On the Complexity of Nucleolus Computation for Bipartite b-Matching Games

Jochen Könemann, Justin Toth, Felix Zhou

Cooperative Game Theory

- Setting: Finite collection N of players, any subset $S \subseteq N$ can collaborate to generate revenue.

Cooperative Game Theory

- Setting: Finite collection N of players, any subset $S \subseteq N$ can collaborate to generate revenue.
- *Cooperative game:* $\Gamma = (N, \nu)$.

Cooperative Game Theory

- Setting: Finite collection N of players, any subset $S \subseteq N$ can collaborate to generate revenue.
- *Cooperative game:* $\Gamma = (N, \nu)$.
- *Player set:* $N = \{1, 2, \ldots, n\}$.

Cooperative Game Theory

- Setting: Finite collection N of players, any subset $S \subseteq N$ can collaborate to generate revenue.
- *Cooperative game:* $\Gamma = (N, \nu)$.
- *Player set:* $N = \{1, 2, \ldots, n\}$.
- *Characteristic function:* $\nu : 2^N \to \mathbb{R}$ with $\nu(\varnothing) = 0$.

Cooperative Game Theory

- Setting: Finite collection N of players, any subset $S \subseteq N$ can collaborate to generate revenue.
- *Cooperative game:* $\Gamma = (N, \nu)$.
- *Player set:* $N = \{1, 2, \ldots, n\}$.
- *Characteristic function:* $\nu : 2^N \to \mathbb{R}$ with $\nu(\varnothing) = 0$.
  - $\nu(S)$ is revenue of coalition S.

Cooperative Game Theory

- What sort of coalitions will form?

Cooperative Game Theory

- What sort of coalitions will form?
- How will the total revenue be shared?

Cooperative Game Theory

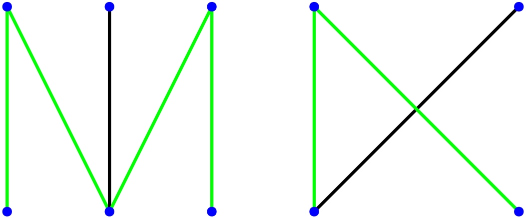
- What sort of coalitions will form?
- How will the total revenue be shared?
- *Allocation:* $x \in \mathbb{R}^N : x(N) = \nu(N)$.

Cooperative Game Theory

- What sort of coalitions will form?
- How will the total revenue be shared?
- *Allocation:* $x \in \mathbb{R}^N : x(N) = \nu(N)$.
- *Imputation:* Subset of allocations such that $x(i) \geq \nu(\{i\})$ for all $i \in N$.

(Non-simple) b-Matching Problem

- Graph $G = (N, E)$.

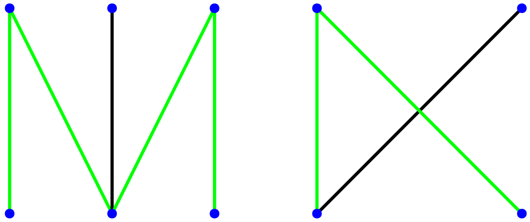


$w \equiv 1$

$b \equiv 2$

(Non-simple) b-Matching Problem

- Graph $G = (N, E)$.
- Edge weights $w : E \to \mathbb{R}$.

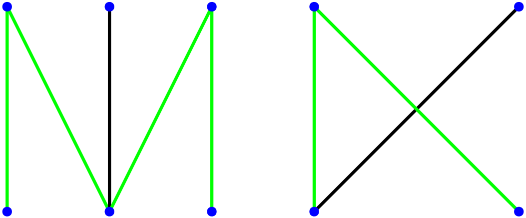


$w \equiv 1$

$b \equiv 2$

(Non-simple) b-Matching Problem

- Graph $G = (N, E)$.
- Edge weights $w : E \to \mathbb{R}$.
- Vertex-incidence capacity $b : N \to \mathbb{Z}_+$.

