Matchings, Covers, and Network Games

Laura Sanità
Combinatorics and Optimization Department
University of Waterloo

29th Cumberland Conference on Combinatorics, Graph Theory and Computing
Matching and Stable Graphs

- A matching of a graph $G = (V, E)$ is a subset $M \subseteq E$ such that each $v \in V$ is incident into at most one edge of $M$. 

A vertex $v \in V$ is called inessential if there exists a matching in $G$ of maximum cardinality that exposes $v$. 


Matching and Stable Graphs

- A matching of a graph $G = (V, E)$ is a subset $M \subseteq E$ such that each $v \in V$ is incident into at most one edge of $M$. 

- A vertex $v \in V$ is called inessential if there exists a matching in $G$ of maximum cardinality that exposes $v$. 

![Graph Diagram]
Matching and Stable Graphs

- A matching of a graph $G = (V, E)$ is a subset $M \subseteq E$ such that each $v \in V$ is incident into at most one edge of $M$.

- A vertex $v \in V$ is called inessential if there exists a matching in $G$ of maximum cardinality that exposes $v$. 

Matching and Stable Graphs

• A matching of a graph $G = (V, E)$ is a subset $M \subseteq E$ such that each $v \in V$ is incident into at most one edge of $M$.

• A vertex $v \in V$ is called inessential if there exists a matching in $G$ of maximum cardinality that exposes $v$. 

Matching and Stable Graphs

• A matching of a graph \( G = (V, E) \) is a subset \( M \subseteq E \) such that each \( v \in V \) is incident into at most one edge of \( M \).

• A vertex \( v \in V \) is called essential if there is no matching in \( G \) of maximum cardinality that exposes \( v \).
Matching and Stable Graphs

- $G$ is said to be stable if the set of its inessential vertices forms a stable set (i.e., are pairwise not adjacent).
Matching and Stable Graphs

- $G$ is said to be **stable** if the set of its inessential vertices forms a stable set (i.e., are pairwise not adjacent).

- Unstable graph
Matching and Stable Graphs

- $G$ is said to be stable if the set of its inessential vertices forms a stable set (i.e., are pairwise not adjacent).

- Unstable graph
Matching and Stable Graphs

- $G$ is said to be *stable* if the set of its inessential vertices forms a stable set (i.e., are pairwise not adjacent).
Matching and Stable Graphs

- $G$ is said to be **stable** if the set of its inessential vertices forms a stable set (i.e., are pairwise not adjacent).

- Stable graph
Matching and Stable Graphs

- $G$ is said to be stable if the set of its inessential vertices forms a stable set (i.e., are pairwise not adjacent).

- Stable graph
Matching and Stable Graphs

- $G$ is said to be stable if the set of its inessential vertices forms a stable set (i.e., are pairwise not adjacent).

- Stable graph → Why are these graphs interesting?
Matching and Network games

- Several interesting game theory problems are defined on networks:
  - Stable graphs play a crucial role in some Network Games:
    - Cooperative matching games [Shapley & Shubik '71]
    - Network bargaining games [Kleinberg & Tardos '08]
  - Instances of such games are described by a graph $G = (V, E)$ where
    - Vertices represent players
    - The cardinality of a maximum matching represents a total value that the players could get by interacting with each other.
  - Goal: find a stable outcome (players do not have incentive to deviate)
Matching and Network games

- Several interesting *game theory* problems are defined on *networks*: the structure of the underlying graph is fundamental to have good outcomes.
Matching and Network games

- Several interesting *game theory* problems are defined on *networks*: the structure of the underlying graph is fundamental to have good outcomes.

- Stable graphs play a crucial role in some Network Games:
Matching and Network games

- Several interesting *game theory* problems are defined on *networks*: the structure of the underlying graph is fundamental to have good outcomes.

- Stable graphs play a crucial role in some Network Games:
  - *Cooperative matching* games [Shapley & Shubik ’71]
  - *Network bargaining* games [Kleinberg & Tardos ’08]
Matching and Network games

• Several interesting game theory problems are defined on networks: the structure of the underlying graph is fundamental to have good outcomes.

• Stable graphs play a crucial role in some Network Games:
  – Cooperative matching games [Shapley & Shubik ’71]
  – Network bargaining games [Kleinberg & Tardos ’08]

• Instances of such games are described by a graph $G = (V, E)$ where
Matching and Network games

• Several interesting game theory problems are defined on networks: the structure of the underlying graph is fundamental to have good outcomes.

• Stable graphs play a crucial role in some Network Games:
  - Cooperative matching games [Shapley & Shubik '71]
  - Network bargaining games [Kleinberg & Tardos '08]

• Instances of such games are described by a graph $G = (V, E)$ where
  - Vertices represent players
Matching and Network games

• Several interesting game theory problems are defined on networks: the structure of the underlying graph is fundamental to have good outcomes.

• Stable graphs play a crucial role in some Network Games:
  - Cooperative matching games [Shapley & Shubik ’71]
  - Network bargaining games [Kleinberg & Tardos ’08]

• Instances of such games are described by a graph $G = (V, E)$ where
  - Vertices represent players
  - The cardinality of a maximum matching represents a total value that the players could get by interacting with each other.
Matching and Network games

- Several interesting game theory problems are defined on networks: the structure of the underlying graph is fundamental to have good outcomes.

- Stable graphs play a crucial role in some Network Games:
  
  - Cooperative matching games [Shapley & Shubik ’71]
  - Network bargaining games [Kleinberg & Tardos ’08]

- Instances of such games are described by a graph $G = (V, E)$ where
  
  - Vertices represent players
  - The cardinality of a maximum matching represents a total value that the players could get by interacting with each other.

- Goal: find a stable outcome (players do not have incentive to deviate)
Matching and Network games

- Several interesting *game theory* problems are defined on *networks*: the structure of the underlying graph is fundamental to have good outcomes.

- Stable graphs play a crucial role in some Network Games:
  - *Cooperative matching* games [Shapley & Shubik ’71]
  - *Network bargaining* games [Kleinberg & Tardos ’08]

- Instances of such games are described by a graph $G = (V, E)$ where
  - *Vertices* represent players
  - The cardinality of a *maximum matching* represents a total *value* that the players could get by interacting with each other.

- Goal: find a *stable outcome* (players do not have incentive to deviate)
Network Bargaining Games

Network bargaining games are described by a graph $G = (V, E)$ where

- Vertices represent players
- Edges represent potential deals of unit value between players

Players can enter in a deal with at most one neighbour $\to$ matching $M$

If players $u$ and $v$ make a deal, they agree on how to split a unit value $\to$ allocation $y \in \mathbb{R}^V$: $y_u + y_v = 1$ for all $\{uv\} \in M$

$y_u = 0$ if $u$ is exposed by $M$.

An outcome for the game is a pair $(M, y)$.
Network Bargaining Games

- *Network bargaining games* are described by a graph $G = (V, E)$ where
  - Vertices represent players
  - Edges represent potential deals of unit value between players
  - Players can enter in a deal with at most one neighbour → matching $M$
  - If players $u$ and $v$ make a deal, they agree on how to split a unit value → allocation $y \in \mathbb{R}^V$: $y_u + y_v = 1$ for all $\{uv\} \in M$
  - $y_u = 0$ if $u$ is exposed by $M$.

