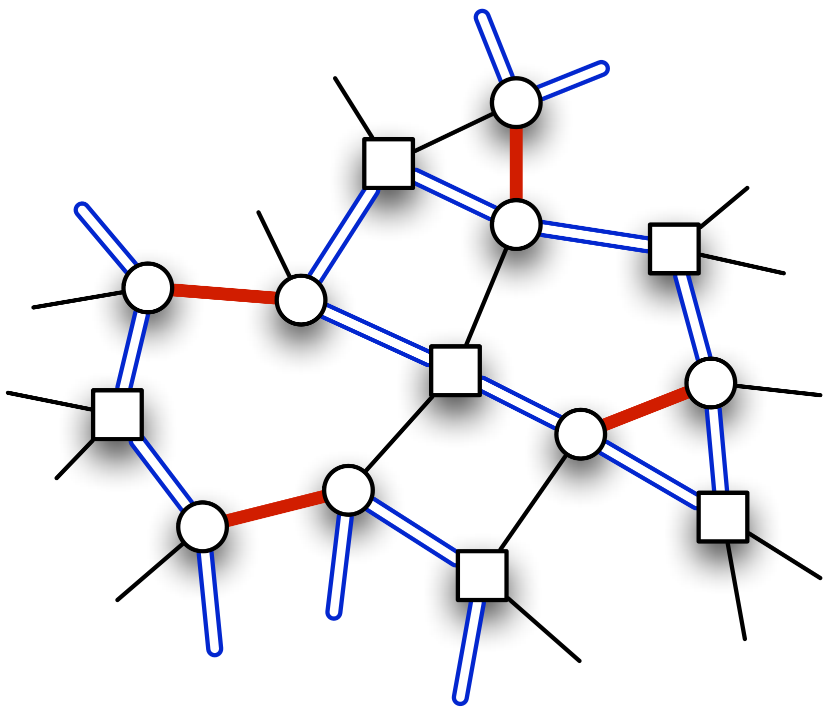
Distributed algorithms for edge dominating sets

Jukka Suomela

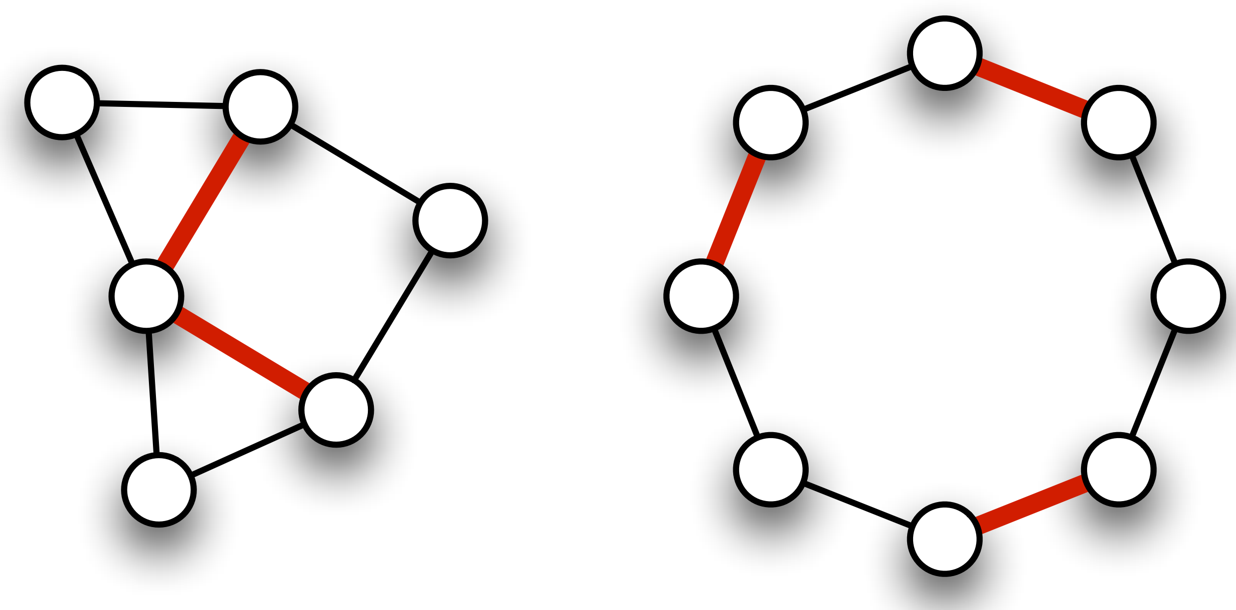
Helsinki Institute for Information Technology HIIT
University of Helsinki, Finland

PODC, Zurich,
28 July 2010



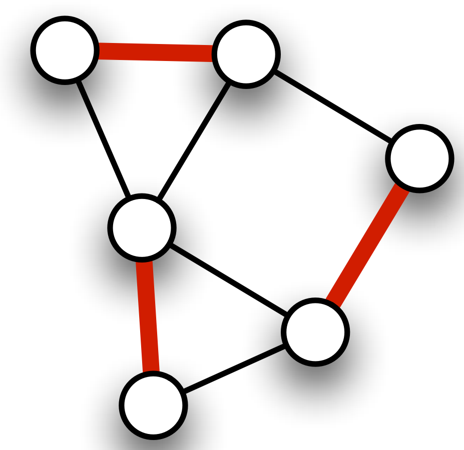
Edge dominating sets

- Simple undirected graph $G = (V, E)$
- **Edge dominating set** $D \subseteq E$: each edge is in D or adjacent at least one edge in D

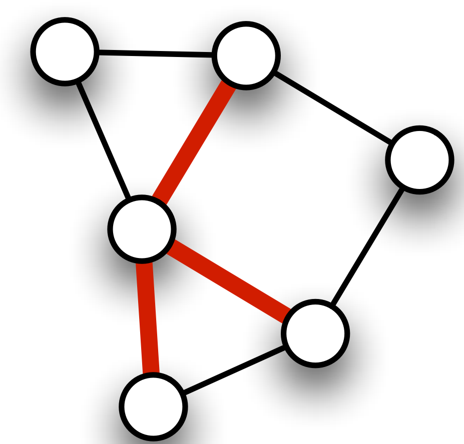


Edge dominating sets

- Any **maximal matching** is an edge dominating set

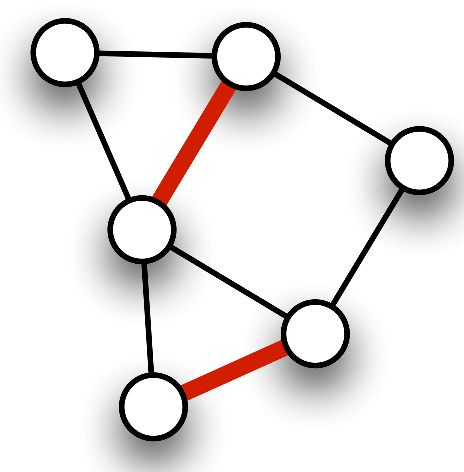


- But edge dominating sets are not necessarily matchings

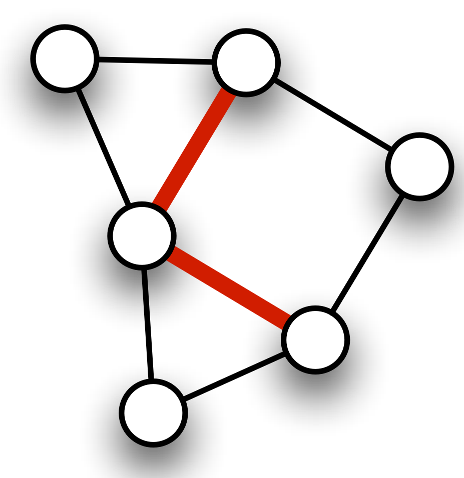


Edge dominating sets

- Any **minimum** maximal matching is a **minimum** edge dominating set
    - Allan & Laskar 1978, Yannakakis & Gavril 1980

