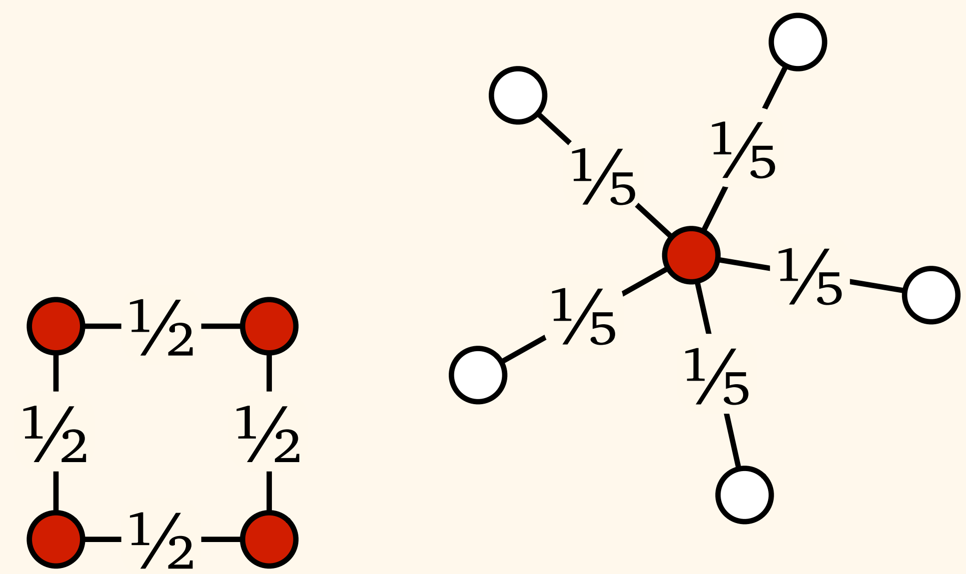
Vertex Covers & Edge Packings

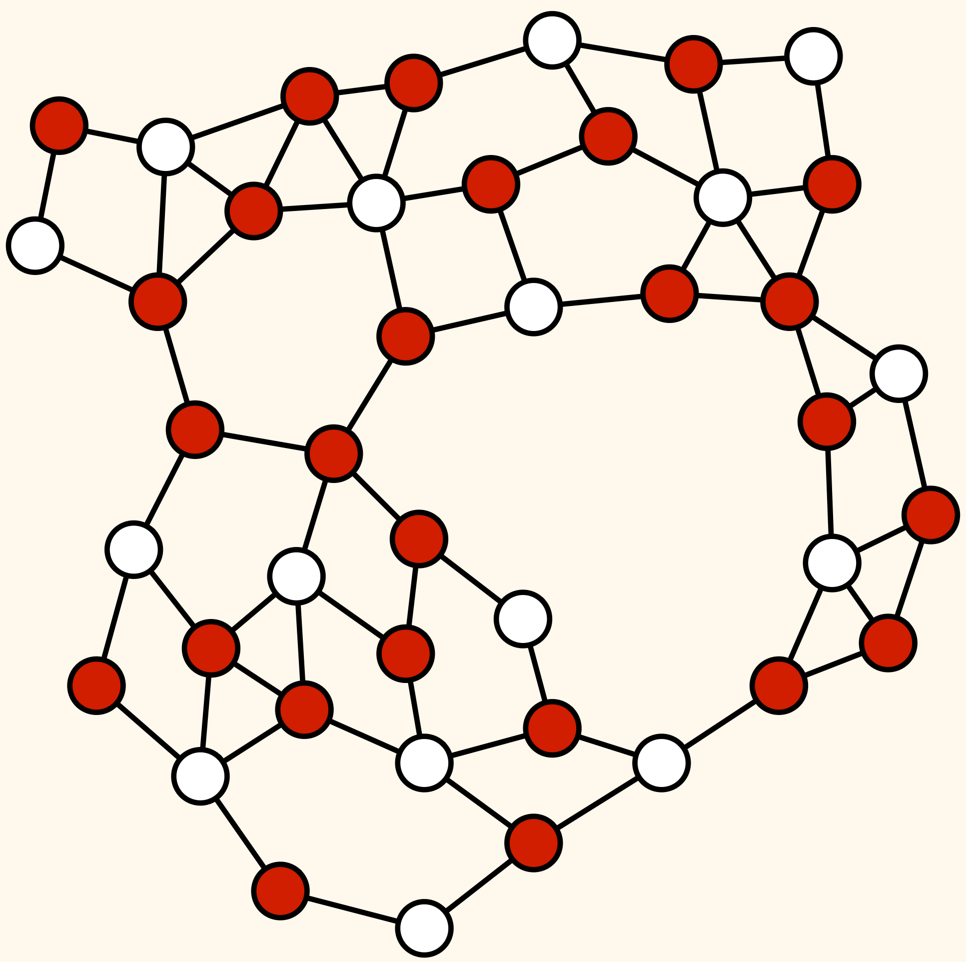


DDA Course

week 4

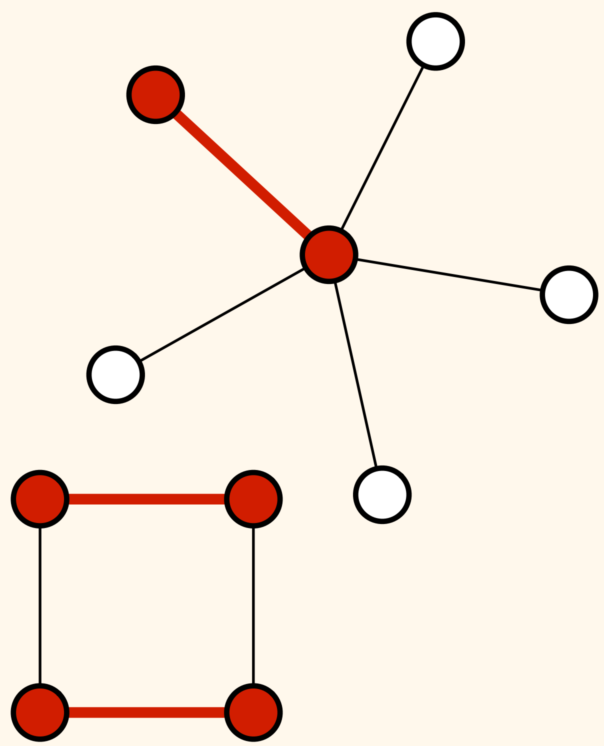
Vertex Cover

- Finding a minimum vertex cover is hard
- How to find good approximations?
- General idea: find something else first, show that it is useful…



Chapter 1

maximal matching

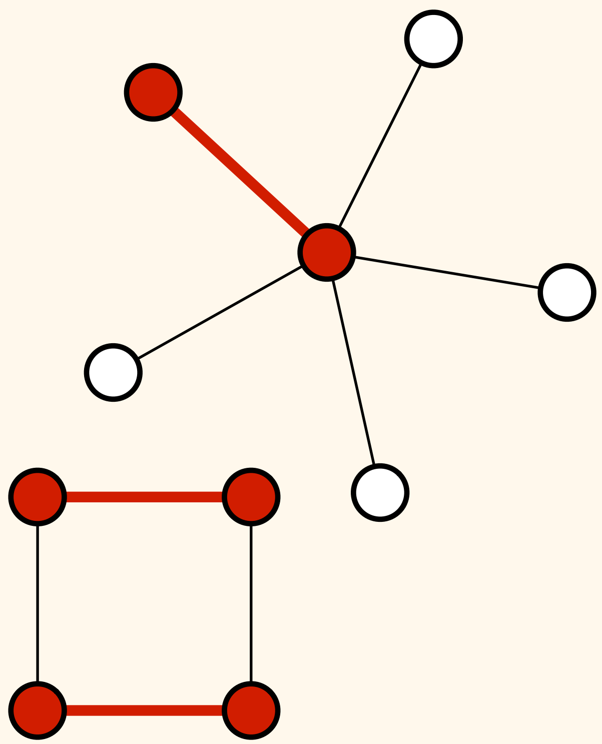