An outcome for the game is a pair $(M, y)$.
Network Bargaining Games

- *Network bargaining games* are described by a graph $G = (V, E)$ where
  - *Vertices* represent players
  - *Edges* represent potential deals of unit value between players

- Players can enter in a deal with at most one neighbour
- Matching $M$
- If players $u$ and $v$ make a deal, they agree on how to split a unit value
  - Allocation $y \in \mathbb{R}^V$: $y_u + y_v = 1$ for all $(uv) \in M$
  - $y_u = 0$ if $u$ is exposed by $M$.
Network Bargaining Games

- *Network bargaining games* are described by a graph $G = (V, E)$ where
  - **Vertices** represent players
  - **Edges** represent potential deals of unit value between players
- Players can enter in a deal with at most one neighbour
Network Bargaining Games

- Network bargaining games are described by a graph \( G = (V, E) \) where
  - Vertices represent players
  - Edges represent potential deals of unit value between players
- Players can enter in a deal with at most one neighbour
  \( \rightarrow \text{matching } M \)
Network Bargaining Games

- Network bargaining games are described by a graph $G = (V, E)$ where
  - **Vertices** represent players
  - **Edges** represent potential deals of unit value between players

- Players can enter in a deal with at most one neighbour
  $\rightarrow$ matching $M$

- If players $u$ and $v$ make a deal, they agree on how to split a unit value
Network Bargaining Games

- *Network bargaining games* are described by a graph $G = (V, E)$ where
  - *Vertices* represent players
  - *Edges* represent potential deals of unit value between players

- Players can enter in a deal with at most one neighbour
  $$\rightarrow \text{matching } M$$

- If players $u$ and $v$ make a deal, they agree on how to split a unit value
  $$\rightarrow \text{allocation } y \in \mathbb{R}^V:$$
  $$y_u + y_v = 1 \text{ for all } \{uv\} \in M$$
  $$y_u = 0 \text{ if } u \text{ is exposed by } M.$$
Network Bargaining Games

- **Network bargaining games** are described by a graph $G = (V, E)$ where
  - **Vertices** represent players
  - **Edges** represent potential deals of unit value between players
- Players can enter in a deal with at most one neighbour
  - $\rightarrow$ **matching** $M$
- If players $u$ and $v$ make a deal, they agree on how to split a unit value
  - $\rightarrow$ **allocation** $y \in \mathbb{R}^V$:
    - $y_u + y_v = 1$ for all $\{uv\} \in M$
    - $y_u = 0$ if $u$ is exposed by $M$.
- An **outcome** for the game is a pair $(M, y)$
Network Bargaining Games

For a given outcome \((M, y)\) player \(u\) gets implicitly an outside alternative ▶ If there exists a neighbour \(v\) of \(u\) with \(1 - y^v > y^u\) → player \(u\) has an incentive to enter in a deal with \(v\)!

An outcome \((M, y)\) is stable if \(y^u + y^v \geq 1\) for all edges \(\{uv\} \in E\). → no player has an incentive to deviate

[Kleinberg & Tardos'08] proved that for network bargaining instances A stable outcome exists ⇔ the correspondent graph \(G\) is stable.
Network Bargaining Games

• For a given outcome \((M, y)\) player \(u\) gets implicitly an outside alternative

\[\text{If there exists a neighbour } v \text{ of } u \text{ with } 1 - y_v > y_u \rightarrow \text{player } u \text{ has an incentive to enter in a deal with } v!\]

• An outcome \((M, y)\) is stable if \(y_u + y_v \geq 1\) for all edges \(\{uv\} \in E\).

\[\rightarrow \text{no player has an incentive to deviate}\]

• [Kleinberg & Tardos'08] proved that for network bargaining instances \(\text{a stable outcome exists } \iff \text{the correspondent graph } G \text{ is stable.}\)
Network Bargaining Games

- For a given outcome \((M, y)\) player \(u\) gets implicitly an **outside alternative**
  - If there exists a neighbour \(v\) of \(u\) with \(1 - y_v > y_u\)

\(\text{[Kleinberg & Tardos'08]}\) proved that for network bargaining instances

\(\text{a stable outcome exists} \iff \text{the correspondent graph } G \text{ is stable.}\)
Network Bargaining Games

- For a given outcome \((M, y)\) player \(u\) gets implicitly an outside alternative
  
  - If there exists a neighbour \(v\) of \(u\) with \(1 - y_v > y_u\)
    
    \(\rightarrow\) player \(u\) has an incentive to enter in a deal with \(v\)!

[Kleinberg & Tardos'08] proved that for network bargaining instances a stable outcome exists \(\iff\) the correspondent graph \(G\) is stable.
Network Bargaining Games

- For a given outcome \((M, y)\) player \(u\) gets implicitly an outside alternative
  
  - If there exists a neighbour \(v\) of \(u\) with \(1 - y_v > y_u\)
    
    \(\rightarrow\) player \(u\) has an incentive to enter in a deal with \(v\)!

- An outcome \((M, y)\) is stable if \(y_u + y_v \geq 1\) for all edges \(\{uv\} \in E\).
Network Bargaining Games

• For a given outcome \((M, y)\) player \(u\) gets implicitly an outside alternative
  
  ▶ If there exists a neighbour \(v\) of \(u\) with \(1 - y_v > y_u\)
    
    \(\rightarrow\) player \(u\) has an incentive to enter in a deal with \(v\)!

• An outcome \((M, y)\) is stable if \(y_u + y_v \geq 1\) for all edges \(\{uv\} \in E\).
    
    \(\rightarrow\) no player has an incentive to deviate
Network Bargaining Games

• For a given outcome \((M, y)\) player \(u\) gets implicitly an outside alternative

  ▶ If there exists a neighbour \(v\) of \(u\) with \(1 - y_v > y_u\)
    
    \(\rightarrow\) player \(u\) has an incentive to enter in a deal with \(v\)!

• An outcome \((M, y)\) is stable if \(y_u + y_v \geq 1\) for all edges \(\{uv\} \in E\).

    \(\rightarrow\) no player has an incentive to deviate

• [Kleinberg & Tardos’08] proved that for network bargaining instances
**Network Bargaining Games**

- For a given outcome \((M, y)\) player \(u\) gets implicitly an **outside alternative**
  - If there exists a neighbour \(v\) of \(u\) with \(1 - y_v > y_u\)
    - \(u\) has an incentive to enter in a deal with \(v\)!

- An outcome \((M, y)\) is **stable** if \(y_u + y_v \geq 1\) for all edges \(\{uv\} \in E\).
  - \(\rightarrow\) no player has an incentive to deviate

- [Kleinberg & Tardos’08] proved that for network bargaining instances
  - A stable outcome exists
Network Bargaining Games

- For a given outcome \((M, y)\) player \(u\) gets implicitly an outside alternative.
  - If there exists a neighbour \(v\) of \(u\) with \(1 - y_v > y_u\)
    \(\rightarrow\) player \(u\) has an incentive to enter in a deal with \(v\)!

- An outcome \((M, y)\) is stable if \(y_u + y_v \geq 1\) for all edges \(\{uv\} \in E\).
  \(\rightarrow\) no player has an incentive to deviate.

- [Kleinberg & Tardos’08] proved that for network bargaining instances
  
  A stable outcome exists
  \(\Leftrightarrow\)
  the correspondent graph \(G\) is stable.
Cooperative matching games

• A similar result holds for cooperative matching games.
Cooperative matching games

- A similar result holds for cooperative matching games.

- In cooperative matching instance, we search for an allocation $y \in \mathbb{R}_{\geq 0}^v$ of the value $\nu(G) := |\text{max matching}|$, such that
  - no subset $S \subseteq V$ has incentive to deviate

[Shubik & Shapley'71] proved A stable allocation exists $\iff$ the correspondent graph $G$ is stable.

Question: [Biró, Kern & Paulusma'10, Kőnemann, Larson & Steiner'12] Can we stabilize unstable games through minimal changes in the underlying network?

$\rightarrow$ Let's look at this question from a graph theory perspective
Cooperative matching games

- A similar result holds for cooperative matching games.

- In cooperative matching instance, we search for an allocation $y \in \mathbb{R}^v_{\geq 0}$ of the value $\nu(G) := |\text{max matching}|$, such that
  - no subset $S \subseteq V$ has incentive to deviate ($y(S) < \nu(G[S])$).

[Shubik & Shapley'71] proved $\text{A stable allocation exists} \iff \text{the correspondent graph } G$ is stable.

Question: [Biró, Kern & Paulusma'10, Kőnemann, Larson & Steiner'12] Can we stabilize unstable games through minimal changes in the underlying network? 

$\rightarrow$ Let's look at this question from a graph theory perspective.
Cooperative matching games

- A similar result holds for cooperative matching games.

- In cooperative matching instance, we search for an allocation $y \in \mathbb{R}^V_{\geq 0}$ of the value $\nu(G) := |\text{max matching}|$, such that no subset $S \subseteq V$ has incentive to deviate ($y(S) < \nu(G[S])$).

- [Shubik & Shapley'71] proved
  
  A stable allocation exists
A similar result holds for cooperative matching games.

In cooperative matching instance, we search for an allocation \( y \in \mathbb{R}_{\geq 0}^V \) of the value \( \nu(G) := |\text{max matching}| \), such that

- no subset \( S \subseteq V \) has incentive to deviate \( (y(S) < \nu(G[S])) \).

[Shubik & Shapley'71] proved

A stable allocation exists \( \iff \) the correspondent graph \( G \) is stable.
Cooperative matching games

- A similar result holds for cooperative matching games.