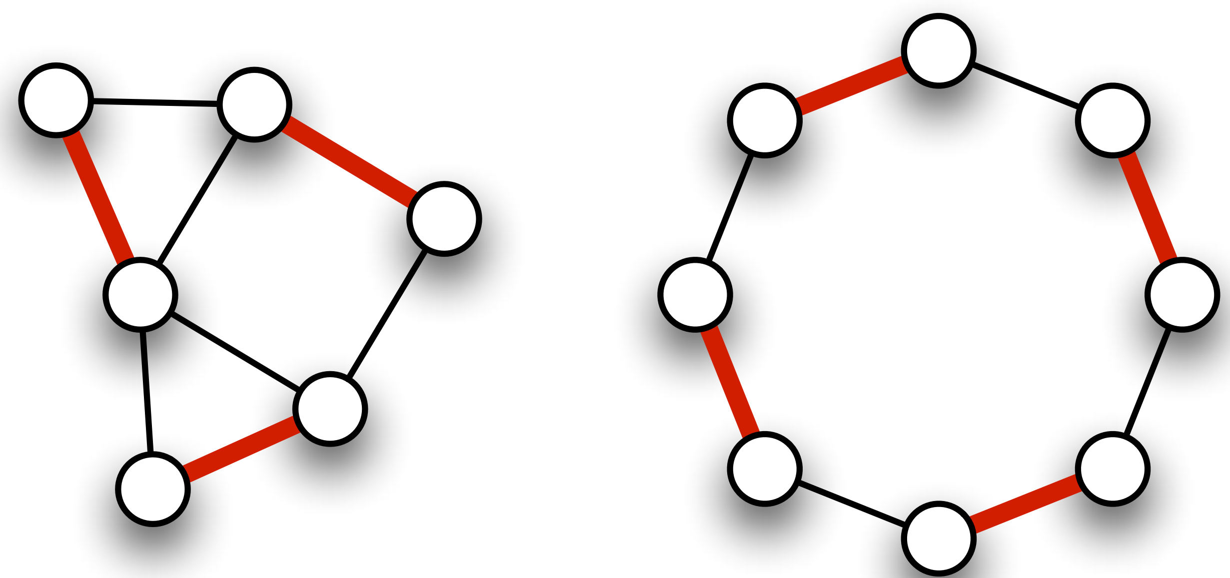


- But minimum edge dominating sets are not necessarily matchings



Edge dominating sets

- NP-hard (and APX-hard) optimisation problem
- Simple **2-approximation algorithm**: find any maximal matching

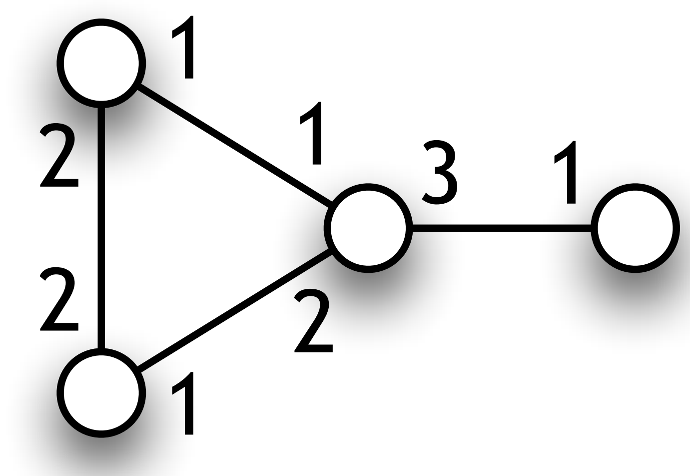


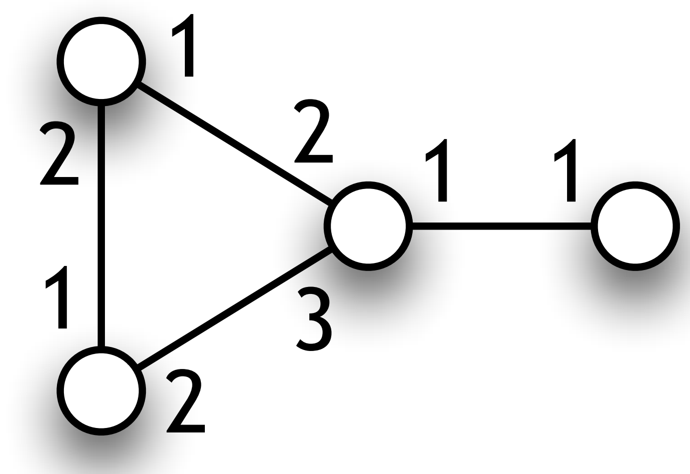
Edge dominating sets

- NP-hard (and APX-hard) optimisation problem
- Simple 2-approximation algorithm:
  find any maximal matching
- What about **distributed** approximation algorithms?
- In **very weak** models of distributed computing
  - Deterministic algorithms, port-numbering model
  - Can’t find maximal matchings…

Port-numbering model

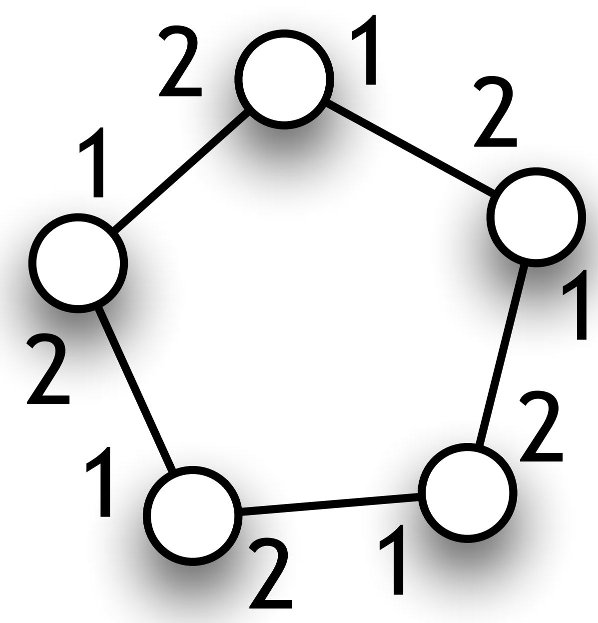
- Identical nodes,
  no unique identifiers
- **Port numbers**:
  - Node of degree *d* can refer to its neighbours by integers 1, 2, …, *d*
- Worst-case analysis:
  - Port-numbering chosen by adversary





Port-numbering model

- Focus:
  - **Deterministic** distributed algorithms
  - **Port-numbering** model
  - No restrictions on message size, local computation, …
- Weak model:
  - Can't break symmetry in cycles
  - Can't find graph colouring, maximal matching, …



Edge dominating sets in port-numbering model

- Problem simple to state:
  exactly how well can we approximate
  minimum edge dominating sets
  - using deterministic distributed algorithms, in the port-numbering model
- But why would we care?
- Let's have a look at some classical graph problems from this perspective…

Some classical graph problems in port-numbering model

	Node-based	Edge-based
Covering problems	vertex cover	edge cover
	dominating set	edge dominating set
Packing problems	independent set	matching

Some classical graph problems in port-numbering model

	Node-based	Edge-based
Cove… prob…		… set
Packing problems	independent set	matching

Many packing problems are unsolvable for trivial reasons

(impossibility of symmetry breaking in cycles)

Some classical graph problems in port-numbering model

	Node-based
Covering problems	vertex cover
	dominating set
Packing problems	independent set

Many non-trivial positive results

(SPAA 2008, DISC 2008, DISC 2009, SPAA 2010, DISC 2010, …)

But trivial lower bounds!

