An Efficient Characterization of Submodular Spanning Tree Games

Zhuan Khye Koh Laura Sanità

THE LONDON SCHOOL OF ECONOMICS AND POLITICAL SCIENCE

UNIVERSITY OF WATERLOO

Cooperative game

Cooperative game

Setting: A set of players N who are allowed to cooperate.

Cooperative game

Setting: A set of players N who are allowed to cooperate.

Goal: Distribute **cost** or revenue among them.

Cooperative game

Setting: A set of players N who are allowed to cooperate.

Goal: Distribute **cost** or revenue among them.

- To model cooperation, we use a characteristic function $\nu : 2^N \rightarrow \mathbb{R}$.

Cooperative game

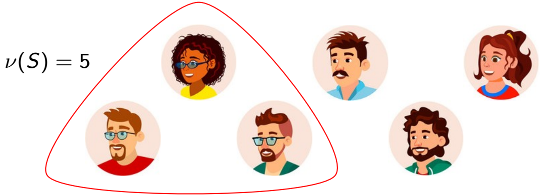
Setting: A set of players N who are allowed to cooperate.

Goal: Distribute **cost** or revenue among them.

- To model cooperation, we use a characteristic function $\nu : 2^N \to \mathbb{R}$.
  - $\nu(S) =$ total cost paid by the players in S if they form a coalition.

Cooperative game

Setting: A set of players N who are allowed to cooperate.

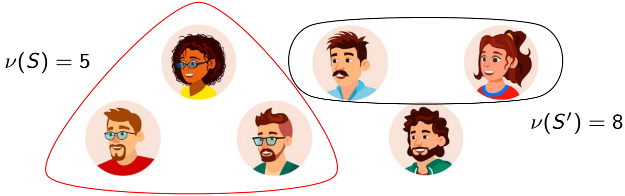


Goal: Distribute **cost** or revenue among them.

- To model cooperation, we use a characteristic function $\nu : 2^N \to \mathbb{R}$.
  - $\nu(S) =$ total cost paid by the players in S if they form a coalition.

Cooperative game

Setting: A set of players N who are allowed to cooperate.

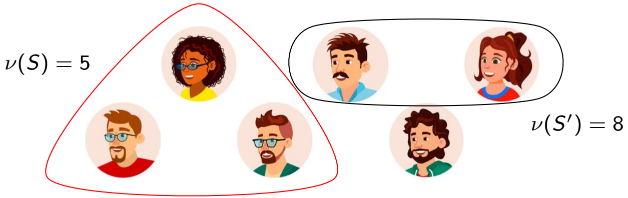


Goal: Distribute **cost** or revenue among them.

- To model cooperation, we use a characteristic function $\nu : 2^N \to \mathbb{R}$.
  - $\nu(S) =$ total cost paid by the players in S if they form a coalition.

Cooperative game

Setting: A set of players N who are allowed to cooperate.

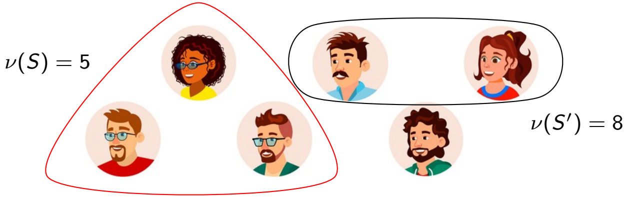


Goal: Distribute **cost** or revenue among them.

- To model cooperation, we use a characteristic function $\nu : 2^N \to \mathbb{R}$.
  - $\nu(S) =$ total cost paid by the players in S if they form a coalition.
- An instance of the game is defined by (N, ν).

Cooperative game

Setting: A set of players N who are allowed to cooperate.



Goal: Distribute **cost** or revenue among them.

- To model cooperation, we use a characteristic function $\nu : 2^N \to \mathbb{R}$.
  - $\nu(S)$ = total cost paid by the players in S if they form a coalition.
- An instance of the game is defined by (N, ν).
- An outcome of the game is an allocation $y \in \mathbb{R}^N$ such that

$$\sum_{v \in N} y_v = \nu(N).$$

How “good” is an allocation?

How “good” is an allocation?

- Many criteria for evaluating an allocation:

How "good" is an allocation?

- Many criteria for evaluating an allocation:
  - **Fairness** - Is every agent charged proportionally to its contribution?

How "good" is an allocation?

- Many criteria for evaluating an allocation:
  - **Fairness** - Is every agent charged proportionally to its contribution?
  - **Stability** - Are there any incentives to cooperate?

How “good” is an allocation?

- Many criteria for evaluating an allocation:
  - **Fairness** - Is every agent charged proportionally to its contribution?
  - **Stability** - Are there any incentives to cooperate?

- We use solution concepts. Some popular ones include:

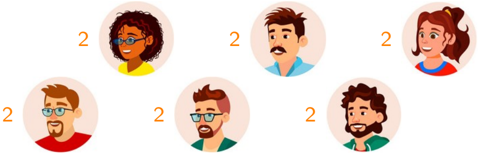
How "good" is an allocation?

- Many criteria for evaluating an allocation:
  - **Fairness** - Is every agent charged proportionally to its contribution?
  - **Stability** - Are there any incentives to cooperate?

- We use solution concepts. Some popular ones include:
  - *Core:* $\sum_{v \in S} y_v \leq \nu(S)$ for all $S \subseteq N$.

How "good" is an allocation?

- Many criteria for evaluating an allocation:
  - **Fairness** - Is every agent charged proportionally to its contribution?
  - **Stability** - Are there any incentives to cooperate?



- We use solution concepts. Some popular ones include:
  - *Core:* $\sum_{v \in S} y_v \leq \nu(S)$ for all $S \subseteq N$.

How "good" is an allocation?

- Many criteria for evaluating an allocation:
  - **Fairness** - Is every agent charged proportionally to its contribution?
  - **Stability** - Are there any incentives to cooperate?

- We use solution concepts. Some popular ones include:
  - *Core:* $\sum_{v \in S} y_v \leq \nu(S)$ for all $S \subseteq N$.

How “good” is an allocation?

- Many criteria for evaluating an allocation:
  - **Fairness** - Is every agent charged proportionally to its contribution?
  - **Stability** - Are there any incentives to cooperate?

- We use solution concepts. Some popular ones include:
  - *Core:* $\sum_{v \in S} y_v \leq \nu(S)$ for all $S \subseteq N$.

How “good” is an allocation?

- Many criteria for evaluating an allocation:
  - **Fairness** - Is every agent charged proportionally to its contribution?
  - **Stability** - Are there any incentives to cooperate?

- We use solution concepts. Some popular ones include:
  - *Core:* $\sum_{v \in S} y_v \leq \nu(S)$ for all $S \subseteq N$.
  - *Shapley value, nucleolus, kernel, bargaining set, stable set, . . .*

How "good" is an allocation?

- Many criteria for evaluating an allocation:
  - **Fairness** - Is every agent charged proportionally to its contribution?
  - **Stability** - Are there any incentives to cooperate?

- We use solution concepts. Some popular ones include:
  - *Core:* $\sum_{v \in S} y_v \leq \nu(S)$ for all $S \subseteq N$.
  - *Shapley value, nucleolus, kernel, bargaining set, stable set, ...*
- Generally hard to compute unless ν satisfies certain properties.