- In cooperative matching instance, we search for an allocation \( y \in \mathbb{R}^v_{\geq 0} \) of the value \( \nu(G) := |\text{max matching}| \), such that
  
  - no subset \( S \subseteq V \) has incentive to deviate \( (y(S) < \nu(G[S])) \).

- [Shubik & Shapley’71] proved

  A stable allocation exists \( \iff \) the correspondent graph \( G \) is stable.

**Question:** [Biró, Kern & Paulusma’10, Könemann, Larson & Steiner’12]

Can we stabilize unstable games through minimal changes in the underlying network?
Cooperative matching games

- A similar result holds for cooperative matching games.

- In cooperative matching instance, we search for an allocation $y \in \mathbb{R}^{V}_{\geq 0}$ of the value $\nu(G) := |\text{max matching}|$, such that
  - no subset $S \subseteq V$ has incentive to deviate ($y(S) < \nu(G[S])$).

- [Shubik & Shapley’71] proved
  
  A stable allocation exists $\iff$ the correspondent graph $G$ is stable.

**Question:** [Biró, Kern & Paulusma’10, Könemann, Larson & Steiner’12]

*Can we stabilize unstable games through minimal changes in the underlying network?*

→ Let’s look at this question from a graph theory perspective
Stabilizers

- Two natural graph operations:
  - edge-removal operation → blocking some potential deals
  - vertex-removal operation → blocking some players

Def. An edge-stabilizer for $G = (V, E)$ is a subset $F \subseteq E$ s.t. $G \setminus F$ is stable.
Stabilizers

- Two natural graph operations:

  ▶ edge-removal operation → blocking some potential deals
  ▶ vertex-removal operation → blocking some players

Def. An edge-stabilizer for $G = (V, E)$ is a subset $F \subseteq E$ s.t. $G \setminus F$ is stable.
Stabilizers

• Two natural graph operations:
  ▶ edge-removal operation
  ▶ vertex-removal operation
Stabilizers

- Two natural graph operations:
  - edge-removal operation  →  blocking some potential deals
  - vertex-removal operation  →  blocking some players

Def. An edge-stabilizer for $G = (V, E)$ is a subset $F \subseteq E$ s.t. $G \setminus F$ is stable.
Stabilizers

- Two natural graph operations:
  - edge-removal operation $\rightarrow$ blocking some potential deals
  - vertex-removal operation $\rightarrow$ blocking some players

**Def.** An edge-stabilizer for $G = (V, E)$ is a subset $F \subseteq E$ s.t. $G \setminus F$ is stable.
Stabilizers

- Two natural graph operations:
  - edge-removal operation → blocking some potential deals
  - vertex-removal operation → blocking some players

**Def.** An edge-stabilizer for $G = (V, E)$ is a subset $F \subseteq E$ s.t. $G \setminus F$ is stable.
Stabilizers

- Two natural graph operations:
  - edge-removal operation → blocking some potential deals
  - vertex-removal operation → blocking some players

**Def.** An edge-stabilizer for $G = (V, E)$ is a subset $F \subseteq E$ s.t. $G \setminus F$ is stable.

**Def.** A vertex-stabilizer for $G = (V, E)$ is a subset $S \subseteq V$ s.t. $G \setminus S$ is stable.
Stabilizers

- Two natural graph operations:
  - **edge-removal operation** → blocking some potential deals
  - **vertex-removal operation** → blocking some players

**Def.** An **edge-stabilizer** for $G = (V, E)$ is a subset $F \subseteq E$ s.t. $G \setminus F$ is stable.

**Def.** A **vertex-stabilizer** for $G = (V, E)$ is a subset $S \subseteq V$ s.t. $G \setminus S$ is stable.

**Combinatorial question:** Can we efficiently find (edge-/vertex-) stabilizers of **minimum cardinality**?
Edge-stabilizers: complexity results

Recall $\nu(G)$ denote the cardinality of a maximum matching in $G$.

Thm: [Bock, Chandrasekaran, Kónemann, Peis, S. '14] For a minimum edge-stabilizer $F$ of $G$ we have $\nu(G \setminus F) = \nu(G)$.

Network Bargaining Interpretation: there is always a way to stabilize the game that ▶ blocks min number of potential deals, and ▶ does not decrease the total value the players can get!

Edge-stabilizers: complexity results

- Recall $\nu(G)$ denote the cardinality of a maximum matching in $G$. 

**Theorem:** [Bock, Chandrasekaran, Könemann, Peis, S. '14] For a minimum edge-stabilizer $F$ of $G$ we have $\nu(G \setminus F) = \nu(G)$. 

**Theorem:** [Bock, Chandrasekaran, Könemann, Peis, S. '14] Finding a minimum cardinality edge-stabilizer is an NP-Hard problem.
Edge-stabilizers: complexity results

- Recall $\nu(G)$ denote the cardinality of a maximum matching in $G$.

**Thm: [Bock, Chandrasekaran, K"onemann, Peis, S. ’14]** For a minimum edge-stabilizer $F$ of $G$ we have

$$\nu(G \setminus F) = \nu(G)$$
Edge-stabilizers: complexity results

- Recall \( \nu(G) \) denote the cardinality of a maximum matching in \( G \).

**Thm: [Bock, Chandrasekaran, Konemann, Peis, S. ’14]** For a minimum edge-stabilizer \( F \) of \( G \) we have

\[
\nu(G \setminus F) = \nu(G)
\]

- Network Bargaining Interpretation: there is always a way to stabilize the game that
Edge-stabilizers: complexity results

- Recall $\nu(G)$ denote the cardinality of a maximum matching in $G$.

**Thm: [Bock, Chandrasekaran, Kömemann, Peis, S. ’14]** For a minimum edge-stabilizer $F$ of $G$ we have

$$\nu(G \setminus F) = \nu(G)$$

- Network Bargaining Interpretation: there is always a way to stabilize the game that
  - blocks min number of potential deals, and
Edge-stabilizers: complexity results

- Recall $\nu(G)$ denote the cardinality of a maximum matching in $G$.

**Thm:** [Bock, Chandrasekaran, Könemann, Peis, S. ’14] For a minimum edge-stabilizer $F$ of $G$ we have

$$\nu(G \setminus F) = \nu(G)$$

- Network Bargaining Interpretation: there is always a way to stabilize the game that
  - blocks min number of potential deals, and
  - does not decrease the total value the players can get!
Edge-stabilizers: complexity results

• Recall $\nu(G)$ denote the cardinality of a maximum matching in $G$.

Thm: [Bock, Chandrasekaran, K"onemann, Peis, S. ’14] For a minimum edge-stabilizer $F$ of $G$ we have

$$\nu(G \setminus F) = \nu(G)$$

• Network Bargaining Interpretation: there is always a way to stabilize the game that
  ▶ blocks min number of potential deals, and
  ▶ does not decrease the total value the players can get!

Vertex-stabilizers: complexity results

- Recall $\nu(G)$ denote the cardinality of a maximum matching in $G$. 

\[\text{Thm: [Ahmadian, Hosseinzadeh, S. '16]}\]
For a minimum vertex-stabilizer $S$ of $G$ we have $\nu(G \setminus S) = \nu(G)$.

\[\text{Network Bargaining Interpretation: there is always a way to stabilize the game that}\]
\[\text{blocks min number of players, and}\]
\[\text{does not decrease the total value the players can get!}\]

\[\text{Thm: [Ito, Kakimura, Kamiyama, Kobayashi, Okamoto '16], [AHS'16]}\]
Finding a minimum cardinality vertex-stabilizer is a polynomial-time solvable problem.

$\rightarrow$ How are these results proved?
Vertex-stabilizers: complexity results

- Recall $\nu(G)$ denote the cardinality of a maximum matching in $G$.

**Thm: [Ahmadian, Hosseinzadeh, S. ’16]** For a minimum vertex-stabilizer $S$ of $G$ we have

$$\nu(G \setminus S) = \nu(G)$$
Vertex-stabilizers: complexity results

- Recall $\nu(G)$ denote the cardinality of a maximum matching in $G$.

**Thm: [Ahmadian, Hosseinzadeh, S. ’16]** For a minimum vertex-stabilizer $S$ of $G$ we have

$$\nu(G \setminus S) = \nu(G)$$

- Network Bargaining Interpretation: there is always a way to stabilize the game that
Vertex-stabilizers: complexity results

- Recall $\nu(G)$ denote the cardinality of a maximum matching in $G$.