(cycles, cliques, etc.)

Some classical graph problems in port-numbering model

	Node-based	Edge-based
Co pr		edge cover
	et	edge dominating set
P problems	independent set	matching

But do we know anything about ***edge-based covering problems*** in this setting?

Edge-based covering problems in port-numbering model

- Minimum *edge cover* seems to be a bit too simple: factor 2 approximation is trivial and tight
- But what about minimum *edge dominating sets*?
- Surprise: both upper bounds and lower bounds are non-trivial!
- Contribution: **full characterisation** of approximability of edge dominating sets in regular graphs and bounded-degree graphs

Edge dominating sets: deterministic algorithms in port-numbering model

Graph family		Approximation ratio
d-regular graphs	$d = 1, 3, \ldots$	$4 - 6/(d + 1)$
	$d = 2, 4, \ldots$	$4 - 2/d$
graphs with degree $\le \Delta$	$\Delta = 3, 5, \ldots$	$4 - 2/(\Delta - 1)$
	$\Delta = 2, 4, \ldots$	$4 - 2/\Delta$

Tight results: these are both lower bounds and upper bounds

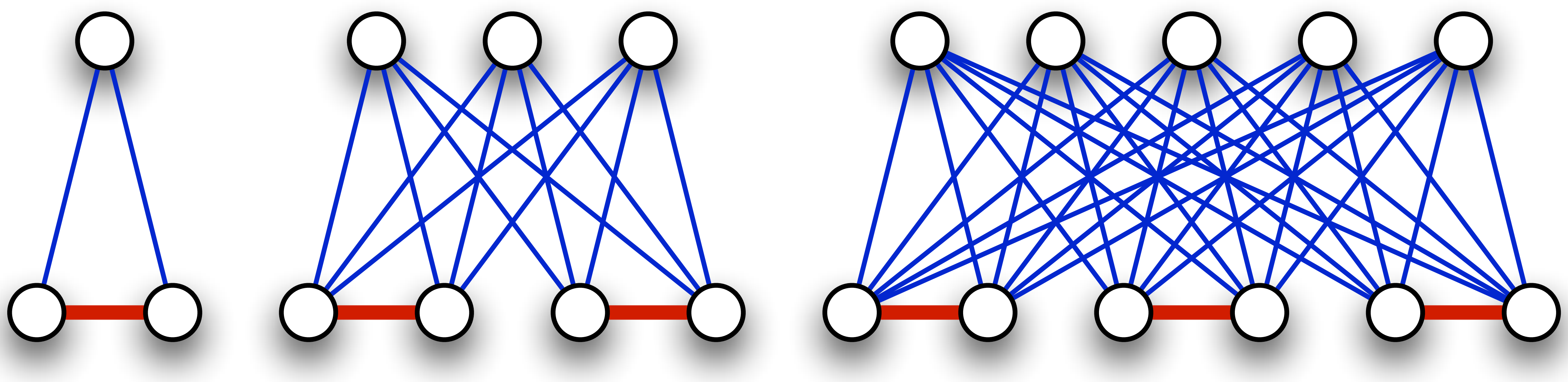
Edge dominating sets: deterministic algorithms in port-numbering model

Graph family		Approximation ratio	Time
d-regular graphs	$d = 1, 3, \ldots$	$4 - 6/(d+1)$	$O(d^2)$
	$d = 2, 4, \ldots$	$4 - 2/d$	$O(1)$
graphs with degree $\leq \Delta$	$\Delta = 3, 5, \ldots$	$4 - 2/(\Delta - 1)$	$O(\Delta^2)$
	$\Delta = 2, 4, \ldots$	$4 - 2/\Delta$	$O(\Delta^2)$

Tight approximation ratios achievable in $f(\Delta)$ time, $f(n)$-time algorithms cannot do any better

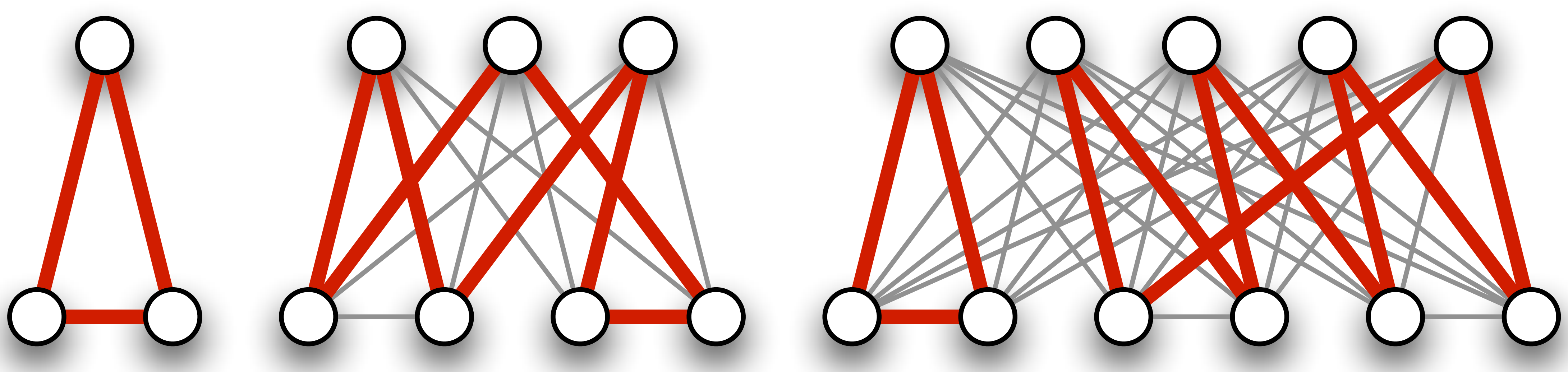
Lower bound construction: some key ideas

- Case: d-regular graphs, $d = 2k$
- Complete bipartite graph $K_{d,d-1}$
- k **extra edges** (optimal solution)



