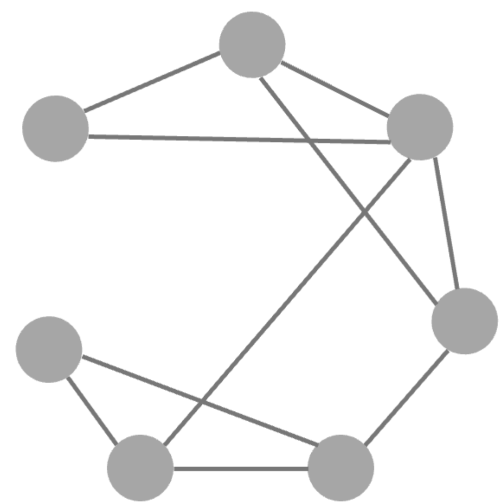
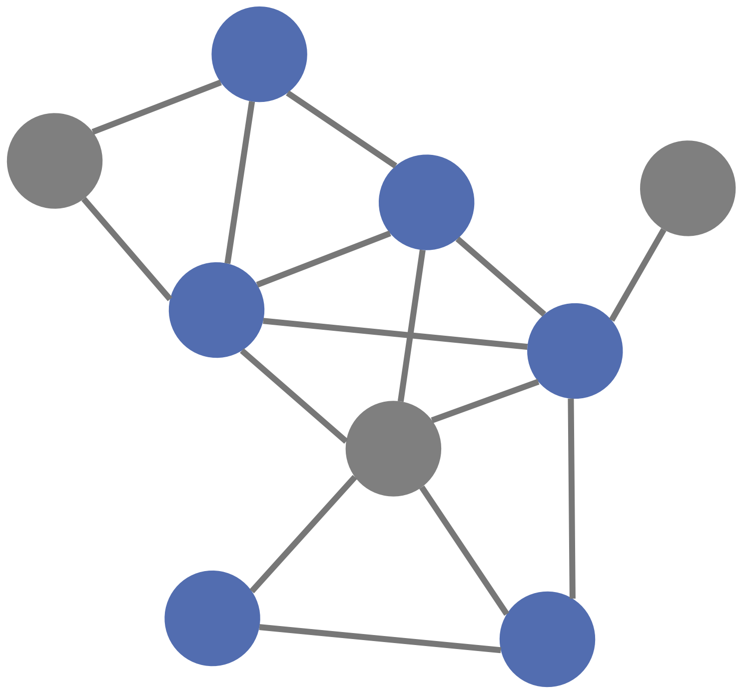
Greedily Growing a Maximal Independent Set to Approximate Vertex Cover



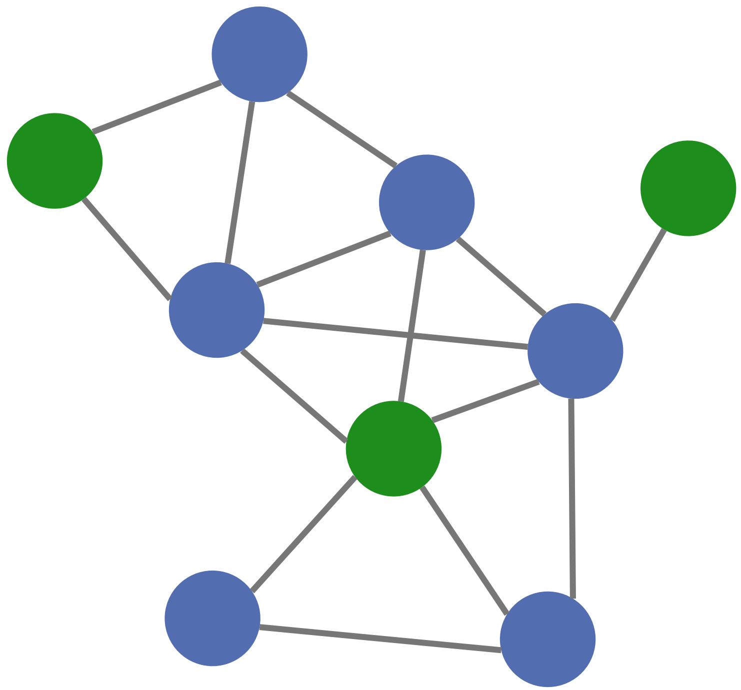
Nate Veldt
Texas A&M University

Finding a minimum vertex cover **(Min-VC)** is one of the most well-studied NP-hard problems.



A **vertex cover** is a set of nodes that “covers” all edges.

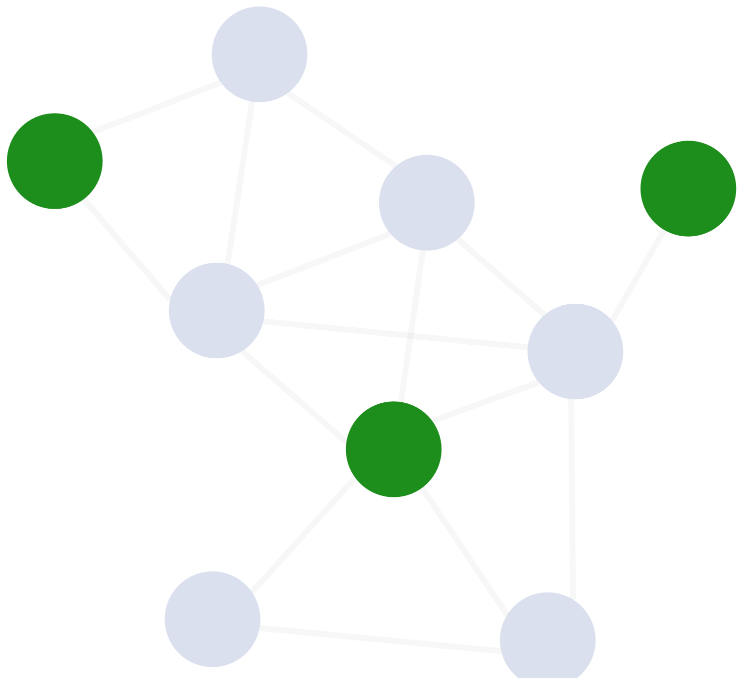
Finding a minimum vertex cover (Min-VC) is one of the most well-studied NP-hard problems.



A **vertex cover** is a set of nodes that "covers" all edges.