Submodularity

Submodularity

Def. A game is submodular/convex if for any $S, T \subseteq N$,

$$\nu(S) + \nu(T) \geq \nu(S \cup T) + \nu(S \cap T).$$

Submodularity

Def. A game is submodular/convex if for any $S, T \subseteq N$,

$$\nu(S) + \nu(T) \geq \nu(S \cup T) + \nu(S \cap T).$$

- Equivalently, for any $S \subseteq T \subseteq N$ and $u \in N \setminus T$,

$$\nu(S \cup u) - \nu(S) \geq \nu(T \cup u) - \nu(T).$$

Submodularity

Def. A game is submodular/convex if for any $S, T \subseteq N$,

$$\nu(S) + \nu(T) \geq \nu(S \cup T) + \nu(S \cap T).$$

- Equivalently, for any $S \subseteq T \subseteq N$ and $u \in N \setminus T$,

$$\nu(S \cup u) - \nu(S) \geq \nu(T \cup u) - \nu(T).$$

- "Snowballing" effect.

Submodularity

Def. A game is submodular/convex if for any $S, T \subseteq N$,

$$\nu(S) + \nu(T) \geq \nu(S \cup T) + \nu(S \cap T).$$

- Equivalently, for any $S \subseteq T \subseteq N$ and $u \in N \setminus T$,

$$\nu(S \cup u) - \nu(S) \geq \nu(T \cup u) - \nu(T).$$

- "Snowballing" effect.
- Some advantages of submodularity:

Submodularity

Def. A game is submodular/convex if for any $S, T \subseteq N$,

$$\nu(S) + \nu(T) \geq \nu(S \cup T) + \nu(S \cap T).$$

- Equivalently, for any $S \subseteq T \subseteq N$ and $u \in N \setminus T$,

$$\nu(S \cup u) - \nu(S) \geq \nu(T \cup u) - \nu(T).$$

- "Snowballing" effect.
- Some advantages of submodularity:
  - [Shapley '71] A core solution exists and can be computed efficiently.

Submodularity

Def. A game is submodular/convex if for any $S, T \subseteq N$,

$$\nu(S) + \nu(T) \geq \nu(S \cup T) + \nu(S \cap T).$$

- Equivalently, for any $S \subseteq T \subseteq N$ and $u \in N \setminus T$,

$$\nu(S \cup u) - \nu(S) \geq \nu(T \cup u) - \nu(T).$$

- "Snowballing" effect.
- Some advantages of submodularity:
  - [Shapley '71] A core solution exists and can be computed efficiently.
  - Core membership is easy.

Submodularity

Def. A game is submodular/convex if for any $S, T \subseteq N$,

$$\nu(S) + \nu(T) \geq \nu(S \cup T) + \nu(S \cap T).$$

- Equivalently, for any $S \subseteq T \subseteq N$ and $u \in N \setminus T$,

$$\nu(S \cup u) - \nu(S) \geq \nu(T \cup u) - \nu(T).$$

- "Snowballing" effect.
- Some advantages of submodularity:
  - [Shapley '71] A core solution exists and can be computed efficiently.
  - Core membership is easy.
  - [Kuipers '96] The nucleolus can be computed efficiently.

Submodularity

Def. A game is submodular/convex if for any $S, T \subseteq N$,

$$\nu(S) + \nu(T) \geq \nu(S \cup T) + \nu(S \cap T).$$

- Equivalently, for any $S \subseteq T \subseteq N$ and $u \in N \setminus T$,

$$\nu(S \cup u) - \nu(S) \geq \nu(T \cup u) - \nu(T).$$

- "Snowballing" effect.
- Some advantages of submodularity:
  - [Shapley '71] A core solution exists and can be computed efficiently.
  - Core membership is easy.
  - [Kuipers '96] The nucleolus can be computed efficiently.

Q. Can we characterize submodular instances of a cooperative game?

Submodularity

Def. A game is submodular/convex if for any $S, T \subseteq N$,

$$\nu(S) + \nu(T) \geq \nu(S \cup T) + \nu(S \cap T).$$

- Equivalently, for any $S \subseteq T \subseteq N$ and $u \in N \setminus T$,

$$\nu(S \cup u) - \nu(S) \geq \nu(T \cup u) - \nu(T).$$

- "Snowballing" effect.
- Some advantages of submodularity:
  - [Shapley '71] A core solution exists and can be computed efficiently.
  - Core membership is easy.
  - [Kuipers '96] The nucleolus can be computed efficiently.

Q. Can we characterize submodular instances of a cooperative game?

- [van den Nouweland & Borm '91] Communication game.

Submodularity

Def. A game is submodular/convex if for any $S, T \subseteq N$,

$$\nu(S) + \nu(T) \geq \nu(S \cup T) + \nu(S \cap T).$$

- Equivalently, for any $S \subseteq T \subseteq N$ and $u \in N \setminus T$,

$$\nu(S \cup u) - \nu(S) \geq \nu(T \cup u) - \nu(T).$$

- "Snowballing" effect.
- Some advantages of submodularity:
  - [Shapley '71] A core solution exists and can be computed efficiently.
  - Core membership is easy.
  - [Kuipers '96] The nucleolus can be computed efficiently.

Q. Can we characterize submodular instances of a cooperative game?

- [van den Nouweland & Borm '91] Communication game.
- [Okamoto '03] Coloring game and vertex cover game.

Spanning tree game

Spanning tree game

- Introduced by [Claus & Kleitman '73].

Spanning tree game

- Introduced by [Claus & Kleitman '73].

Setting: A set of clients N would like to be connected to a source r.

Cheapest solution is a minimum spanning tree.

Goal: Distribute the cost of the tree.

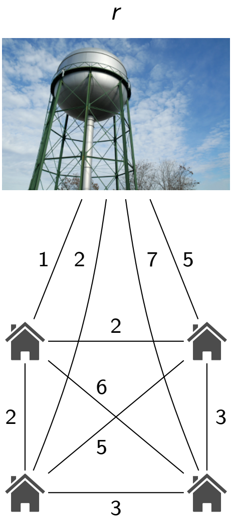
Spanning tree game

- Introduced by [Claus & Kleitman '73].

Setting: A set of clients N would like to be connected to a source r.

Cheapest solution is a minimum spanning tree.

Goal: Distribute the cost of the tree.



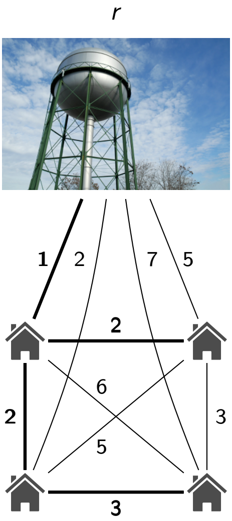
Spanning tree game

- Introduced by [Claus & Kleitman '73].

Setting: A set of clients N would like to be connected to a source r.

Cheapest solution is a minimum spanning tree.

Goal: Distribute the cost of the tree.



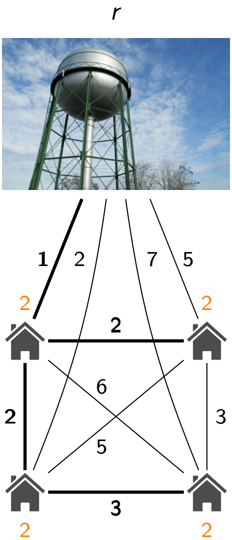
Spanning tree game

- Introduced by [Claus & Kleitman '73].

Setting: A set of clients N would like to be connected to a source r.

Cheapest solution is a minimum spanning tree.

Goal: Distribute the cost of the tree.



Spanning tree game

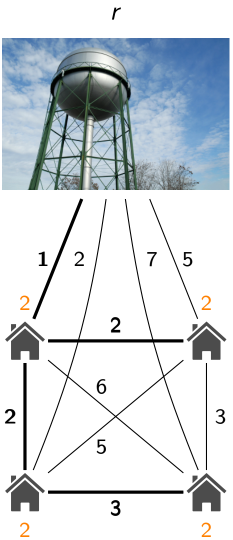
- Introduced by [Claus & Kleitman '73].

