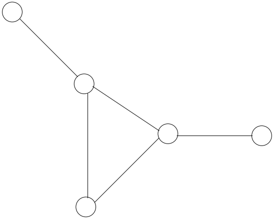
# Finding small stabilizer for unstable graphs

Adrian Bock[1], Karthik Chandrasekaran[2], Jochen Könemann[3], Britta Peis[4], and Laura Sanitá[3]

([1]Lausanne, [2]Boston, [3]Waterloo, [4]Aachen)

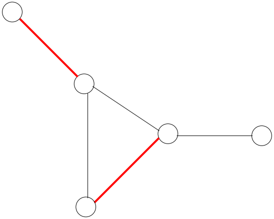
Let's recall some basics on matchings and vertex cover:

Given an undirected graph $G = (V, E)$,



Let's recall some basics on matchings and vertex cover:

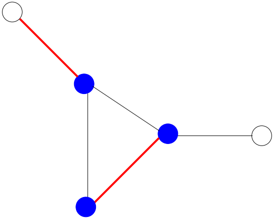
Given an undirected graph $G = (V, E)$,



- a matching is a set $M \subseteq E$ of non-adjacent edges,

Let's recall some basics on matchings and vertex cover:

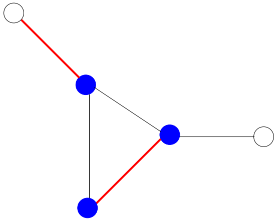
Given an undirected graph $G = (V, E)$,



- a matching is a set $M \subseteq E$ of non-adjacent edges,
- a vertex cover is a set of vertices $C \subseteq V$ such that each edge has at least one endpoint in $C$.

Let's recall some basics on matchings and vertex cover:

Given an undirected graph $G = (V, E)$,



- a matching is a set $M \subseteq E$ of non-adjacent edges,
- a vertex cover is a set of vertices $C \subseteq V$ such that each edge has at least one endpoint in $C$.
- Finding a maximum matching is "easy" whereas finding a minimum cover is "hard".

As usual, let

- $\nu(G) = \max\{|M| \mid M \text{ matching in } G\}$, and
- $\tau(G) = \min\{|C| \mid C \text{ vertex cover in } G\}$.

As usual, let

- $\nu(G) = \max\{|M| \mid M \text{ matching in } G\}$, and
- $\tau(G) = \min\{|C| \mid C \text{ vertex cover in } G\}$.

Consider the corresponding linear relaxations

- $\nu_f(G) = \max\{\sum_{e \in E} y_e \mid y(\delta(v)) \leq 1 \ \forall v \in V;\ y \in \mathbb{R}_+^E\}$,
- $\tau_f(G) = \min\{\sum_{v \in V} x_v \mid x_u + x_v \geq 1 \ \forall uv \in E;\ x \in \mathbb{R}_+^V\}$.

As usual, let

- $\nu(G) = \max\{|M| \mid M \text{ matching in } G\}$, and
- $\tau(G) = \min\{|C| \mid C \text{ vertex cover in } G\}$.

Consider the corresponding linear relaxations

- $\nu_f(G) = \max\{\sum_{e\in E} y_e \mid y(\delta(v)) \leq 1 \ \forall v \in V;\ y \in \mathbb{R}_+^E\}$,
- $\tau_f(G) = \min\{\sum_{v\in V} x_v \mid x_u + x_v \geq 1 \ \forall uv \in E;\ x \in \mathbb{R}_+^V\}$.

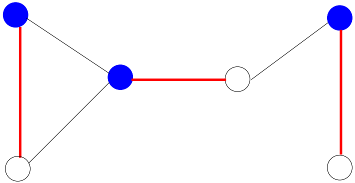
By duality theory:

$$\nu(G) \leq \nu_f(G) = \tau_f(G) \leq \tau(G).$$

In general: $\nu(G) \leq \nu_f(G) = \tau_f(G) \leq \tau(G)$.

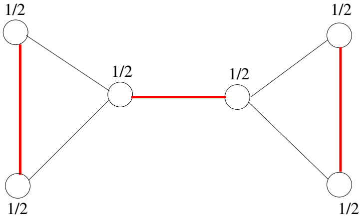
In general: $\nu(G) \leq \nu_f(G) = \tau_f(G) \leq \tau(G)$.

If $\nu(G) = \tau(G)$, graph $G$ is called König-Egerváry graph.



In general: $\nu(G) \leq \nu_f(G) = \tau_f(G) \leq \tau(G)$.

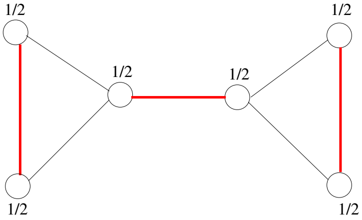
If $\nu(G) = \tau(G)$, graph $G$ is called König-Egerváry graph.




If $\nu(G) = \tau_f(G)$, graph $G$ is called stable.

In general: $\nu(G) \leq \nu_f(G) = \tau_f(G) \leq \tau(G)$.

If $\nu(G) = \tau(G)$, graph $G$ is called König-Egerváry graph.

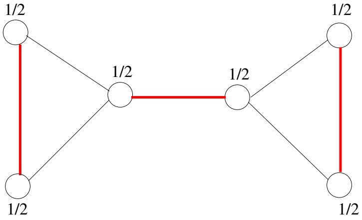



If $\nu(G) = \tau_f(G)$, graph $G$ is called stable.

Stable graphs have a rich history in game theory:

In general: $\nu(G) \leq \nu_f(G) = \tau_f(G) \leq \tau(G)$.

If $\nu(G) = \tau(G)$, graph $G$ is called König-Egerváry graph.

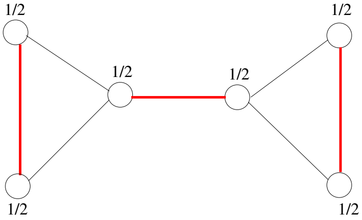



If $\nu(G) = \tau_f(G)$, graph $G$ is called stable.

Stable graphs have a rich history in game theory: In various graph-theoretic settings, stable graph are exactly those instances for which stable outcomes exist.

In general: $\nu(G) \leq \nu_f(G) = \tau_f(G) \leq \tau(G)$.

If $\nu(G) = \tau(G)$, graph $G$ is called König-Egerváry graph.




If $\nu(G) = \tau_f(G)$, graph $G$ is called stable.

Stable graphs have a rich history in game theory: In various graph-theoretic settings, stable graph are exactly those instances for which stable outcomes exist.

This talk: Given an unstable graph, how can we find a small stabilizer, i.e., a subset $F \subseteq E$ s.t. $G \setminus F$ is stable.