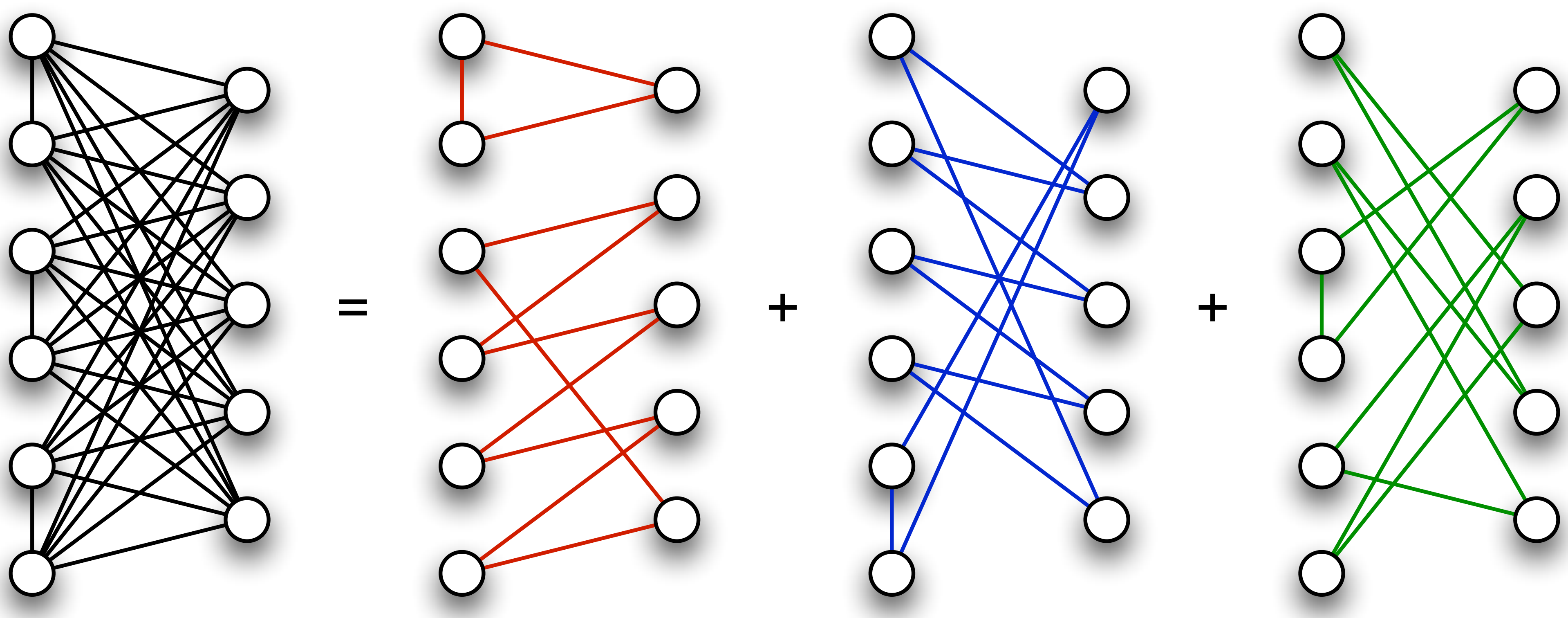
Lower bound construction: some key ideas

- Idea: show that there is a port-numbering s.t. any deterministic algorithm has to output a **spanning 2-regular subgraph**
  - I.e., a **2-factor** (spanning set of disjoint cycles)



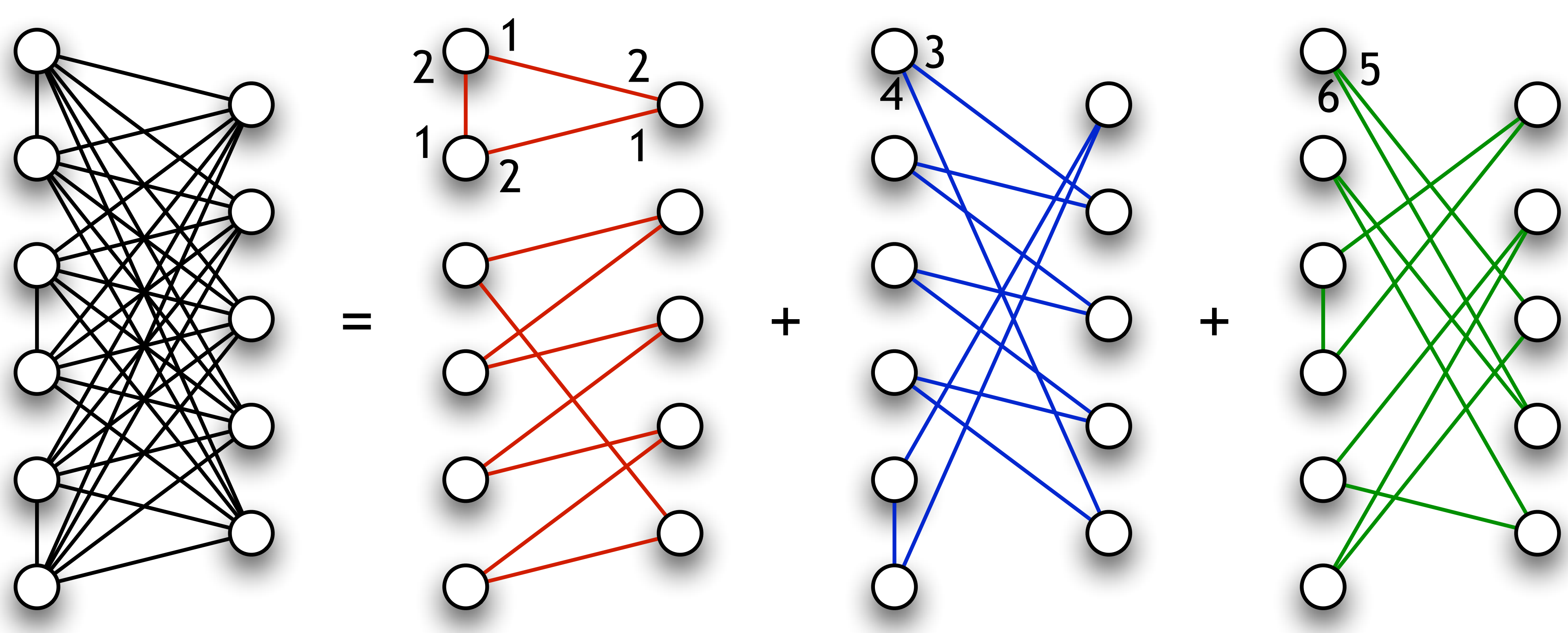
Lower bound construction: some key ideas

- Petersen (1891): any $2k$-regular graph admits a **2-factorisation** (partition in 2-factors)