Setting: A set of clients N would like to be connected to a source r.

Cheapest solution is a minimum spanning tree.

Goal: Distribute the cost of the tree.

- An instance is defined by an edge-weighted complete graph $G = (V, E)$ where $V = N \cup r$.



Spanning tree game

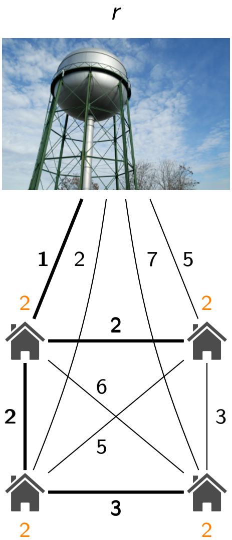
- Introduced by [Claus & Kleitman '73].

Setting: A set of clients N would like to be connected to a source r.

Cheapest solution is a minimum spanning tree.

Goal: Distribute the cost of the tree.

- An instance is defined by an edge-weighted complete graph $G = (V, E)$ where $V = N \cup r$.
- The clients can cooperate.



Spanning tree game

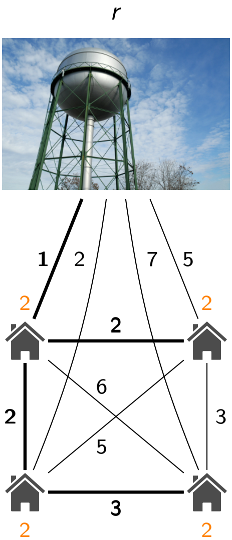
- Introduced by [Claus & Kleitman '73].

Setting: A set of clients N would like to be connected to a source r.

Cheapest solution is a minimum spanning tree.

Goal: Distribute the cost of the tree.

- An instance is defined by an edge-weighted complete graph $G = (V, E)$ where $V = N \cup r$.
- The clients can cooperate.
- For $S \subseteq N$, $\nu(S)$ is the cost of a minimum spanning tree in $G[S \cup r]$.



Spanning tree game

• Introduced by [Claus & Kleitman '73].

Setting: A set of clients N would like to be connected to a source r.

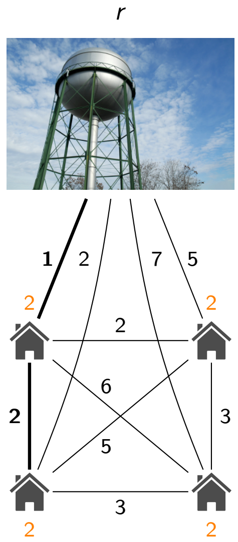
Cheapest solution is a minimum spanning tree.

Goal: Distribute the cost of the tree.

• An instance is defined by an edge-weighted complete graph $G = (V, E)$ where $V = N \cup r$.

• The clients can cooperate.

• For $S \subseteq N$, $\nu(S)$ is the cost of a minimum spanning tree in $G[S \cup r]$.



Spanning tree game

Spanning tree game

- Not submodular.

Spanning tree game

- Not submodular.
- [Bird '76] proposed an allocation scheme.

Spanning tree game

- Not submodular.
- [Bird '76] proposed an allocation scheme.
- [Granot & Huberman '81] Bird's allocation is a core solution.

Spanning tree game

- Not submodular.
- [Bird '76] proposed an allocation scheme.
- [Granot & Huberman '81] Bird's allocation is a core solution.
- [Granot & Huberman '82] The game is permutationally convex.

Spanning tree game

- Not submodular.
- [Bird '76] proposed an allocation scheme.
- [Granot & Huberman '81] Bird's allocation is a core solution.
- [Granot & Huberman '82] The game is permutationally convex.
  - There exists an ordering $1, 2, \ldots, n$ of the players such that for any $j \leq k$ and $S \subseteq N \setminus [k]$,

$$\nu([j] \cup S) - \nu([j]) \geq \nu([k] \cup S) - \nu([k]).$$

Spanning tree game

- Not submodular.
- [Bird '76] proposed an allocation scheme.
- [Granot & Huberman '81] Bird's allocation is a core solution.
- [Granot & Huberman '82] The game is permutationally convex.
  - There exists an ordering $1, 2, \ldots, n$ of the players such that for any $j \leq k$ and $S \subseteq N \setminus [k]$,

$$\nu([j] \cup S) - \nu([j]) \geq \nu([k] \cup S) - \nu([k]).$$

  - Generalizes submodularity.

Spanning tree game

- Not submodular.
- [Bird '76] proposed an allocation scheme.
- [Granot & Huberman '81] Bird's allocation is a core solution.
- [Granot & Huberman '82] The game is permutationally convex.
  - There exists an ordering $1, 2, \ldots, n$ of the players such that for any $j \leq k$ and $S \subseteq N \setminus [k]$,

$$\nu([j] \cup S) - \nu([j]) \geq \nu([k] \cup S) - \nu([k]).$$

  - Generalizes submodularity.
- [Faigle et al. '97] Core membership is co-NP-hard.

Spanning tree game

- Not submodular.
- [Bird '76] proposed an allocation scheme.
- [Granot & Huberman '81] Bird's allocation is a core solution.
- [Granot & Huberman '82] The game is permutationally convex.
  - There exists an ordering $1, 2, \ldots, n$ of the players such that for any $j \leq k$ and $S \subseteq N \setminus [k]$,

    $$\nu([j] \cup S) - \nu([j]) \geq \nu([k] \cup S) - \nu([k]).$$

  - Generalizes submodularity.
- [Faigle et al. '97] Core membership is co-NP-hard.
- [Faigle et al. '98] Computing the nucleolus is NP-hard.

Spanning tree game

- Not submodular.
- [Bird '76] proposed an allocation scheme.
- [Granot & Huberman '81] Bird's allocation is a core solution.
- [Granot & Huberman '82] The game is permutationally convex.
  - There exists an ordering $1, 2, \ldots, n$ of the players such that for any $j \leq k$ and $S \subseteq N \setminus [k]$,

$$\nu([j] \cup S) - \nu([j]) \geq \nu([k] \cup S) - \nu([k]).$$

  - Generalizes submodularity.
- [Faigle et al. '97] Core membership is co-NP-hard.
- [Faigle et al. '98] Computing the nucleolus is NP-hard.

Can we find an **efficient** *characterization of submodular instances?*

State of the art

State of the art

- [Kobayashi & Okamoto '14] characterized submodularity when G has only **two** distinct edge-weights.

State of the art

- [Kobayashi & Okamoto '14] characterized submodularity when G has only **two** distinct edge-weights.
  - Let G_1 be the subgraph spanned by the cheaper edges.

State of the art

- [Kobayashi & Okamoto '14] characterized submodularity when G has only **two** distinct edge-weights.
  - Let G_1 be the subgraph spanned by the cheaper edges.
  - Submodular $\Leftrightarrow$ The vertices of every cycle in G_1 are adjacent to r or pairwise adjacent.

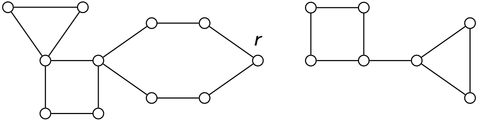
State of the art

- [Kobayashi & Okamoto '14] characterized submodularity when G has only **two** distinct edge-weights.
  - Let G_1 be the subgraph spanned by the cheaper edges.
  - Submodular $\Leftrightarrow$ The vertices of every cycle in G_1 are adjacent to r or pairwise adjacent.
  - Efficiently testable using block decomposition.