Exercise 1.3:

- find any maximal matching
- take all matched nodes
- 2-approximation of minimum vertex cover

Chapter 1

maximal matching



2-approx.

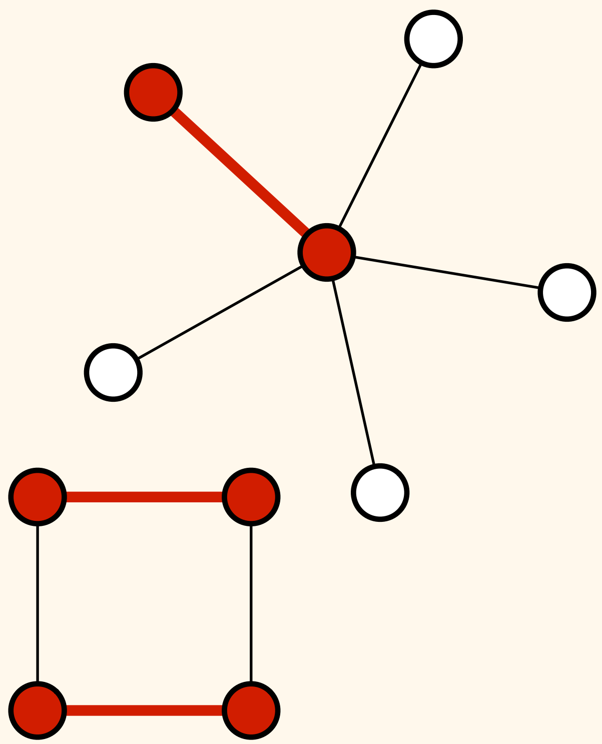
no distributed algorithm

Corollary 3.3:

- there is no distributed algorithm that finds a maximal matching

Chapter 1

maximal matching

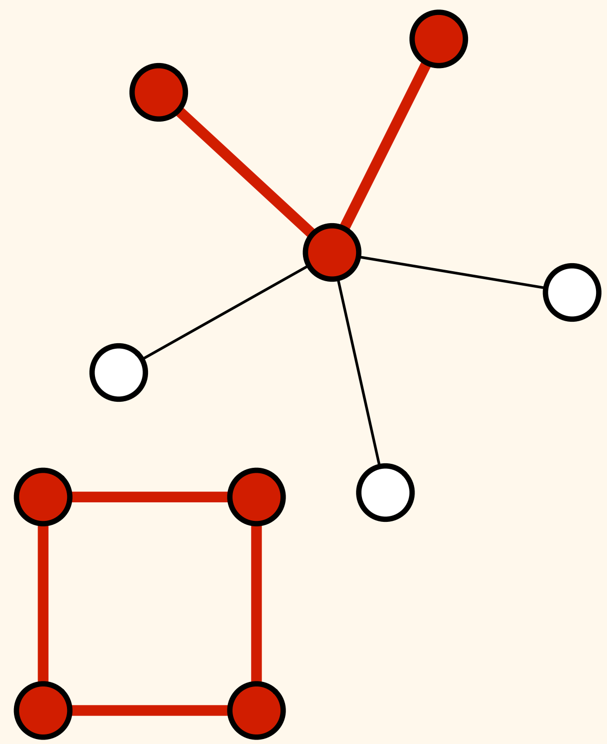


2-approx.