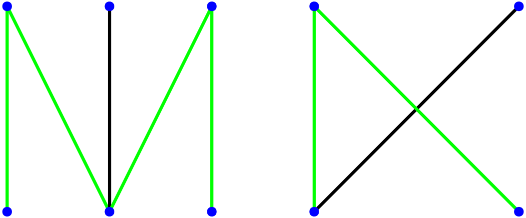


$w \equiv 1$

$b \equiv 2$

(Non-simple) b-Matching Problem

- Graph $G = (N, E)$.
- Edge weights $w : E \to \mathbb{R}$.
- Vertex-incidence capacity $b : N \to \mathbb{Z}_+$.
- Find a (multi-)set of edges M maximizing $w(M)$ such that each $v \in N$ is in incident to at most b_v edges of M.

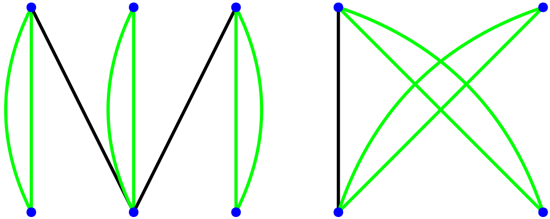


$w \equiv 1$

$b \equiv 2$

(Non-simple) b-Matching Problem

- Graph $G = (N, E)$.
- Edge weights $w : E \to \mathbb{R}$.
- Vertex-incidence capacity $b : N \to \mathbb{Z}_+$.
- Find a (multi-)set of edges M maximizing $w(M)$ such that each $v \in N$ is in incident to at most b_v edges of M.

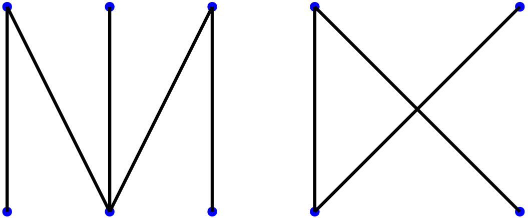


$w \equiv 1$

$b \equiv 2$

b-Matching Games

- Instance of b-Matching: G, w, b.



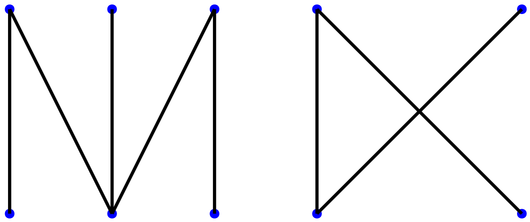
$\nu = 4$ $\nu = 2$

$w \equiv 1$

$b \equiv 2$

b-Matching Games

- Instance of b-Matching: G, w, b.
- Players: Vertices.



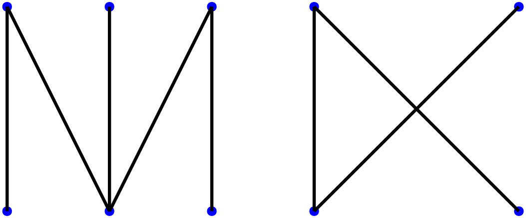
$\nu = 4$ $\nu = 2$

$w \equiv 1$

$b \equiv 2$

b-Matching Games

- Instance of b-Matching: G, w, b.
- Players: Vertices.
- Characteristic Function: $\nu(S)$ is the weight of a maximum weight b-matching in $G[S]$.



$\nu = 4$ $\quad$ $\nu = 2$

$w \equiv 1$

$b \equiv 2$

Core

- *Excess:* $e(S, x) := x(S) - \nu(S)$.

Core

- *Excess:* $e(S, x) := x(S) - \nu(S)$.
- "Satisfaction" of a coalition with respect to x.

Core

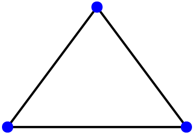
- *Excess:* $e(S, x) := x(S) - \nu(S)$.
- "Satisfaction" of a coalition with respect to x.
- Imputations: Non-negative singleton excess.