**Thm: [Ahmadian, Hosseinzadeh, S. ’16]** For a minimum vertex-stabilizer $S$ of $G$ we have

$$\nu(G \setminus S) = \nu(G)$$

- Network Bargaining Interpretation: there is always a way to stabilize the game that
  - blocks *min number* of players, and
Vertex-stabilizers: complexity results

• Recall $\nu(G)$ denote the cardinality of a maximum matching in $G$.

**Thm: [Ahmadian, Hosseinzadeh, S. '16]** For a minimum vertex-stabilizer $S$ of $G$ we have

$$\nu(G \setminus S) = \nu(G)$$

• Network Bargaining Interpretation: there is always a way to stabilize the game that
  ▶ blocks min number of players, and
  ▶ does not decrease the total value the players can get!
**Vertex-stabilizers: complexity results**

- Recall $\nu(G)$ denote the cardinality of a maximum matching in $G$.

**Thm: [Ahmadian, Hosseinzadeh, S. ’16]** For a minimum vertex-stabilizer $S$ of $G$ we have

$$\nu(G \setminus S) = \nu(G)$$

- Network Bargaining Interpretation: there is always a way to stabilize the game that
  - blocks **min number** of players, and
  - does not decrease the total value the players can get!

**Thm: [Ito,Kakimura,Kamiyama,Kobayashi,Okamoto ’16], [AHS’16]**

*Finding a minimum cardinality vertex-stabilizer is a polynomial-time solvable problem.*
**Vertex-stabilizers: complexity results**

- Recall $\nu(G)$ denote the cardinality of a maximum matching in $G$.

**Thm: [Ahmadian, Hosseinzadeh, S. ’16]** For a minimum vertex-stabilizer $S$ of $G$ we have

\[ \nu(G \setminus S) = \nu(G) \]

- Network Bargaining Interpretation: there is always a way to stabilize the game that
  - blocks min number of players, and
  - does not decrease the total value the players can get!

**Thm: [Ito,Kakimura,Kamiyama,Kobayashi,Okamoto ’16], [AHS’16]**

*Finding a minimum cardinality vertex-stabilizer is a polynomial-time solvable problem.*

$\rightarrow$ How are these results proved?
Matchings and Covers

• A “dual” problem to matchings is finding a minimum vertex-cover of a graph.
Matchings and Covers

• A “dual” problem to matchings is finding a minimum vertex-cover of a graph.

**Recall:** A vertex-cover of $G = (V, E)$ is a subset $C \subseteq V$ s.t. each $e \in E$ is incident into at least one vertex of $C$. 
A “dual” problem to matchings is finding a minimum vertex-cover of a graph. Recall: A vertex-cover of $G = (V, E)$ is a subset $C \subseteq V$ s.t. each $e \in E$ is incident into at least one vertex of $C$. Let $\tau(G)$ denote the minimum cardinality of a vertex-cover. It is well known that the inequality $\nu(G) \leq \tau(G)$ holds for all graphs $G$. 
Matchings and Covers

- A “dual” problem to matchings is finding a minimum vertex-cover of a graph.

**Recall:** A vertex-cover of $G = (V, E)$ is a subset $C \subseteq V$ s.t. each $e \in E$ is incident into at least one vertex of $C$.

- Let $\tau(G)$ denote the minimum cardinality of a vertex-cover.
**Matchings and Covers**

- A "dual" problem to matchings is finding a **minimum vertex-cover** of a graph.

**Recall:** A vertex-cover of $G = (V, E)$ is a subset $C \subseteq V$ s.t. each $e \in E$ is incident into at least one vertex of $C$.

- Let $\tau(G)$ denote the minimum cardinality of a vertex-cover.
- It is well known that the inequality $\nu(G) \leq \tau(G)$ holds for all graphs $G$. 
Matchings and Covers

- A “dual” problem to matchings is finding a minimum vertex-cover of a graph.

**Recall:** A vertex-cover of $G = (V, E)$ is a subset $C \subseteq V$ s.t. each $e \in E$ is incident into at least one vertex of $C$.

- Let $\tau(G)$ denote the minimum cardinality of a vertex-cover.
- It is well known that the inequality $\nu(G) \leq \tau(G)$ holds for all graphs $G$. 
There are some graphs where the inequality $\nu(G) \leq \tau(G)$ holds tight. 

\textbf{Konig's theorem (1931)}: For any bipartite graph $G$, $\nu(G) = \tau(G)$.

The inequality holds tight for a superclass of bipartite graphs. A graph $G$ satisfying $\nu(G) = \tau(G)$, is called a Konig-Egervary graph.

Stable graphs are a superclass of Konig-Egervary graphs.
Matchings and Covers

- There are some graphs where the inequality $\nu(G) \leq \tau(G)$ holds tight.

[König's theorem (1931)]: For any bipartite graph $G$, $\nu(G) = \tau(G)$.

- The inequality holds tight for a superclass of bipartite graphs.

A graph $G$ satisfying $\nu(G) = \tau(G)$, is called a König-Egerváry graph.
Matchings and Covers

- There are some graphs where the inequality $\nu(G) \leq \tau(G)$ holds **tight**.

**[König’s theorem (1931)]:** For any bipartite graph $G$, $\nu(G) = \tau(G)$. 
Matchings and Covers

- There are some graphs where the inequality \( \nu(G) \leq \tau(G) \) holds tight.

[König’s theorem (1931)]: For any bipartite graph \( G \), \( \nu(G) = \tau(G) \).

- The inequality holds tight for a superclass of bipartite graphs.
Matchings and Covers

• There are some graphs where the inequality $\nu(G) \leq \tau(G)$ holds tight.

[König’s theorem (1931)]: For any bipartite graph $G$, $\nu(G) = \tau(G)$.

• The inequality holds tight for a superclass of bipartite graphs. A graph $G$ satisfying $\nu(G) = \tau(G)$, is called a König-Egerváry graph.
Matchings and Covers

• There are some graphs where the inequality $\nu(G) \leq \tau(G)$ holds tight.

[König’s theorem (1931)]: For any bipartite graph $G$, $\nu(G) = \tau(G)$.

• The inequality holds tight for a superclass of bipartite graphs. A graph $G$ satisfying $\nu(G) = \tau(G)$, is called a König-Egerváry graph.
Matchings and Covers

- There are some graphs where the inequality $\nu(G) \leq \tau(G)$ holds tight.

**[König’s theorem (1931)]:** For any bipartite graph $G$, $\nu(G) = \tau(G)$.

- The inequality holds tight for a superclass of bipartite graphs. A graph $G$ satisfying $\nu(G) = \tau(G)$, is called a **König-Egerváry** graph.

- Stable graphs are a superclass of Königs-Egerváry graphs,
Matchings and Covers

- There are some graphs where the inequality $\nu(G) \leq \tau(G)$ holds tight.

**[König’s theorem (1931)]:** For any bipartite graph $G$, $\nu(G) = \tau(G)$.

- The inequality holds tight for a *superclass* of bipartite graphs. A graph $G$ satisfying $\nu(G) = \tau(G)$, is called a König-Egerváry graph.

![Diagram showing the relationship between bipartite, Konig-Egerváry, stable, and general graphs]

- Stable graphs are a superclass of König-Egerváry graphs,
• There are some graphs where the inequality \( \nu(G) \leq \tau(G) \) holds tight.

[\textit{König’s theorem (1931)}]: \textit{For any bipartite graph } \(G\), \( \nu(G) = \tau(G) \).

• The inequality holds tight for a \textit{superclass} of bipartite graphs. A graph \( G \) satisfying \( \nu(G) = \tau(G) \), is called a \textit{König-Egerváry} graph.