no distributed algorithm

Chapter 2

paths & cycles



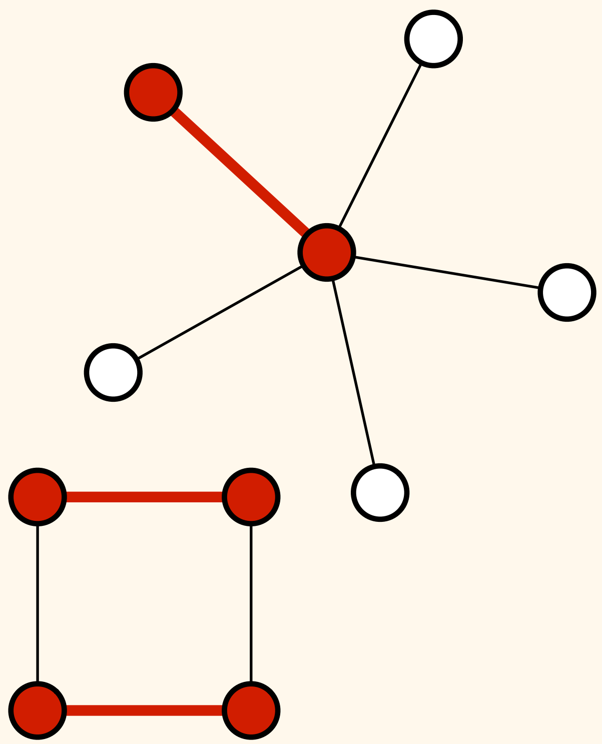
3-approx.

fast distributed algorithm

VC3

Chapter 1

maximal matching

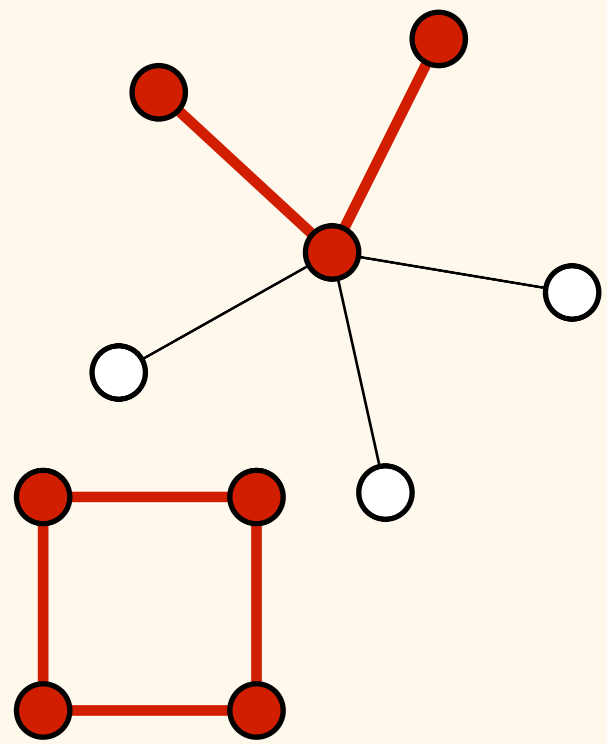


2-approx.

no distributed algorithm

Chapter 2

paths & cycles

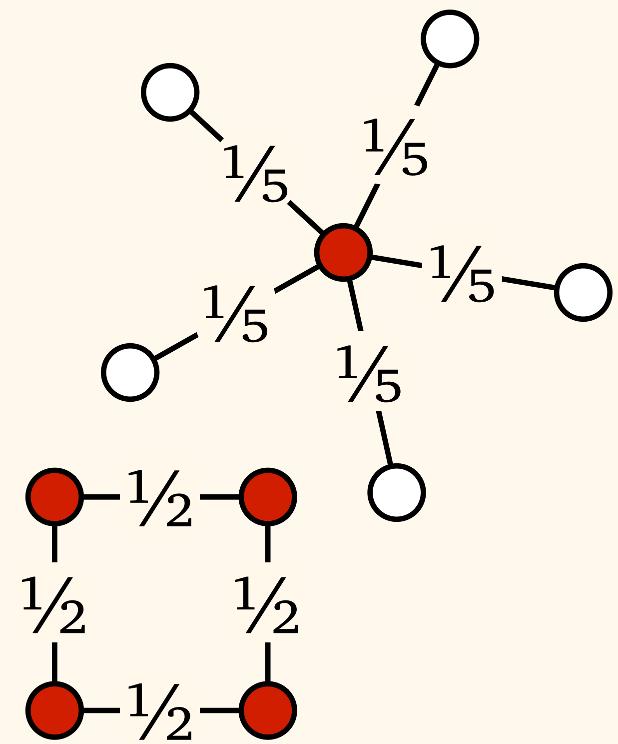


3-approx.