State of the art

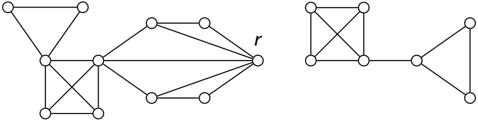
• [Kobayashi & Okamoto '14] characterized submodularity when G has only **two** distinct edge-weights.

- Let G_1 be the subgraph spanned by the cheaper edges.
- Submodular $\Leftrightarrow$ The vertices of every cycle in G_1 are adjacent to r or pairwise adjacent.
- Efficiently testable using block decomposition.



State of the art

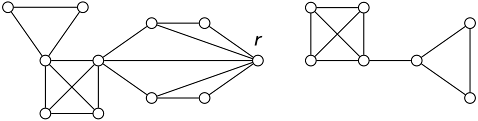
- [Kobayashi & Okamoto '14] characterized submodularity when G has only **two** distinct edge-weights.
  - Let G_1 be the subgraph spanned by the cheaper edges.
  - Submodular $\Leftrightarrow$ The vertices of every cycle in G_1 are adjacent to r or pairwise adjacent.
  - Efficiently testable using block decomposition.



State of the art

• [Kobayashi & Okamoto '14] characterized submodularity when G has only **two** distinct edge-weights.

- Let G_1 be the subgraph spanned by the cheaper edges.
- Submodular $\Leftrightarrow$ The vertices of every cycle in G_1 are adjacent to r or pairwise adjacent.
- Efficiently testable using block decomposition.

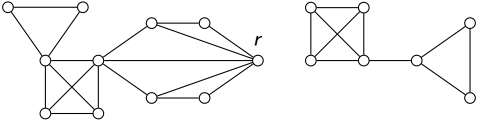


• For general weights, they stated some sufficient conditions and some necessary conditions. [Trudeau '12] also gave a sufficient condition.

State of the art

• [Kobayashi & Okamoto '14] characterized submodularity when G has only **two** distinct edge-weights.

- Let G_1 be the subgraph spanned by the cheaper edges.
- Submodular $\Leftrightarrow$ The vertices of every cycle in G_1 are adjacent to r or pairwise adjacent.
- Efficiently testable using block decomposition.



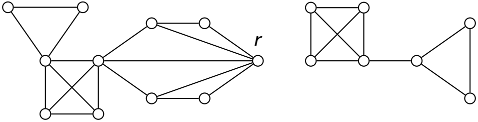
• For general weights, they stated some sufficient conditions and some necessary conditions. [Trudeau '12] also gave a sufficient condition.

• It was conjectured that testing submodularity is co-NP-complete.

State of the art

• [Kobayashi & Okamoto '14] characterized submodularity when G has only **two** distinct edge-weights.

- Let G_1 be the subgraph spanned by the cheaper edges.
- Submodular $\Leftrightarrow$ The vertices of every cycle in G_1 are adjacent to r or pairwise adjacent.
- Efficiently testable using block decomposition.



• For general weights, they stated some sufficient conditions and some necessary conditions. [Trudeau '12] also gave a sufficient condition.

• It was conjectured that testing submodularity is co-NP-complete.

• In this work, we **fully** characterize submodular instances. This characterization can be verified in **polynomial time**.

Preliminaries

Preliminaries

Def. An instance is submodular if for any $S \subseteq N$ and $u, v \in N \setminus S$,

$$\nu(S \cup u) + \nu(S \cup v) - \nu(S) - \nu(S \cup \{u, v\}) \geq 0$$

Preliminaries

Def. An instance is submodular if for any $S \subseteq N$ and $u, v \in N \setminus S$,

$$\nu(S \cup u) + \nu(S \cup v) - \nu(S) - \nu(S \cup \{u, v\}) \geq 0$$

- For $S \subseteq V$, $\text{mst}(S) :=$ cost of a minimum spanning tree in $G[S]$.

Preliminaries

Def. An instance is submodular if for any $S \subseteq N$ and $u, v \in N \setminus S$,

$$\nu(S \cup u) + \nu(S \cup v) - \nu(S) - \nu(S \cup \{u, v\}) \geq 0$$

- For $S \subseteq V$, $\mathrm{mst}(S) :=$ cost of a minimum spanning tree in $G[S]$.
- For $u, v \in N$, $\mathcal{S}_{uv} := \{S \subseteq V : r \in S \text{ and } u, v \notin S\}$.

Preliminaries

Def. An instance is submodular if for any $S \subseteq N$ and $u, v \in N \setminus S$,

$$\mathrm{mst}(S \cup u) + \mathrm{mst}(S \cup v) - \mathrm{mst}(S) - \mathrm{mst}(S \cup \{u, v\}) \geq 0$$

- For $S \subseteq V$, $\mathrm{mst}(S) :=$ cost of a minimum spanning tree in $G[S]$.
- For $u, v \in N$, $\mathcal{S}_{uv} := \{S \subseteq V : r \in S \text{ and } u, v \notin S\}$.

Preliminaries

Def. An instance is submodular if for any $u, v \in N$ and $S \subseteq \mathcal{S}_{uv}$,

$$\mathrm{mst}(S \cup u) + \mathrm{mst}(S \cup v) - \mathrm{mst}(S) - \mathrm{mst}(S \cup \{u, v\}) \geq 0$$

- For $S \subseteq V$, $\mathrm{mst}(S) :=$ cost of a minimum spanning tree in $G[S]$.
- For $u, v \in N$, $\mathcal{S}_{uv} := \{S \subseteq V : r \in S \text{ and } u, v \notin S\}$.

Preliminaries

Def. An instance is submodular if for any $u, v \in N$ and $S \subseteq \mathcal{S}_{uv}$,

$$f_{uv}(S) := \mathrm{mst}(S \cup u) + \mathrm{mst}(S \cup v) - \mathrm{mst}(S) - \mathrm{mst}(S \cup \{u, v\}) \geq 0$$

- For $S \subseteq V$, $\mathrm{mst}(S) :=$ cost of a minimum spanning tree in $G[S]$.
- For $u, v \in N$, $\mathcal{S}_{uv} := \{S \subseteq V : r \in S \text{ and } u, v \notin S\}$.

Preliminaries

Def. An instance is submodular if for any $u, v \in N$ and $S \subseteq \mathcal{S}_{uv}$,

$$f_{uv}(S) := \text{mst}(S \cup u) + \text{mst}(S \cup v) - \text{mst}(S) - \text{mst}(S \cup \{u, v\}) \geq 0$$

- For $S \subseteq V$, $\text{mst}(S) :=$ cost of a minimum spanning tree in $G[S]$.
- For $u, v \in N$, $\mathcal{S}_{uv} := \{S \subseteq V : r \in S \text{ and } u, v \notin S\}$.
- Sort the edge weights $w_1 < w_2 < \cdots < w_k$.

Preliminaries

Def. An instance is submodular if for any $u, v \in N$ and $S \subseteq \mathcal{S}_{uv}$,

$$f_{uv}(S) := \mathrm{mst}(S \cup u) + \mathrm{mst}(S \cup v) - \mathrm{mst}(S) - \mathrm{mst}(S \cup \{u, v\}) \geq 0$$

- For $S \subseteq V$, $\mathrm{mst}(S) :=$ cost of a minimum spanning tree in $G[S]$.
- For $u, v \in N$, $\mathcal{S}_{uv} := \{S \subseteq V : r \in S \text{ and } u, v \notin S\}$.
- Sort the edge weights $w_1 < w_2 < \cdots < w_k$.
- Define the subgraph $G_i := (V, E_i)$ where $E_i = \{e \in E : w(e) \leq w_i\}$.