## Assignment games [Shapley & Shubik '71]:

The assignment game is a cooperative game where

Assignment games [Shapley & Shubik '71]:

The assignment game is a cooperative game where

- the players are represented by vertices of some given instance $G = (V, E)$, and

## Assignment games [Shapley & Shubik '71]:

The assignment game is a cooperative game where

- the players are represented by vertices of some given instance $G = (V, E)$, and
- the value of each coalition $S \subseteq V$ is $\nu(G[S])$, i.e., the maximum size of a matching achieved solely the players in $S$.

## Assignment games [Shapley & Shubik '71]:

The assignment game is a cooperative game where

- the players are represented by vertices of some given instance $G = (V, E)$, and
- the value of each coalition $S \subseteq V$ is $\nu(G[S])$, i.e., the maximum size of a matching achieved solely the players in $S$.

The core of the game consists of all "fair" allocations of $\nu(G)$ among the player set $V$, i.e.,

$$Core(G) = \{x \in \mathbb{R}^V \mid x(V) = \nu(G);\ x(S) \geq \nu(G[S])\ \forall S \subseteq V\}.$$

## Assignment games [Shapley & Shubik '71]:

The assignment game is a cooperative game where

- the players are represented by vertices of some given instance $G = (V, E)$, and
- the value of each coalition $S \subseteq V$ is $\nu(G[S])$, i.e., the maximum size of a matching achieved solely the players in $S$.

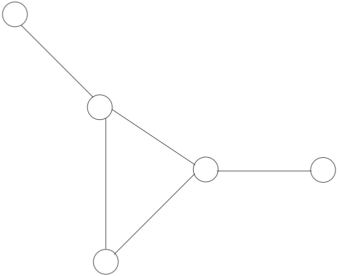
The core of the game consists of all "fair" allocations of $\nu(G)$ among the player set $V$, i.e.,

$$Core(G) = \{x \in \mathbb{R}^V \mid x(V) = \nu(G);\ x(S) \geq \nu(G[S])\ \forall S \subseteq V\}.$$

Note: Core$(G) \neq \emptyset \iff G$ is stable.

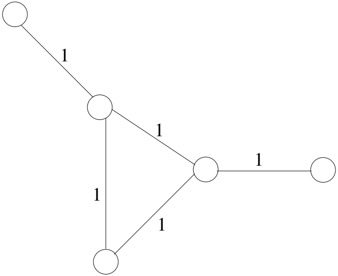
Network bargaining games [Kleinberg/Tardos'08]:

Players are nodes of an undirected graph $G = (V, E)$.



Network bargaining games [Kleinberg/Tardos'08]:

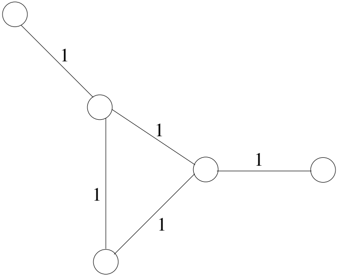
Players are nodes of an undirected graph $G = (V, E)$.




Edges indicate the possible unit-valued collaborations.

Network bargaining games [Kleinberg/Tardos'08]:

Players are nodes of an undirected graph $G = (V, E)$.

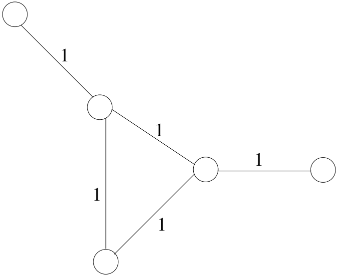



Edges indicate the possible unit-valued collaborations.

Each player can collaborate with **at most one** player.

Network bargaining games [Kleinberg/Tardos'08]:

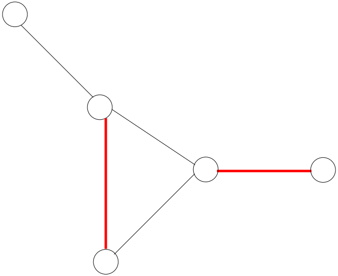
Players are nodes of an undirected graph $G = (V, E)$.




Edges indicate the possible unit-valued collaborations.

Each player can collaborate with **at most one** player. Players start negotiating ...

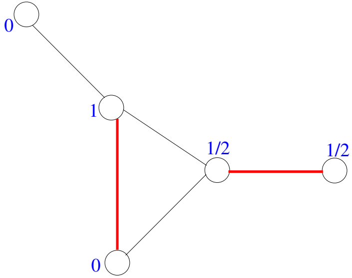
Network bargaining games [Kleinberg/Tardos'08]:
Players are nodes of an undirected graph $G = (V, E)$.



Edges indicate the possible unit-valued collaborations.
Each player can collaborate with **at most one** player. Players start negotiating ...

Solution/outcome: A tuple $(M, x)$ consisting of matching $M \subseteq E$

Network bargaining games [Kleinberg/Tardos'08]:
Players are nodes of an undirected graph $G = (V, E)$.

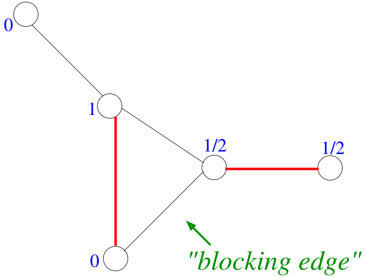



Edges indicate the possible unit-valued collaborations.
Each player can collaborate with **at most one** player. Players start negotiating ...

Solution/outcome: A tuple $(M, x)$ consisting of matching $M \subseteq E$ and an allocation $x \in \mathbb{R}^V$ of value $|M|$ among the endpoints of $M$.

## Network bargaining games [Kleinberg/Tardos'08]:

Players are nodes of an undirected graph $G = (V, E)$.

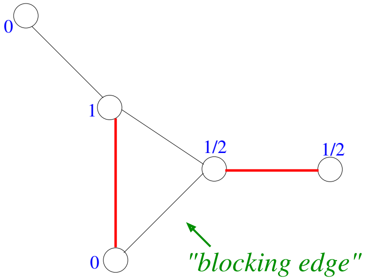



Edges indicate the possible unit-valued collaborations.
Each player can collaborate with **at most one** player. Players start negotiating . . .

Solution/outcome: A tuple $(M, x)$ consisting of matching $M \subseteq E$ and an allocation $x \in \mathbb{R}^V$ of value $|M|$ among the endpoints of $M$.

Network bargaining games [Kleinberg/Tardos'08]:

Players are nodes of an undirected graph $G = (V, E)$.



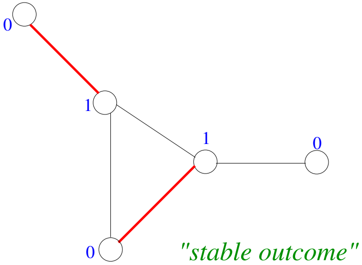

Edges indicate the possible unit-valued collaborations.

Each player can collaborate with **at most one** player. Players start negotiating ...