• Stable graphs are a superclass of König-Egerváry graphs, and can be characterized in terms of \textit{fractional} matchings and covers.
Fractional matchings and covers

Finding a maximum matching of a graph $G = (V, E)$ can be formulated as the following Integer Program (IP):

$$\nu(G) := \max \left\{ \sum_{v \in V} x(\delta(v)) : \sum_{v \in V} x(\delta(v)) \leq 1 \\forall v \in V, x \in \{0, 1\} \right\}$$

Finding a minimum vertex-cover can be formulated as the following IP:

$$\tau(G) := \min \left\{ \sum_{u \in V} y(u) + \sum_{v \in V} y(v) : \sum_{e = \{u, v\} \in E} y(u) + y(v) \geq 1 \\forall e \in E, y \in \{0, 1\} \right\}$$

If we relax the integrality constraints, we get a pair of Linear Programs (LP).

$$\nu_f(G) := \max \left\{ \sum_{v \in V} x(\delta(v)) : \sum_{v \in V} x(\delta(v)) \leq 1 \\forall v \in V, x \in \mathbb{R} \geq 0 \right\}$$

$$\tau_f(G) := \min \left\{ \sum_{u \in V} y(u) + \sum_{v \in V} y(v) : \sum_{e = \{u, v\} \in E} y(u) + y(v) \geq 1 \\forall e \in E, y \in \mathbb{R} \geq 0 \right\}$$

Feasible solutions to these LPs yield fractional matchings and covers!
Fractional matchings and covers

• Finding a maximum matching of a graph $G = (V, E)$ can be formulated as the following Integer Program (IP):

$$\nu(G) := \max \{1^T x : x(\delta(v)) \leq 1 \ \forall v \in V, \ x \in \{0, 1\}^E \}$$
Fractional matchings and covers

• Finding a maximum matching of a graph $G = (V, E)$ can be formulated as the following Integer Program (IP):

$$\nu(G) := \max \{1^T x : x(\delta(v)) \leq 1 \ \forall v \in V, \ x \in \{0, 1\}^E\}$$

• Finding a minimum vertex-cover can be formulated as the following IP:

$$\tau(G) := \min \{1^T y : y_u + y_v \geq 1 \ \forall e = \{u, v\} \in E, \ y \in \{0, 1\}^V\}$$
Fractional matchings and covers

- Finding a maximum matching of a graph $G = (V, E)$ can be formulated as the following Integer Program (IP):

$$\nu(G) := \max \{1^T x : x(\delta(v)) \leq 1 \quad \forall v \in V, \quad x \in \{0, 1\}^E\}$$

- Finding a minimum vertex-cover can be formulated as the following IP:

$$\tau(G) := \min \{1^T y : y_u + y_v \geq 1 \quad \forall e = \{u, v\} \in E, \quad y \in \{0, 1\}^V\}$$

- If we relax the integrality constraints, we get a pair of Linear Programs (LP).

$$\nu_f(G) := \max \{1^T x : x(\delta(v)) \leq 1 \quad \forall v \in V, \quad x \in \mathbb{R}^E_{\geq 0}\}$$

$$\tau_f(G) := \min \{1^T y : y_u + y_v \geq 1 \quad \forall e = \{u, v\} \in E, \quad y \in \mathbb{R}^V_{\geq 0}\}$$
Fractional matchings and covers

- Finding a maximum matching of a graph $G = (V, E)$ can be formulated as the following Integer Program (IP):

$$\nu(G) := \max \{ 1^T x : x(\delta(v)) \leq 1 \ \forall v \in V, \ x \in \{0, 1\}^E \}$$

- Finding a minimum vertex-cover can be formulated as the following IP:

$$\tau(G) := \min \{ 1^T y : y_u + y_v \geq 1 \ \forall e = \{u, v\} \in E, \ y \in \{0, 1\}^V \}$$

- If we relax the integrality constraints, we get a pair of Linear Programs (LP).

$$\nu_f(G) := \max \{ 1^T x : x(\delta(v)) \leq 1 \ \forall v \in V, \ x \in \mathbb{R}_\geq^E \}$$

$$\tau_f(G) := \min \{ 1^T y : y_u + y_v \geq 1 \ \forall e = \{u, v\} \in E, \ y \in \mathbb{R}_\geq^V \}$$

- Feasible solutions to these LPs yield *fractional* matchings and covers!
Fractional matchings and covers
**Fractional matchings and covers**

**Def.** A vector $x \in \mathbb{R}^E$ is a **fractional matching** if it is a feasible solution to:

$$\nu_f(G) := \max \{1^T x : x(\delta(v)) \leq 1 \forall v \in V, x \in \mathbb{R}_{\geq 0}^E\}$$
Fractional matchings and covers

**Def.** a vector $x \in \mathbb{R}^E$ is a **fractional matching** if it is a feasible solution to:

$$
\nu_f(G) := \max \{1^T x : x(\delta(v)) \leq 1 \ \forall v \in V, \ x \in \mathbb{R}_{\geq 0}^E \}
$$

**Def.** a vector $y \in \mathbb{R}^V$ is called a **fractional vertex-cover** if it is a feasible solution to its dual:

$$
\tau_f(G) := \min \{1^T y : y_u + y_v \geq 1 \ \forall e = \{u, v\} \in E, \ y \in \mathbb{R}_{\geq 0}^V \}$$
Fractional matchings and covers

**Def.** A vector $x \in \mathbb{R}^E$ is a **fractional matching** if it is a feasible solution to:

$$\nu_f(G) := \max\{1^T x : x(\delta(v)) \leq 1 \forall v \in V, x \in \mathbb{R}^E_{\geq 0}\}$$

**Def.** A vector $y \in \mathbb{R}^V$ is called a **fractional vertex-cover** if it is a feasible solution to its dual:

$$\tau_f(G) := \min\{1^T y : y_u + y_v \geq 1 \forall e = \{u, v\} \in E, y \in \mathbb{R}^V_{\geq 0}\}$$

- **By duality:** We know that the following chain of inequalities holds for all $G$:

  $$\nu(G) \leq \nu_f(G) = \tau_f(G) \leq \tau(G)$$
Fractional matchings and covers

- Example:

\[ \nu(G) \leq \nu_f(G) = \tau_f(G) \leq \tau(G) \]
Fractional matchings and covers

- Example:

\[ \nu(G) \leq \nu_f(G) = \tau_f(G) \leq \tau(G) \]

\[ \nu(G) = 1 \]
Fractional matchings and covers

- Example:

\[ \nu(G) \leq \nu_f(G) = \tau_f(G) \leq \tau(G) \]

- \( \nu(G) = 1 \)
- \( \nu_f(G) = 1.5 \)
Fractional matchings and covers

- **Example:**

\[ \nu(G) \leq \nu_f(G) = \tau_f(G) \leq \tau(G) \]

- \( \nu(G) = 1 \)
- \( \nu_f(G) = 1.5 \)
- \( \tau_f(G) = 1.5 \)
Fractional matchings and covers

• Example:

\[ \nu(G) \leq \nu_f(G) \leq \tau_f(G) \leq \tau(G) \]

- \( \nu(G) = 1 \)
- \( \nu_f(G) = 1.5 \)
- \( \tau_f(G) = 1.5 \)
- \( \tau(G) = 2 \)
Fractional matchings and covers

**Proposition:** $G$ is stable if and only if $\nu(G) = \nu_f(G) = \tau_f(G)$.

(it follows from classical results e.g. [Uhry'75, Balas'81, Pulleyblank'87].)

- In other words, $G$ is stable if and only if the cardinality of a max matching equals the min size of a fractional vertex cover.
- Note: such a $y$ does not necessarily have integer coordinates!
- In fact, $\text{General graphs} \supset \text{Stable graphs} \supset \text{König-Egerváry graphs} \supset \text{Bipartite graphs}$.
- The fact that $\nu(G) = \nu_f(G)$ allows us to exploit properties of max matchings and max fractional matchings to stabilize graphs.
- Key ingredient: Edmonds-Gallai Decomposition of a graph.
Fractional matchings and covers

Proposition: $G$ is stable if and only if $\nu(G) = \nu_f(G) = \tau_f(G)$.

(It follows from classical results e.g. [Uhry'75, Balas'81, Pulleyblank'87])
Fractional matchings and covers

**Proposition:** $G$ is stable if and only if $\nu(G) = \nu_f(G) = \tau_f(G)$.

(It follows from classical results e.g. [Uhry’75, Balas’81, Pulleyblank’87])

- In other words, $G$ is stable if and only if
  
  cardinality of a max matching $= \min$ size of a fractional vertex cover $y$. 

Fractional matchings and covers

**Proposition:** $G$ is stable if and only if $\nu(G) = \nu_f(G) = \tau_f(G)$.

(It follows from classical results e.g. [Uhry’75, Balas’81, Pulleyblank’87])

- In other words, $G$ is stable if and only if
  
  cardinality of a max matching = min size of a fractional vertex cover $y$.

