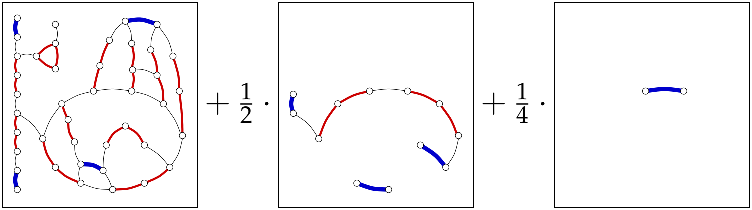
# A local 2-approximation algorithm for the vertex cover problem

Matti Åstrand · Patrik Floréen · Valentin Polishchuk
Joel Rybicki · Jukka Suomela · Jara Uitto

HIIT · University of Helsinki · Finland



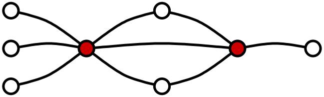

DISC, Elche, Spain, 23 September 2009

# Vertex cover

Given a graph $\mathcal{G} = (V, E)$, find a smallest $C \subseteq V$ that covers every edge of $\mathcal{G}$

- i.e., each edge $e \in E$ incident to at least one node in $C$

Classical NP-hard optimisation problem



# Vertex cover in a distributed setting

Node $=$ computer, edge $=$ communication link,
each node must decide whether it is in the cover $C$

Goals:

- deterministic algorithm
- running time independent of $n = |V|$
  (but may depend on maximum degree $\Delta$)
- *the best possible approximation ratio*

## □ Prior work

Kuhn et al. (2006):

- $(2+\epsilon)$-approximation in $O(\log \Delta/\epsilon^4)$ rounds

Czygrinow et al. (2008), Lenzen & Wattenhofer (2008):

- $(2-\epsilon)$-approximation requires $\Omega(\log^* n)$ rounds, even if $\Delta = 2$

What about 2-approximation?

Is it possible in $f(\Delta)$ rounds, for some $f$?

# Prior work

Kuhn et al. (2006):

- $(2+\epsilon)$-approximation in $O(\log \Delta/\epsilon^4)$ rounds

Czygrinow et al. (2008), Lenzen & Wattenhofer (2008):

- $(2-\epsilon)$-approximation requires $\Omega(\log^* n)$ rounds, even if $\Delta = 2$

What about 2-approximation?

Is it possible in $f(\Delta)$ rounds, for some $f$? – *Yes!*

## Contribution

Deterministic $2$-approximation algorithm for vertex cover

- Running time $(\Delta + 1)^2$ synchronous rounds

No $O$-notation needed here…

# Contribution

Deterministic $2$-approximation algorithm for vertex cover

- Running time $(\Delta + 1)^2$ synchronous rounds

Surprise: node identifiers not needed

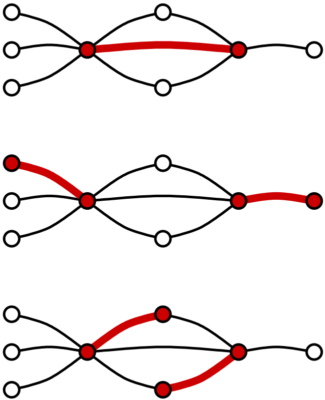
- Negative result for $(2 - \epsilon)$-approximation holds even if there are unique node identifiers
- Our algorithm can be used in *anonymous networks*

## Background: maximal matching

In a centralised setting, 2-approximation is easy: find a *maximal matching*, take all matched nodes

But matching requires $\Omega(\log^* n)$ rounds and unique identifiers

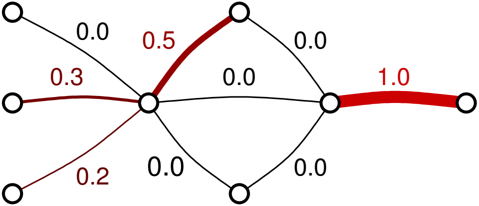
- symmetry breaking!