The complement set is an **independent set**

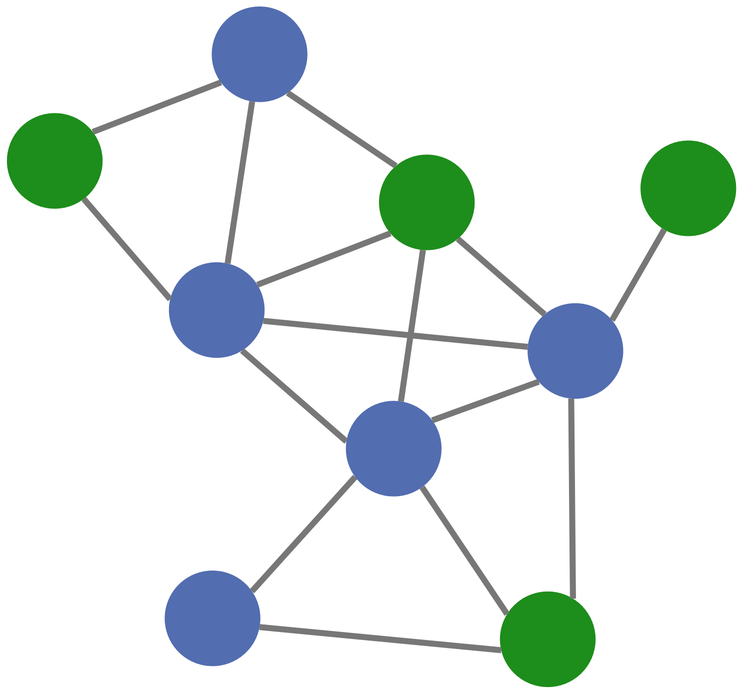
Finding a minimum vertex cover (Min-VC) is one of the most well-studied NP-hard problems.



A **vertex cover** is a set of nodes that "covers" all edges.

The complement set is an **independent set**

Finding a minimum vertex cover (Min-VC) is one of the most well-studied NP-hard problems.



Simple linear-time 2-approximation algorithms for **Min-VC** have existed since the 1970s and 1980s.

For every fixed ε > 0, it is UGC-hard to obtain a 2-ε approximation. (KR '08)

This talk will cover a simple new 2-approximation algorithm for minimum vertex cover.

Khot & Regev, Jour. Comp. and Sys. Sci. 2008

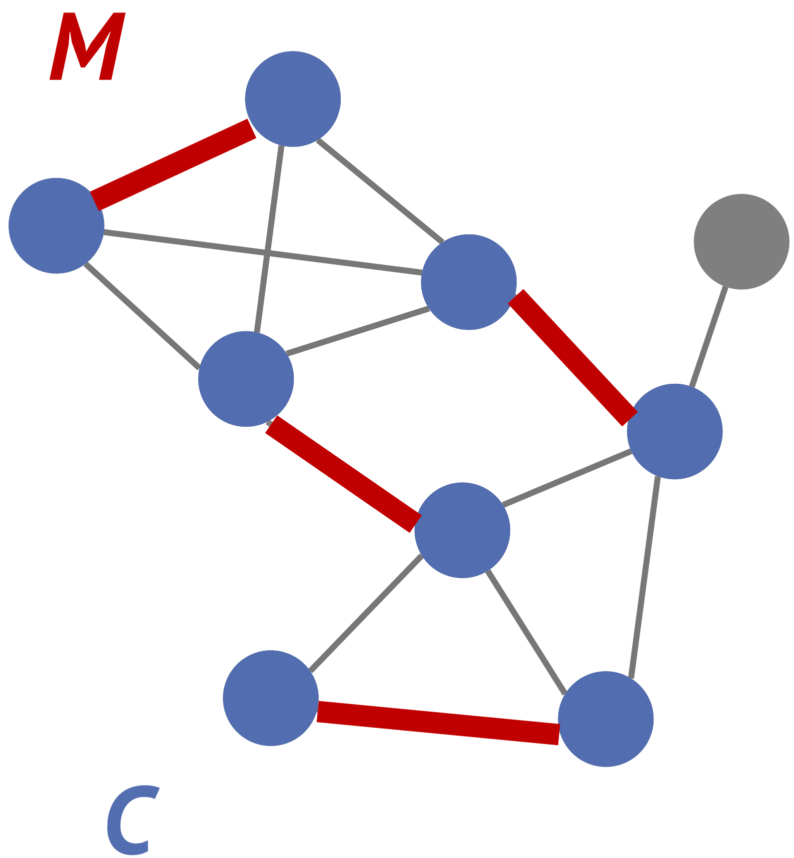
Simple, linear-time Min-VC algorithms have been known for decades. So what is new, and what's this talk about?

1. New and unifying connections
   - Equivalence with an existing greedy independent set algorithm
   - New connections to *correlation clustering* algorithms
   - This has implications for a parallel MIS algorithm that simultaneously solves multiple problems

2. Simple and fast algorithms for problems that can be reduced to **Min-VC**

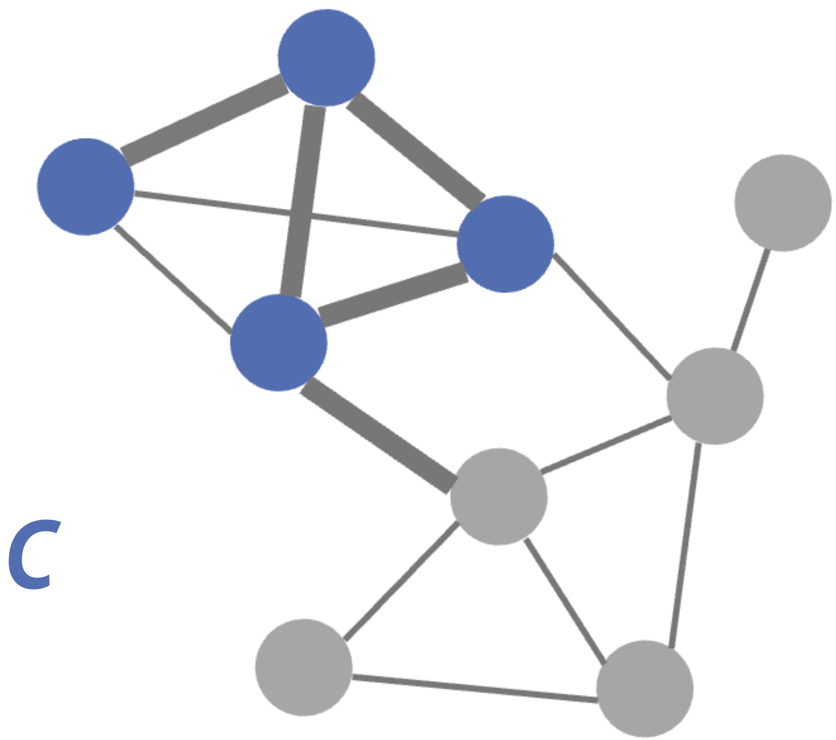
3. Open questions

A standard 2-approximation is to include all nodes from a maximal matching



Theorem.
For every **maximal matching,**
the **endpoints** are 2-approx for **Min-VC.**