Core

- *Excess:* $e(S, x) := x(S) - \nu(S)$.
- "Satisfaction" of a coalition with respect to x.
- Imputations: Non-negative singleton excess.
- *Core:* Subset of imputations such that $e(S, x) \geq 0$ for all $S \subseteq N$.

Core

- 2012; Biro, Kern, Paulusma: Stable matchings with payments (variant of stable marriage problem) correspond to core allocations.

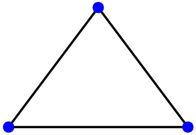


$w \equiv 1$

$b \equiv 1$

Core

- 2012; Biro, Kern, Paulusma: Stable matchings with payments (variant of stable marriage problem) correspond to core allocations.
- 1999; Deng, Ibaraki, Nagamochi: The core is non-empty in bipartite b-matching games.

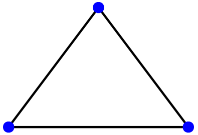


$w \equiv 1$

$b \equiv 1$

Core

- 2012; Biro, Kern, Paulusma: Stable matchings with payments (variant of stable marriage problem) correspond to core allocations.
- 1999; Deng, Ibaraki, Nagamochi: The core is non-empty in bipartite b-matching games.
  - The core of a combinatorial optimization game is non-empty if and only if the fractional LP of the underlying optimization problem has integral optimal solutions.

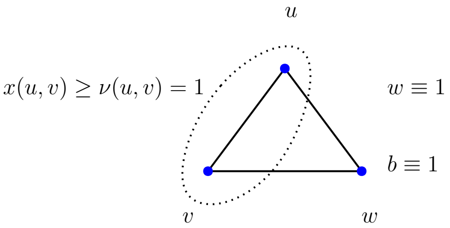


$w \equiv 1$

$b \equiv 1$

Core

- 2012; Biro, Kern, Paulusma: Stable matchings with payments (variant of stable marriage problem) correspond to core allocations.
- 1999; Deng, Ibaraki, Nagamochi: The core is non-empty in bipartite b-matching games.
  - The core of a combinatorial optimization game is non-empty if and only if the fractional LP of the underlying optimization problem has integral optimal solutions.
- The core can be empty, even for 1-matching games.



Nucleolus

- Alternative definition of "fairness"?

$$\begin{aligned} & \max \epsilon && LP_1 \\ x(N) &= \nu(N) && \\ x(S) &\geq \nu(S) + \epsilon && \forall \varnothing \neq S \subsetneq N \end{aligned}$$

Nucleolus

- Alternative definition of “fairness”?
- Idea: Maximize the satisfaction among the worst-case coalitions.

$$
\begin{aligned}
&\max \epsilon && LP_1 \\
x(N) &= \nu(N) && \\
x(S) &\geq \nu(S) + \epsilon && \forall \varnothing \neq S \subsetneq N
\end{aligned}
$$

Nucleolus

- Why stop at the worst-case coalitions?

Nucleolus

- Why stop at the worst-case coalitions?
- $\Theta(x) \in \mathbb{R}^{2^n-2}$: Entries are $e(S,x), \varnothing \neq S \subset N$, sorted in non-decreasing order.

Nucleolus

- Why stop at the worst-case coalitions?
- $\Theta(x) \in \mathbb{R}^{2^n-2}$: Entries are $e(S,x), \varnothing \neq S \subset N$, sorted in non-decreasing order.
- *Nucleolus:* (Unique) imputation maximizing $\Theta(x)$ lexicographically.

Nucleolus

- The nucleolus always exists.

Nucleolus

- The nucleolus always exists.
- The nucleolus is unique.

Nucleolus

- The nucleolus always exists.
- The nucleolus is unique.
- If core is non-empty, nucleolus is a member of the core.