Preliminaries

Def. An instance is submodular if for any $u, v \in N$ and $S \subseteq \mathcal{S}_{uv}$,

$$f_{uv}(S) := \text{mst}(S \cup u) + \text{mst}(S \cup v) - \text{mst}(S) - \text{mst}(S \cup \{u, v\}) \geq 0$$

- For $S \subseteq V$, $\text{mst}(S) :=$ cost of a minimum spanning tree in $G[S]$.
- For $u, v \in N$, $\mathcal{S}_{uv} := \{S \subseteq V : r \in S \text{ and } u, v \notin S\}$.
- Sort the edge weights $w_1 < w_2 < \cdots < w_k$.
- Define the subgraph $G_i := (V, E_i)$ where $E_i = \{e \in E : w(e) \leq w_i\}$.

Def. The expensive neighborhood of an edge uv is

$$\hat{N}(uv) := \{s \in V : w(su) > w(uv) \text{ or } w(sv) > w(uv)\}.$$

Preliminaries

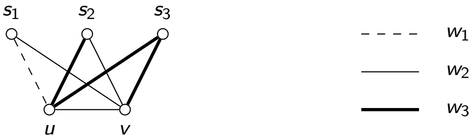
Def. An instance is submodular if for any $u, v \in N$ and $S \subseteq \mathcal{S}_{uv}$,

$$f_{uv}(S) := \text{mst}(S \cup u) + \text{mst}(S \cup v) - \text{mst}(S) - \text{mst}(S \cup \{u, v\}) \geq 0$$

- For $S \subseteq V$, $\text{mst}(S) :=$ cost of a minimum spanning tree in $G[S]$.
- For $u, v \in N$, $\mathcal{S}_{uv} := \{S \subseteq V : r \in S \text{ and } u, v \notin S\}$.
- Sort the edge weights $w_1 < w_2 < \cdots < w_k$.
- Define the subgraph $G_i := (V, E_i)$ where $E_i = \{e \in E : w(e) \leq w_i\}$.

Def. The expensive neighborhood of an edge uv is

$$\hat{N}(uv) := \{s \in V : w(su) > w(uv) \text{ or } w(sv) > w(uv)\}.$$



Preliminaries

Def. An instance is submodular if for any $u, v \in N$ and $S \subseteq \mathcal{S}_{uv}$,

$$f_{uv}(S) := \mathrm{mst}(S \cup u) + \mathrm{mst}(S \cup v) - \mathrm{mst}(S) - \mathrm{mst}(S \cup \{u, v\}) \geq 0$$

- For $S \subseteq V$, $\mathrm{mst}(S)$:= cost of a minimum spanning tree in $G[S]$.
- For $u, v \in N$, $\mathcal{S}_{uv} := \{S \subseteq V : r \in S \text{ and } u, v \notin S\}$.
- Sort the edge weights $w_1 < w_2 < \cdots < w_k$.
- Define the subgraph $G_i := (V, E_i)$ where $E_i = \{e \in E : w(e) \leq w_i\}$.

Def. The expensive neighborhood of an edge uv is

$$\hat{N}(uv) := \{s \in V : w(su) > w(uv) \text{ or } w(sv) > w(uv)\}.$$

Main result

Theorem: The spanning tree game on G is submodular if and only if:

1. There are no violated cycles in G_i for all $i < k$.
2. For every candidate edge uv, $f_{uv}(\hat{N}(uv)) \geq 0$.

Furthermore, these conditions can be verified in polynomial time.

First step

First step

- Submodularity characterization for $k = 2$:

> *The vertices of every cycle in G_1 are adjacent to r or pairwise adjacent.*

First step

- Submodularity characterization for $k = 2$:

  The vertices of every cycle in G_1 are adjacent to r or pairwise adjacent.

- A natural extension:

  The vertices of every cycle in G_i are adjacent to r or pairwise adjacent, for all $i < k$.

First step

- Submodularity characterization for $k = 2$:

> *The vertices of every cycle in G_1 are adjacent to r or pairwise adjacent.*

- A natural extension:

> *The vertices of every cycle in G_i are adjacent to r or pairwise adjacent, for all $i < k$.*

- This condition is too strong.

First step

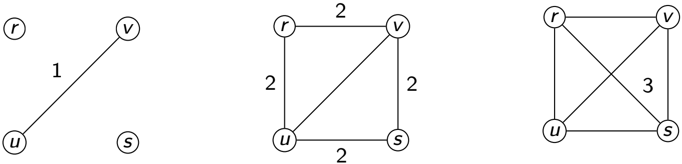
- Submodularity characterization for $k = 2$:

The vertices of every cycle in G_1 are adjacent to r or pairwise adjacent.

- A natural extension:

The vertices of every cycle in G_i are adjacent to r or pairwise adjacent, for all $i < k$.

- This condition is too strong.



G_1 G_2 $G_3 = G$

First step

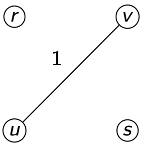
- Submodularity characterization for $k = 2$:

> *The vertices of every cycle in G_1 are adjacent to r or pairwise adjacent.*

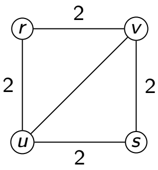
- A natural extension:

> *The vertices of every cycle in G_i are adjacent to r or pairwise adjacent, for all $i < k$.*

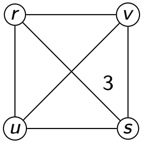
- This condition is too strong.



G_1



G_2



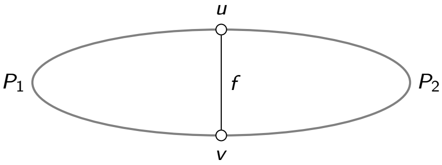
$G_3 = G$

- G_2 violates the condition, yet the instance is submodular.

Violated cycles

Violated cycles

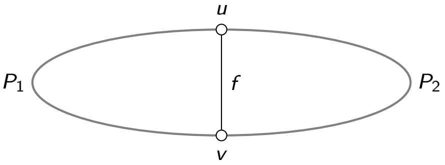
Def. Given a cycle C and a chord $f = uv$, let P_1 and P_2 denote the two u-v paths in C.



Violated cycles

Def. Given a cycle C and a chord $f = uv$, let P_1 and P_2 denote the two u-v paths in C.

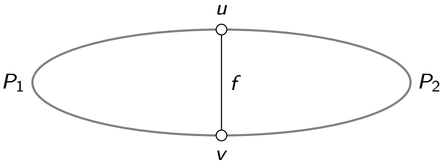
- f covers C if $w(f) \geq w(e)$ for all $e \in E(P_1)$ or $e \in E(P_2)$.



Violated cycles

Def. Given a cycle C and a chord $f = uv$, let P_1 and P_2 denote the two u-v paths in C.

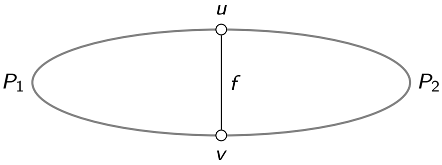
- f covers C if $w(f) \geq w(e)$ for all $e \in E(P_1)$ or $e \in E(P_2)$.
- C is well-covered if it is covered by all of its chords.

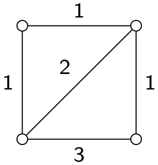


Violated cycles

Def. Given a cycle C and a chord $f = uv$, let P_1 and P_2 denote the two u-v paths in C.

- f covers C if $w(f) \geq w(e)$ for all $e \in E(P_1)$ or $e \in E(P_2)$.
- C is well-covered if it is covered by all of its chords.