This can be implemented by iterating through edges



O(E) runtime

for each $(u, v) \in E$

  if (u, v) is not covered

    $C \leftarrow C \cup \{u, v\}$

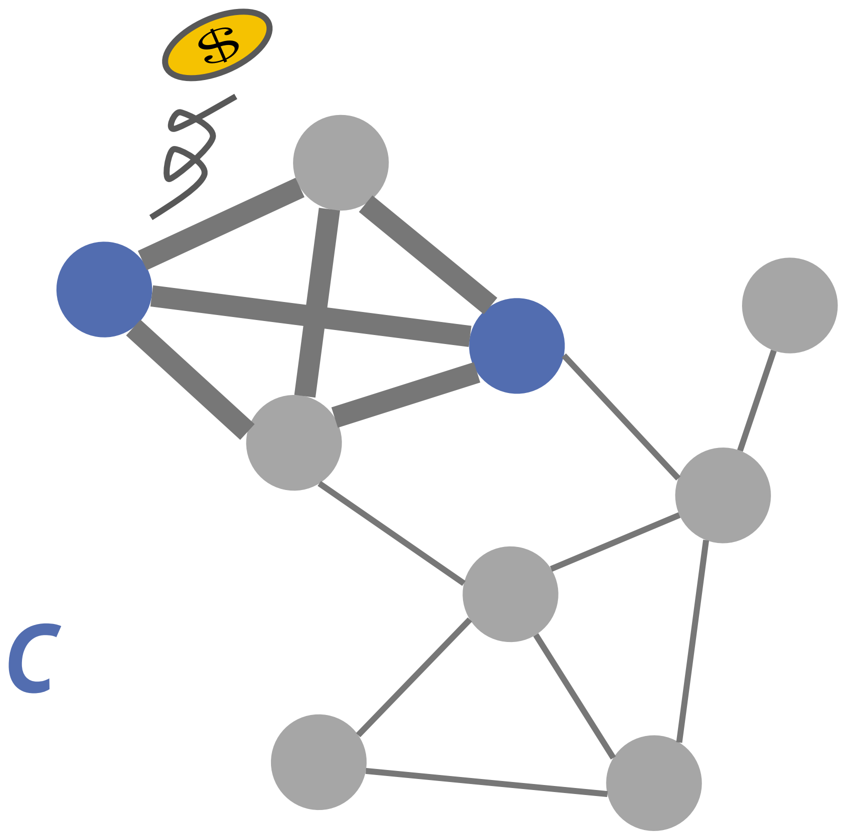
  end

end

This has also been generalized to node-weighted Min-VC. (B-Y, E '85)

Bar-Yehuda & Even, Ann. Dis. Math 1985

Pitt's algorithm is a randomized 2-approximation



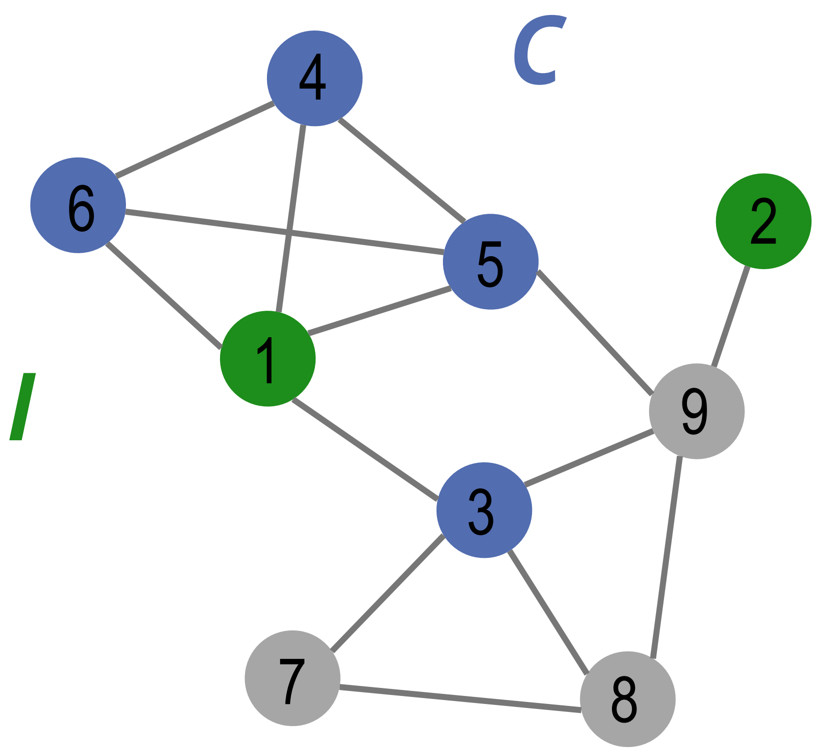
O(E) runtime

for each $(u, v) \in E$
    if (u, v) is not covered
        choose u or v at random
    end
end

This has also been generalized to node-weighted Min-VC

Leonard Pitt, Yale University, 1985

The new* algorithm generates a random node permutation and greedily grows a maximal independent set



