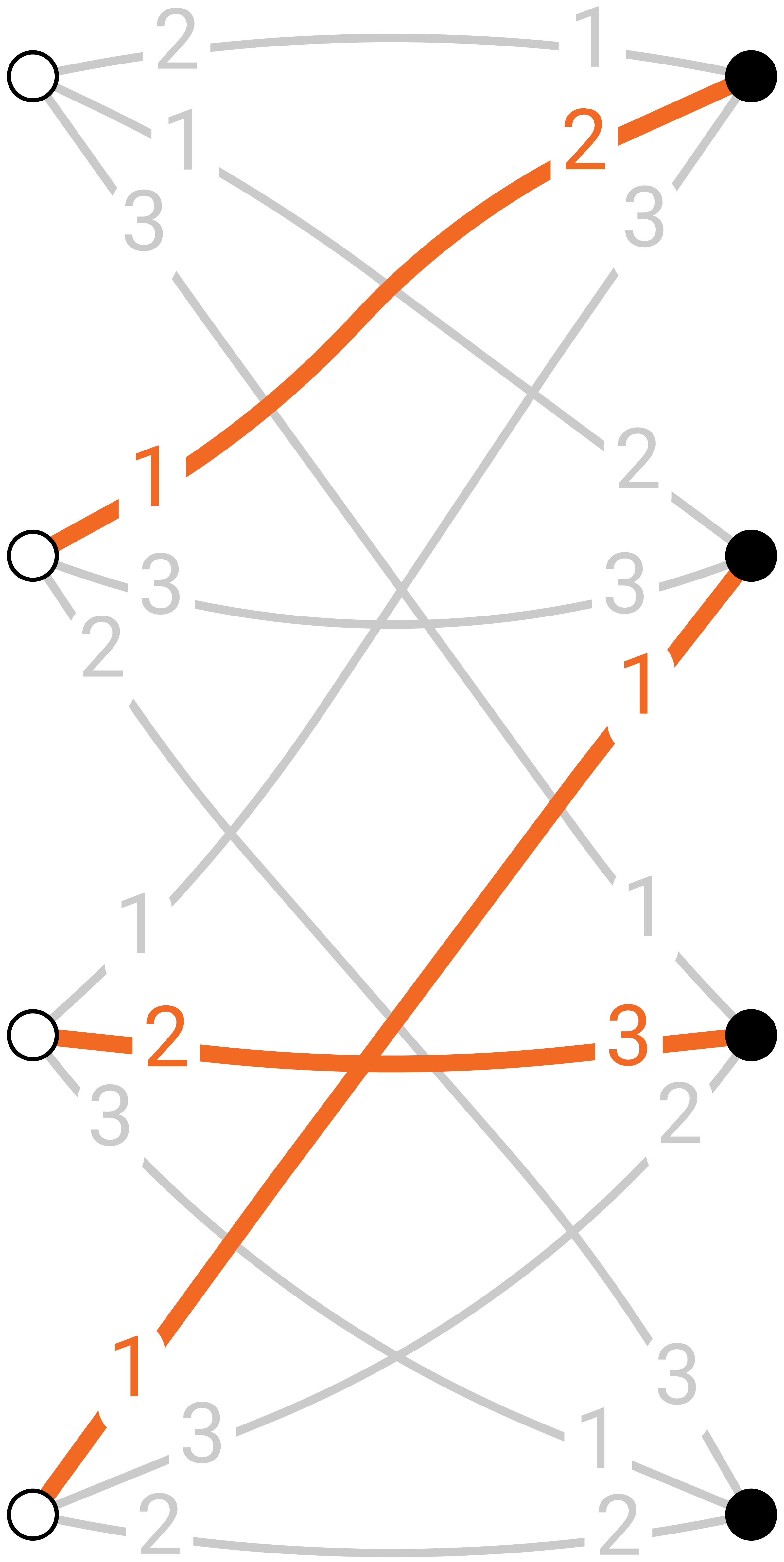
unmatched white nodes:
send ***proposal*** to port 3

black nodes:
accept the first proposal you get, ***reject*** everything else (break ties with port numbers)



Very simple algorithm

Finds a ***maximal matching*** in $O(\Delta)$ communication rounds

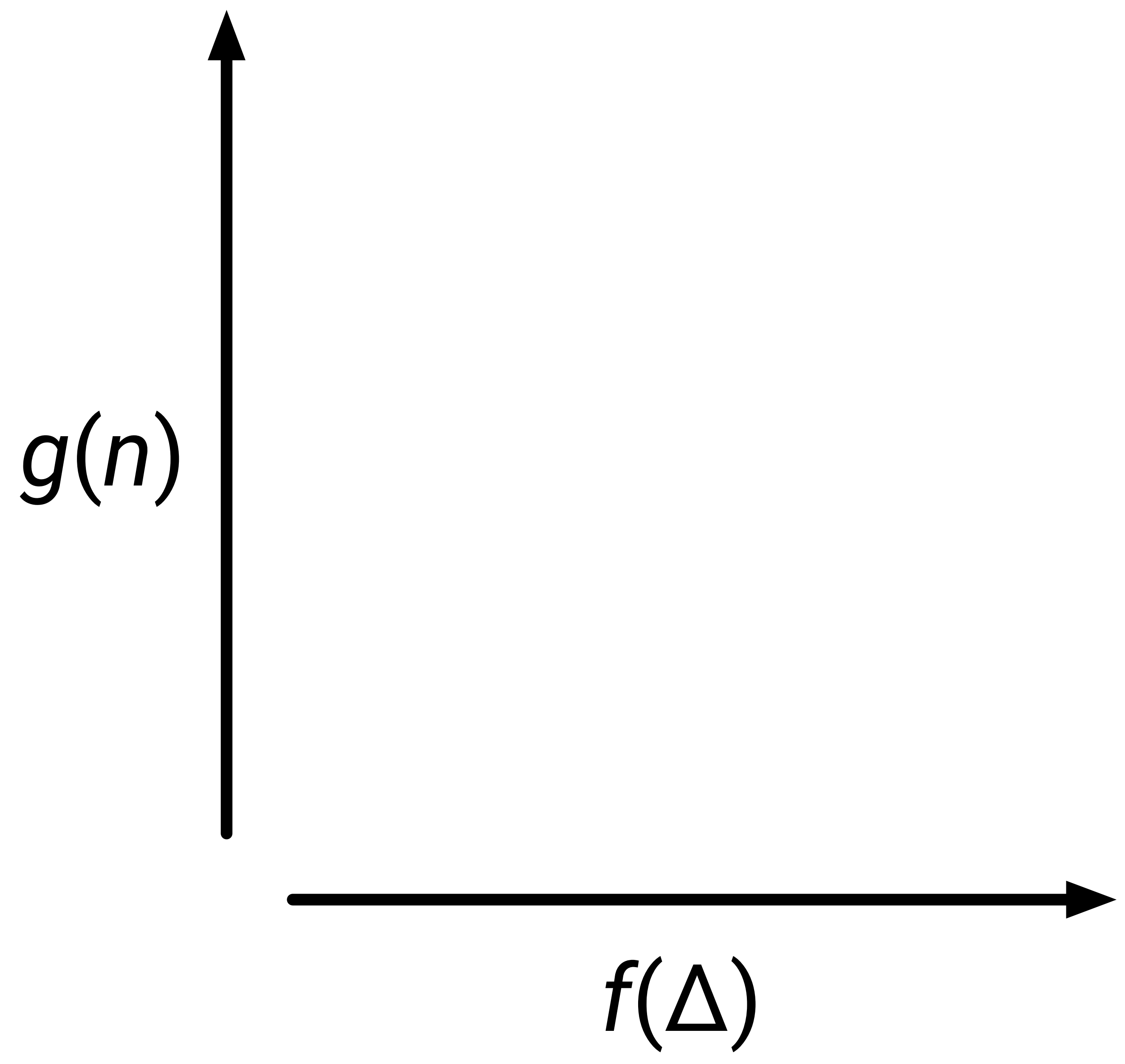


Very simple algorithm

Finds a ***maximal matching*** in $O(\Delta)$ communication rounds

This is optimal!