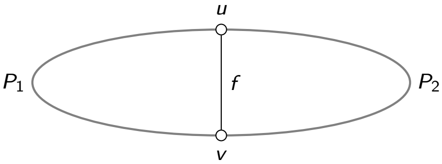


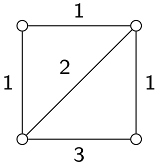


Violated cycles

Def. Given a cycle C and a chord $f = uv$, let P_1 and P_2 denote the two u-v paths in C.

- f covers C if $w(f) \geq w(e)$ for all $e \in E(P_1)$ or $e \in E(P_2)$.
- C is well-covered if it is covered by all of its chords.



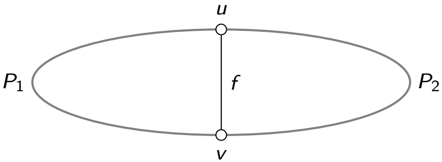


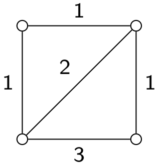
Def. A cycle is violated if it is well-covered but its vertices are neither adjacent to r nor pairwise adjacent.

Violated cycles

Def. Given a cycle C and a chord $f = uv$, let P_1 and P_2 denote the two u-v paths in C.

- f covers C if $w(f) \geq w(e)$ for all $e \in E(P_1)$ or $e \in E(P_2)$.
- C is well-covered if it is covered by all of its chords.





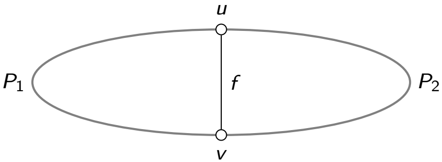
Def. A cycle is violated if it is well-covered but its vertices are neither adjacent to r nor pairwise adjacent.

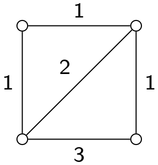
Lemma 1: If the instance is submodular, then there are no **violated cycles** in G_i for all $i < k$.

Violated cycles

Def. Given a cycle C and a chord $f = uv$, let P_1 and P_2 denote the two u-v paths in C.

- f covers C if $w(f) \geq w(e)$ for all $e \in E(P_1)$ or $e \in E(P_2)$.
- C is well-covered if it is covered by all of its chords.





Def. A cycle is violated if it is well-covered but its vertices are neither adjacent to r nor pairwise adjacent.

Lemma 1: If the instance is submodular, then there are no **violated cycles** in G_i for all $i < k$.

- Coincides with [Kobayashi & Okamoto '14] when $k = 2$.

Violated cycles

Lemma 1: If the instance is submodular, then there are no violated cycles in G_i for all $i < k$.

Violated cycles

Lemma 1: If the instance is submodular, then there are no violated cycles in G_i for all $i < k$.

- Recall some basic structures:

Violated cycles

Lemma 1: If the instance is submodular, then there are no violated cycles in G_i for all $i < k$.

- Recall some basic structures:
  - Hole.

Violated cycles

Lemma 1: If the instance is submodular, then there are no violated cycles in G_i for all $i < k$.

- Recall some basic structures:
  - Hole.
  - Diamond. The degree-two vertices are called tips.

Violated cycles

Lemma 1: If the instance is submodular, then there are no violated cycles in G_i for all $i < k$.

- Recall some basic structures:
  - Hole.
  - Diamond. The degree-two vertices are called tips.

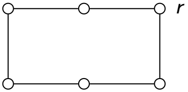
Def. A hole is bad if at least one of its vertices is not adjacent to r.

Violated cycles

Lemma 1: If the instance is submodular, then there are no violated cycles in G_i for all $i < k$.

- Recall some basic structures:
  - Hole.
  - Diamond. The degree-two vertices are called tips.

Def. A hole is bad if at least one of its vertices is not adjacent to r.

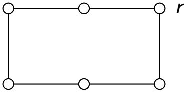


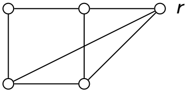
Violated cycles

Lemma 1: If the instance is submodular, then there are no violated cycles in G_i for all $i < k$.

- Recall some basic structures:
  - Hole.
  - Diamond. The degree-two vertices are called tips.

Def. A hole is bad if at least one of its vertices is not adjacent to r.





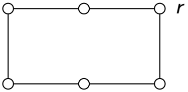
Violated cycles

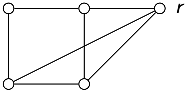
Lemma 1: If the instance is submodular, then there are no violated cycles in G_i for all $i < k$.

- Recall some basic structures:
  - Hole.
  - Diamond. The degree-two vertices are called tips.

Def. A hole is bad if at least one of its vertices is not adjacent to r.

Def. An induced diamond is bad if its hamiltonian cycle is well-covered but at least one of its tips is not adjacent to r.





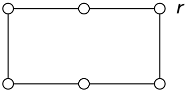
Violated cycles

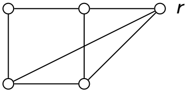
Lemma 1: If the instance is submodular, then there are no violated cycles in G_i for all $i < k$.

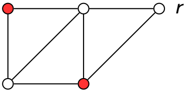
- Recall some basic structures:
  - Hole.
  - Diamond. The degree-two vertices are called tips.

Def. A hole is bad if at least one of its vertices is not adjacent to r.

Def. An induced diamond is bad if its hamiltonian cycle is well-covered but at least one of its tips is not adjacent to r.







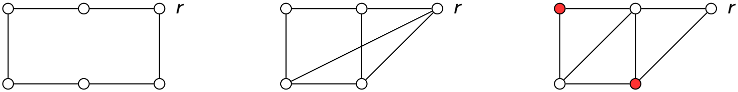
Violated cycles

Lemma 1: If the instance is submodular, then there are no violated cycles in G_i for all $i < k$.

- Recall some basic structures:
  - Hole.
  - Diamond. The degree-two vertices are called tips.

Def. A hole is bad if at least one of its vertices is not adjacent to r.

Def. An induced diamond is bad if its hamiltonian cycle is well-covered but at least one of its tips is not adjacent to r.



Lemma 2: If the instance is submodular, then there are no bad holes or bad induced diamonds in G_i for all $i < k$.

Violated cycles

Lemma 1: If the instance is submodular, then there are no violated cycles in G_i for all $i < k$.

Violated cycles

Lemma 1: If the instance is submodular, then there are no violated cycles in G_i for all $i < k$.

Proof: Contrapositive.

Violated cycles

Lemma 1: If the instance is submodular, then there are no violated cycles in G_i for all $i < k$.

Proof: Contrapositive.

- Let j be the smallest integer such that G_j has a violated cycle.

Violated cycles

Lemma 1: If the instance is submodular, then there are no violated cycles in G_i for all $i < k$.

Proof: Contrapositive.

- Let j be the smallest integer such that G_j has a violated cycle.
- Pick the smallest violated cycle C in G_j.

Violated cycles

Lemma 1: If the instance is submodular, then there are no violated cycles in G_i for all $i < k$.

Proof: Contrapositive.

- Let j be the smallest integer such that G_j has a violated cycle.
- Pick the smallest violated cycle C in G_j.
- We may assume that C has a chord.

Violated cycles

Lemma 1: If the instance is submodular, then there are no violated cycles in G_i for all $i < k$.

Proof: Contrapositive.

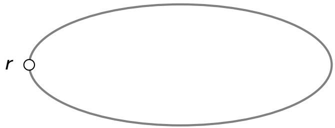
- Let j be the smallest integer such that G_j has a violated cycle.
- Pick the smallest violated cycle C in G_j.
- We may assume that C has a chord.
- **Claim:** The subcycles of C formed by any chord are well-covered.

Violated cycles

Lemma 1: If the instance is submodular, then there are no violated cycles in G_i for all $i < k$.