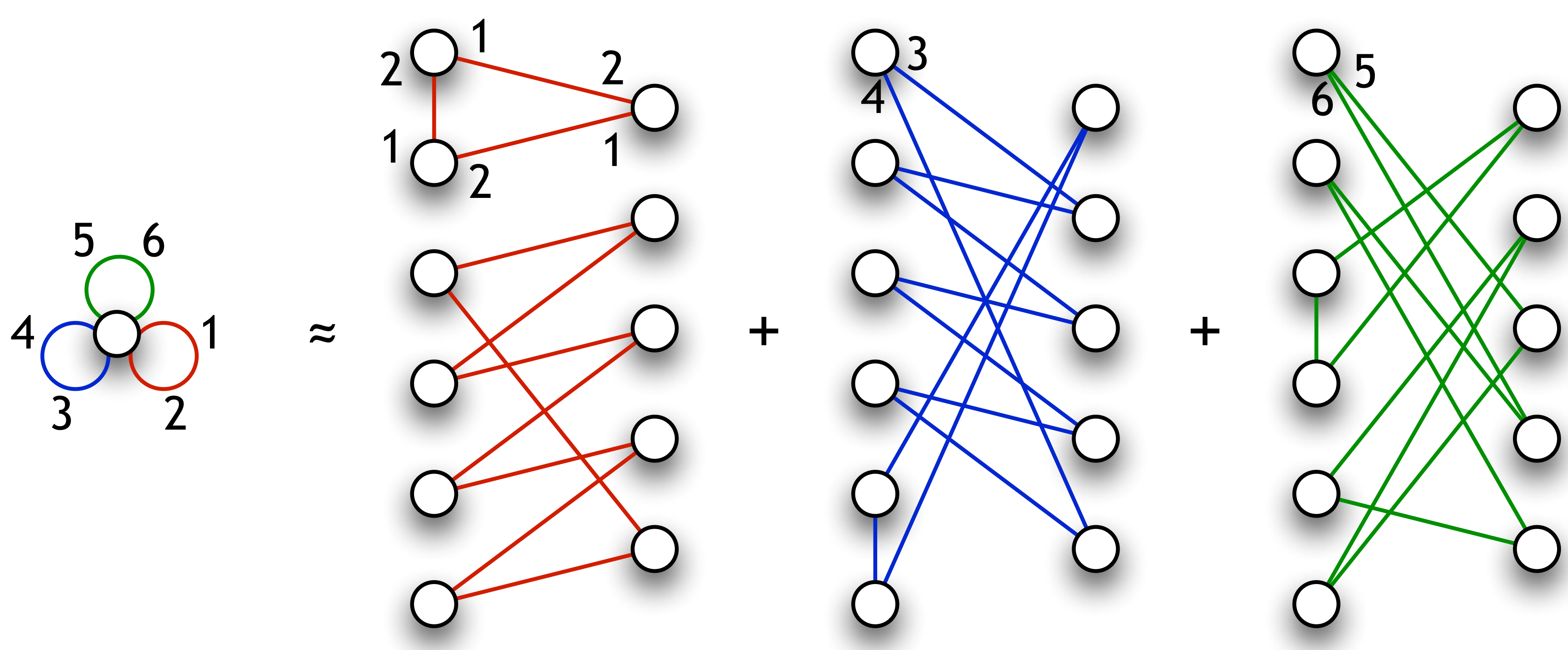
Lower bound construction: some key ideas

- Use 2-factorisation to assign **port numbers**:
  - 1, 2, 1, 2, … in each cycle of 1st factor, 3, 4, 3, 4, … in each cycle of 2nd factor, etc.



Lower bound construction: some key ideas

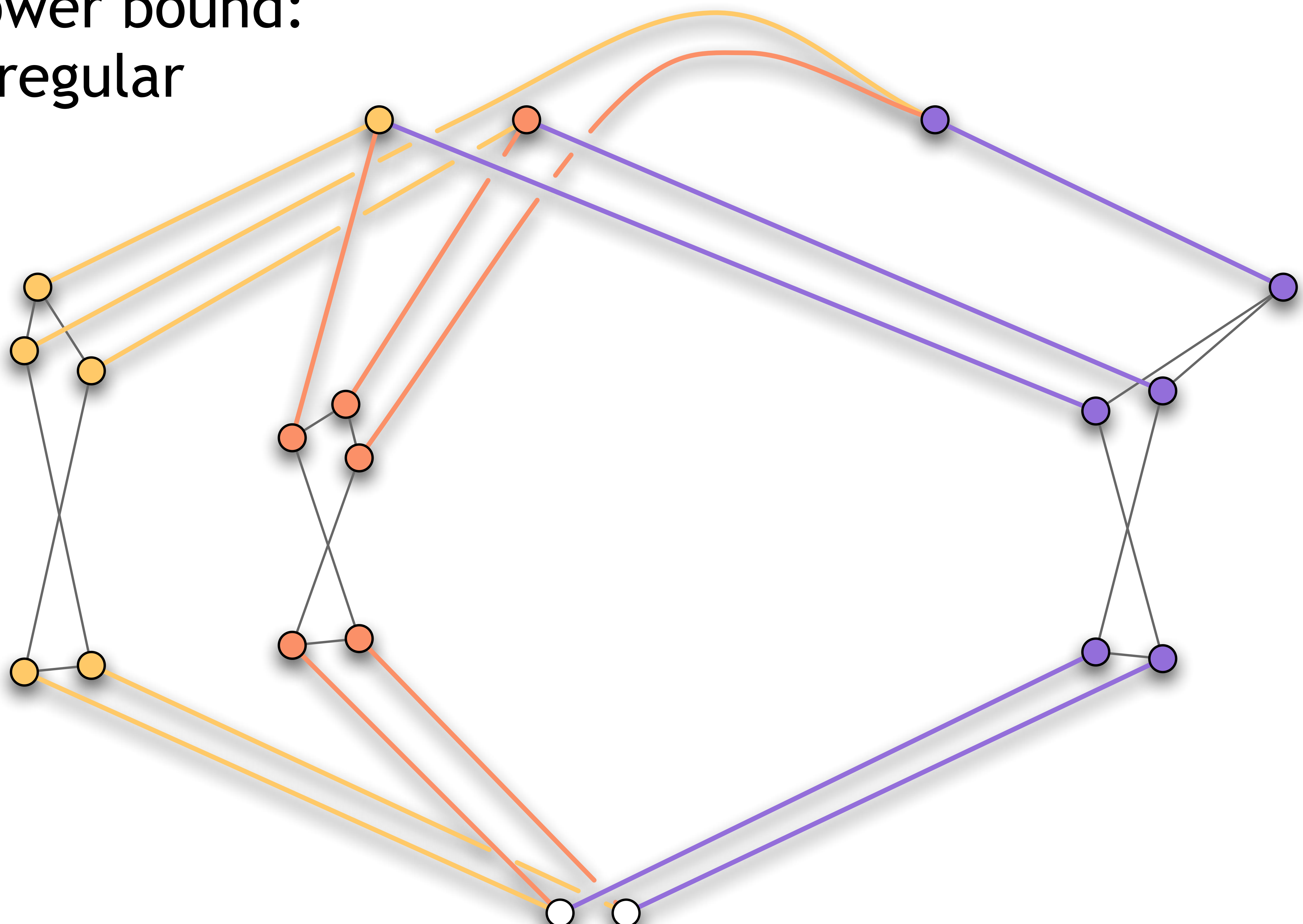
- Then we can use **covering maps** to argue that any algorithm must take all or nothing from each 2-factor