O(E) runtime (unweighted case)

generate random permutation σ

for $i = 1$ to $|V|$

  visit node $\sigma(i)$

  add to I if possible

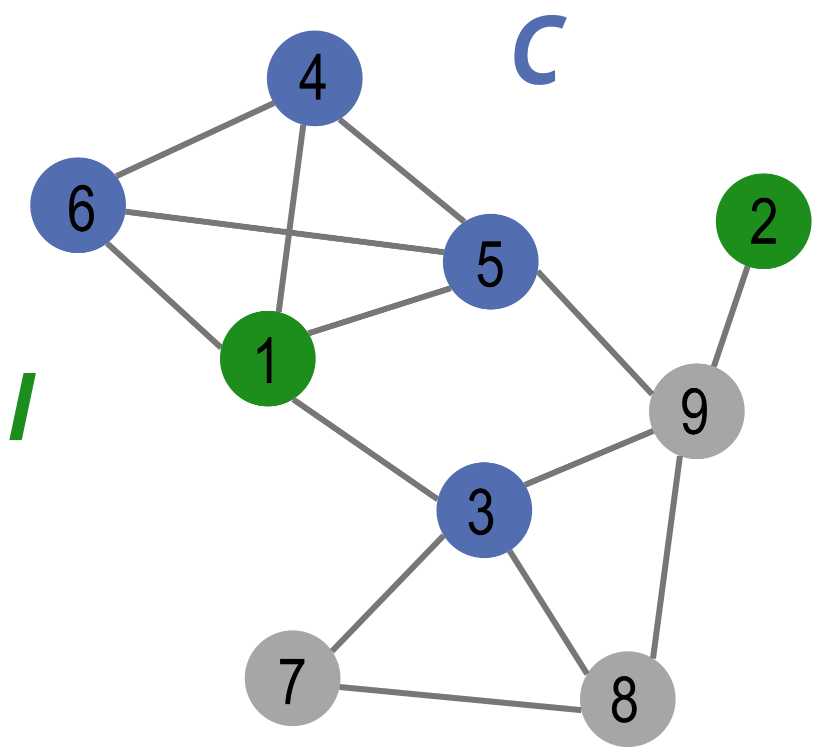
  else add to C

end

Permutation depends on node weight

Can be easily generalized to weighted Min-VC!
The runtime becomes O(V log V + E)

The unweighted version has been used for decades as a Greedy Maximal Independent Set algorithm



A sample of references on **GreedyMIS**

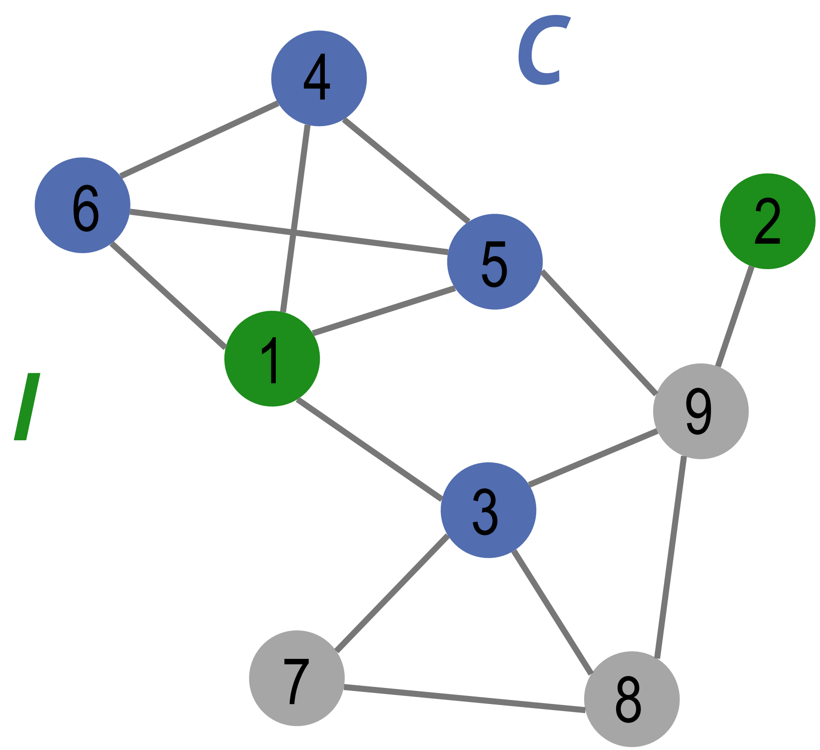
- Coppersmith, Raghavan & Tompa, FOCS 1987
- Gamarnik & Goldberg, Prob. & Computing 2010
- Blelloch, Fineman & Shun, SPAA 2012
- Bennet & Bohman, RSA 2016
- Fischer & Noever, TALG 2019, SODA 2018
- **J. Shi,** Wang, Shang, (survey) 2018

These focus on MIS, no mention of Min-VC

Theorem (Veldt, 2022). GreedyMIS is a randomized 2-approximation algorithm for Min-VC (and can be generalized to node-weighted graphs).

https://arxiv.org/abs/2209.04673

A corollary regarding Parallel GreedyMIS



A simple parallel version of GreedyMIS selects multiple IS nodes in each round.

Returns same output as the sequential version!

WHP, it terminates in O(log n) rounds (FN '18).

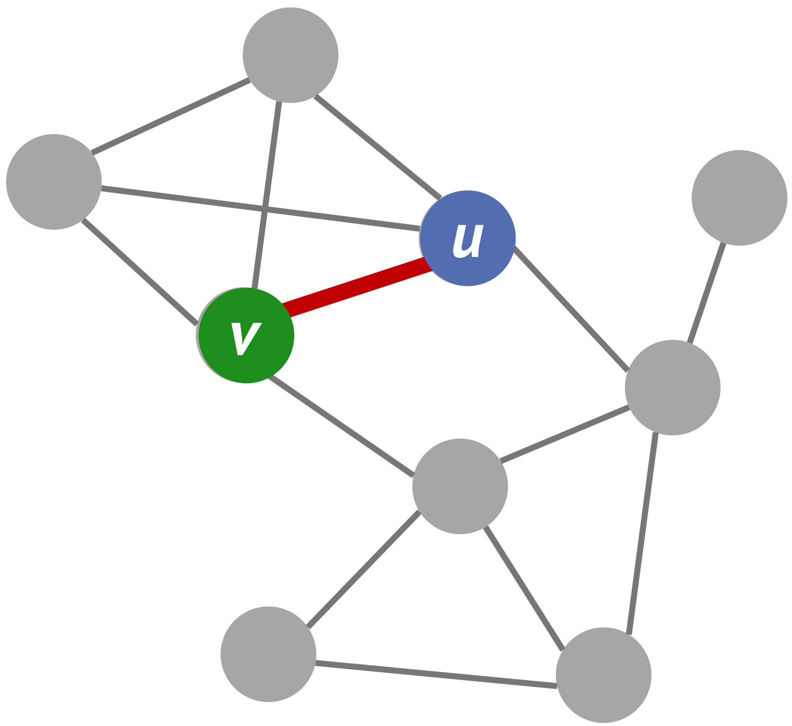
Corollary. Parallel GreedyMIS returns:

- A maximal independent set
- 2-approximate vertex cover
- 3-approximation for correlation clustering
- 3-approximation for STC+

(related to Lutz's talk)

Fischer & Noever, SODA 2018

Proof sketch



We need to "pay" for nodes added to **C**.