Proof: Contrapositive.

- Let j be the smallest integer such that G_j has a violated cycle.
- Pick the smallest violated cycle C in G_j.
- We may assume that C has a chord.
- **Claim:** The subcycles of C formed by any chord are well-covered.
- Suppose $r \in V(C)$.

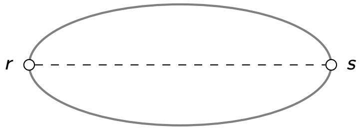


Violated cycles

Lemma 1: If the instance is submodular, then there are no violated cycles in G_i for all $i < k$.

Proof: Contrapositive.

- Let j be the smallest integer such that G_j has a violated cycle.
- Pick the smallest violated cycle C in G_j.
- We may assume that C has a chord.
- **Claim:** The subcycles of C formed by any chord are well-covered.
- Suppose $r \in V(C)$.

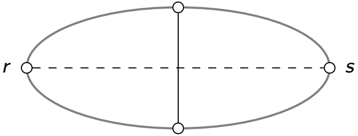


Violated cycles

Lemma 1: If the instance is submodular, then there are no violated cycles in G_i for all $i < k$.

Proof: Contrapositive.

- Let j be the smallest integer such that G_j has a violated cycle.
- Pick the smallest violated cycle C in G_j.
- We may assume that C has a chord.
- **Claim:** The subcycles of C formed by any chord are well-covered.
- Suppose $r \in V(C)$.

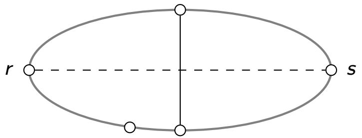


Violated cycles

Lemma 1: If the instance is submodular, then there are no violated cycles in G_i for all $i < k$.

Proof: Contrapositive.

- Let j be the smallest integer such that G_j has a violated cycle.
- Pick the smallest violated cycle C in G_j.
- We may assume that C has a chord.
- **Claim:** The subcycles of C formed by any chord are well-covered.
- Suppose $r \in V(C)$.

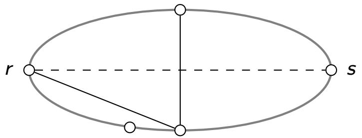


Violated cycles

Lemma 1: If the instance is submodular, then there are no violated cycles in G_i for all $i < k$.

Proof: Contrapositive.

- Let j be the smallest integer such that G_j has a violated cycle.
- Pick the smallest violated cycle C in G_j.
- We may assume that C has a chord.
- **Claim:** The subcycles of C formed by any chord are well-covered.
- Suppose $r \in V(C)$.

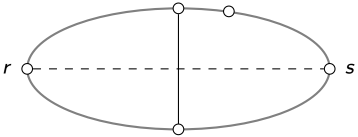


Violated cycles

Lemma 1: If the instance is submodular, then there are no violated cycles in G_i for all $i < k$.

Proof: Contrapositive.

- Let j be the smallest integer such that G_j has a violated cycle.
- Pick the smallest violated cycle C in G_j.
- We may assume that C has a chord.
- **Claim:** The subcycles of C formed by any chord are well-covered.
- Suppose $r \in V(C)$.

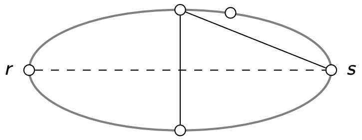


Violated cycles

Lemma 1: If the instance is submodular, then there are no violated cycles in G_i for all $i < k$.

Proof: Contrapositive.

- Let j be the smallest integer such that G_j has a violated cycle.
- Pick the smallest violated cycle C in G_j.
- We may assume that C has a chord.
- **Claim:** The subcycles of C formed by any chord are well-covered.
- Suppose $r \in V(C)$.

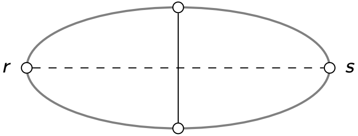


Violated cycles

Lemma 1: If the instance is submodular, then there are no violated cycles in G_i for all $i < k$.

Proof: Contrapositive.

- Let j be the smallest integer such that G_j has a violated cycle.
- Pick the smallest violated cycle C in G_j.
- We may assume that C has a chord.
- **Claim:** The subcycles of C formed by any chord are well-covered.
- Suppose $r \in V(C)$.



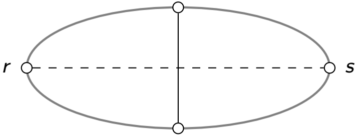
- C is a bad induced diamond. (We skip the case $r \notin V(C)$.) ■

Violated cycles

Lemma 1: If the instance is submodular, then there are no violated cycles in G_i for all $i < k$.

Proof: Contrapositive.

- Let j be the smallest integer such that G_j has a violated cycle.
- Pick the smallest violated cycle C in G_j.
- We may assume that C has a chord.
- **Claim:** The subcycles of C formed by any chord are well-covered.
- Suppose $r \in V(C)$.



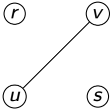
- C is a bad induced diamond. (We skip the case $r \notin V(C)$.) ■

Remark: Violated cycles can be detected in polynomial time.

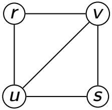
Is this sufficient?

Is this sufficient?

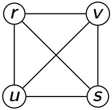
- Recall our example:



G_1



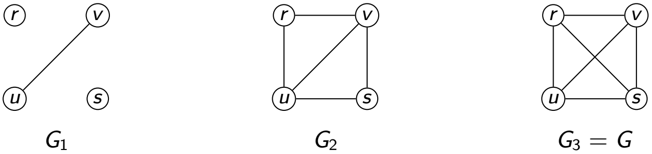
G_2



$G_3 = G$

Is this sufficient?

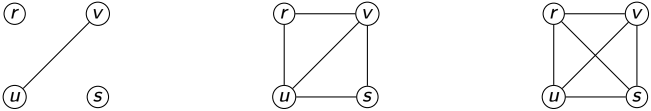
- Recall our example:



$$f_{uv}(\{r,s\}) = \mathrm{mst}(\{r,s,u\}) + \mathrm{mst}(\{r,s,v\}) - \mathrm{mst}(\{r,s\}) - \mathrm{mst}(\{r,s,u,v\})$$

Is this sufficient?

- Recall our example:

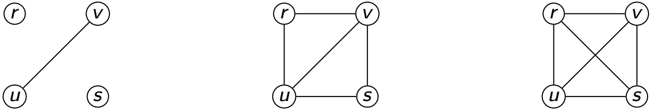


G_1 G_2 $G_3 = G$

$$
\begin{aligned}
f_{uv}(\{r,s\}) &= \mathrm{mst}(\{r,s,u\}) + \mathrm{mst}(\{r,s,v\}) - \mathrm{mst}(\{r,s\}) - \mathrm{mst}(\{r,s,u,v\}) \\
&= 2w_2 + 2w_2 - w_3 - (w_1 + 2w_2)
\end{aligned}
$$

Is this sufficient?

- Recall our example:

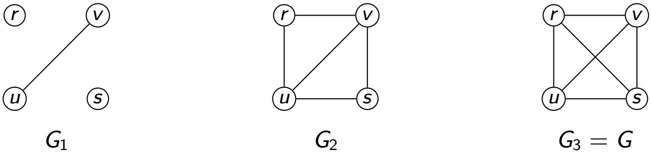


G_1 $\qquad$ G_2 $\qquad$ $G_3 = G$

$$\begin{aligned}
f_{uv}(\{r,s\}) &= \mathrm{mst}(\{r,s,u\}) + \mathrm{mst}(\{r,s,v\}) - \mathrm{mst}(\{r,s\}) - \mathrm{mst}(\{r,s,u,v\}) \\
&= 2w_2 + 2w_2 - w_3 - (w_1 + 2w_2) \\
&= 2w_2 - w_1 - w_3
\end{aligned}$$

Is this sufficient?