Solution/outcome: A tuple $(M, x)$ consisting of matching $M \subseteq E$ and an allocation $x \in \mathbb{R}^V$ of value $|M|$ among the endpoints of $M$.

Outcome $(M, x)$ is stable if $x_u + x_v \geq 1$ for each edge $\{u, v\} \in E$.

Network bargaining games [Kleinberg/Tardos'08]:
Players are nodes of an undirected graph $G = (V, E)$.



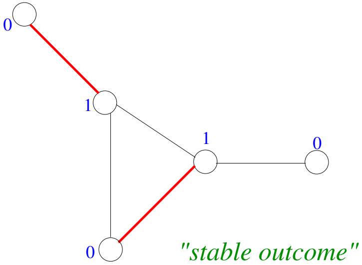

Edges indicate the possible unit-valued collaborations.
Each player can collaborate with **at most one** player. Players start negotiating ...

Solution/outcome: A tuple $(M, x)$ consisting of matching $M \subseteq E$ and an allocation $x \in \mathbb{R}^V$ of value $|M|$ among the endpoints of $M$.

Outcome $(M, x)$ is stable if $x_u + x_v \geq 1$ for each edge $\{u, v\} \in E$.

Network bargaining games [Kleinberg/Tardos'08]:

Players are nodes of an undirected graph $G = (V, E)$.




Edges indicate the possible unit-valued collaborations.

Each player can collaborate with **at most one** player. Players start negotiating ...

Solution/outcome: A tuple $(M, x)$ consisting of matching $M \subseteq E$ and an allocation $x \in \mathbb{R}^V$ of value $|M|$ among the endpoints of $M$.

Outcome $(M, x)$ is stable if $x_u + x_v \geq 1$ for each edge $\{u, v\} \in E$.

Note: $G$ admits a stable outcome $\iff \nu(G) = \tau_f(G)$.

Known: The following statements are equivalent:

- $\nu(G) = \tau_f(G)$, i.e., $G$ is stable;

Known: The following statements are equivalent:

- $\nu(G) = \tau_f(G)$, i.e., $G$ is stable;
- $\mathrm{Core}(G) \neq \emptyset$;

Known: The following statements are equivalent:

- $\nu(G) = \tau_f(G)$, i.e., $G$ is stable;
- $\mathrm{Core}(G) \neq \emptyset$;
- Graph $G = (V, E)$ admits a stable outcome;

Known: The following statements are equivalent:

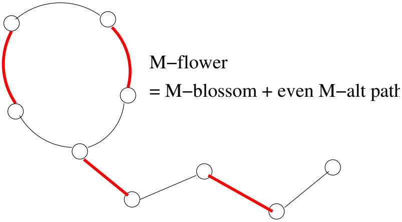
- $\nu(G) = \tau_f(G)$, i.e., $G$ is stable;
- $\text{Core}(G) \neq \emptyset$;
- Graph $G = (V, E)$ admits a stable outcome;
- $G$ admits a "balanced" outcome;

Known: The following statements are equivalent:

- $\nu(G) = \tau_f(G)$, i.e., $G$ is stable;
- $\mathrm{Core}(G) \neq \emptyset$;
- Graph $G = (V, E)$ admits a stable outcome;
- $G$ admits a "balanced" outcome;
- the set of inessential vertices forms an independent set;

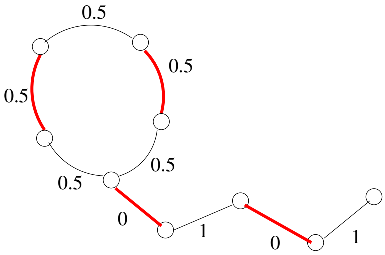
Known: The following statements are equivalent:

- $\nu(G) = \tau_f(G)$, i.e., $G$ is stable;
- $\mathrm{Core}(G) \neq \emptyset$;
- Graph $G = (V, E)$ admits a stable outcome;
- $G$ admits a "balanced" outcome;
- the set of inessential vertices forms an independent set;
- $G$ contains no $M$-flower for a maximum matching $M$;



Known: The following statements are equivalent:

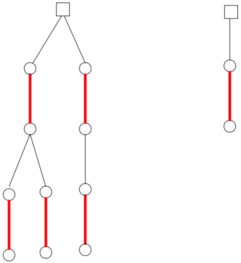
- $\nu(G) = \tau_f(G)$, i.e., $G$ is stable;
- $\mathrm{Core}(G) \neq \emptyset$;
- Graph $G = (V, E)$ admits a stable outcome;
- $G$ admits a "balanced" outcome;
- the set of inessential vertices forms an independent set;
- $G$ contains no $M$-flower for a maximum matching $M$;



Edmonds' algorithm either computes a stable solution $(M, x)$,

Edmonds' algorithm either computes a stable solution $(M, x)$, or gives a certificate ($M$-flower) proving that none exists:

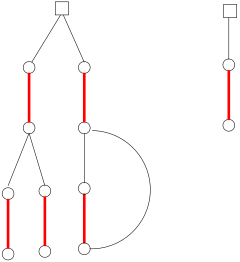
Edmonds' algorithm either computes a stable solution $(M, x)$, or gives a certificate ($M$-flower) proving that none exists:



Algorithm:

1. compute a max matching $M$;
2. consider the $M$-alternating forest;

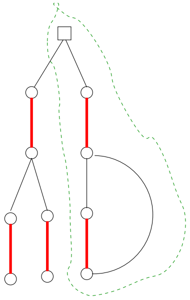
Edmonds' algorithm either computes a stable solution $(M, x)$, or gives a certificate ($M$-flower) proving that none exists:



Algorithm:

1. compute a max matching $M$;
2. consider the $M$-alternating forest;
3. either there is an $M$-flower

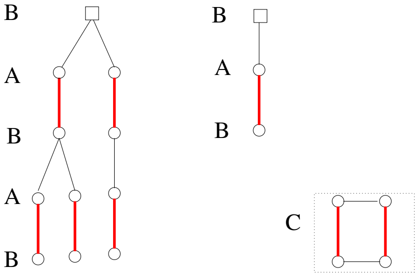
Edmonds' algorithm either computes a stable solution $(M, x)$, or gives a certificate ($M$-flower) proving that none exists:



Algorithm:

1. compute a max matching $M$;
2. consider the $M$-alternating forest;
3. either there is an $M$-flower proving that $G$ is not stable, or

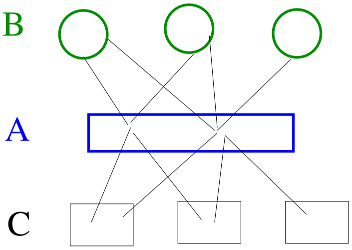
Edmonds' algorithm either computes a stable solution $(M, x)$, or gives a certificate ($M$-flower) proving that none exists:




Algorithm:

1. compute a max matching $M$;
2. consider the $M$-alternating forest;
3. either there is an $M$-flower proving that $G$ is not stable, or
4. the Gallai-Edmonds-decomposition $V = B \cup A \cup C$

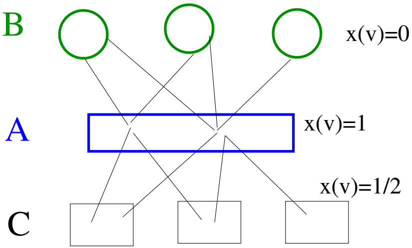
Edmonds' algorithm either computes a stable solution $(M, x)$, or gives a certificate ($M$-flower) proving that none exists:




Algorithm:

1. compute a max matching $M$;
2. consider the $M$-alternating forest;
3. either there is an $M$-flower proving that $G$ is not stable, or
4. the Gallai-Edmonds-decomposition $V = B \cup A \cup C$

Edmonds' algorithm either computes a stable solution $(M, x)$, or gives a certificate ($M$-flower) proving that none exists:




Algorithm:

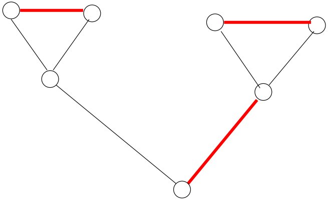
1. compute a max matching $M$;
2. consider the $M$-alternating forest;
3. either there is an $M$-flower proving that $G$ is not stable, or
4. the Gallai-Edmonds-decomposition $V = B \cup A \cup C$defines a cover $x$ of size $|M|$;

Our focus: graphs that are not stable.

Our focus: graphs that are not stable. Can we stabilize unstable graphs by altering the graph in as few places as possible?

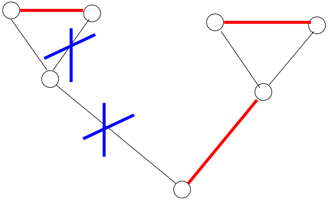
Our focus: graphs that are not stable. Can we stabilize unstable graphs by altering the graph in as few places as possible?

Suppose $G$ admits no matching and fractional cover of equal size.



Our focus: graphs that are not stable. Can we stabilize unstable graphs by altering the graph in as few places as possible?

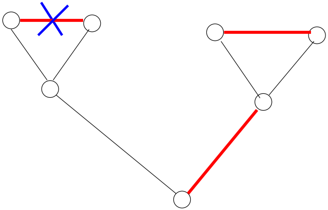
Suppose $G$ admits no matching and fractional cover of equal size.



Call $F \subseteq E$ a stabilizer if $G \setminus F = G[E \setminus F]$ is stable.

Our focus: graphs that are not stable. Can we stabilize unstable graphs by altering the graph in as few places as possible?

Suppose $G$ admits no matching and fractional cover of equal size.

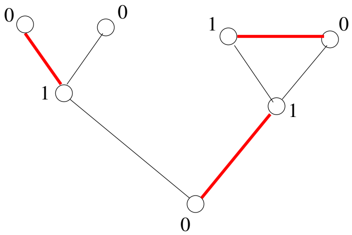


Call $F \subseteq E$ a stabilizer if $G \setminus F = G[E \setminus F]$ is stable.

Min Stabilizer problem: find a stabilizer $F$ of minimum cardinality

Our focus: graphs that are not stable. Can we stabilize unstable graphs by altering the graph in as few places as possible?

Suppose $G$ admits no matching and fractional cover of equal size.

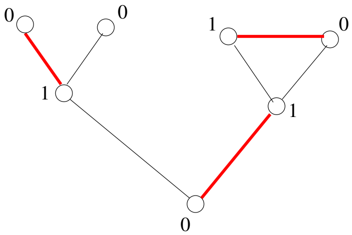



Call $F \subseteq E$ a stabilizer if $G \setminus F = G[E \setminus F]$ is stable.

Min Stabilizer problem: find a stabilizer $F$ of minimum cardinality

Our focus: graphs that are not stable. Can we stabilize unstable graphs by altering the graph in as few places as possible?

Suppose $G$ admits no matching and fractional cover of equal size.

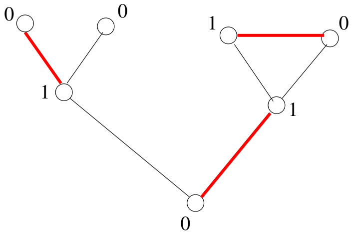



Call $F \subseteq E$ a stabilizer if $G \setminus F = G[E \setminus F]$ is stable.

Min Stabilizer problem: find a stabilizer $F$ of minimum cardinality If possible such that $\nu(G) = \nu(G \setminus F)$.

Our focus: graphs that are not stable. Can we stabilize unstable graphs by altering the graph in as few places as possible?

Suppose $G$ admits no matching and fractional cover of equal size.




Call $F \subseteq E$ a stabilizer if $G \setminus F = G[E \setminus F]$ is stable.

Min Stabilizer problem: find a stabilizer $F$ of minimum cardinality If possible such that $\nu(G) = \nu(G \setminus F)$.

### Theorem (Key)

*If $F$ is a minimum stabilizer, then $\nu(G \setminus F) = \nu(G)$.*

Thus, there is a minimum stabilizer $F$ so that at least one maximum matching $M$ survives.

Thus, there is a minimum stabilizer $F$ so that at least one maximum matching $M$ survives.
We use this structural insight to show:

### Theorem

*The* MINIMUM STABILIZER PROBLEM *is NP-hard (vertex cover).*

Thus, there is a minimum stabilizer $F$ so that at least one maximum matching $M$ survives.
We use this structural insight to show:

### Theorem

*The* MINIMUM STABILIZER PROBLEM *is NP-hard (vertex cover). Even if the matching $M$ is given.*

Thus, there is a minimum stabilizer $F$ so that at least one maximum matching $M$ survives.
We use this structural insight to show:

**Theorem**

*The* MINIMUM STABILIZER PROBLEM *is NP-hard (vertex cover). Even if the matching $M$ is given.*

**Theorem**

*There is a 2-approximation for the stabilizer problem with given $M$.*

Thus, there is a minimum stabilizer $F$ so that at least one maximum matching $M$ survives.
We use this structural insight to show:

**Theorem**

*The* MINIMUM STABILIZER PROBLEM *is NP-hard (vertex cover). Even if the matching $M$ is given.*

**Theorem**

*There is a 2-approximation for the stabilizer problem with given $M$.*

Open: Constant factor approximation for the general MIN STABILIZER PROBLEM?

Thus, there is a minimum stabilizer $F$ so that at least one maximum matching $M$ survives.
We use this structural insight to show:

**Theorem**

*The* MINIMUM STABILIZER PROBLEM *is NP-hard (vertex cover). Even if the matching $M$ is given.*

**Theorem**

*There is a 2-approximation for the stabilizer problem with given $M$.*

Open: Constant factor approximation for the general MIN STABILIZER PROBLEM?