Related work



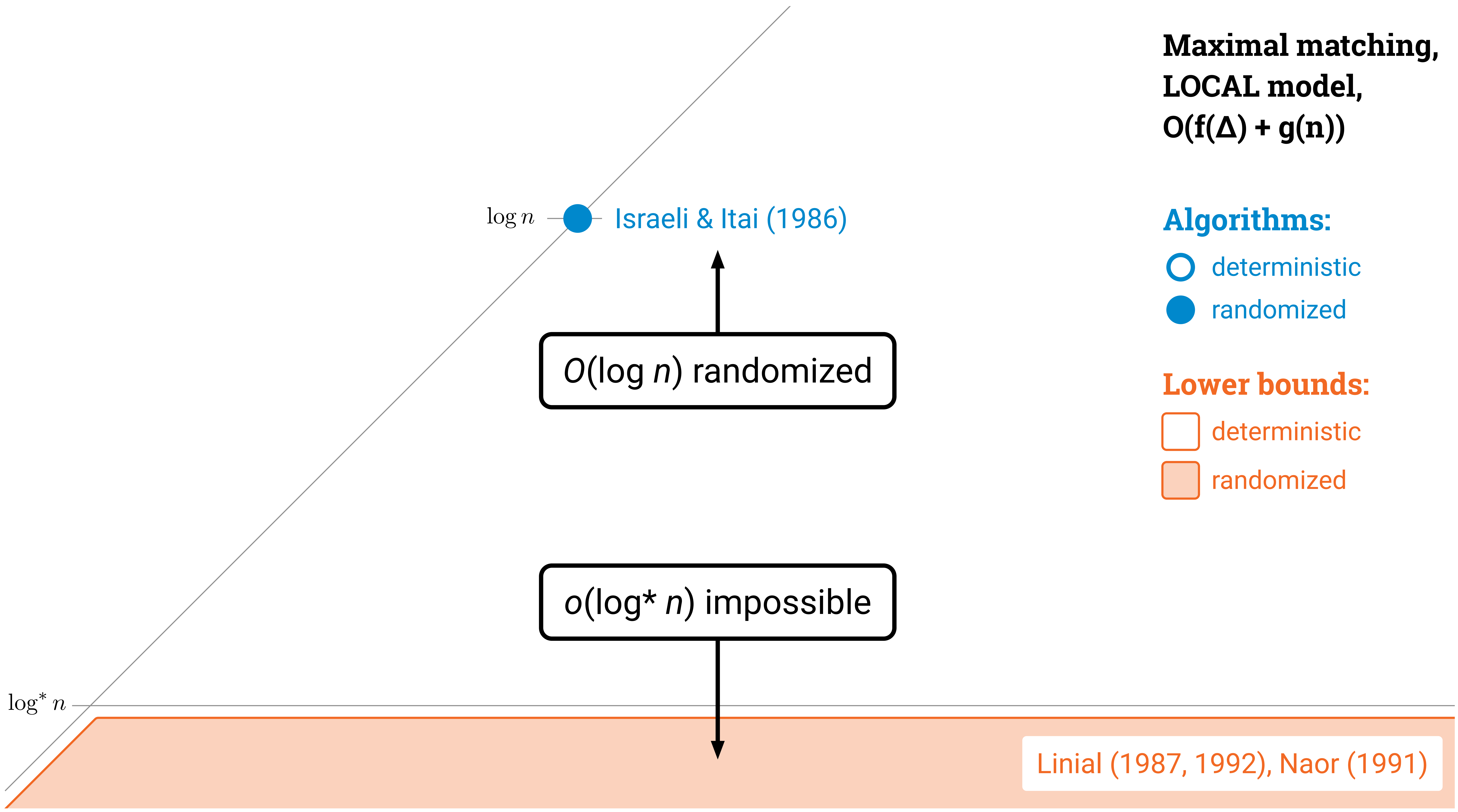
Maximal matching, LOCAL model, $O(f(\Delta) + g(n))$

Algorithms:

- deterministic
- randomized

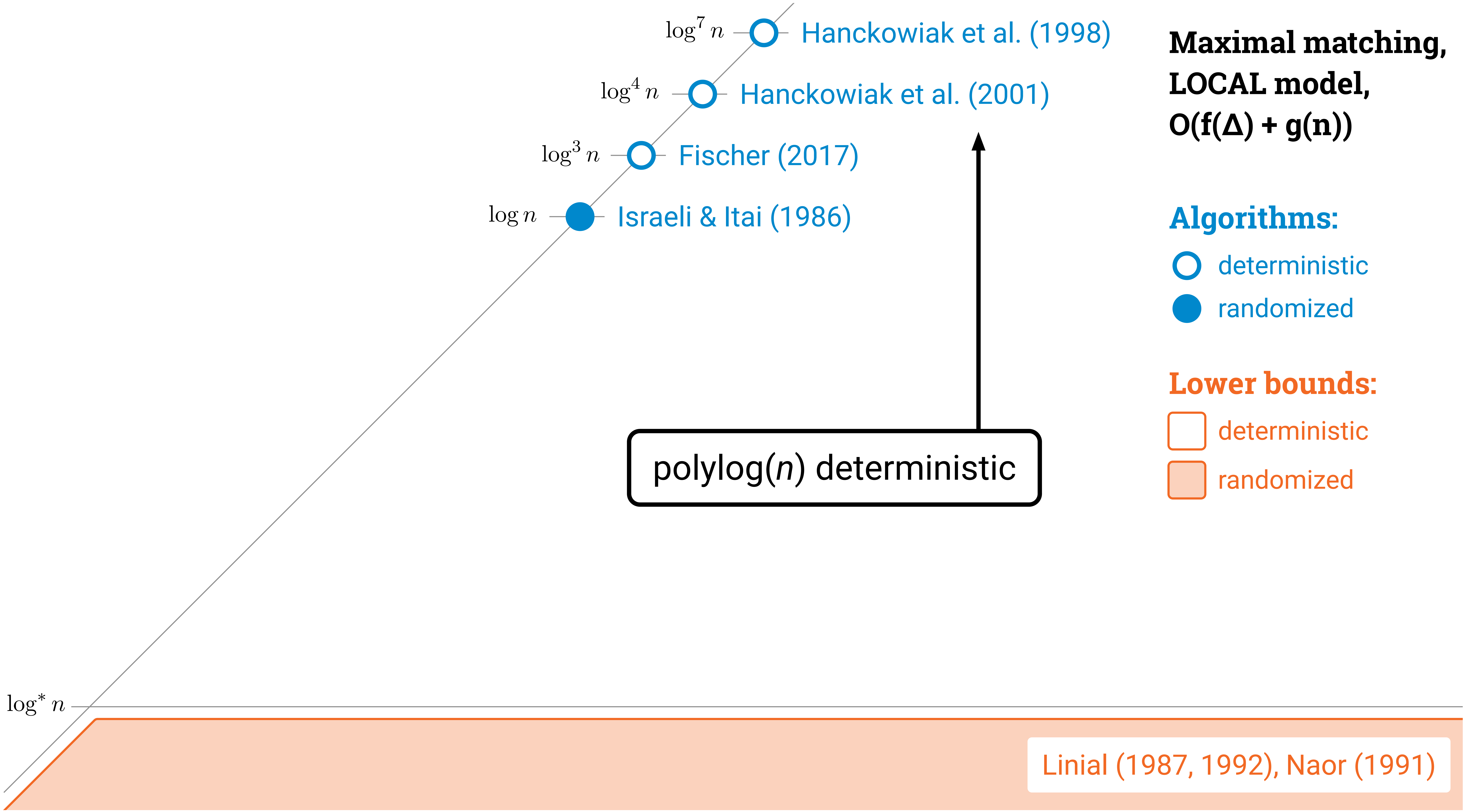
Lower bounds:

- deterministic
- randomized



Maximal matching,
LOCAL model,
O(f(Δ) + g(n))
Algorithms:
deterministic
randomized
Lower bounds:
deterministic
randomized
log n
Israeli & Itai (1986)
O(log n) randomized
o(log* n) impossible
log* n
Linial (1987, 1992), Naor (1991)

Maximal matching, LOCAL model, O(f(Δ) + g(n))

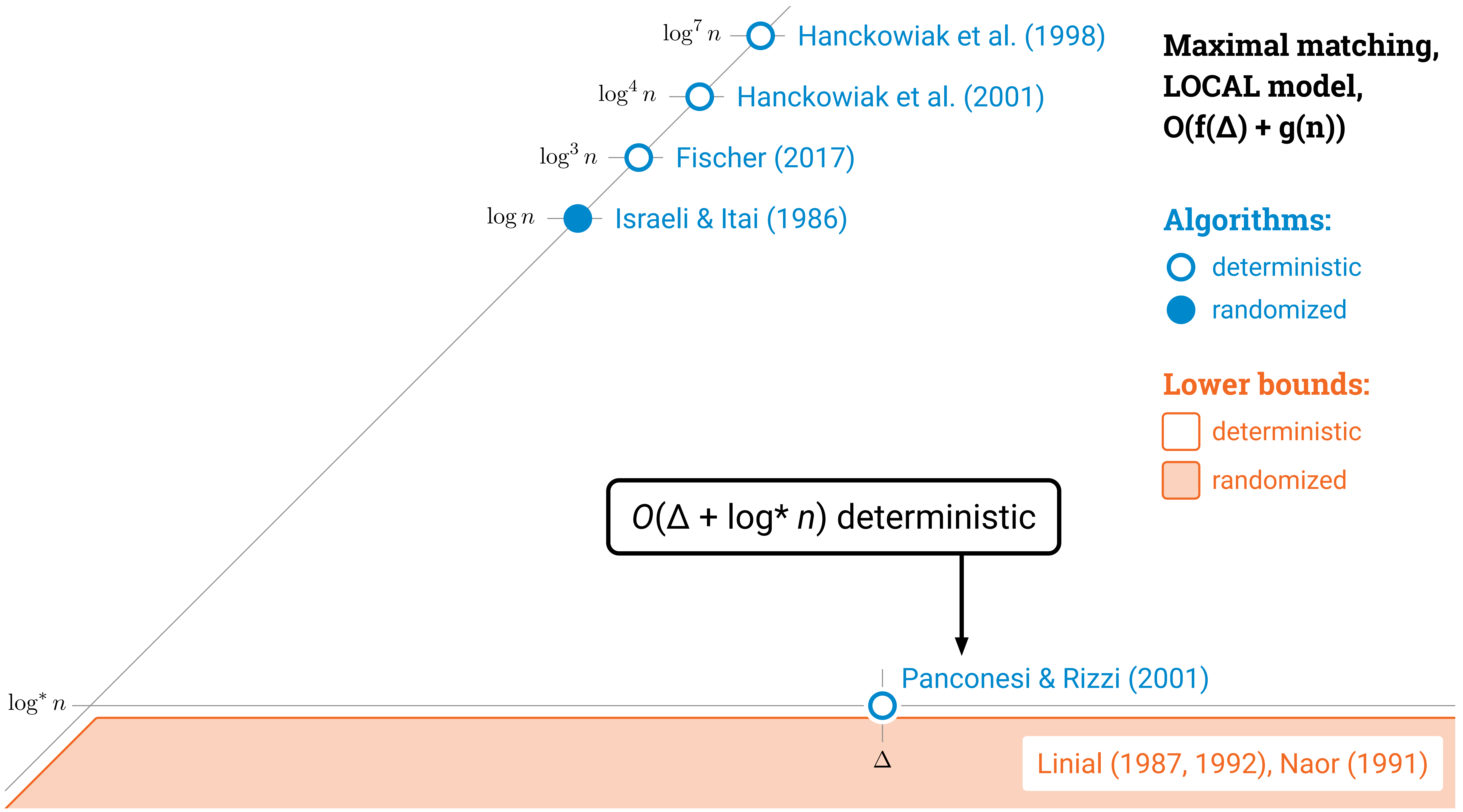


Algorithms:

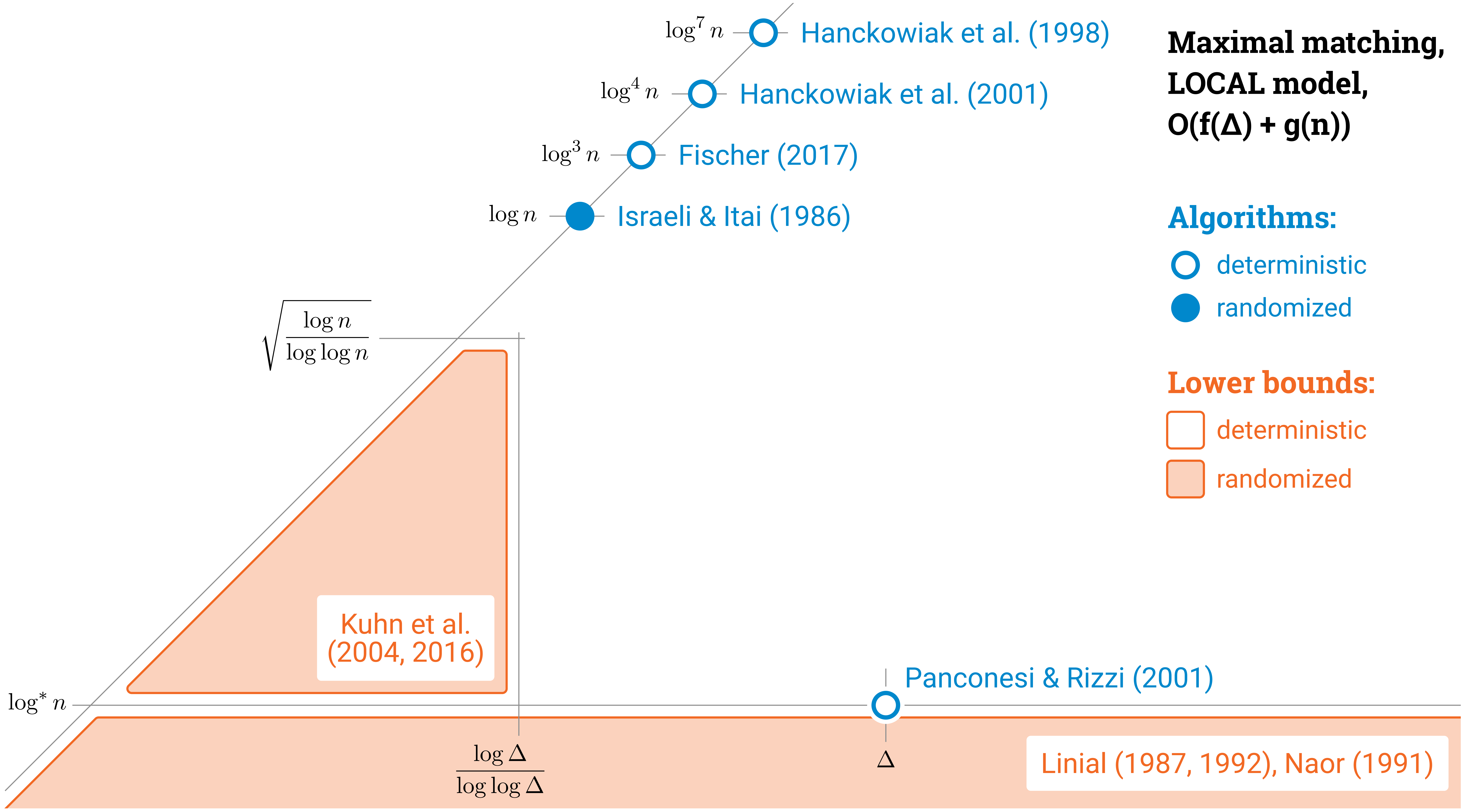
- deterministic
- randomized

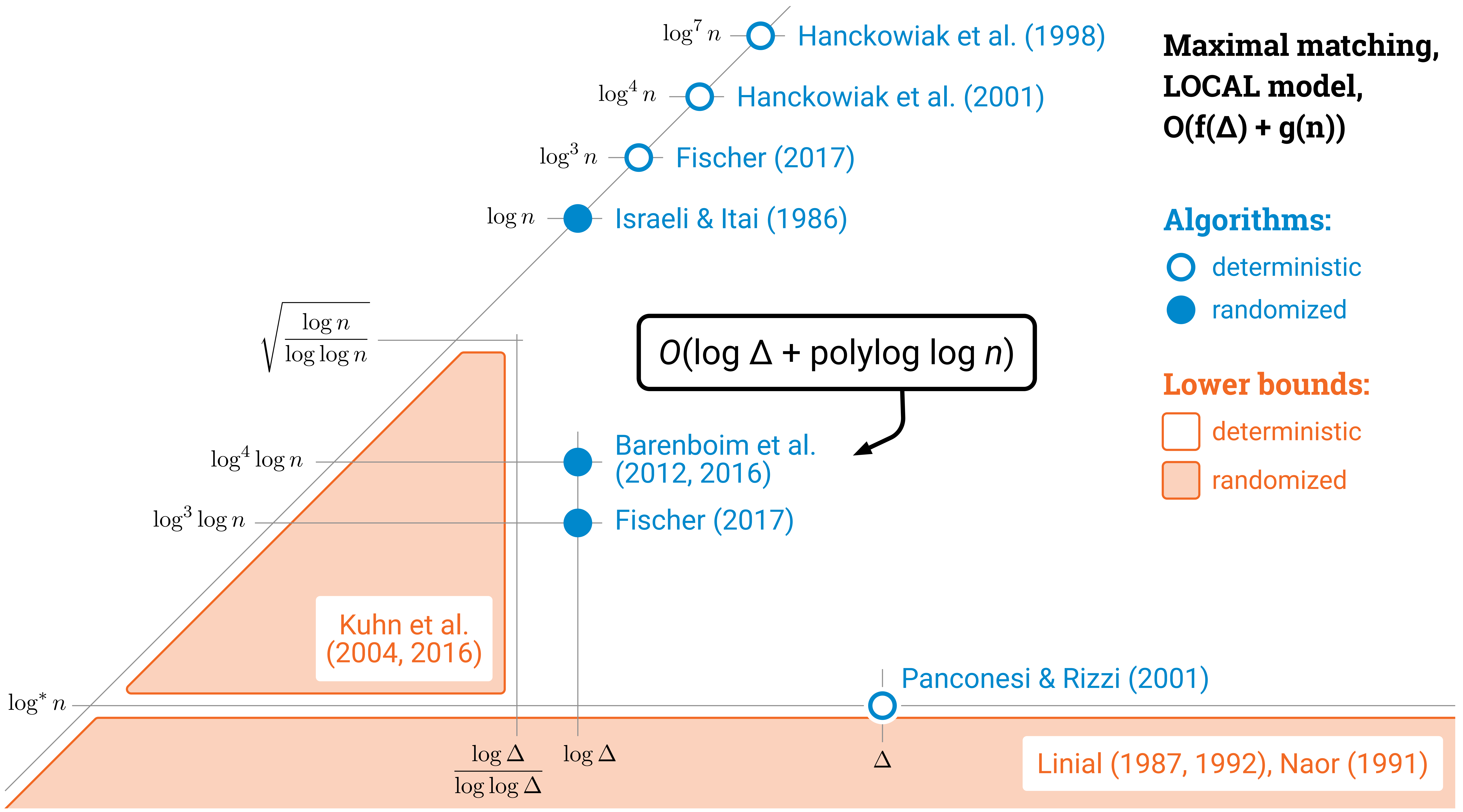
Lower bounds:

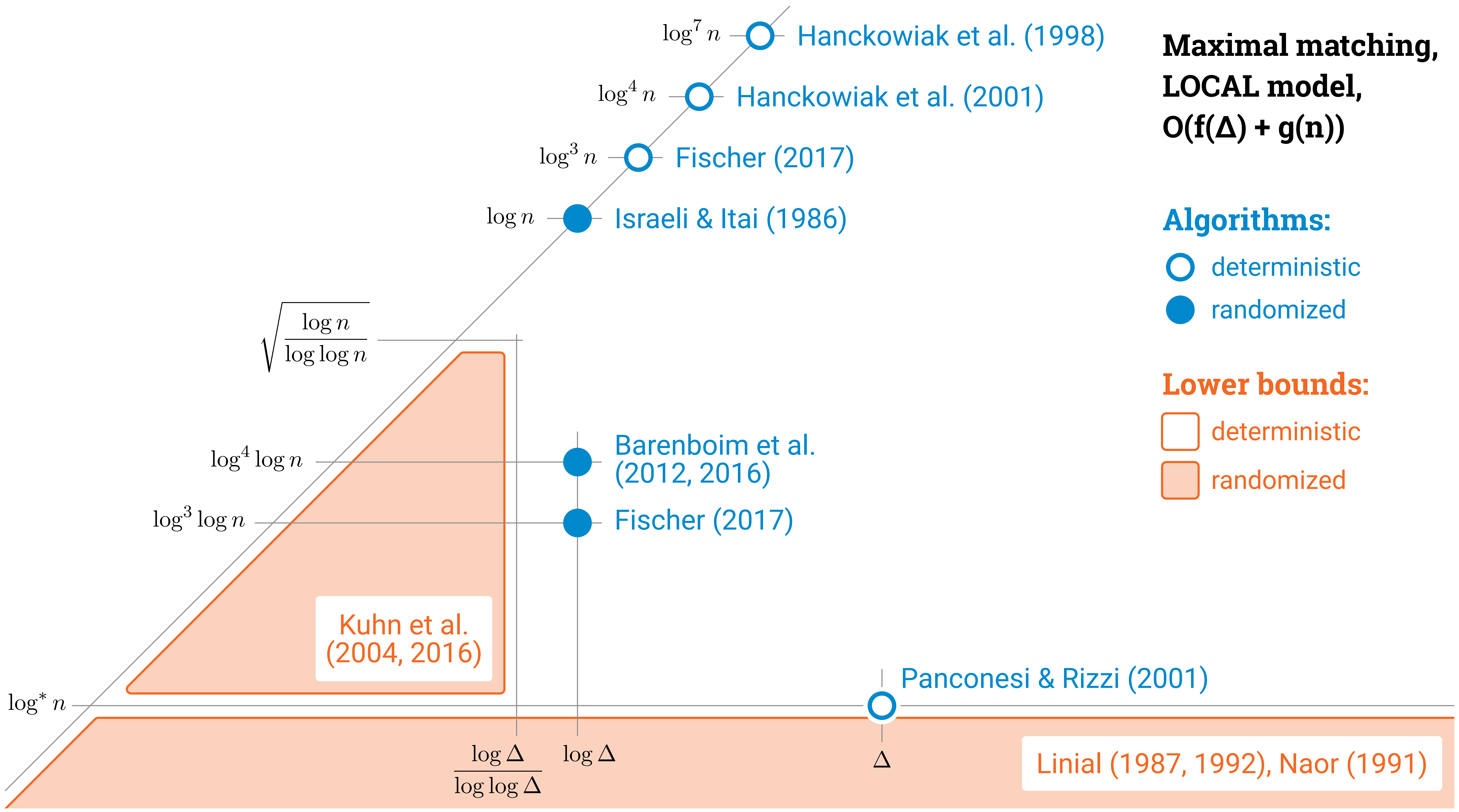
- deterministic
- randomized



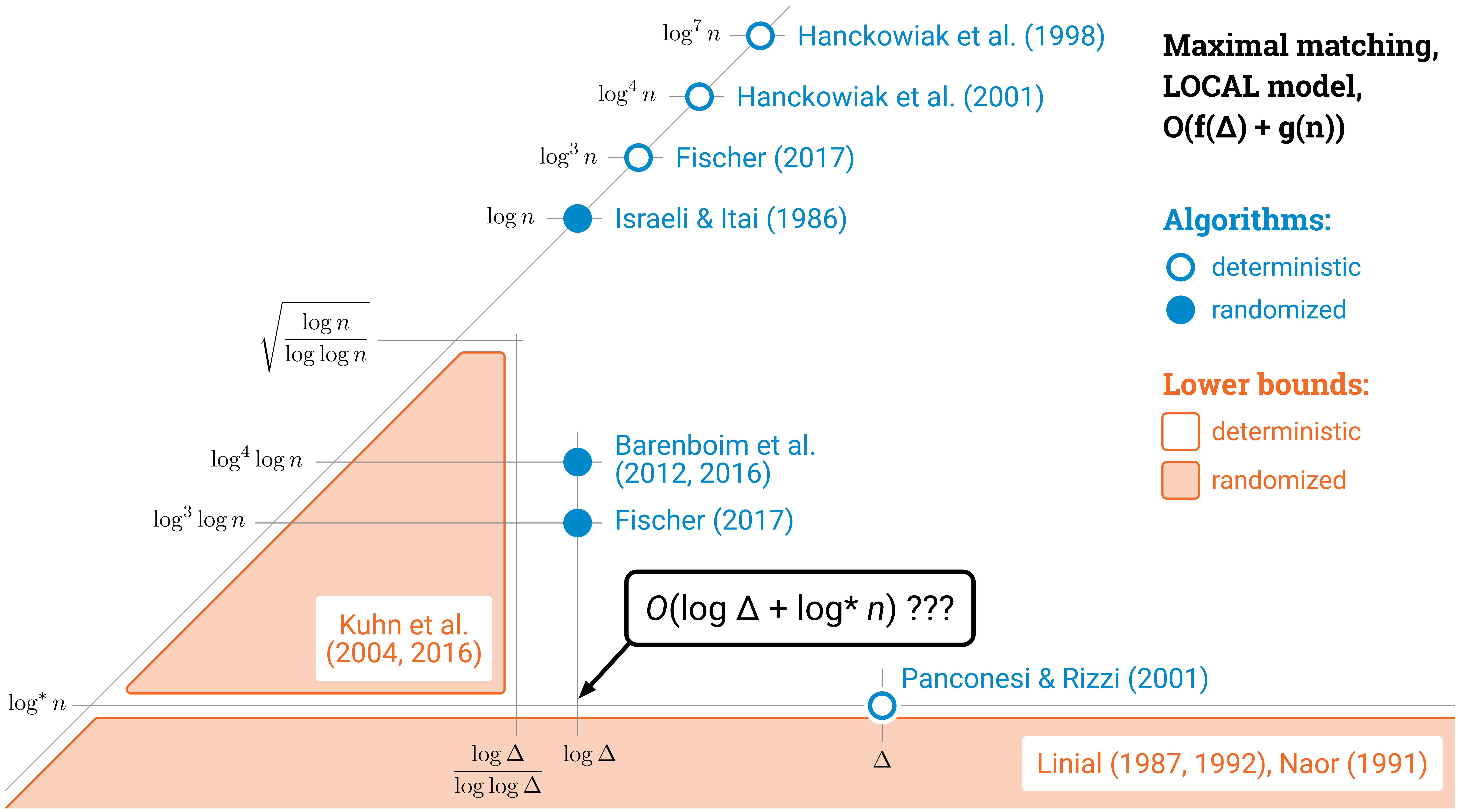
Maximal matching,
LOCAL model,
O(f(Δ) + g(n))
Algorithms:
deterministic
randomized
Lower bounds:
deterministic
randomized
log^7 n
Hanckowiak et al. (1998)
log^4 n
Hanckowiak et al. (2001)
log^3 n
Fischer (2017)
log n
Israeli & Itai (1986)
O(Δ + log* n) deterministic
Panconesi & Rizzi (2001)
log* n
Δ
Linial (1987, 1992), Naor (1991)