- **Note:** such $y$ does not necessarily have integer coordinates!
Fractional matchings and covers

**Proposition:** $G$ is stable if and only if $\nu(G) = \nu_f(G) = \tau_f(G)$.

(It follows from classical results e.g. [Uhry’75, Balas’81, Pulleyblank’87])

- In other words, $G$ is stable if and only if
  \[
  \text{cardinality of a max matching} = \text{min size of a fractional vertex cover} \ y.
  \]

- **Note:** such $y$ does not necessarily have integer coordinates! In fact,

  General graphs $\supset$ Stable graphs $\supset$ König-Egervary graphs $\supset$ Bipartite graphs.
Fractional matchings and covers

Proposition: $G$ is stable if and only if $\nu(G) = \nu_f(G) = \tau_f(G)$.

(It follows from classical results e.g. [Uhry'75, Balas'81, Pulleyblank'87])

- In other words, $G$ is stable if and only if
  
  \[
  \text{cardinality of a max matching } = \min \text{ size of a fractional vertex cover } y.
  \]

- Note: such $y$ does not necessarily have integer coordinates! In fact,

  General graphs $\supset$ Stable graphs $\supset$ König-Egervary graphs $\supset$ Bipartite graphs.

- The fact that $\nu(G) = \nu_f(G)$ allows us to exploit properties of \textit{max matchings}
  and \textit{max fractional matchings} to stabilize graphs.
Fractional matchings and covers

Proposition: \( G \) is stable if and only if \( \nu(G) = \nu_f(G) = \tau_f(G) \).

(It follows from classical results e.g. [Uhry’75, Balas’81, Pulleyblank’87])

- In other words, \( G \) is stable if and only if the cardinality of a max matching equals the min size of a fractional vertex cover \( y \).

- Note: such \( y \) does not necessarily have integer coordinates! In fact,

  General graphs \( \supset \) Stable graphs \( \supset \) König-Egervary graphs \( \supset \) Bipartite graphs.

- The fact that \( \nu(G) = \nu_f(G) \) allows us to exploit properties of max matchings and max fractional matchings to stabilize graphs.

  Key ingredient: Edmonds-Gallai Decomposition of a graph.
Edmonds-Gallai decomposition

• The Edmonds-Gallai decomposition of \( G = (V, E) \) is a partition of \( V \) into three sets \( B, C, D \) such that:
  - \( B \) contains the set of inessential vertices of \( G \)
  - \( C \) contains the set of neighbors of \( B \)
  - \( D \) contains all remaining vertices
Edmonds-Gallai decomposition

- The Edmonds-Gallai decomposition of $G = (V, E)$ is a partition of $V$ into three sets $B, C, D$ such that:

  - $B$ contains the set of inessential vertices of $G$,
  - $C$ contains the set of neighbors of $B$,
  - $D$ contains all remaining vertices.
The Edmonds-Gallai decomposition of $G = (V, E)$ is a partition of $V$ into three sets $B, C, D$ such that:

- $B$ contains the set of inessential vertices of $G$
- $C$ contains the set of neighbors of $B$
- $D$ contains all remaining vertices
Edmonds-Gallai decomposition

- The Edmonds-Gallai decomposition of $G = (V, E)$ is a partition of $V$ into three sets $B, C, D$ such that:

  ▶ $B$ contains the set of inessential vertices of $G$

![Diagram of Edmonds-Gallai decomposition](image)
The Edmonds-Gallai decomposition of $G = (V, E)$ is a partition of $V$ into three sets $B, C, D$ such that:

- $B$ contains the set of inessential vertices of $G$
- $C$ contains the set of neighbors of $B$
The Edmonds-Gallai decomposition of $G = (V, E)$ is a partition of $V$ into three sets $B, C, D$ such that:

- **B** contains the set of inessential vertices of $G$
- **C** contains the set of neighbors of **B**
- **D** contains all remaining vertices
Edmonds-Gallai decomposition

- The Edmonds-Gallai decomposition of $G = (V, E)$ is a partition of $V$ into three sets $B, C, D$ such that:

- **Note:** The Edmonds-Gallai decomposition of a graph can be computed in polynomial-time.
Edmonds-Gallai decomposition

- The Edmonds-Gallai decomposition of $G = (V, E)$ is a partition of $V$ into three sets $B$, $C$, $D$ such that:

- **Note:** The Edmonds-Gallai decomposition of a graph can be computed in polynomial-time.

- What is the relation between this decomposition and max matchings?
Edmonds-Gallai decomposition

• Let $M$ be any maximum matching of $G$. 
• Let $M$ be any maximum matching of $G$. 
Edmonds-Gallai decomposition

- Let $M$ be any maximum matching of $G$. Then
Edmonds-Gallai decomposition


![Diagram showing Edmonds-Gallai decomposition]
Let $M$ be any maximum matching of $G$. Then

- $M$ induces a near-perfect matching in each component of $G[B]$
- $M$ matches $C$ to distinct components of $G[B]$
Let $M$ be *any* maximum matching of $G$. Then

- $M$ induces a near-perfect matching in each component of $G[B]$.
- $M$ matches $C$ to distinct components of $G[B]$.
- $M$ induces a perfect matching in $G[D]$. 
Edmonds-Gallai decomposition

- Let $M$ be any maximum matching of $G$. Then

  - $M$ induces a near-perfect matching in each component of $G[B]$.
  - $M$ matches $C$ to distinct components of $G[B]$.
  - $M$ induces a perfect matching in $G[D]$. 
Edmonds-Gallai decomposition

- Let $M$ be a maximum matching of $G$, that **covers the maximum number of singletons in** $G[B]$.
Edmonds-Gallai decomposition

• Let $M$ be a maximum matching of $G$, that covers the maximum number of singletons in $G[B]$. 

![Diagram](image-url)
Edmonds-Gallai decomposition

- Let $M$ be a maximum matching of $G$, that covers the maximum number of singletons in $G[B]$.
Edmonds-Gallai decomposition

• Let $M$ be a maximum matching of $G$, that covers the maximum number of singletons in $G[B]$.

Construct a fractional matching $x \in \mathbb{R}^E$ as follows:
Edmonds-Gallai decomposition

- Let $M$ be a maximum matching of $G$, that covers the maximum number of singletons in $G[B]$.

Construct a fractional matching $x \in \mathbb{R}^E$ as follows:

- Find odd cycles in $G[B]$ containing $M$-exposed vertices,
Edmonds-Gallai decomposition

- Let $M$ be a maximum matching of $G$, that covers the maximum number of singletons in $G[B]$.

Construct a fractional matching $x \in \mathbb{R}^E$ as follows:

- Find odd cycles in $G[B]$ containing $M$-exposed vertices,
Edmonds-Gallai decomposition

- Let $M$ be a maximum matching of $G$, that covers the maximum number of singletons in $G[B]$.

Construct a fractional matching $x \in \mathbb{R}^E$ as follows:

- Find odd cycles in $G[B]$ containing $M$-exposed vertices, set $x_e = \frac{1}{2}$ for such edges. Set $x_e = 1$ for all other edges of $M$. 
Edmonds-Gallai decomposition

• Let $M$ be a maximum matching of $G$, that covers the maximum number of singletons in $G[B]$. 

```
B C D
0.5 0.5 0.5
0.5 0.5 0.5
1 1 1
1 1 1
```

Construct a fractional matching $x \in \mathbb{R}^E$ as follows:

- Find odd cycles in $G[B]$ containing $M$-exposed vertices, set $x_e = \frac{1}{2}$ for such edges. Set $x_e = 1$ for all other edges of $M$.

Then, $x$ is maximum fractional matching.
Edmonds-Gallai decomposition

- We can use this insight to prove our theorems.
Edmonds-Gallai decomposition

- We can use this insight to prove our theorems. Recall the structural results.
Edmonds-Gallai decomposition

- We can use this insight to prove our theorems. Recall the structural results.

Thm: [BCKPS '14] For a minimum edge-stabilizer $F$ of $G$, $\nu(G \setminus F) = \nu(G)$.

Thm: [AHS '16] For a minimum vertex-stabilizer $S$ of $G$, $\nu(G \setminus S) = \nu(G)$. 
Edmonds-Gallai decomposition

- We can use this insight to prove our theorems. Recall the structural results.

Thm: [BCKPS '14] For a minimum edge-stabilizer $F$ of $G$, $\nu(G \setminus F) = \nu(G)$.

Thm: [AHS '16] For a minimum vertex-stabilizer $S$ of $G$, $\nu(G \setminus S) = \nu(G)$.