fast distributed algorithm

Chapter 4

edge packing

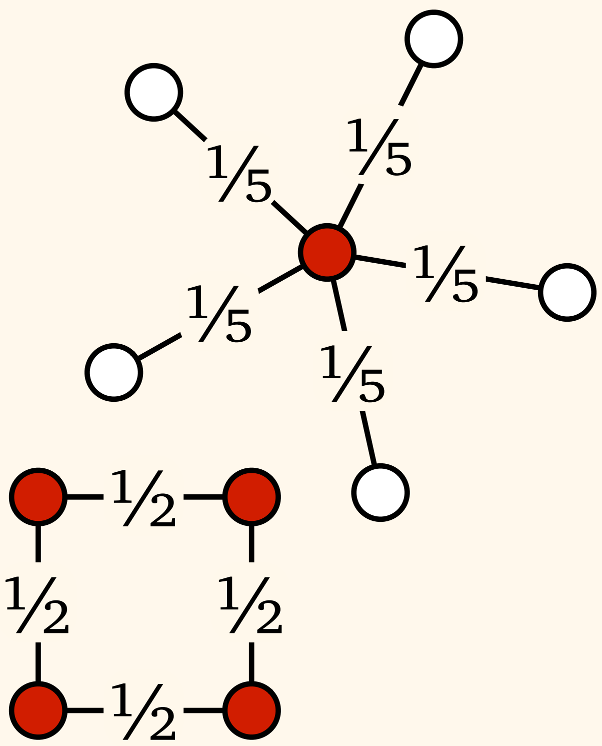


2-approx.

fast distributed algorithm

Edge Packing

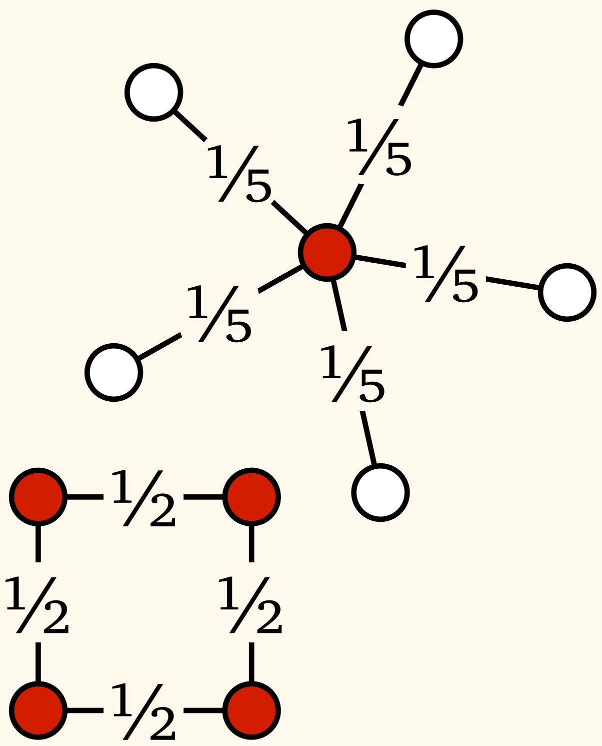
- Function $f: E \to [0, 1]$
  - $f[v]$ = sum of $f(e)$ over all edges e incident to v
- Constraints: $f[v] \leq 1$



$f[\circ] = 1/5$

$f[\bullet] = 1$

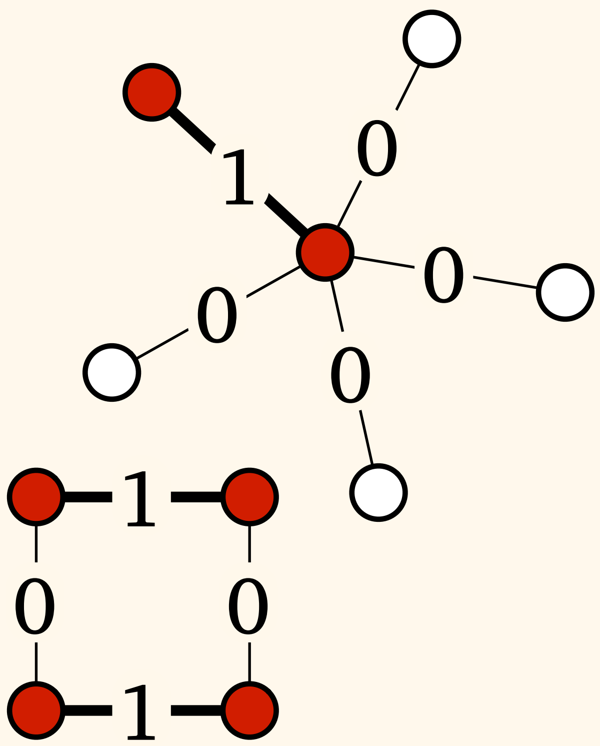
Edge Packing



- Function $f: E \to [0, 1]$
  - $f[v]$ = sum of $f(e)$ over all edges e incident to v
- Constraints: $f[v] \leq 1$
  - v is *saturated* if $f[v] = 1$
  - edge $e = \{u, v\}$ is *saturated* if u or v is saturated
  - edge packing is *maximal* if all edges are saturated