# Background: maximal edge packing

*Edge packing* = edge weights from $[0, 1]$,
for each node $v \in V$, total weight on incident edges $\leq 1$

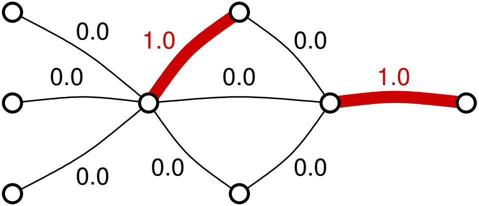
*Maximal*, if no weight can be increased



# ☐ Background: maximal edge packing

Maximal matching $\implies$ maximal edge packing

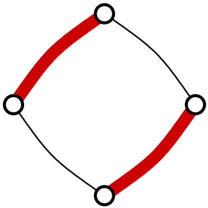
(matched: weight $1$, unmatched: weight $0$)

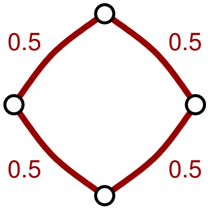


## Background: maximal edge packing

Maximal matching requires symmetry breaking

Maximal edge packing does not

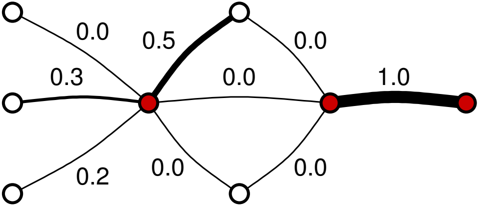




# Background: maximal edge packing

Node *saturated* if total weight on incident edges $= 1$

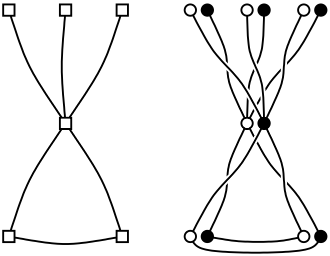
Saturated nodes: $2$-approximation of vertex cover
(proof: LP duality)



# Finding an edge packing

Construct a 2-coloured *bipartite double cover*

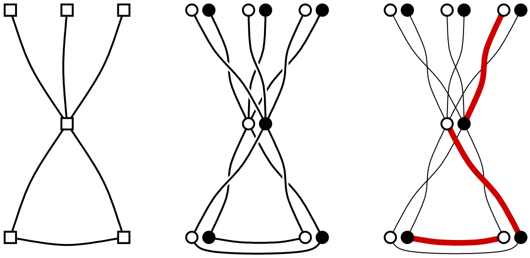
Each original node simulates two nodes of the cover



# ☐ Finding an edge packing

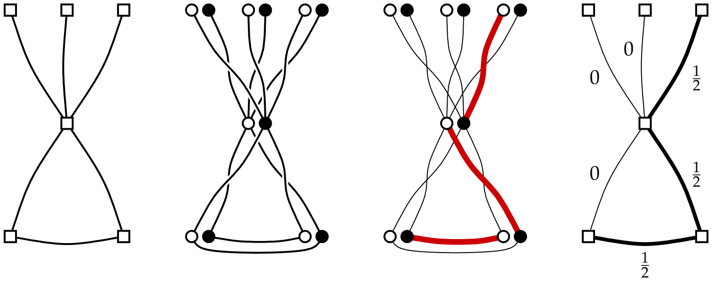
Find a maximal matching in the 2-coloured graph