**Theorem**

*$4\omega$-approximation, where $\omega$ is the sparsity of the graph.*

Thus, there is a minimum stabilizer $F$ so that at least one maximum matching $M$ survives.
We use this structural insight to show:

**Theorem**

*The* MINIMUM STABILIZER PROBLEM *is NP-hard (vertex cover). Even if the matching $M$ is given.*

**Theorem**

*There is a 2-approximation for the stabilizer problem with given $M$.*

Open: Constant factor approximation for the general MIN STABILIZER PROBLEM?

**Theorem**

*$4\omega$-approximation, where $\omega$ is the sparsity of the graph.*

**Theorem**

*2-approximation for regular graphs.*

### Theorem (Key)

*If $F$ is a minimum stabilizer, then $\nu(G \setminus F) = \nu(G)$.*

## Theorem (Key)

*If $F$ is a minimum stabilizer, then $\nu(G \setminus F) = \nu(G)$.*

## Proof.

Let $F$ be a minimum stabilizer. Choose a max matching $M$ with $|F \cap M|$ minimal.

### Theorem (Key)

*If $F$ is a minimum stabilizer, then $\nu(G \setminus F) = \nu(G)$.*

### Proof.

Let $F$ be a minimum stabilizer. Choose a max matching $M$ with $|F \cap M|$ minimal. Suppose $F \cap M \neq \emptyset$.

**Theorem (Key)**

*If $F$ is a minimum stabilizer, then $\nu(G \setminus F) = \nu(G)$.*

**Proof.**

Let $F$ be a minimum stabilizer. Choose a max matching $M$ with $|F \cap M|$ minimal. Suppose $F \cap M \neq \emptyset$. Delete edges in $F \setminus M$.

## Theorem (Key)

*If $F$ is a minimum stabilizer, then $\nu(G \setminus F) = \nu(G)$.*

## Proof.

Let $F$ be a minimum stabilizer. Choose a max matching $M$ with $|F \cap M|$ minimal. Suppose $F \cap M \neq \emptyset$. Delete edges in $F \setminus M$.

The resulting graph $G'$ is not stable.

## Theorem (Key)

*If $F$ is a minimum stabilizer, then $\nu(G \setminus F) = \nu(G)$.*

## Proof.

Let $F$ be a minimum stabilizer. Choose a max matching $M$ with $|F \cap M|$ minimal. Suppose $F \cap M \neq \emptyset$. Delete edges in $F \setminus M$.

The resulting graph $G'$ is not stable. Moreover, $M$ is max matching in $G'$. Thus, there exists some $M$-flower in $G'$,

## Theorem (Key)

*If $F$ is a minimum stabilizer, then $\nu(G \setminus F) = \nu(G)$.*

## Proof.

Let $F$ be a minimum stabilizer. Choose a max matching $M$ with $|F \cap M|$ minimal. Suppose $F \cap M \neq \emptyset$. Delete edges in $F \setminus M$.

The resulting graph $G'$ is not stable. Moreover, $M$ is max matching in $G'$. Thus, there exists some $M$-flower in $G'$, not hit by $F$.

## Theorem (Key)

*If $F$ is a minimum stabilizer, then $\nu(G \setminus F) = \nu(G)$.*

## Proof.

Let $F$ be a minimum stabilizer. Choose a max matching $M$ with $|F \cap M|$ minimal. Suppose $F \cap M \neq \emptyset$. Delete edges in $F \setminus M$.

The resulting graph $G'$ is not stable. Moreover, $M$ is max matching in $G'$. Thus, there exists some $M$-flower in $G'$, not hit by $F$.

Since $G \setminus F$ is stable, $M \setminus F$ isn't a max matching in $G \setminus F$.

## Theorem (Key)

*If $F$ is a minimum stabilizer, then $\nu(G \setminus F) = \nu(G)$.*

## Proof.

Let $F$ be a minimum stabilizer. Choose a max matching $M$ with $|F \cap M|$ minimal. Suppose $F \cap M \neq \emptyset$. Delete edges in $F \setminus M$.

The resulting graph $G'$ is not stable. Moreover, $M$ is max matching in $G'$. Thus, there exists some $M$-flower in $G'$, not hit by $F$.

Since $G \setminus F$ is stable, $M \setminus F$ isn't a max matching in $G \setminus F$.Thus, there exists some $M \setminus F$-augmenting path $P$ in $G \setminus F$.

### Theorem (Key)

*If $F$ is a minimum stabilizer, then $\nu(G \setminus F) = \nu(G)$.*

### Proof.

Let $F$ be a minimum stabilizer. Choose a max matching $M$ with $|F \cap M|$ minimal. Suppose $F \cap M \neq \emptyset$. Delete edges in $F \setminus M$.

The resulting graph $G'$ is not stable. Moreover, $M$ is max matching in $G'$. Thus, there exists some $M$-flower in $G'$, not hit by $F$.

Since $G \setminus F$ is stable, $M \setminus F$ isn't a max matching in $G \setminus F$.Thus, there exists some $M \setminus F$-augmenting path $P$ in $G \setminus F$.

However, as $M$ is max matching in $G$, one end of the path must be adjacent to $f \in M \cap F$.

## Theorem (Key)

*If $F$ is a minimum stabilizer, then $\nu(G \setminus F) = \nu(G)$.*

## Proof.

Let $F$ be a minimum stabilizer. Choose a max matching $M$ with $|F \cap M|$ minimal. Suppose $F \cap M \neq \emptyset$. Delete edges in $F \setminus M$.

The resulting graph $G'$ is not stable. Moreover, $M$ is max matching in $G'$. Thus, there exists some $M$-flower in $G'$, not hit by $F$.

Since $G \setminus F$ is stable, $M \setminus F$ isn't a max matching in $G \setminus F$.Thus, there exists some $M \setminus F$-augmenting path $P$ in $G \setminus F$.