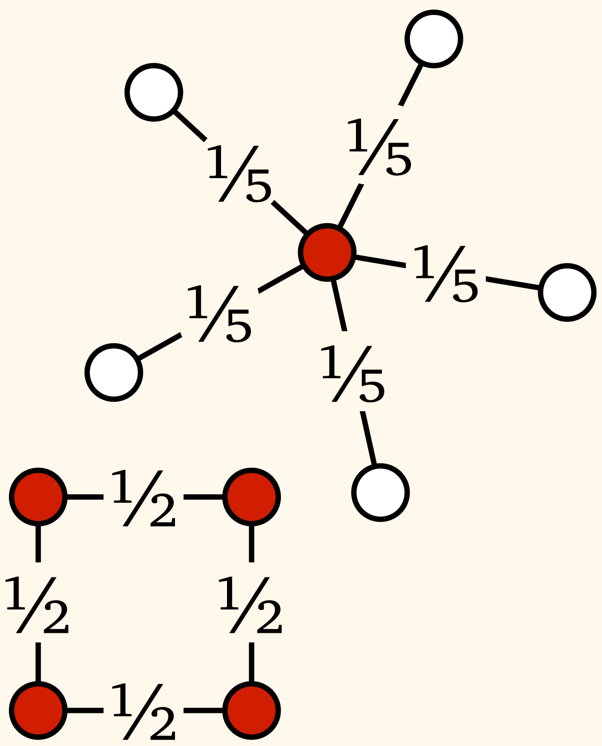
Edge Packing

- Function $f: E \to [0, 1]$
  - $f[v]$ = sum of $f(e)$ over all edges e incident to v
- Constraints: $f[v] \leq 1$
- "Fractional" matching



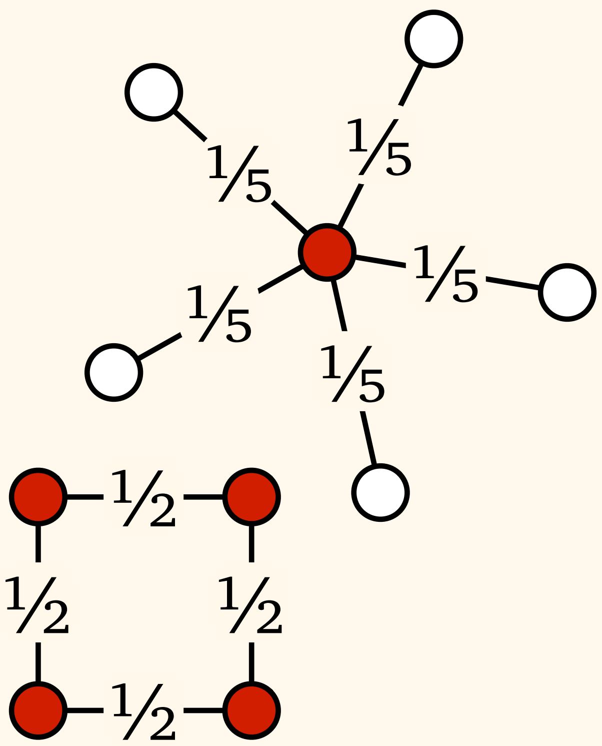
Edge Packing

- Find any maximal edge packing
- Set of saturated nodes: *vertex cover*
  - *Proof*: maximal
    = each edge saturated
    = each edge has a saturated endpoint
    = saturated nodes form a vertex cover

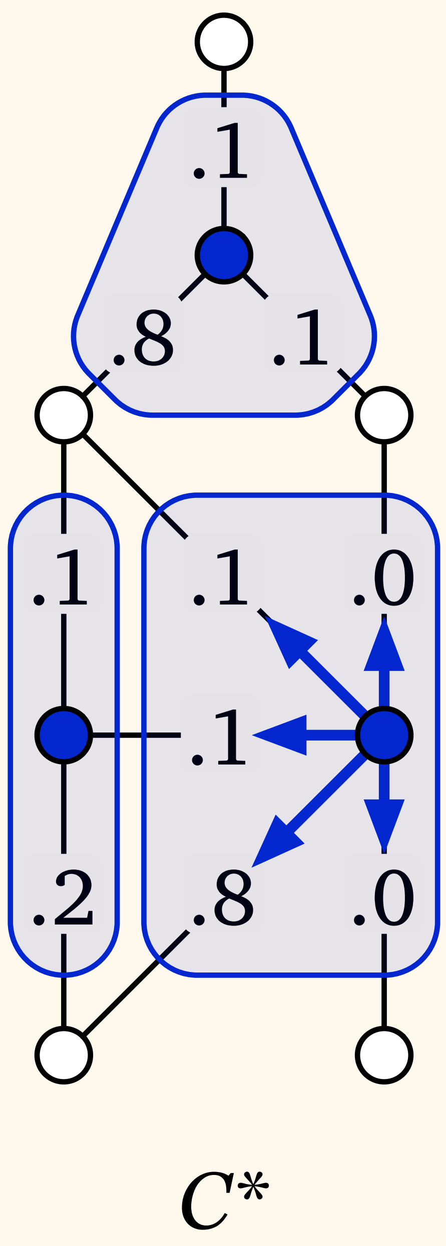


Edge Packing

- Find any maximal edge packing
- Set of saturated nodes: *2-approximation of minimum vertex cover*