Easy in $O(\Delta)$ rounds



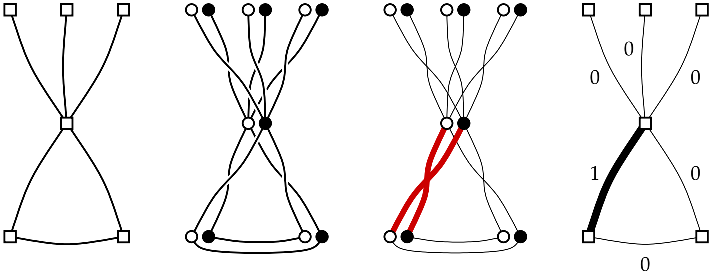
# ☐ Finding an edge packing

Give $\frac{1}{2}$ units of weight to each edge in matching



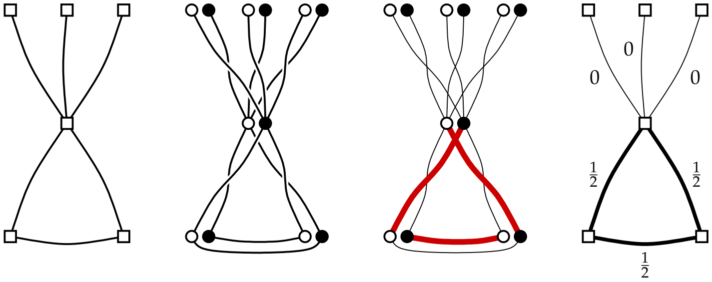
# ☐ Finding an edge packing

Many possibilities. . .



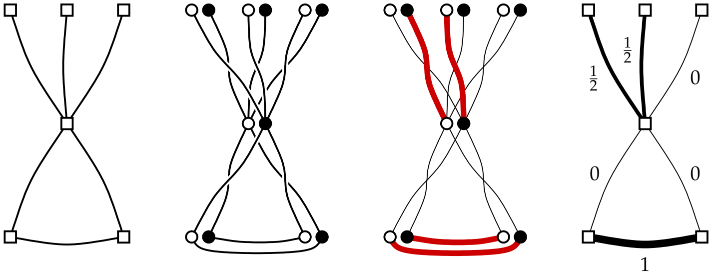
# Finding an edge packing

Many possibilities. . .



# Finding an edge packing

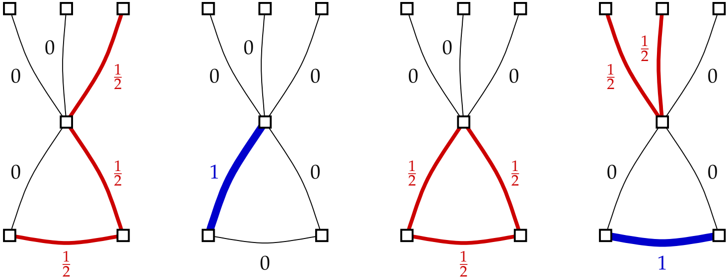
Many possibilities…



# Finding an edge packing

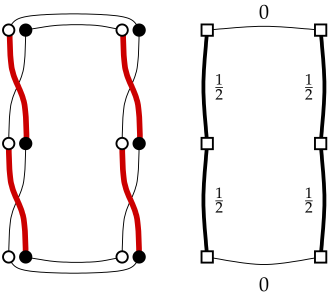
Always: weight $\frac{1}{2}$ paths and cycles and weight $1$ edges

Valid edge packing



# Finding a maximal edge packing

Not necessarily maximal – but all unsaturated edges adjacent to two weight $\frac{1}{2}$ edges

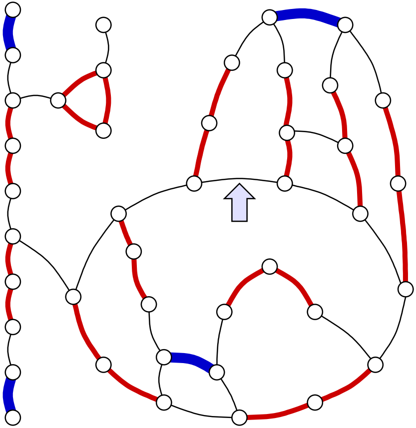


# Finding a maximal edge packing

In any graph:

Unsaturated edges adjacent to two weight $\frac{1}{2}$ edges

$\Delta = 3$



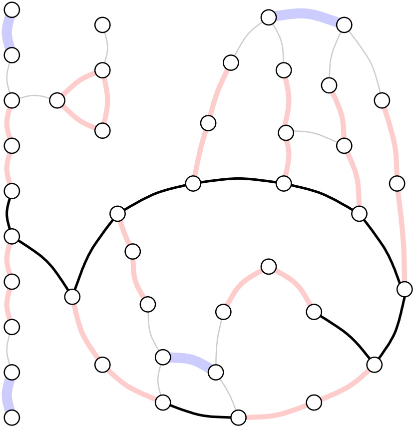
# Finding a maximal edge packing

In any graph:

Unsaturated edges adjacent to two weight $\frac{1}{2}$ edges

Delete saturated edges

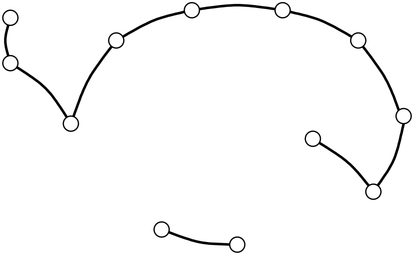
$\Delta = 3 \longrightarrow \Delta = 2$



# Finding a maximal edge packing

Each node has lost
at least one neighbour

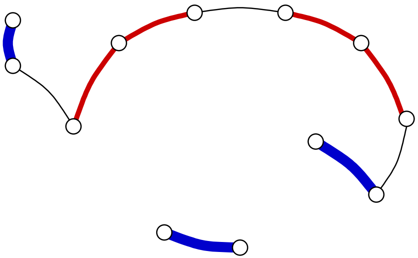
Remaining capacity
of each node is
exactly $\frac{1}{2}$



$\Delta = 3 \;\rightarrow\; \Delta = 2$

# Finding a maximal edge packing

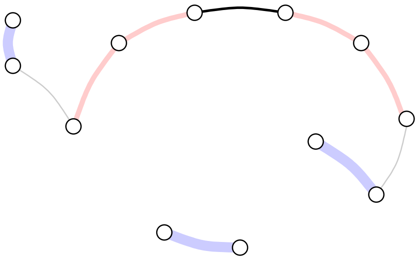
Repeat



$\Delta = 2$

## ☐ Finding a maximal edge packing

Delete saturated edges



$\Delta = 2 \longrightarrow \Delta = 1$

# Finding a maximal edge packing

Each node has lost
at least one neighbour

Remaining capacity
of each node is
exactly $\frac{1}{4}$

$\Delta = 2 \rightarrow \Delta = 1$

# Finding a maximal edge packing

Repeat. . .

$\Delta = 1$

## □ Finding a maximal edge packing

Repeat…

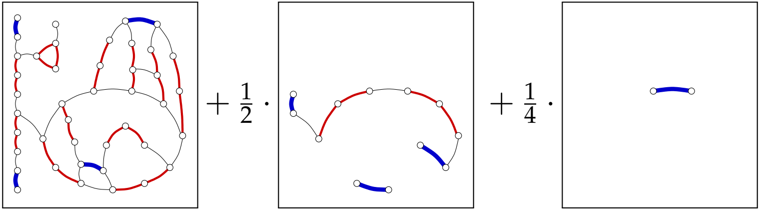
Maximum degree decreases
on each iteration

Everything saturated in
$\Delta$ iterations

# Summary

Maximal edge packing in $(\Delta+1)^2$ rounds

$\implies$ 2-approximation of vertex cover



$+\ \frac{1}{2}\cdot$ $\quad$ $+\ \frac{1}{4}\cdot$

$\Delta = 3$ $\quad$ $\Delta = 2$ $\quad$ $\Delta = 1$