However, as $M$ is max matching in $G$, one end of the path must be adjacent to $f \in M \cap F$. It follows that $\hat{M} = M\Delta(P + f)$ is max matching in $G$ with $|\hat{M} \cap F| < |M \cap F|$. Contradiction! $\square$

Theorem

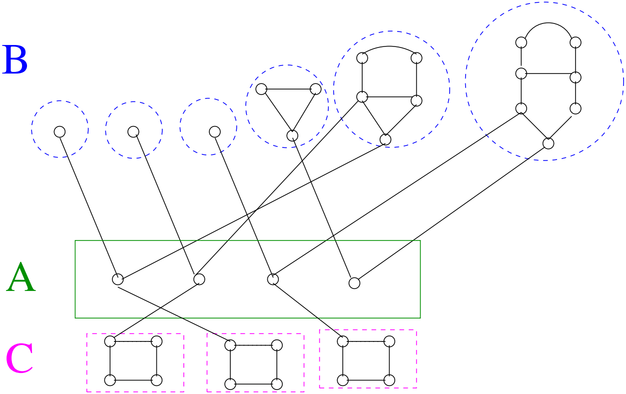
*F stabilizer of G* $\implies |F| \geq 2(\nu_f(G) - \nu(G))$

Theorem

$F$ *stabilizer of* $G \Longrightarrow |F| \geq 2(\nu_f(G) - \nu(G))$

Proof:

Consider the Gallai–Edmonds' Decomposition V=B+A+C

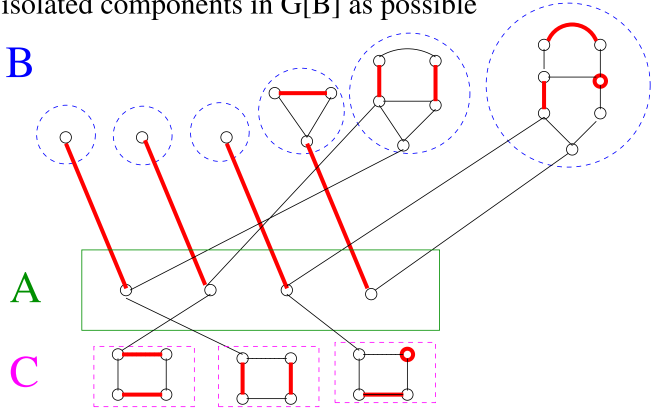


Theorem

*F stabilizer of G* $\implies |F| \geq 2(\nu_f(G) - \nu(G))$

Proof:

Choose max matching M linking as much isolated components in G[B] as possible

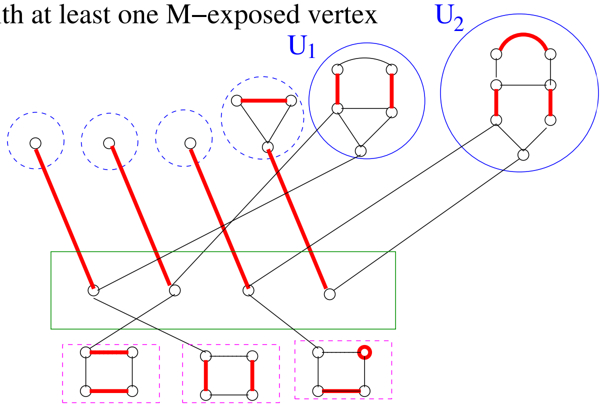


## Theorem

*F stabilizer of G* $\implies |F| \geq 2(\nu_f(G) - \nu(G))$

## Proof:

Let $U_1$, ..., $U_k$ denote the non–trivial components in G[B] with at least one M–exposed vertex

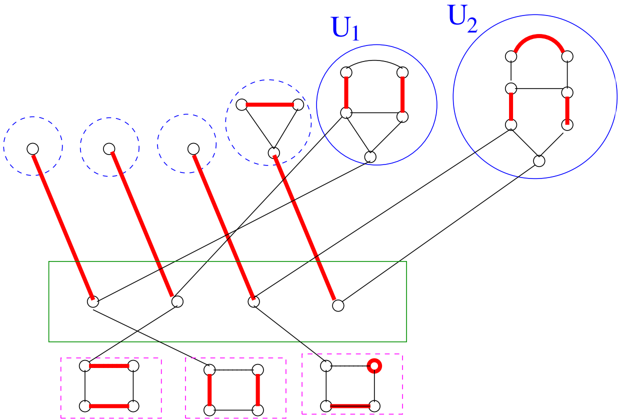


## Theorem

*F stabilizer of G* $\Longrightarrow |F| \geq 2(\nu_f(G) - \nu(G))$

## Proof:

Each $U_i$ has at least one vertex $v_i$ essential in G\F

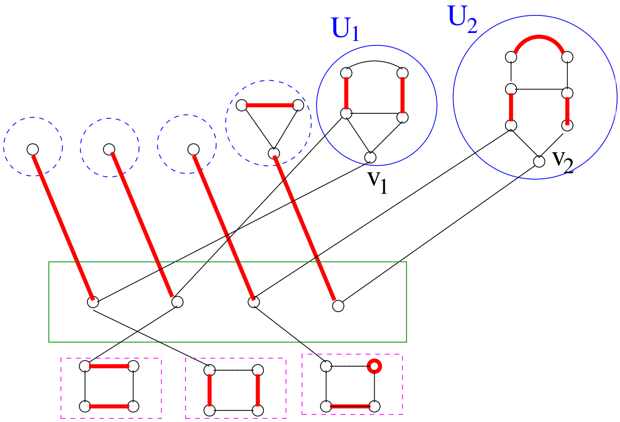


**Theorem**

*$F$ stabilizer of $G \implies |F| \geq 2(\nu_f(G) - \nu(G))$*

**Proof:**

Can assume that $v_i$ is M–exposed

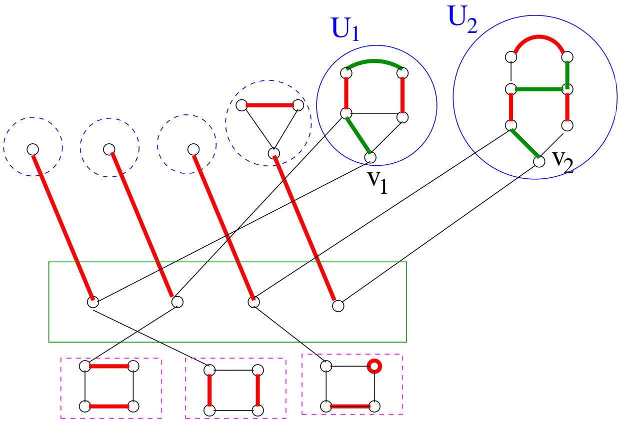


## Theorem

*F stabilizer of G* $\implies |F| \geq 2(\nu_f(G) - \nu(G))$

## Proof:

Choose max matching N in G\F

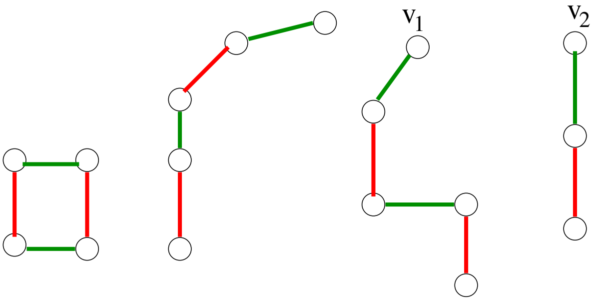


Theorem

$F$ *stabilizer of* $G \implies |F| \geq 2(\nu_f(G) - \nu(G))$

Proof:

N△M is disjoint union of even cycles and even paths

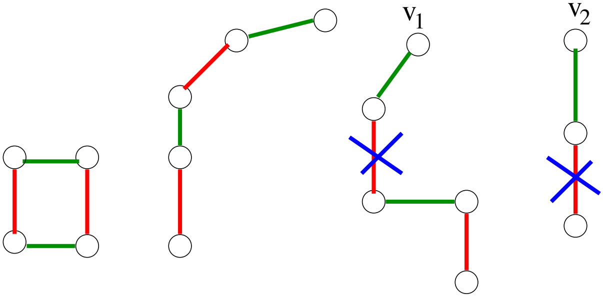


## Theorem

*F stabilizer of G* $\Longrightarrow |F| \geq 2(\nu_f(G) - \nu(G))$

## Proof:

Note: each of the k paths starting at $v_i$ has at least one edge in F.