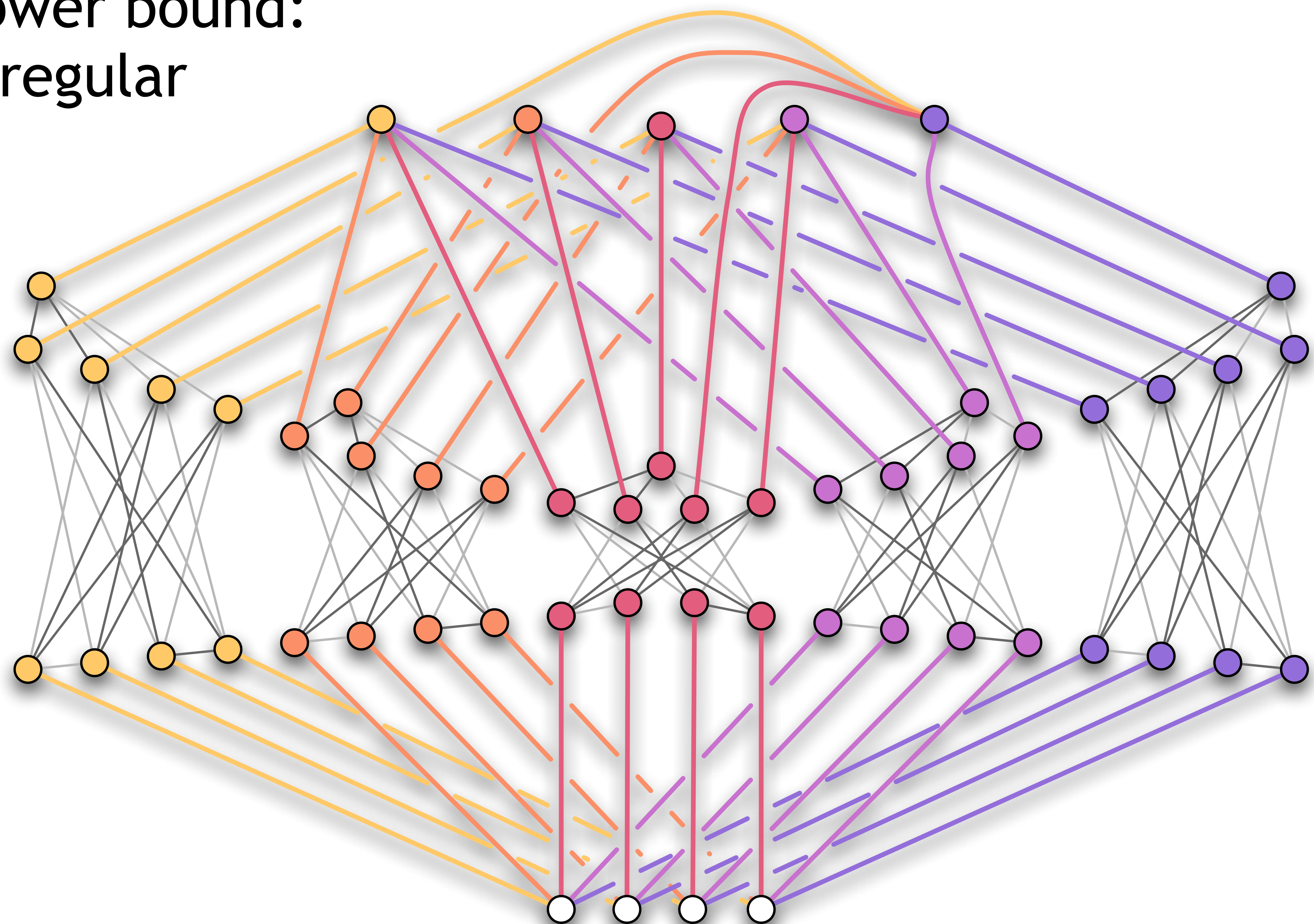
Lower bound construction: some key ideas

- Then we can use covering graphs to argue that any algorithm must take all or nothing from each 2-factor
- That's it for *even* degrees — the case of *odd* degrees is more difficult
  - There is always some amount of symmetry-breaking information in port-numbered graphs of odd degree (recall Naor & Stockmeyer 1995)