- Intuition: We need to “kill” the fractional cycles.
Edmonds-Gallai decomposition

- We can use this insight to prove our theorems. Recall the structural results.

\[ \text{Thm: [BCKPS '14]} \quad \text{For a minimum edge-stabilizer } F \text{ of } G, \nu(G \setminus F) = \nu(G). \]

\[ \text{Thm: [AHS '16]} \quad \text{For a minimum vertex-stabilizer } S \text{ of } G, \nu(G \setminus S) = \nu(G). \]

- Intuition: We need to “kill” the fractional cycles. Edges/vertices achieving this goal can be chosen to be disjoint by at least one max matching.
We can use this insight to prove our theorems.
Edmonds-Gallai decomposition

- We can use this insight to prove our theorems. Recall the algorithmic results.
Edmonds-Gallai decomposition

- We can use this insight to prove our theorems. Recall the algorithmic results.

\[\text{Thm: [AHS '16, IKKKO '16]}\] Finding a minimum vertex-stabilizer is solvable in polynomial-time.
Edmonds-Gallai decomposition

- We can use this insight to prove our theorems. Recall the algorithmic results.

**Thm:** [AHS ’16, IKKKO ’16] *Finding a minimum vertex-stabilizer is solvable in polynomial-time.*

- **Intuition:** We construct the fractional matching $x$ as before.
Edmonds-Gallai decomposition

- We can use this insight to prove our theorems. Recall the algorithmic results.

Thm: [AHS ’16, IKKKO ’16] Finding a minimum vertex-stabilizer is solvable in polynomial-time.

- Intuition: We construct the fractional matching $x$ as before.
Edmonds-Gallai decomposition

- We can use this insight to prove our theorems. Recall the algorithmic results.

Thm: [AHS ’16, IKKKO ’16] Finding a minimum vertex-stabilizer is solvable in polynomial-time.

- Intuition: We construct the fractional matching $x$ as before. We remove one distinct vertex from each fractional cycle.
Edmonds-Gallai decomposition

• We can use this insight to prove our theorems. Recall the algorithmic results.

Thm: [AHS '16, IKKKO '16] Finding a minimum vertex-stabilizer is solvable in polynomial-time.

▶ Intuition: We construct the fractional matching $x$ as before. We remove one distinct vertex from each fractional cycle.
Edmonds-Gallai decomposition

- We can use this insight to prove our theorems. Recall the algorithmic results.

**Thm: [BCKPS '14]** Finding a minimum edge-stabilizer is an NP-hard problem.
Edmonds-Gallai decomposition

- We can use this insight to prove our theorems. Recall the algorithmic results.

Thm: [BCKPS '14] Finding a minimum edge-stabilizer is an NP-hard problem.

- Intuition: Though the number of fractional cycle is a lower bound on the number of edges to remove,
Edmonds-Gallai decomposition

- We can use this insight to prove our theorems. Recall the algorithmic results.

**Thm: [BCKPS '14]** Finding a minimum edge-stabilizer is an NP-hard problem.

- **Intuition:** Though the number of fractional cycle is a lower bound on the number of edges to remove, it is not clear how many of them to select...
Edmonds-Gallai decomposition

- We can use this insight to prove our theorems. Recall the algorithmic results.

**Thm: [BCKPS ’14]** Finding a minimum edge-stabilizer is an NP-hard problem.

- **Intuition:** Though the number of fractional cycle is a lower bound on the number of edges to remove, it is not clear how many of them to select...

How about approximation algorithms?
Approximation algorithms

**Def.** An algorithm is called an \( \alpha \)-approximation algorithm for a minimization problem \( \Pi \) if for every instance of \( \Pi \), it computes in polynomial-time a feasible solution of value at most \( \alpha \)-times the value of an optimal solution.

- A graph \( G \) is called \( \omega \)-sparse if \( \forall S \subseteq V, |E(S)| \leq \omega |S| \).

- **Thm [Bock, Chandrasekaran, Köhnen, Peis, S.'14]:** There is an \( O(\omega) \)-approximation algorithm for finding a minimum edge-stabilizer.

  - The algorithm relies on the following **Lemma**.

  **Lemma:** Let \( G \) be s.t. \( \nu_f(G) > \nu(G) \). We can find \( L \subseteq E \) with \( |L| \leq O(\omega) \) s.t.

  \[
  \begin{align*}
  (i) & \quad G \setminus L \text{ has a matching of size } \nu(G), \\
  (ii) & \quad \nu_f(G \setminus L) \leq \nu_f(G) - \frac{1}{2}.
  \end{align*}
  \]

  - In other words, we can find a small subset of edges to remove from \( G \) that

    - **(i)** does not decrease the value of a max matching,
    - **(ii)** reduces the minimum size of a fractional vertex cover.
Approximation algorithms

**Def.** An algorithm is called an $\alpha$-approximation algorithm for a minimization problem $\Pi$ if for every instance of $\Pi$, it computes in polynomial-time a feasible solution of value at most $\alpha$-times the value of an optimal solution.

- A graph $G$ is called $\omega$-sparse if $\forall S \subseteq V, |E(S)| \leq \omega |S|$.
Approximation algorithms

**Def.** An algorithm is called an $\alpha$-approximation algorithm for a minimization problem $\Pi$ if for every instance of $\Pi$, it computes in polynomial-time a feasible solution of value at most $\alpha$-times the value of an optimal solution.

- A graph $G$ is called $\omega$-sparse if $\forall S \subseteq V$, $|E(S)| \leq \omega|S|$.

**Thm [Bock, Chandrasekaran, Könemann, Peis, S.’14]:** There is a $O(\omega)$-approximation algorithm for finding a minimum edge-stabilizer.
**Approximation algorithms**

**Def.** An algorithm is called an $\alpha$-approximation algorithm for a minimization problem $\Pi$ if for every instance of $\Pi$, it computes in polynomial-time a feasible solution of value at most $\alpha$-times the value of an optimal solution.

- A graph $G$ is called $\omega$-sparse if $\forall S \subseteq V, |E(S)| \leq \omega |S|$.

**Thm [Bock, Chandrasekaran, Königemann, Peis, S.’14]:** There is a $O(\omega)$-approximation algorithm for finding a minimum edge-stabilizer.

- The algorithm relies on the following Lemma.
Def. An algorithm is called an $\alpha$-approximation algorithm for a minimization problem $\Pi$ if for every instance of $\Pi$, it computes in polynomial-time a feasible solution of value at most $\alpha$-times the value of an optimal solution.

- A graph $G$ is called $\omega$-sparse if $\forall S \subseteq V$, $|E(S)| \leq \omega |S|$.

Thm [Bock, Chandrasekaran, Könemann, Peis, S.’14]: There is a $O(\omega)$-approximation algorithm for finding a minimum edge-stabilizer.

- The algorithm relies on the following Lemma.

Lemma: Let $G$ be s.t. $\nu_f(G) > \nu(G)$. 

Approximation algorithms
Approximation algorithms

**Def.** An algorithm is called an \(\alpha\)-approximation algorithm for a minimization problem \(\Pi\) if for every instance of \(\Pi\), it computes in polynomial-time a feasible solution of value at most \(\alpha\)-times the value of an optimal solution.

- A graph \(G\) is called \(\omega\)-sparse if \(\forall S \subseteq V, |E(S)| \leq \omega|S|\).

**Thm [Bock, Chandrasekaran, K"{o}nemann, Peis, S.’14]:** There is a \(O(\omega)\)-approximation algorithm for finding a minimum edge-stabilizer.

- The algorithm relies on the following Lemma.

**Lemma:** Let \(G\) be s.t. \(\nu_f(G) > \nu(G)\). We can find \(L \subseteq E\) with \(|L| \leq O(\omega)\) s.t.
  - (i) \(G \setminus L\) has a matching of size \(\nu(G)\),
Approximation algorithms

**Def.** An algorithm is called an $\alpha$-approximation algorithm for a minimization problem $\Pi$ if for every instance of $\Pi$, it computes in polynomial-time a feasible solution of value at most $\alpha$-times the value of an optimal solution.

- A graph $G$ is called $\omega$-sparse if $\forall S \subseteq V, |E(S)| \leq \omega |S|$.

**Thm [Bock, Chandrasekaran, Könemann, Peis, S.’14]:** There is a $O(\omega)$-approximation algorithm for finding a minimum edge-stabilizer.