Maximal matching,
LOCAL model,
O(f(Δ) + g(n))
Algorithms:
deterministic
randomized
Lower bounds:
deterministic
randomized
$\log^7 n$
Hanckowiak et al. (1998)
$\log^4 n$
Hanckowiak et al. (2001)
$\log^3 n$
Fischer (2017)
$\log n$
Israeli & Itai (1986)
$\sqrt{\frac{\log n}{\log \log n}}$
Kuhn et al. (2004, 2016)
$\log^* n$
Panconesi & Rizzi (2001)
$\frac{\log \Delta}{\log \log \Delta}$
Δ
Linial (1987, 1992), Naor (1991)

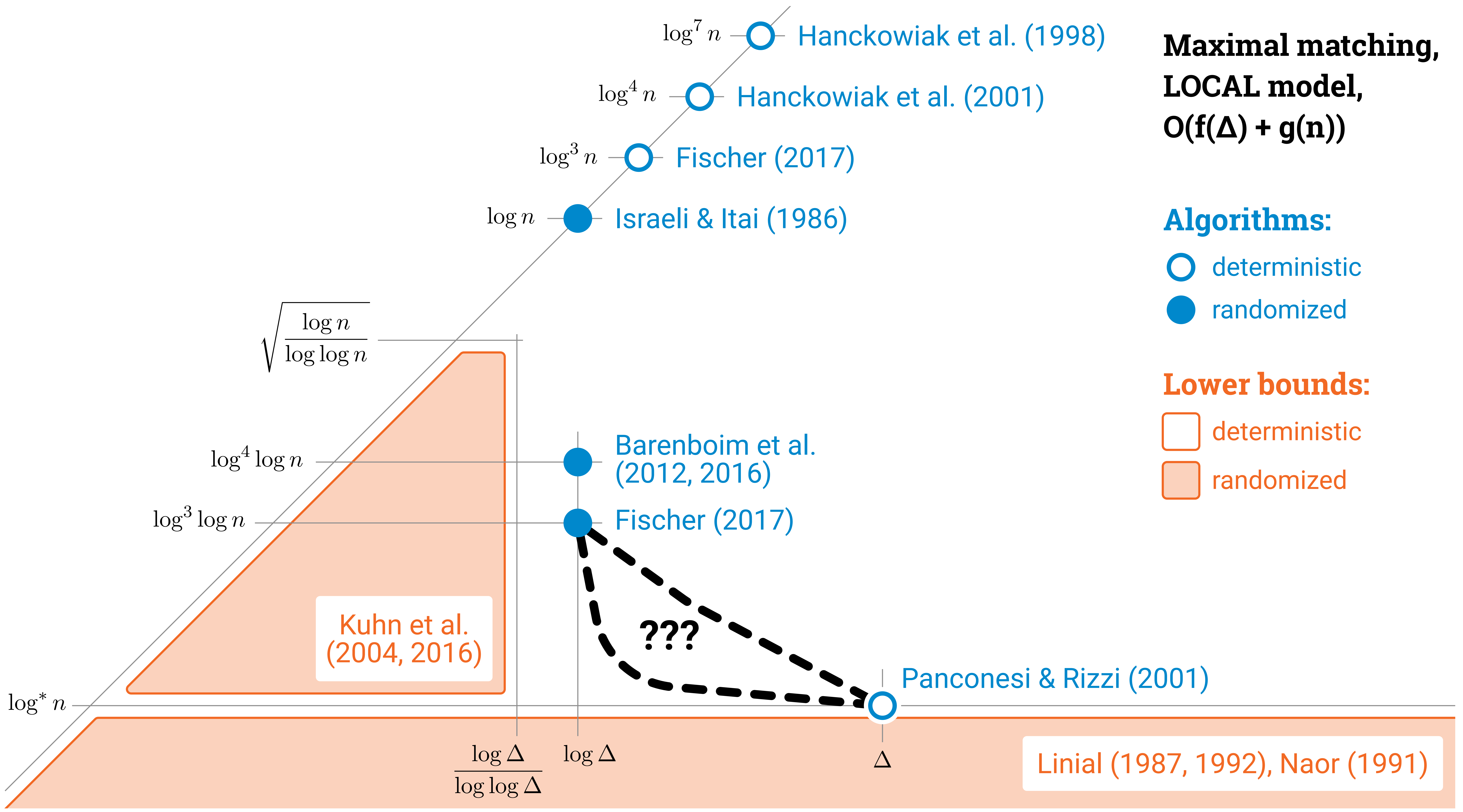

Maximal matching, LOCAL model, O(f(Δ) + g(n))
Algorithms:
deterministic
randomized
Lower bounds:
deterministic
randomized
$\log^7 n$
Hanckowiak et al. (1998)
$\log^4 n$
Hanckowiak et al. (2001)
$\log^3 n$
Fischer (2017)
$\log n$
Israeli & Itai (1986)
$\sqrt{\frac{\log n}{\log \log n}}$
O(log Δ + polylog log n)
$\log^4 \log n$
Barenboim et al. (2012, 2016)
$\log^3 \log n$
Fischer (2017)
Kuhn et al. (2004, 2016)
$\log^* n$
Panconesi & Rizzi (2001)
$\frac{\log \Delta}{\log \log \Delta}$
$\log \Delta$
Δ
Linial (1987, 1992), Naor (1991)