Edge Packing



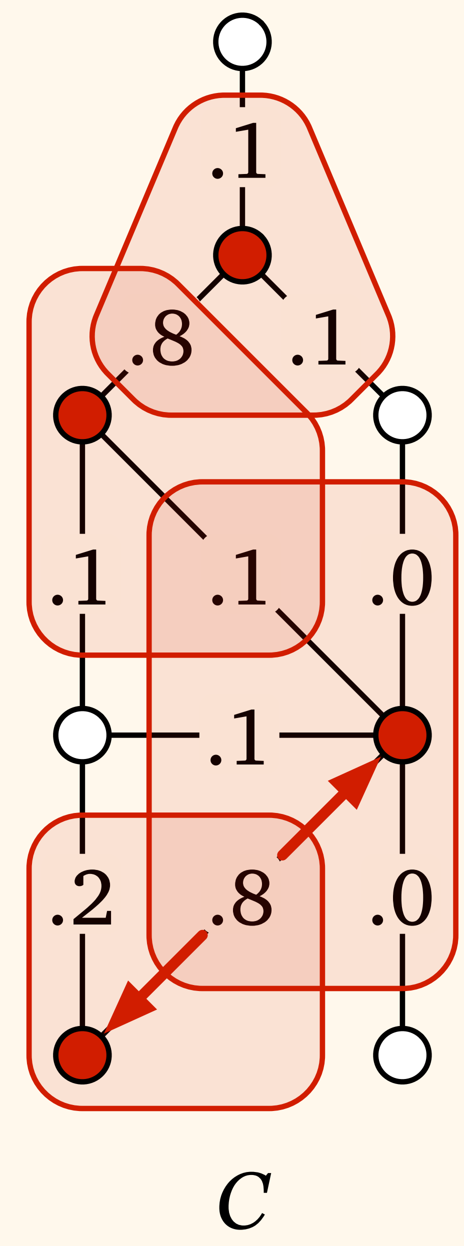
Each node $v \in C^*$
has 1 unit of money

Give $f(e)$ units
to each edge e

Double all money

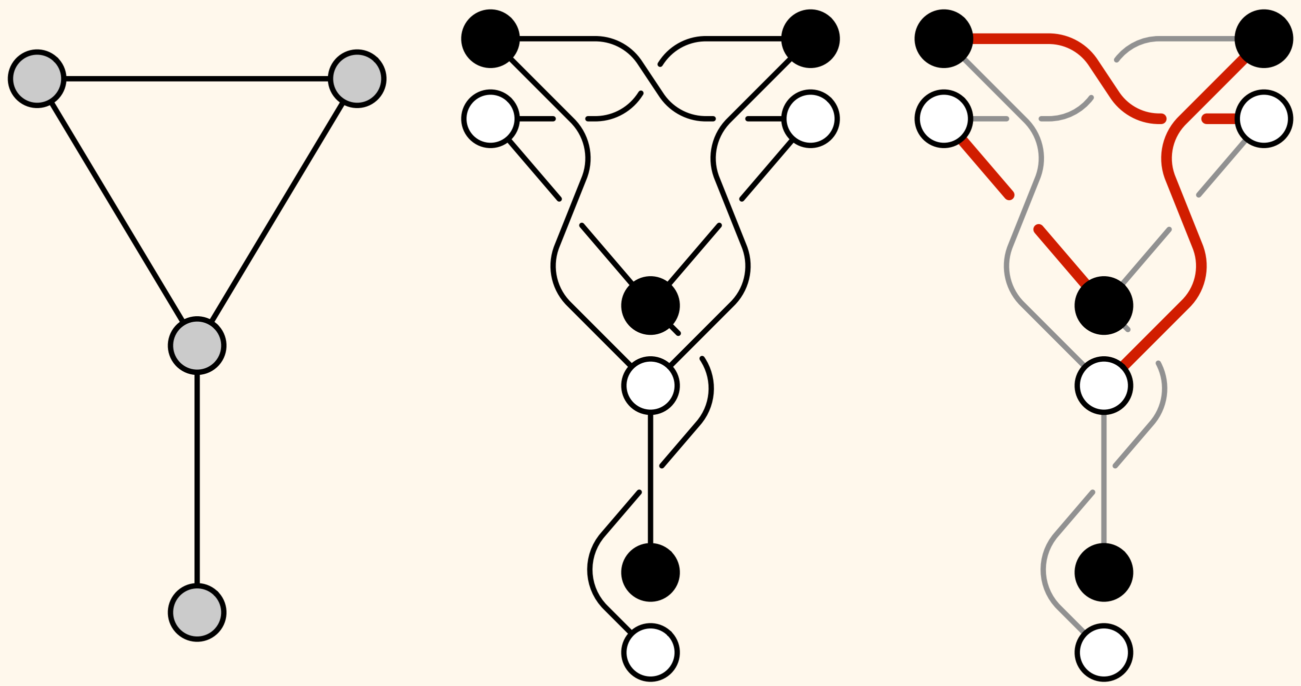
Give $f[v] = 1$ units to each
saturated node $v \in C$