Lower bound: 3-regular

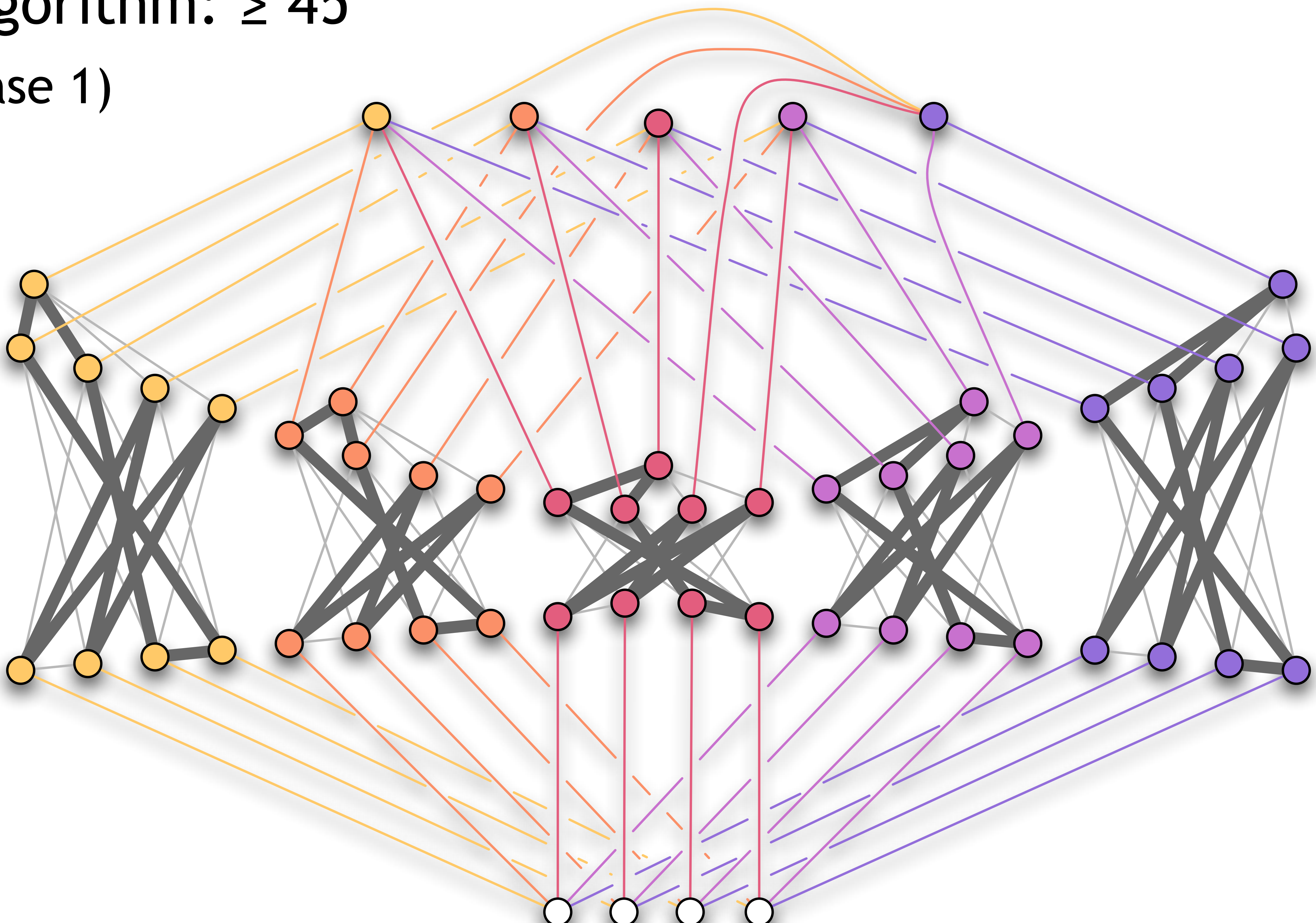


Lower bound:
5-regular



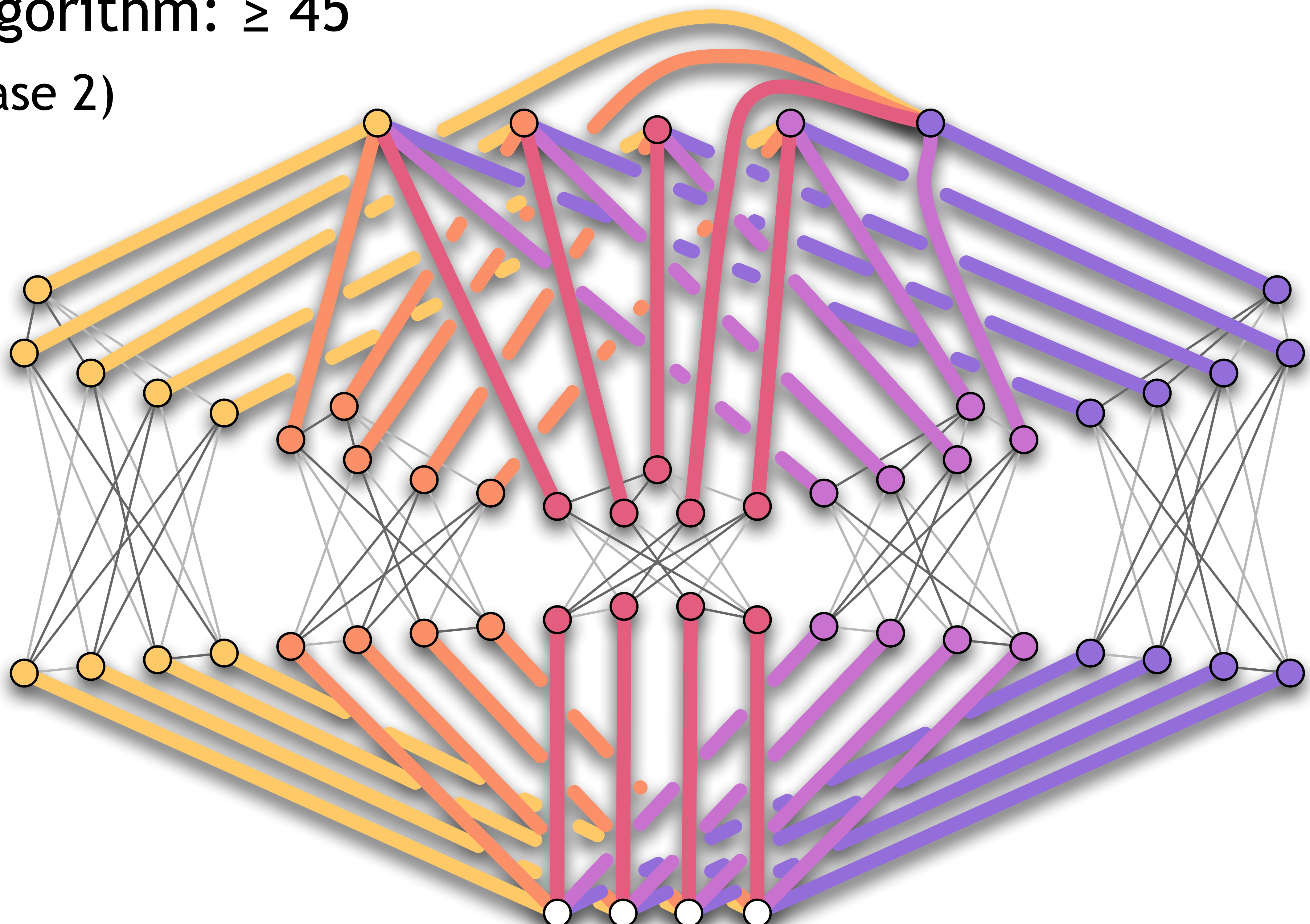
Algorithm: ≥ 45

(case 1)

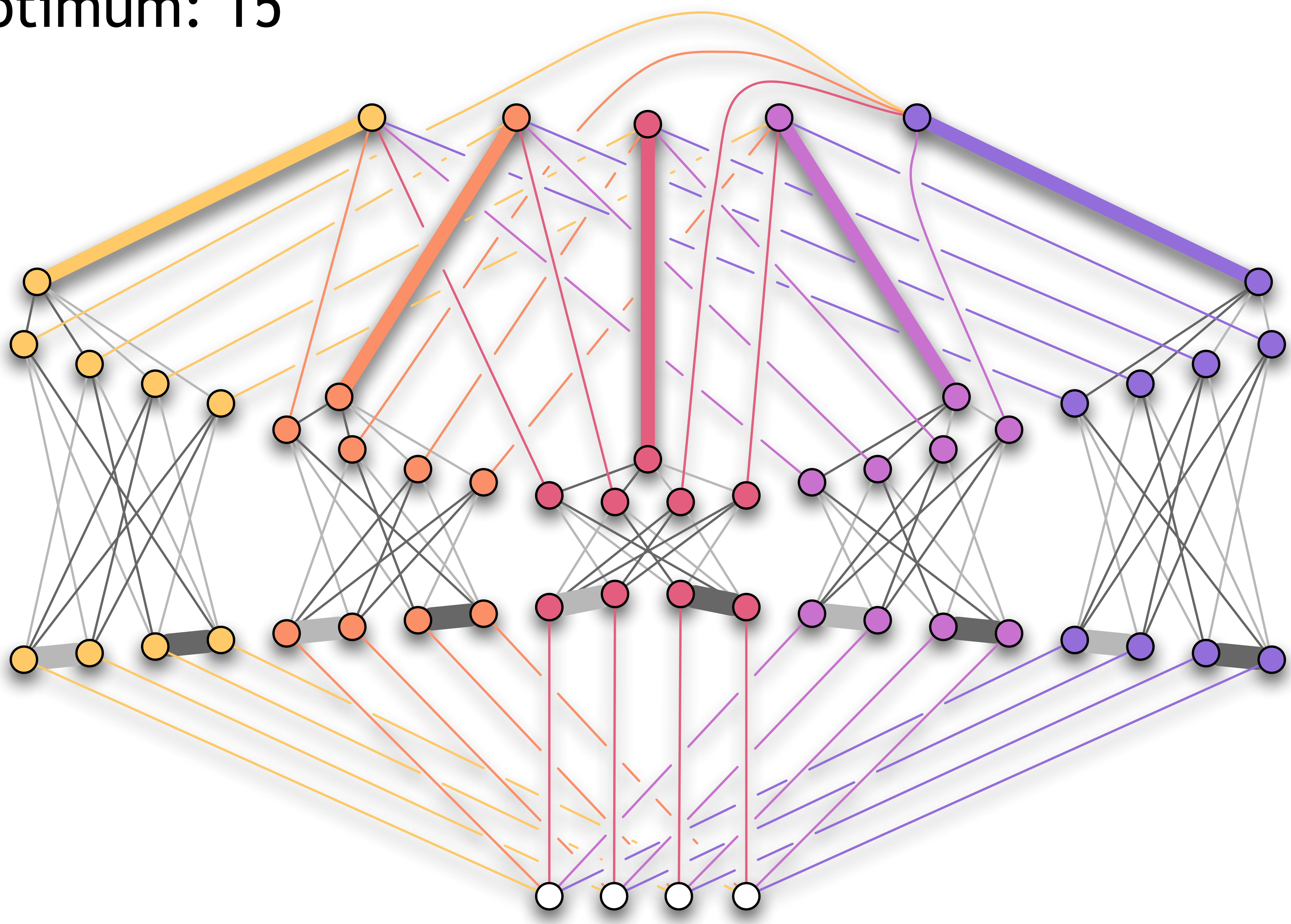


Algorithm: ≥ 45

(case 2)