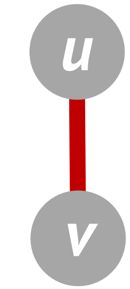
Say *u* is added to **C.**

It must have had a neighbor *v* that was added to **I.**

Charge edge *(u,v)* for adding *u* to **C**.

Proof sketch

p_{uv} = probability (u,v) is charged



Step 1: algorithm's expected cost is:

$$\mathbb{E}[cost] = \sum_{(u,v)\in E} p_{uv}$$

Dual LP (fractional matching)

$$\begin{aligned} \max \quad & \sum_{(u,v)\in E} y_{uv} \\ \text{s.t. } \forall u \in V: \quad & \textstyle\sum_{i:(u,i)\in E} y_{ui} \leq 1 \\ & y_{uv} \geq 0 \text{ for } (u,v) \in E. \end{aligned}$$

Step 2: Prove (implicitly) that

$y_{uv} = \frac{1}{2} p_{uv}$ is feasible for dual.

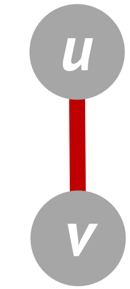
$$\text{Min-VC} \geq \text{OPT Dual LP} \geq \sum_{(u,v)\in E} \frac{p_{uv}}{2} = \frac{\mathbb{E}[cost]}{2}$$

LP duality

Proof

Inspired by the proof that Pivot is a 3-approximation for correlation clustering!

$$p_{uv} = \text{probability } (u,v) \text{ is charged}$$



…ithm's expected cost is:

$$\mathbb{E}[cost] = \sum_{(u,v)\in E} p_{uv}$$

Dual LP (fractional matching)

$$\max \sum_{(u,v)\in E} y_{uv}$$
$$\text{s.t. } \forall u \in V: \quad \textstyle\sum_{i:(u,i)\in E} y_{ui} \leq 1$$
$$y_{uv} \geq 0 \text{ for } (u,v) \in E.$$

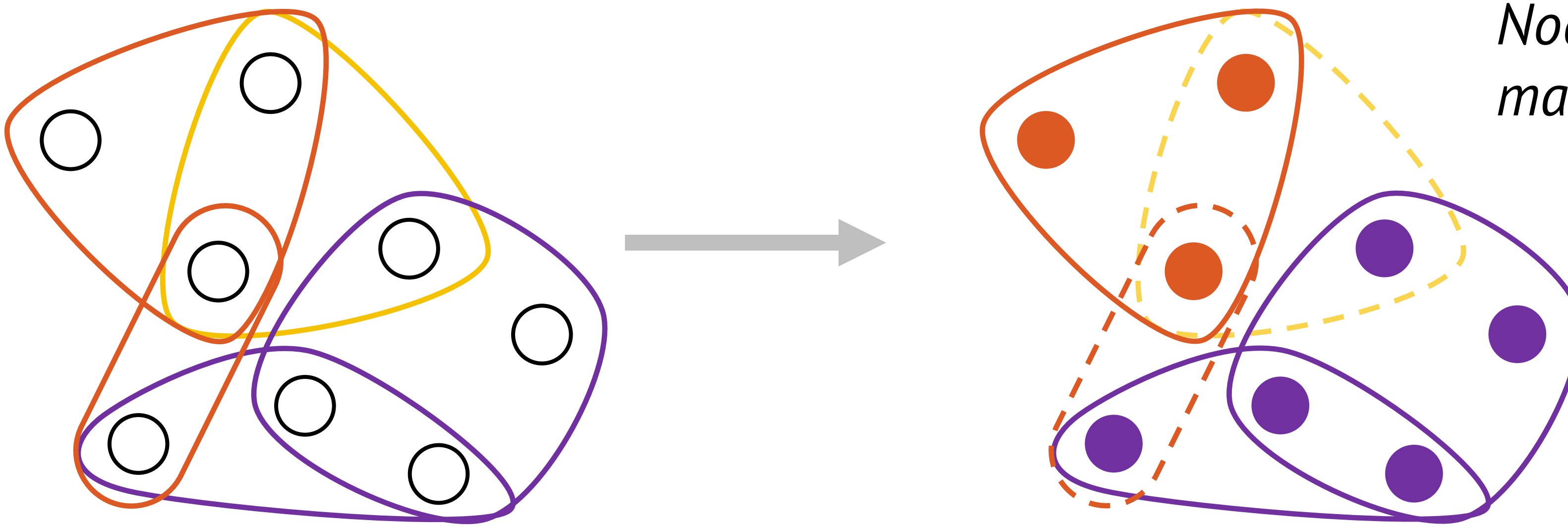
Step 2: Prove (implicitly) that

$y_{uv} = \frac{1}{2} p_{uv}$ is feasible for dual.

$$\text{Min-VC} \geq \text{OPT Dual LP} \geq \sum_{(u,v)\in E} \frac{p_{uv}}{2} = \frac{\mathbb{E}[cost]}{2}$$

LP duality

Colored Hypergraph Clustering can be reduced to Min-VC in an approximation preserving way