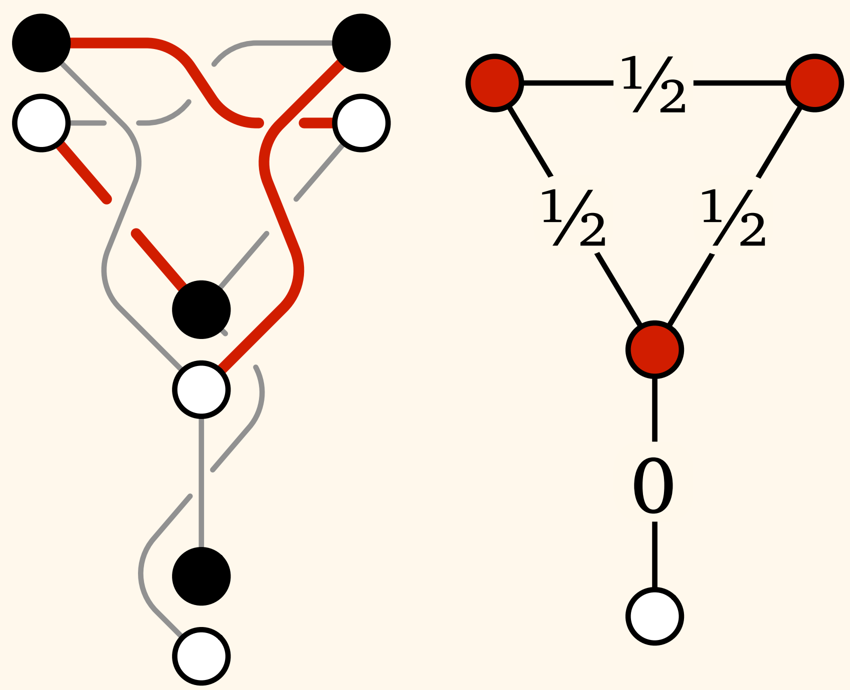
$|C| \leq 2\,|C^*|$

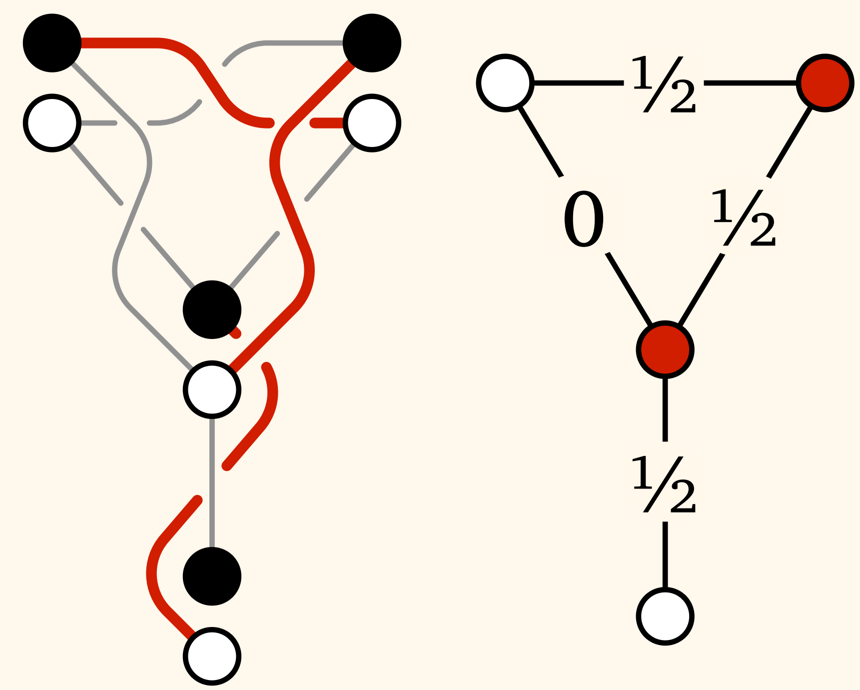


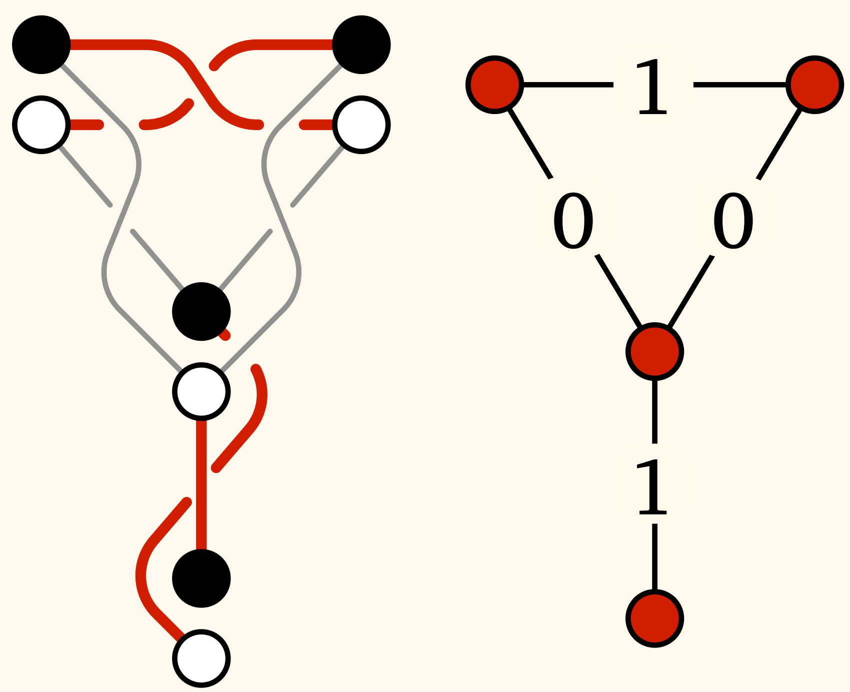
Edge Packing

- How to find maximal edge packings?
- Basic idea:
  - bipartite double covers
  - maximal matching
  - recursively!