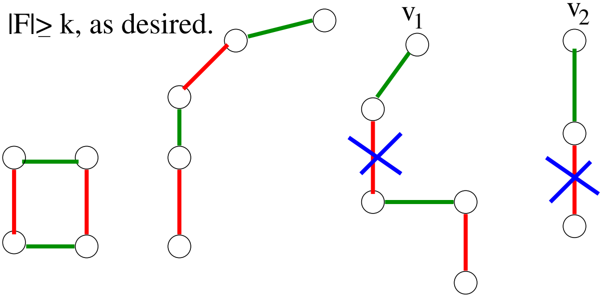
## Theorem

*F stabilizer of G* $\implies |F| \geq 2(\nu_f(G) - \nu(G))$

## Proof:

Note: each of the k paths starting at v has at least one edge in F.

Thus, |F|≥ k, as desired.



Theorem

*There is a poly-time algorithm that finds a stabilizer $F$ with $\nu(G) = \nu(G \setminus F)$ and $|F| \leq 4\omega \cdot OPT$,*

Theorem

*There is a poly-time algorithm that finds a stabilizer* $F$ *with* $\nu(G) = \nu(G \setminus F)$ *and* $|F| \leq 4\omega \cdot OPT$, *where* $\omega$ *is the sparsity of* $G$.

**Theorem**

*There is a poly-time algorithm that finds a stabilizer $F$ with $\nu(G) = \nu(G \setminus F)$ and $|F| \leq 4\omega \cdot OPT$, where $\omega$ is the sparsity of $G$.*

**Lemma**

*If $\nu(G) < \nu_f(G)$, can find in poly-time a set $L \subseteq E$ with $|L| \leq 4\omega$, $\nu(G) = \nu(G \setminus L)$, and $\nu_f(G \setminus L) = \nu_f(G) - \frac{1}{2}$.*

## Theorem

*There is a poly-time algorithm that finds a stabilizer $F$ with $\nu(G) = \nu(G \setminus F)$ and $|F| \leq 4\omega \cdot OPT$, where $\omega$ is the sparsity of $G$.*

## Lemma

*If $\nu(G) < \nu_f(G)$, can find in poly-time a set $L \subseteq E$ with $|L| \leq 4\omega$, $\nu(G) = \nu(G \setminus L)$, and $\nu_f(G \setminus L) = \nu_f(G) - \frac{1}{2}$.*

## Algorithm:

- Initialize $F \leftarrow \emptyset$;
- WHILE $G \setminus F$ is not stable:
  - apply Lemma to find $L$;
  - $F \leftarrow F \cup L$;

## Theorem

*There is a poly-time algorithm that finds a stabilizer $F$ with $\nu(G) = \nu(G \setminus F)$ and $|F| \leq 4\omega \cdot OPT$, where $\omega$ is the sparsity of $G$.*

## Lemma

*If $\nu(G) < \nu_f(G)$, can find in poly-time a set $L \subseteq E$ with $|L| \leq 4\omega$, $\nu(G) = \nu(G \setminus L)$, and $\nu_f(G \setminus L) = \nu_f(G) - \frac{1}{2}$.*

Algorithm:

- Initialize $F \leftarrow \emptyset$;
- WHILE $G \setminus F$ is not stable:
  - apply Lemma to find $L$;
  - $F \leftarrow F \cup L$;

Note: In each iteration, at most $4\omega$ edges are added to $F$;
At most $2(\nu_f(G) - \nu(G)) \leq$OPT iterations.

## Lemma

*If* $\nu(G) < \nu_f(G)$*, the following algorithm returns a set* $L \subseteq E$ *with* $|L| \leq 4\omega$, $\nu(G) = \nu(G \setminus L)$*, and* $\nu_f(G \setminus L) = \nu_f(G) - \frac{1}{2}$.

**Lemma**

*If $\nu(G) < \nu_f(G)$, the following algorithm returns a set $L \subseteq E$ with $|L| \leq 4\omega$, $\nu(G) = \nu(G \setminus L)$, and $\nu_f(G \setminus L) = \nu_f(G) - \frac{1}{2}$.*

**Algorithm:**

1. Compute optimal (half-integral) fractional matching $\hat{y}$ with least number of odd cycles $C_1, \ldots, C_m$ in support.

### Lemma

*If $\nu(G) < \nu_f(G)$, the following algorithm returns a set $L \subseteq E$ with $|L| \leq 4\omega$, $\nu(G) = \nu(G \setminus L)$, and $\nu_f(G \setminus L) = \nu_f(G) - \frac{1}{2}$.*

### Algorithm:

1. Compute optimal (half-integral) fractional matching $\hat{y}$ with least number of odd cycles $C_1, \ldots, C_m$ in support.
2. Compute optimal (half-integral) fractional vertex cover $\hat{x}$.

### Lemma

*If $\nu(G) < \nu_f(G)$, the following algorithm returns a set $L \subseteq E$ with $|L| \leq 4\omega$, $\nu(G) = \nu(G \setminus L)$, and $\nu_f(G \setminus L) = \nu_f(G) - \frac{1}{2}$.*

### Algorithm:

1. Compute optimal (half-integral) fractional matching $\hat{y}$ with least number of odd cycles $C_1, \ldots, C_m$ in support.
2. Compute optimal (half-integral) fractional vertex cover $\hat{x}$.
3. Construct max matching $M$ in support of $\hat{y}$.