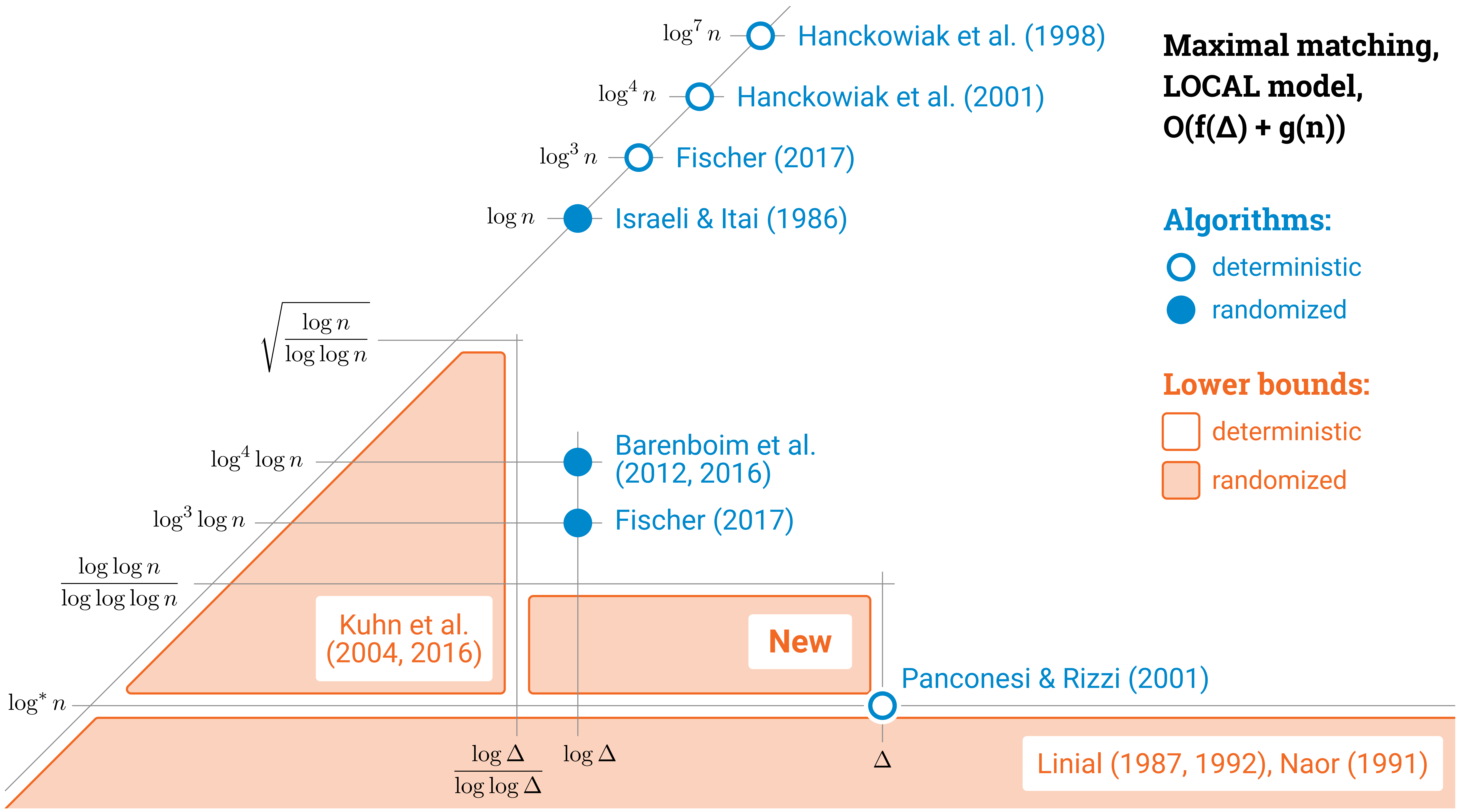


Maximal matching,
LOCAL model,
O(f(Δ) + g(n))
Algorithms:
deterministic
randomized
Lower bounds:
deterministic
randomized
$\log^7 n$
Hanckowiak et al. (1998)
$\log^4 n$
Hanckowiak et al. (2001)
$\log^3 n$
Fischer (2017)
$\log n$
Israeli & Itai (1986)
$\sqrt{\frac{\log n}{\log \log n}}$
$\log^4 \log n$
Barenboim et al. (2012, 2016)
$\log^3 \log n$
Fischer (2017)
Kuhn et al. (2004, 2016)
$\log^* n$
Panconesi & Rizzi (2001)
$\frac{\log \Delta}{\log \log \Delta}$
$\log \Delta$
Δ
Linial (1987, 1992), Naor (1991)

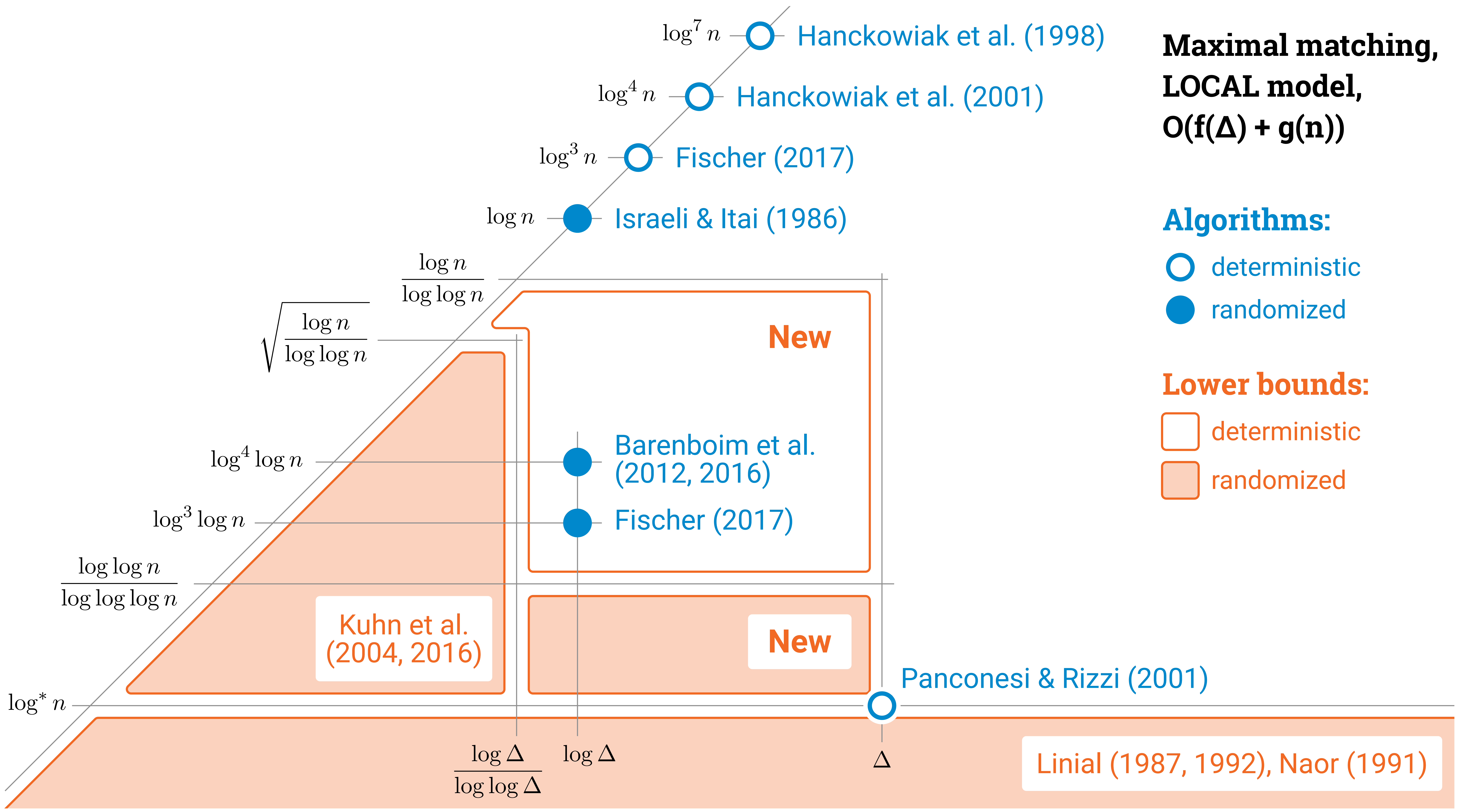


Maximal matching,
LOCAL model,
O(f(Δ) + g(n))
Algorithms:
deterministic
randomized
Lower bounds:
deterministic
randomized
log^7 n
Hanckowiak et al. (1998)
log^4 n
Hanckowiak et al. (2001)
log^3 n
Fischer (2017)
log n
Israeli & Itai (1986)
√(log n / log log n)
log^4 log n
Barenboim et al. (2012, 2016)
log^3 log n
Fischer (2017)
Kuhn et al. (2004, 2016)
O(log Δ + log* n) ???
Panconesi & Rizzi (2001)
log* n
log Δ / log log Δ
log Δ
Δ
Linial (1987, 1992), Naor (1991)