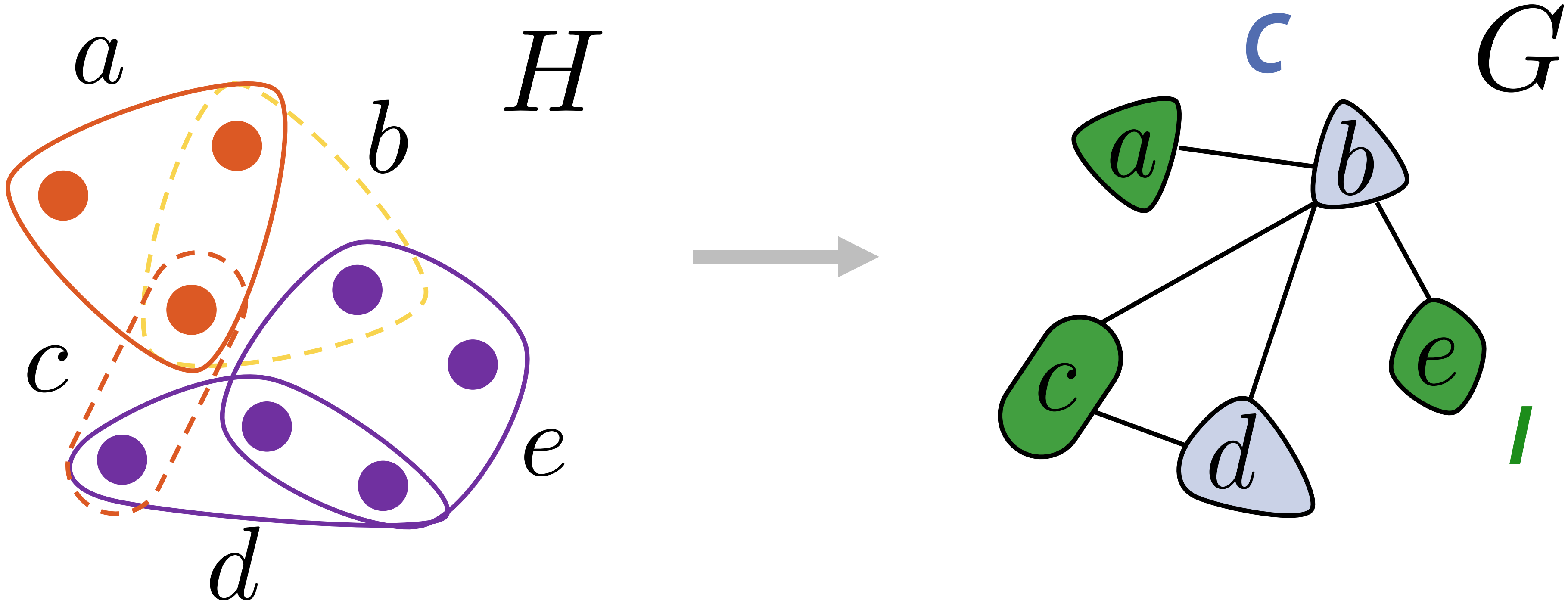
Two unsatisfied edges:
Node colors don't match edge color

Edge-Colored Hypergraph Clustering. Given an edge-colored hypergraph, color nodes in a way that leaves the fewest number of edges *unsatisfied*.

Equivalently: *delete (or cover) min # number of edges to destroy pairs of edges that overlap and are a different color.*

Amburg, Veldt, & Benson, WWW 2020, Veldt 2022

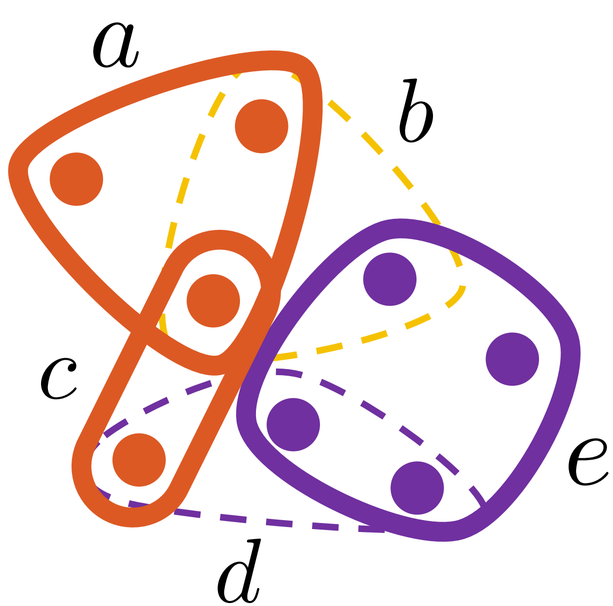
Colored Hypergraph Clustering can be reduced to Min-VC in an approximation preserving way

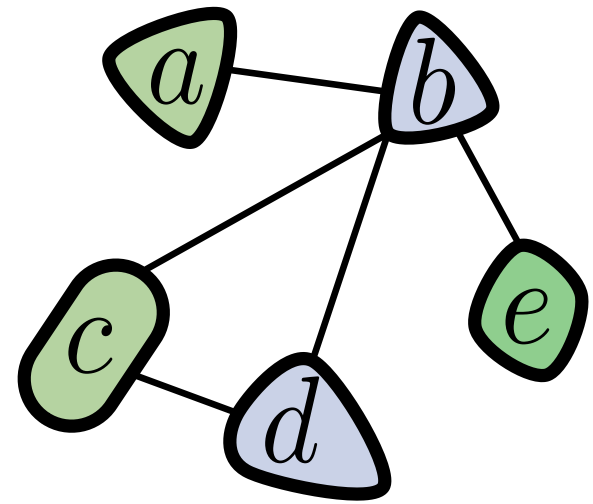


Naive 2-approximation. Explicitly form *G* and run an existing 2-approximation algorithm that iterates through all edges in *G*.

Runtime. $O(\sum_{v \in V} d_v^2 + |E|^2)$, *|E| = # hyperedges*

We can implicitly implement GreedyMIS in linear-time!