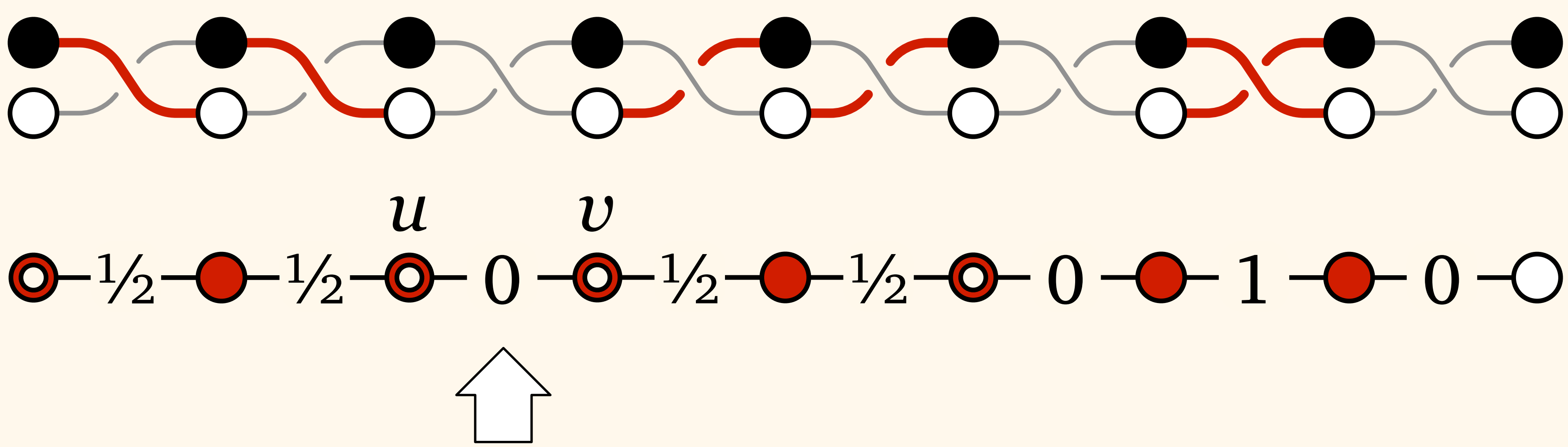


One edge: 1/2

Two edges: 1

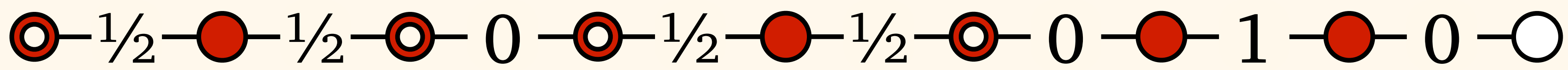
Edge Packing

- In general only "half-saturating"



unsaturated edge $e = \{u, v\}$
$f[u] = f[v] = 1/2$

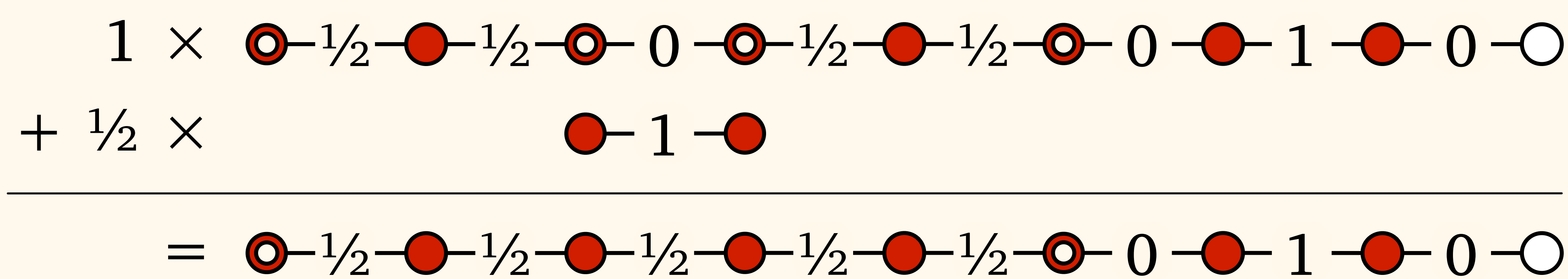
Half-saturating edge packing:



Unsaturated subgraph (*lower degrees*):

Recursively, find a *maximal* edge packing:

Combine solutions — *maximal* edge packing:



Edge Packing

- Recursion by maximum degree Δ
- Case $\Delta = 1$ trivial
- Assuming that case $\Delta - 1$ has been solved:
  - find a *half-saturating* edge packing f
  - recursively, find a *maximal* edge packing g for unsaturated subgraph (maximum degree $\Delta - 1$)
  - return *maximal* edge packing $h = f + g/2$

Summary

- Distributed algorithms that finds a *maximal edge packing*
    - in any graph of maximum degree Δ in time $O(\Delta^2)$
- Saturated nodes: *2-approximation of minimum vertex cover*