Maximal matching,
LOCAL model,
O(f(Δ) + g(n))
Algorithms:
deterministic
randomized
Lower bounds:
deterministic
randomized
$\log^7 n$
Hanckowiak et al. (1998)
$\log^4 n$
Hanckowiak et al. (2001)
$\log^3 n$
Fischer (2017)
$\log n$
Israeli & Itai (1986)
$\sqrt{\frac{\log n}{\log \log n}}$
$\log^4 \log n$
Barenboim et al. (2012, 2016)
$\log^3 \log n$
Fischer (2017)
Kuhn et al. (2004, 2016)
???
$\log^* n$
Panconesi & Rizzi (2001)
$\frac{\log \Delta}{\log \log \Delta}$
$\log \Delta$
Δ
Linial (1987, 1992), Naor (1991)



Maximal matching, LOCAL model, O(f(Δ) + g(n))
Algorithms:
deterministic
randomized
Lower bounds:
deterministic
randomized
$\log^7 n$
Hanckowiak et al. (1998)
$\log^4 n$
Hanckowiak et al. (2001)
$\log^3 n$
Fischer (2017)
$\log n$
Israeli & Itai (1986)
$\sqrt{\frac{\log n}{\log \log n}}$
$\log^4 \log n$
Barenboim et al. (2012, 2016)
$\log^3 \log n$
Fischer (2017)
$\frac{\log \log n}{\log \log \log n}$
Kuhn et al. (2004, 2016)
New
Panconesi & Rizzi (2001)
$\log^* n$
$\frac{\log \Delta}{\log \log \Delta}$
$\log \Delta$
Δ
Linial (1987, 1992), Naor (1991)


Maximal matching, LOCAL model, O(f(Δ) + g(n))
Algorithms:
deterministic
randomized
Lower bounds:
deterministic
randomized
$\log^7 n$
Hanckowiak et al. (1998)
$\log^4 n$
Hanckowiak et al. (2001)
$\log^3 n$
Fischer (2017)
$\log n$
Israeli & Itai (1986)
$\frac{\log n}{\log \log n}$
$\sqrt{\frac{\log n}{\log \log n}}$
New
$\log^4 \log n$
Barenboim et al. (2012, 2016)
$\log^3 \log n$
Fischer (2017)
$\frac{\log \log n}{\log \log \log n}$
Kuhn et al. (2004, 2016)
New
$\log^* n$
Panconesi & Rizzi (2001)
$\frac{\log \Delta}{\log \log \Delta}$
$\log \Delta$
Δ
Linial (1987, 1992), Naor (1991)

Main results

Maximal matching and **maximal independent set** cannot be solved in

- $o(\Delta + \log \log n / \log \log \log n)$ rounds with randomized algorithms
- $o(\Delta + \log n / \log \log n)$ rounds with deterministic algorithms

Upper bound: $O(\Delta + \log^* n)$

Proof sketch

What we really care about

Maximal matching in $o(\Delta)$ rounds

Proof sketch

Maximal matching in $o(\Delta)$ rounds

→ **“$\Delta^{1/2}$ matching”** in $o(\Delta^{1/2})$ rounds

k-matching:
select at most
k edges per node

Proof sketch

Maximal matching in $o(\Delta)$ rounds

→ **"$\Delta^{1/2}$ matching"** in $o(\Delta^{1/2})$ rounds

→ $P(\Delta^{1/2}, 0)$ in $o(\Delta^{1/2})$ rounds

P(x,y):
unnatural relaxed variant of maximal matching

Proof sketch

Maximal matching in $o(\Delta)$ rounds

→ "$\Delta^{1/2}$ **matching**" in $o(\Delta^{1/2})$ rounds

→ $P(\Delta^{1/2}, 0)$ in $o(\Delta^{1/2})$ rounds

→ $P(O(\Delta^{1/2}), o(\Delta))$ in **0** rounds

P(x,y): unnatural relaxed variant of maximal matching

Proof sketch

Maximal matching in $o(\Delta)$ rounds

→ **"$\Delta^{1/2}$ matching"** in $o(\Delta^{1/2})$ rounds

→ $P(\Delta^{1/2}, 0)$ in $o(\Delta^{1/2})$ rounds

→ $P(O(\Delta^{1/2}), o(\Delta))$ in **0** rounds

→ contradiction

P(x,y):
unnatural relaxed variant of maximal matching

Round elimination technique

- **Given:**
  - algorithm A_0 solves problem P_0 in T rounds
- **We construct:**
  - algorithm A_1 solves problem P_1 in $T - 1$ rounds
  - algorithm A_2 solves problem P_2 in $T - 2$ rounds
  - algorithm A_3 solves problem P_3 in $T - 3$ rounds

    …
  - algorithm A_T solves problem P_T in 0 rounds
- But P_T is nontrivial, so A_0 cannot exist

Round elimination technique

- **Given:**
  - algorithm A_0 solves problem P_0 in T rounds
- **We construct:**
  - algorithm A_1 solves problem P_1 in $T - 1$ rounds
  - algorithm A_2 solves problem P_2 in $T - 2$ rounds
  - algorithm A_3 solves problem P_3 in $T - 3$ rounds

  …
  - algorithm A_T solves problem P_T in 0 rounds
- But P_T is nontrivial, so A_0 cannot exist

[B. 2019]: Given any P_i, it is possible to find P_{i+1} automatically, but the description of the problem may grow exponentially

Proof sketch

Maximal matching in $o(\Delta)$ rounds

→ **"$\Delta^{1/2}$ matching"** in $o(\Delta^{1/2})$ rounds

→ $P(\Delta^{1/2}, 0)$ in $o(\Delta^{1/2})$ rounds

→ $P(O(\Delta^{1/2}), o(\Delta))$ in **0** rounds

→ contradiction

Apply round elimination technique

Main Lemma

- Given: ***A*** solves ***P*(*x*, *y*)** in ***T*** rounds
- We can construct: ***A'*** solves ***P*(*x* + 1, *y* + *x*)** in ***T* – 1** rounds

$$W_\Delta(x,y) = \left(\mathsf{MO}^{d-1} \;\middle|\; \mathsf{P}^{d}\right)\mathsf{O}^{y}\mathsf{X}^{x},$$

$$B_\Delta(x,y) = \left([\mathsf{MX}][\mathsf{POX}]^{d-1} \;\middle|\; [\mathsf{OX}]^{d}\right)[\mathsf{POX}]^{y}[\mathsf{MPOX}]^{x},$$

$$d = \Delta - x - y$$

Lower bound for the LOCAL model

- The lower bound holds for the ***simple scenario*** where randomness is not allowed and nodes are anonymous
- ***Additional steps*** are required to handle:
  - randomness
  - non anonymous nodes

Conclusions and open problems

- ***Linear-in-Δ lower bounds*** for **maximal matchings** and **maximal independent sets**
- Maximal matchings can not be solved fast:
  - The simple proposal algorithm is optimal
  - Randomization and large messages do not help
- How about a lower bound for distributed ***Δ+1 coloring***?