Optimum: 15

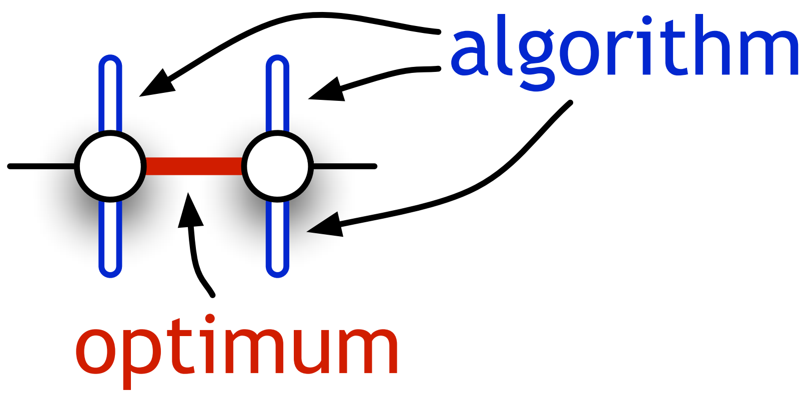


Upper bounds: some key ideas

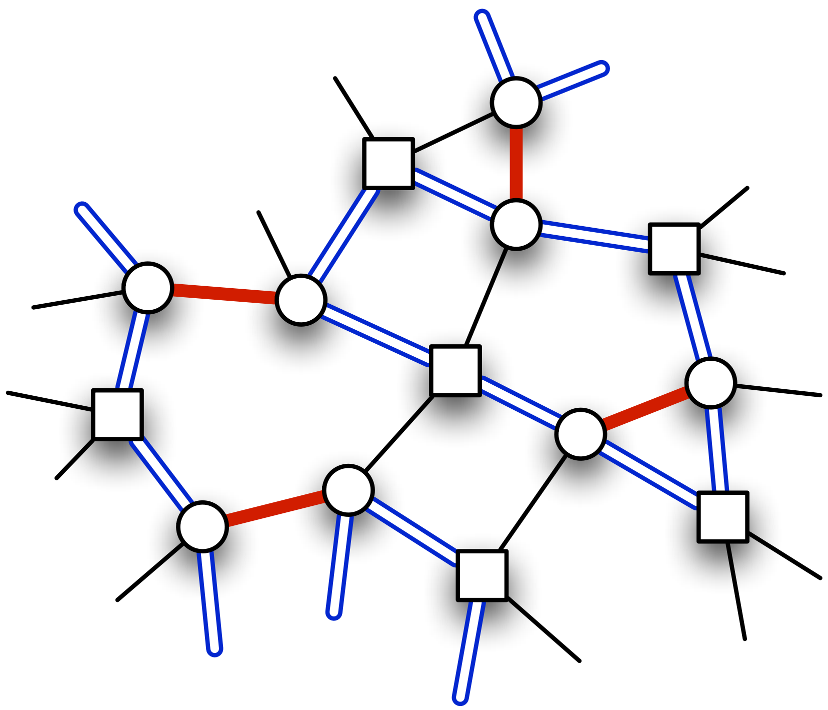
- Exploit all possible **sources of symmetry-breaking** information:
  - Different node degrees: interpret degrees as colours
  - Odd degrees: there is a “distinguishable neighbour”
- And when symmetry can’t be broken, find a **2-matching** (paths and cycles)
  - On average 1 edge per node
- Tricky part: show that this is enough!

Upper bounds: some key ideas

- Some intuition…
- A really bad case:
  - 4 edges in algorithm output
  - 1 edge in optimal solution

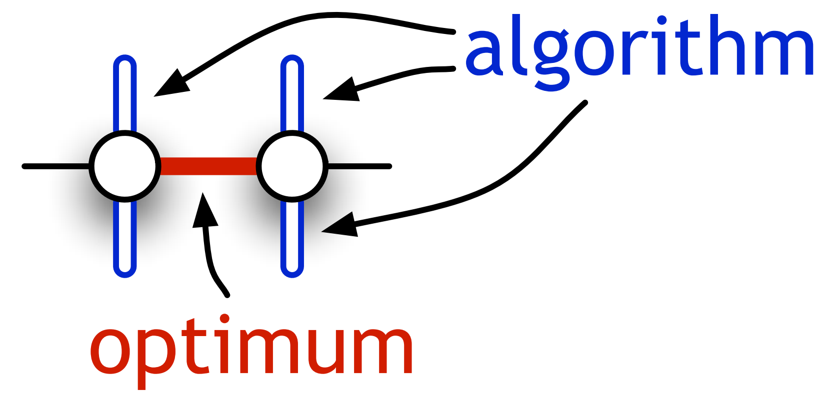


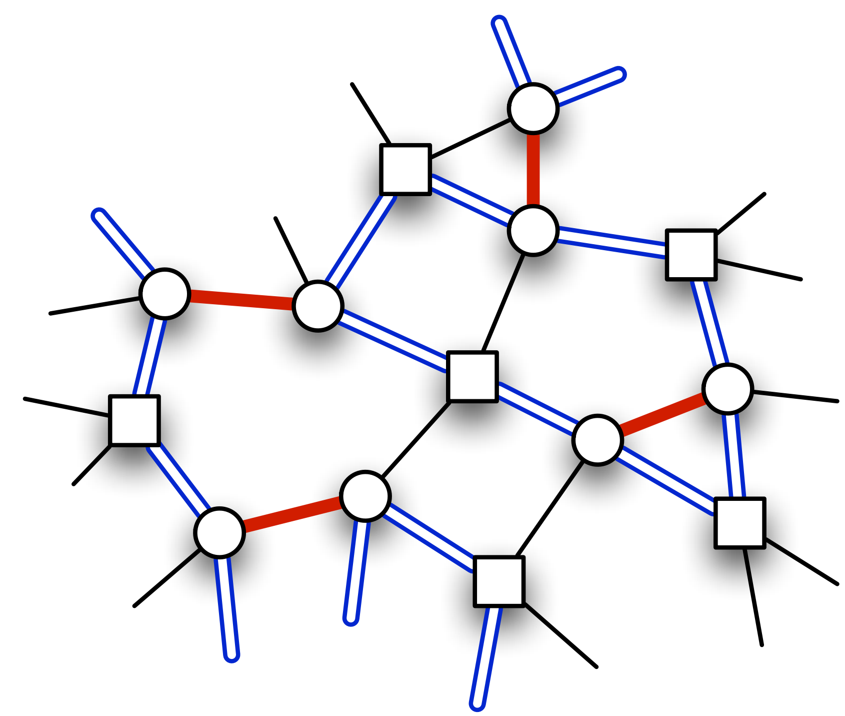
- What if we had this kind of configuration “everywhere” in a regular graph?
  - Approximation factor = 4?



Upper bounds: some key ideas

- This could happen in an infinite graph but not in a *finite* graph!
  - Simple counting argument, different types of endpoints
- We can always achieve better than 4-approximation
  - General case: a bit tedious case analysis, double-counting...





Distributed algorithms for edge dominating sets — summary

- Small edge dominating sets, port-numbering model, deterministic algorithms
  - Best possible approximation factors, exactly matching upper and lower bounds
- Open problem:
  - Can you do better in time $f(\Delta)$ if you have ***unique identifiers*** instead of mere port numbering?