- The algorithm relies on the following Lemma.

**Lemma:** Let $G$ be s.t. $\nu_f(G) > \nu(G)$. We can find $L \subseteq E$ with $|L| \leq O(\omega)$ s.t.
  - (i) $G \setminus L$ has a matching of size $\nu(G)$, and (ii) $\nu_f(G \setminus L) \leq \nu_f(G) - \frac{1}{2}$. 

Approximation algorithms

**Def.** An algorithm is called an \( \alpha \)-approximation algorithm for a minimization problem \( \Pi \) if for every instance of \( \Pi \), it computes in polynomial-time a feasible solution of value at most \( \alpha \)-times the value of an optimal solution.

- A graph \( G \) is called \( \omega \)-sparse if \( \forall S \subseteq V, |E(S)| \leq \omega|S| \).

**Thm [Bock, Chandrasekaran, K"{o}nemann, Peis, S.’14]:** There is a \( O(\omega) \)-approximation algorithm for finding a minimum edge-stabilizer.

- The algorithm relies on the following Lemma.

**Lemma:** Let \( G \) be s.t. \( \nu_f(G) > \nu(G) \). We can find \( L \subseteq E \) with \( |L| \leq O(\omega) \) s.t.
  - (i) \( G \setminus L \) has a matching of size \( \nu(G) \), and (ii) \( \nu_f(G \setminus L) \leq \nu_f(G) - \frac{1}{2} \).

- In other words, we can find a small subset of edges to remove from \( G \) that
Approximation algorithms

**Def.** An algorithm is called an $\alpha$-approximation algorithm for a minimization problem $\Pi$ if for every instance of $\Pi$, it computes in polynomial-time a feasible solution of value at most $\alpha$-times the value of an optimal solution.

• A graph $G$ is called $\omega$-sparse if $\forall S \subseteq V$, $|E(S)| \leq \omega|S|$.

**Thm [Bock, Chandrasekaran, Könenmann, Peis, S.’14]:** There is a $O(\omega)$-approximation algorithm for finding a minimum edge-stabilizer.

• The algorithm relies on the following Lemma.

**Lemma:** Let $G$ be s.t. $\nu_f(G) > \nu(G)$. We can find $L \subseteq E$ with $|L| \leq O(\omega)$ s.t.

- (i) $G \setminus L$ has a matching of size $\nu(G)$, and (ii) $\nu_f(G \setminus L) \leq \nu_f(G) - \frac{1}{2}$.

• In other words, we can find a small subset of edges to remove from $G$ that
  - (i) does not decrease the value of a max matching,
Approximation algorithms

**Def.** An algorithm is called an $\alpha$-approximation algorithm for a minimization problem $\Pi$ if for every instance of $\Pi$, it computes in polynomial-time a feasible solution of value at most $\alpha$-times the value of an optimal solution.

- A graph $G$ is called $\omega$-sparse if $\forall S \subseteq V$, $|E(S)| \leq \omega|S|$.

**Thm [Bock, Chandrasekaran, K"onnen, Peis, S.’14]:** There is a $O(\omega)$-approximation algorithm for finding a minimum edge-stabilizer.

- The algorithm relies on the following Lemma.

**Lemma:** Let $G$ be s.t. $\nu_f(G) > \nu(G)$. We can find $L \subseteq E$ with $|L| \leq O(\omega)$ s.t.
  - (i) $G \setminus L$ has a matching of size $\nu(G)$, and (ii) $\nu_f(G \setminus L) \leq \nu_f(G) - \frac{1}{2}$.

- In other words, we can find a small subset of edges to remove from $G$ that
  - (i) does not decrease the value of a max matching, and (ii) reduces the minimum size of a fractional vertex cover.
Further remarks
Further remarks

**Open question:** Is a $O(1)$-approximation possible for min-edge stabilizer problem?
Further remarks

**Open question:** Is a $O(1)$-approximation possible for min-edge stabilizer problem?

- A graph $G = (V, E)$ is **factor-critical** if for every $v \in V$, $G \setminus \{v\}$ has a perfect matching.
Further remarks

Open question: Is a \( O(1) \)-approximation possible for min-edge stabilizer problem?

• A graph \( G = (V, E) \) is factor-critical if for every \( v \in V \), \( G \setminus \{v\} \) has a perfect matching. Here one can find a maximum fractional matching with one odd cycle in the support.
Further remarks

Open question: Is a $O(1)$-approximation possible for min-edge stabilizer problem?

- A graph $G = (V, E)$ is factor-critical if for every $v \in V$, $G \setminus \{v\}$ has a perfect matching. Here one can find a maximum fractional matching with one odd cycle in the support. Is an $O(1)$-approximation possible here?
Further remarks

**Open question:** Is a $O(1)$-approximation possible for min-edge stabilizer problem?

- A graph $G = (V, E)$ is **factor-critical** if for every $v \in V$, $G \setminus \{v\}$ has a perfect matching. Here one can find a maximum fractional matching with one odd cycle in the support. Is an $O(1)$-approximation possible here?

- Subclasses of graphs?
  - For $d$-regular graphs (→ each player has the same number of potential deals), previous algorithm of [BCKPS'14] yields a 2-approximation
Further remarks

**Open question:** Is a $O(1)$-approximation possible for min-edge stabilizer problem?

- A graph $G = (V, E)$ is **factor-critical** if for every $v \in V$, $G \setminus \{v\}$ has a perfect matching. Here one can find a maximum fractional matching with one odd cycle in the support. Is an $O(1)$-approximation possible here?

- Subclasses of graphs?
  - For $d$-regular graphs (→ each player has the same number of potential deals), previous algorithm of [BCKPS'14] yields a 2-approximation

- What about $b$-matchings? (→ each player $v$ can enter in $b_v$ deals)
Further remarks

Network bargaining games are more generally defined on weighted graphs (each edge represents a deal of value $w$).

Stable graphs: In this setting, a graph $G$ is stable if the max-weight matching equals the cardinality of a min-fractional $w$-cover.

In this setting, min-stabilizers may reduce the weight of max-matching, and finding a min vertex-stabilizer preserving a max-weight matching is no longer poly-time solvable [Koh, S.'17].

Good (approximation) algorithms in this case?
Further remarks

- Network bargaining games are more generally defined on weighted graphs.
Further remarks

• Network bargaining games are more generally defined on weighted graphs
  (→ each edge represents a deal of value $w_e$)
Further remarks

- Network bargaining games are more generally defined on weighted graphs
  \((\rightarrow \text{each edge represents a deal of value } w_e)\)

**Stable graphs:** In this setting, a graph \(G\) is stable if the max-weight matching equals the cardinality of a min-fractional \(w\)-cover.
Further remarks

• Network bargaining games are more generally defined on weighted graphs (→ each edge represents a deal of value $w_e$)

Stable graphs: In this setting, a graph $G$ is stable if the max-weight matching equals the cardinality of a min-fractional $w$-cover.

• In this setting, min-stabilizers may reduce the weight of max-matching,
Further remarks

- Network bargaining games are more generally defined on weighted graphs (→ each edge represents a deal of value $w_e$)

**Stable graphs**: In this setting, a graph $G$ is stable if the max-weight matching equals the cardinality of a min-fractional $w$-cover.

- In this setting, min-stabilizers may reduce the weight of max-matching, and finding a min vertex-stabilizer preserving a max-weight matching is no longer poly-time solvable [Koh, S.’17]
Further remarks

• Network bargaining games are more generally defined on weighted graphs
  (→ each edge represents a deal of value \( w_e \))

**Stable graphs:** In this setting, a graph \( G \) is stable if the max-weight matching
equals the cardinality of a min-fractional \( w \)-cover.

• In this setting, min-stabilizers may reduce the weight of max-matching,
  and finding a min vertex-stabilizer preserving a max-weight matching is no longer
  poly-time solvable [Koh, S.’17]

• Good (approximation) algorithms in this case?
Further remarks

• Network bargaining games are more generally defined on weighted graphs
  (→ each edge represents a deal of value $w_e$)

**Stable graphs:** In this setting, a graph $G$ is stable if the max-weight matching equals the cardinality of a min-fractional $w$-cover.

• In this setting, min-stabilizers may reduce the weight of max-matching, and finding a min vertex-stabilizer preserving a max-weight matching is no longer poly-time solvable [Koh, S.’17]

• Good (approximation) algorithms in this case?

Thank you!