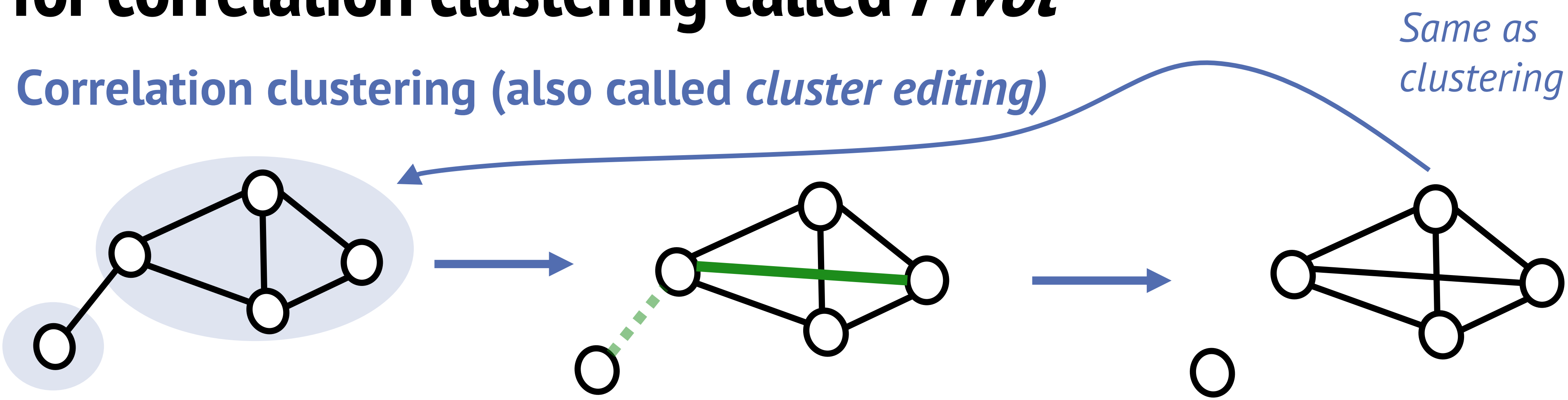
When we visit an edge *e*, we just check each node in *e*, which takes O(|*e*|) time

We never actually form this graph or consider all its edges.

Applying this procedure to H takes $O\left(\sum_{e \in E} |e|\right)$ time.

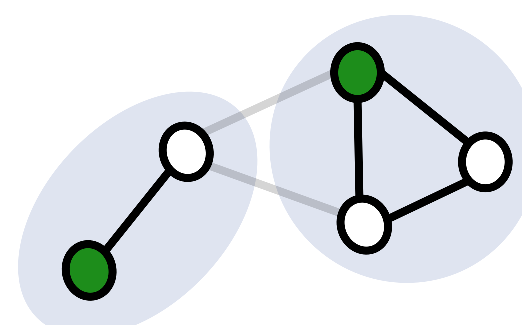
The proof is inspired by a 3-approximation algorithm for correlation clustering called *Pivot*

Correlation clustering (also called *cluster editing*)

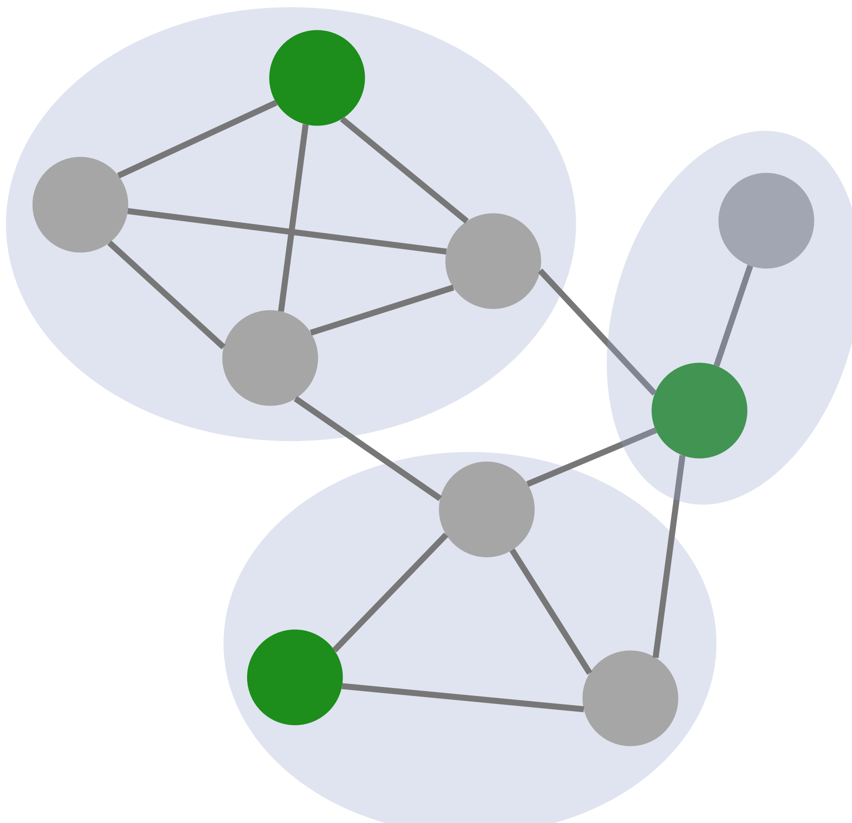


Goal: add/delete a minimum # of edges to change into disjoint clique graph

Pivot algorithm: cluster random **pivot** nodes with their unclustered neighbors



Pivot is (basically) equivalent to GreedyMIS!



The random **pivot nodes** are a greedy maximal independent set.

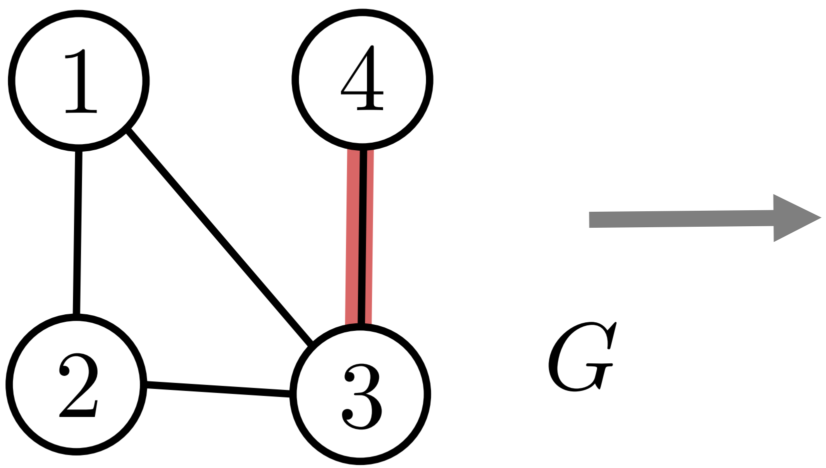
This has been observed in previous work on parallel MIS algorithms.