- Recall our example:



$$
\begin{aligned}
f_{uv}(\{r,s\}) &= \mathrm{mst}(\{r,s,u\}) + \mathrm{mst}(\{r,s,v\}) - \mathrm{mst}(\{r,s\}) - \mathrm{mst}(\{r,s,u,v\}) \\
&= 2w_2 + 2w_2 - w_3 - (w_1 + 2w_2) \\
&= 2w_2 - w_1 - w_3 \longleftarrow \textit{can be made negative}
\end{aligned}
$$

Candidate edge

Candidate edge

- Denote $(\star)$ as the following property:

$$\boxed{\textit{There are no violated cycles in } G_i \textit{ for all } i < k.}$$

Candidate edge

- Denote $(\star)$ as the following property:

$$\boxed{\textit{There are no violated cycles in } G_i \textit{ for all } i < k.}$$

Lemma 4*: If $S \not\subseteq \hat{N}_{uv}$, then $f_{uv}(S) = 0$.

Candidate edge

- Denote $(\star)$ as the following property:

$$\boxed{\textit{There are no violated cycles in } G_i \textit{ for all } i < k.}$$

Lemma 4*: If $S \not\subseteq \hat{N}_{uv}$, then $f_{uv}(S) = 0$.

- If $r \notin \hat{N}(uv)$, then $S \not\subseteq \hat{N}(uv)$ for all $S \in \mathcal{S}_{uv}$.

Candidate edge

- Denote $(\star)$ as the following property:

$$\boxed{\textit{There are no violated cycles in } G_i \textit{ for all } i < k.}$$

Lemma 4*: If $S \not\subseteq \hat{N}_{uv}$, then $f_{uv}(S) = 0$.

- If $r \notin \hat{N}(uv)$, then $S \not\subseteq \hat{N}(uv)$ for all $S \in \mathcal{S}_{uv}$.
- So we can skip these edges!

Candidate edge

- Denote $(\star)$ as the following property:

$$\boxed{\textit{There are no violated cycles in } G_i \textit{ for all } i < k.}$$

Lemma 4*: If $S \not\subseteq \hat{N}_{uv}$, then $f_{uv}(S) = 0$.

- If $r \notin \hat{N}(uv)$, then $S \not\subseteq \hat{N}(uv)$ for all $S \in \mathcal{S}_{uv}$.
- So we can skip these edges!

Def. An edge uv is a candidate edge if $r \in \hat{N}(uv)$.

Candidate edge

- Denote $(\star)$ as the following property:

$$\boxed{\textit{There are no violated cycles in } G_i \textit{ for all } i < k.}$$

Lemma 4*: If $S \not\subseteq \hat{N}_{uv}$, then $f_{uv}(S) = 0$.

- If $r \notin \hat{N}(uv)$, then $S \not\subseteq \hat{N}(uv)$ for all $S \in \mathcal{S}_{uv}$.
- So we can skip these edges!

Def. An edge uv is a candidate edge if $r \in \hat{N}(uv)$.

Lemma 5*: Assume $f_{uv}(\hat{N}(uv)) \geq 0$ for every candidate edge uv. Then, f_{uv} is inclusion-wise nonincreasing in $\hat{N}(uv)$ for every candidate edge uv.

Putting it all together

Putting it all together

Theorem: The spanning tree game on G is submodular if and only if:

1. There are no violated cycles in G_i for all $i < k$.
2. For every candidate edge uv, $f_{uv}(\hat{N}(uv)) \geq 0$.

Furthermore, these conditions can be verified in polynomial time.

Putting it all together

Theorem: The spanning tree game on G is submodular if and only if:

1. There are no violated cycles in G_i for all $i < k$.
2. For every candidate edge uv, $f_{uv}(\hat{N}(uv)) \geq 0$.

Furthermore, these conditions can be verified in polynomial time.

Proof:
($\Rightarrow$) Assume the game is submodular.

Putting it all together

Theorem: The spanning tree game on G is submodular if and only if:

1. There are no violated cycles in G_i for all $i < k$.
2. For every candidate edge uv, $f_{uv}(\hat{N}(uv)) \geq 0$.

Furthermore, these conditions can be verified in polynomial time.

Proof:

($\Rightarrow$) Assume the game is submodular.

- Condition 1 is satisfied by Lemma 1.

Putting it all together

Theorem: The spanning tree game on G is submodular if and only if:

1. There are no violated cycles in G_i for all $i < k$.
2. For every candidate edge uv, $f_{uv}(\hat{N}(uv)) \geq 0$.

Furthermore, these conditions can be verified in polynomial time.

Proof:

($\Rightarrow$) Assume the game is submodular.

- Condition 1 is satisfied by Lemma 1.
- Condition 2 is satisfied trivially.

Putting it all together

Theorem: The spanning tree game on G is submodular if and only if:

1. There are no violated cycles in G_i for all $i < k$.
2. For every candidate edge uv, $f_{uv}(\hat{N}(uv)) \geq 0$.

Furthermore, these conditions can be verified in polynomial time.

Proof:

($\Rightarrow$) Assume the game is submodular.

- Condition 1 is satisfied by Lemma 1.
- Condition 2 is satisfied trivially.

($\Leftarrow$) Assume Conditions 1 and 2 are satisfied.

Putting it all together

Theorem: The spanning tree game on G is submodular if and only if:

1. There are no violated cycles in G_i for all $i < k$.
2. For every candidate edge uv, $f_{uv}(\hat{N}(uv)) \geq 0$.

Furthermore, these conditions can be verified in polynomial time.

Proof:

($\Rightarrow$) Assume the game is submodular.

- Condition 1 is satisfied by Lemma 1.
- Condition 2 is satisfied trivially.

($\Leftarrow$) Assume Conditions 1 and 2 are satisfied.

- Let $u, v \in N$ and $S \in \mathcal{S}_{uv}$.

Putting it all together

Theorem: The spanning tree game on G is submodular if and only if:

1. There are no violated cycles in G_i for all $i < k$.
2. For every candidate edge uv, $f_{uv}(\hat{N}(uv)) \geq 0$.

Furthermore, these conditions can be verified in polynomial time.

Proof:

($\Rightarrow$) Assume the game is submodular.

- Condition 1 is satisfied by Lemma 1.
- Condition 2 is satisfied trivially.

($\Leftarrow$) Assume Conditions 1 and 2 are satisfied.

- Let $u, v \in N$ and $S \in \mathcal{S}_{uv}$.
- If $S \not\subseteq \hat{N}(uv)$, then $f_{uv}(S) = 0$ by Lemma 4.

Putting it all together

Theorem: The spanning tree game on G is submodular if and only if:

1. There are no violated cycles in G_i for all $i < k$.
2. For every candidate edge uv, $f_{uv}(\hat{N}(uv)) \geq 0$.

Furthermore, these conditions can be verified in polynomial time.

Proof:

($\Rightarrow$) Assume the game is submodular.

- Condition 1 is satisfied by Lemma 1.
- Condition 2 is satisfied trivially.

($\Leftarrow$) Assume Conditions 1 and 2 are satisfied.

- Let $u, v \in N$ and $S \in \mathcal{S}_{uv}$.
- If $S \not\subseteq \hat{N}(uv)$, then $f_{uv}(S) = 0$ by Lemma 4.
- If $S \subseteq \hat{N}(uv)$, then $f_{uv}(S) \geq f_{uv}(\hat{N}(uv)) \geq 0$ by Lemma 5. ∎

Thank you!