## Lemma

*If $\nu(G) < \nu_f(G)$, the following algorithm returns a set $L \subseteq E$ with $|L| \leq 4\omega$, $\nu(G) = \nu(G \setminus L)$, and $\nu_f(G \setminus L) = \nu_f(G) - \frac{1}{2}$.*

## Algorithm:

1. Compute optimal (half-integral) fractional matching $\hat{y}$ with least number of odd cycles $C_1, \ldots, C_m$ in support.
2. Compute optimal (half-integral) fractional vertex cover $\hat{x}$.
3. Construct max matching $M$ in support of $\hat{y}$.
4. Mark all vertices in $H := C_1$;

## Lemma

*If $\nu(G) < \nu_f(G)$, the following algorithm returns a set $L \subseteq E$ with $|L| \leq 4\omega$, $\nu(G) = \nu(G \setminus L)$, and $\nu_f(G \setminus L) = \nu_f(G) - \frac{1}{2}$.*

## Algorithm:

1. Compute optimal (half-integral) fractional matching $\hat{y}$ with least number of odd cycles $C_1, \ldots, C_m$ in support.
2. Compute optimal (half-integral) fractional vertex cover $\hat{x}$.
3. Construct max matching $M$ in support of $\hat{y}$.
4. Mark all vertices in $H := C_1$;
5. If exists marked $u \in H$ with $|L_u := \{uv \in E \mid \hat{x}_v = \frac{1}{2}\}| \leq 4\omega$, STOP; Return $L = L_u$;

**Lemma**

*If $\nu(G) < \nu_f(G)$, the following algorithm returns a set $L \subseteq E$ with $|L| \leq 4\omega$, $\nu(G) = \nu(G \setminus L)$, and $\nu_f(G \setminus L) = \nu_f(G) - \frac{1}{2}$.*

**Algorithm:**

1. Compute optimal (half-integral) fractional matching $\hat{y}$ with least number of odd cycles $C_1, \ldots, C_m$ in support.
2. Compute optimal (half-integral) fractional vertex cover $\hat{x}$.
3. Construct max matching $M$ in support of $\hat{y}$.
4. Mark all vertices in $H := C_1$;
5. If exists marked $u \in H$ with $|L_u := \{uv \in E \mid \hat{x}_v = \frac{1}{2}\}| \leq 4\omega$, STOP; Return $L = L_u$;
6. Else, choose marked vertex in $H$ being adjacent to some $w \notin H$ with $\hat{x}_w = \frac{1}{2}$;

## Lemma

*If $\nu(G) < \nu_f(G)$, the following algorithm returns a set $L \subseteq E$ with $|L| \leq 4\omega$, $\nu(G) = \nu(G \setminus L)$, and $\nu_f(G \setminus L) = \nu_f(G) - \frac{1}{2}$.*

## Algorithm:

1. Compute optimal (half-integral) fractional matching $\hat{y}$ with least number of odd cycles $C_1, \ldots, C_m$ in support.
2. Compute optimal (half-integral) fractional vertex cover $\hat{x}$.
3. Construct max matching $M$ in support of $\hat{y}$.
4. Mark all vertices in $H := C_1$;
5. If exists marked $u \in H$ with $|L_u := \{uv \in E \mid \hat{x}_v = \frac{1}{2}\}| \leq 4\omega$, STOP; Return $L = L_u$;
6. Else, choose marked vertex in $H$ being adjacent to some $w \notin H$ with $\hat{x}_w = \frac{1}{2}$;
7. Add $w$ and its matching neighbour $z$ to $H$; Mark $z$ and iterate;

### Lemma

*If $\nu(G) < \nu_f(G)$, the following algorithm returns a set $L \subseteq E$ with $|L| \leq 4\omega$, $\nu(G) = \nu(G \setminus L)$, and $\nu_f(G \setminus L) = \nu_f(G) - \frac{1}{2}$.*

### Algorithm:

1. Compute optimal (half-integral) fractional matching $\hat{y}$ with least number of odd cycles $C_1, \ldots, C_m$ in support.
2. Compute optimal (half-integral) fractional vertex cover $\hat{x}$.
3. Construct max matching $M$ in support of $\hat{y}$.
4. Mark all vertices in $H := C_1$;
5. If exists marked $u \in H$ with $|L_u := \{uv \in E \mid \hat{x}_v = \frac{1}{2}\}| \leq 4\omega$, STOP; Return $L = L_u$;
6. Else, choose marked vertex in $H$ being adjacent to some $w \notin H$ with $\hat{x}_w = \frac{1}{2}$;
7. Add $w$ and its matching neighbour $z$ to $H$; Mark $z$ and iterate;

Invariant: Isolating a marked vertex won't decrease $\nu(G)$,

## Lemma

*If $\nu(G) < \nu_f(G)$, the following algorithm returns a set $L \subseteq E$ with $|L| \leq 4\omega$, $\nu(G) = \nu(G \setminus L)$, and $\nu_f(G \setminus L) = \nu_f(G) - \frac{1}{2}$.*

## Algorithm:

1. Compute optimal (half-integral) fractional matching $\hat{y}$ with least number of odd cycles $C_1, \ldots, C_m$ in support.
2. Compute optimal (half-integral) fractional vertex cover $\hat{x}$.
3. Construct max matching $M$ in support of $\hat{y}$.
4. Mark all vertices in $H := C_1$;
5. If exists marked $u \in H$ with $|L_u := \{uv \in E \mid \hat{x}_v = \frac{1}{2}\}| \leq 4\omega$, STOP; Return $L = L_u$;
6. Else, choose marked vertex in $H$ being adjacent to some $w \notin H$ with $\hat{x}_w = \frac{1}{2}$;
7. Add $w$ and its matching neighbour $z$ to $H$; Mark $z$ and iterate;

Invariant: Isolating a marked vertex won't decrease $\nu(G)$,but $\tau_f(G)$ will decrease by $\frac{1}{2}$.

# Summary

Stable graphs are nicely characterized.

# Summary

Stable graphs are nicely characterized.

The problem to find a stabilizer of minimum cardinality is at least as hard as vertex cover.

# Summary

Stable graphs are nicely characterized.

The problem to find a stabilizer of minimum cardinality is at least as hard as vertex cover.

We have (so far)

- 2-approximation for regular graphs;
- 2-approximation for the min stabilizer problem w.r.t. a fixed matching $M$ (also vertex-cover-hard);
- $4\omega$-approximation;
- $\nu(G) = \nu(G \setminus F)$ for any minimum stabilizer $F$;

# Summary

Stable graphs are nicely characterized.

The problem to find a stabilizer of minimum cardinality is at least as hard as vertex cover.

We have (so far)

- 2-approximation for regular graphs;
- 2-approximation for the min stabilizer problem w.r.t. a fixed matching $M$ (also vertex-cover-hard);
- $4\omega$-approximation;
- $\nu(G) = \nu(G \setminus F)$ for any minimum stabilizer $F$;

We would love to find a constant-factor approximation!!!

# Summary

Stable graphs are nicely characterized.

The problem to find a stabilizer of minimum cardinality is at least as hard as vertex cover.

We have (so far)

- 2-approximation for regular graphs;
- 2-approximation for the min stabilizer problem w.r.t. a fixed matching $M$ (also vertex-cover-hard);
- $4\omega$-approximation;
- $\nu(G) = \nu(G \setminus F)$ for any minimum stabilizer $F$;

We would love to find a constant-factor approximation!!!

## Thank you!