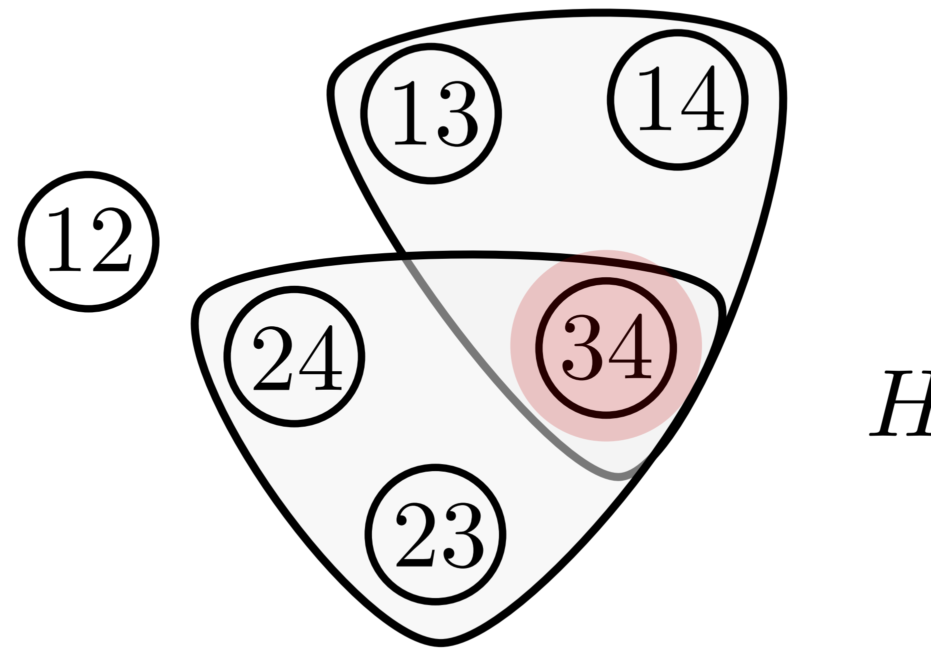
Fischer & Noever, TALG 2019

However, this is not the inspiration for the new Min-VC algorithm!

That is related to a more subtle relationship.

Pivot can be implicitly viewed as a 3-approximation for a 3-uniform hypergraph Vertex Cover algorithm





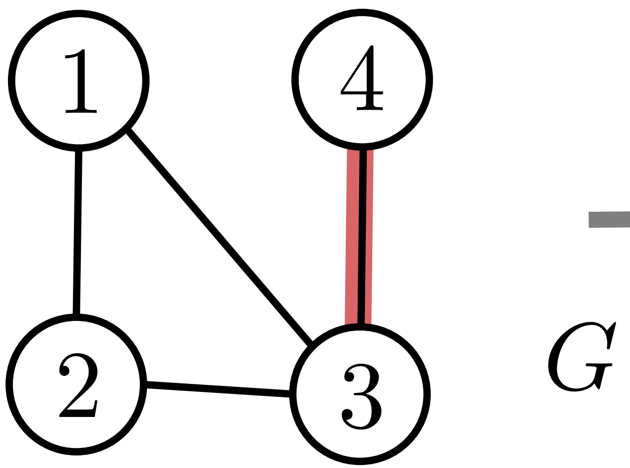
Correlation clustering in *G*...

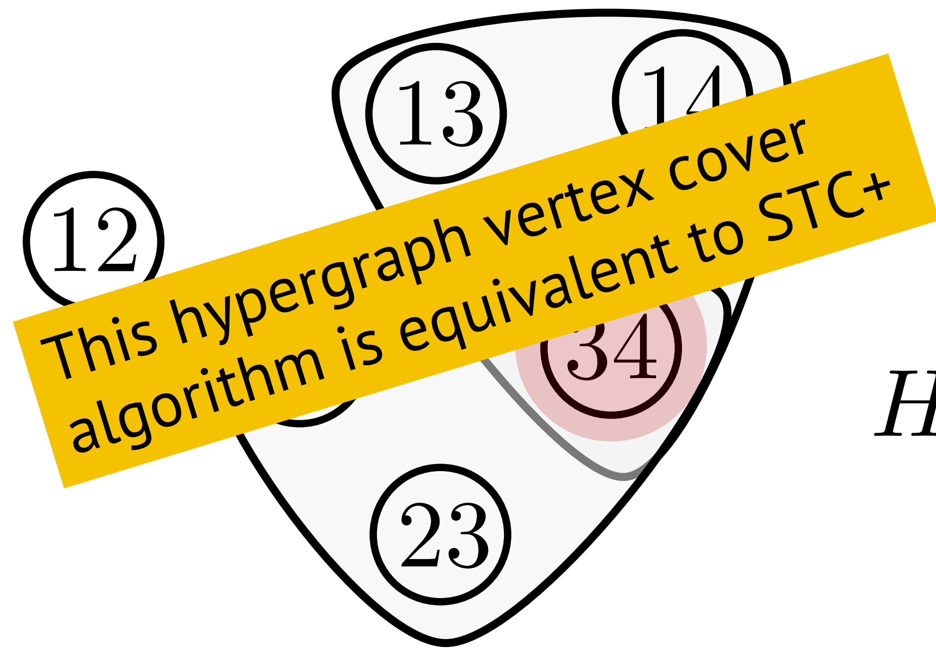
...can be lower-bounded by a Min-VC problem in a 3-uniform hypergraph *H*

Adding/deleting an edge in *G*...

....means covering a node in *H*

Pivot can be implicitly viewed as a 3-approximation for a 3-uniform hypergraph Vertex Cover algorithm





Correlation clustering in *G*...

...can be lower-bounded by a Min-VC problem in a 3-uniform hypergraph *H*

Adding/deleting an edge in *G*...

....means covering a node in *H*

Open questions

Can we further simplify/unify existing parallel algorithms for finding maximal independent sets, approximating Min-VC, and approximating correlation clustering?

Can this maximal independent set approach for Min-VC be extended to hypergraph vertex cover?

Can we exploit this relationship between Min-VC and correlation clustering to develop new combinatorial and parallel algorithms for weighted correlation clustering?

Growing a Random Maximal Independent Set Produces a 2-approximate Vertex Cover
Nate Veldt, https://arxiv.org/abs/2209.04673

Thanks!