Kopelowitz Scheme

- How can we compute the nucleolus?

$$LP_1 : \max \epsilon$$

$$x(N) = \nu(N)$$

$$x(S) \geq \nu(S) + \epsilon \quad \forall \varnothing \neq S \subsetneq N$$

Kopelowitz Scheme

- How can we compute the nucleolus?
- Idea: Solve a sequence of recursively defined linear programs $LP_k, k \geq 1$.

$$LP_1 : \max \epsilon$$

$$x(N) = \nu(N)$$

$$x(S) \geq \nu(S) + \epsilon \quad \forall \varnothing \neq S \subsetneq N$$

Kopelowitz Scheme

- How can we compute the nucleolus?
- Idea: Solve a sequence of recursively defined linear programs $LP_k, k \geq 1$.
- Tight coalitions $\mathcal{J}_k \subseteq 2^N$: For all optimal solutions (x, ϵ_k) of LP_k, $x(S) = \nu(S) + \epsilon_k$.

$$LP_1 : \max \epsilon$$

$$x(N) = \nu(N)$$

$$x(S) \geq \nu(S) + \epsilon \quad \forall \varnothing \neq S \subsetneq N$$

Kopelowitz Scheme

- How can we compute the nucleolus?
- Idea: Solve a sequence of recursively defined linear programs $LP_k, k \geq 1$.
- Tight coalitions $\mathcal{J}_k \subseteq 2^N$: For all optimal solutions (x, ϵ_k) of LP_k, $x(S) = \nu(S) + \epsilon_k$.
- Define LP_{k+1} by fixing new tight constraints.

$$LP_{k+1} : \max \epsilon$$

$$x(S) = \nu(S) + \epsilon_k \quad \forall S \in \mathcal{J}_r, 1 \leq r \leq k$$

$$x(S) \geq \nu(S) + \epsilon \quad \forall S \in 2^N \setminus \bigcup_{r=1}^{k} \mathcal{J}_r$$

Kopelowitz Scheme

- Each LP has $O\left(2^{|N|}\right)$ constraints.

Kopelowitz Scheme

- Each LP has $O\left(2^{|N|}\right)$ constraints.
- At least one coalition is added to $\mathcal{J}_k$ for every k.

Kopelowitz Scheme

- Each LP has $O\left(2^{|N|}\right)$ constraints.
- At least one coalition is added to $\mathcal{J}_k$ for every k.
- Solve $O\left(2^{|N|}\right)$ LPs until solution is unique.

Kopelowitz Scheme

- Each LP has $O\left(2^{|N|}\right)$ constraints.
- At least one coalition is added to $\mathcal{J}_k$ for every k.
- Solve $O\left(2^{|N|}\right)$ LPs until solution is unique.
- Can use Kopelowitz scheme to characterize the nucleolus.

Kopelowitz Scheme

- Each LP has $O\left(2^{|N|}\right)$ constraints.
- At least one coalition is added to $\mathcal{J}_k$ for every k.
- Solve $O\left(2^{|N|}\right)$ LPs until solution is unique.
- Can use Kopelowitz scheme to characterize the nucleolus.
- Maschler's scheme: Variant of Kopelowitz scheme which guarantees termination after $O(|N|)$ iterations.

Main Results

Theorem [Könemann, Toth, Zhou '21]

Deciding whether an allocation is the nucleolus of an unweighted bipartite 3-matching game is NP-hard, even in graphs of maximum degree 7.

Theorem [Könemann, Toth, Zhou '21]

Computing the nucleolus of a bipartite b-matching game is NP-hard, even when $b \leq 3$ and the underlying graph is sparse.

Positive Results

Theorem [**Könemann, Toth, Zhou '21**]

Let $G = (N, E), w$, and $b \leq 2$ be an instance of b-matching. Suppose G has bipartition $N = A \cup B$. Let $k \geq 0$ be a universal constant.

- Suppose $b_v = 2$ for all $v \in A$ but $b_v = 2$ for at most k vertices of B, then the nucleolus of the b-matching game in G is polynomial-time computable.

Positive Results

Theorem [**Könemann, Toth, Zhou ’21**]

Let $G = (N, E), w$, and $b \leq 2$ be an instance of b-matching. Suppose G has bipartition $N = A \cup B$. Let $k \geq 0$ be a universal constant.

- Suppose $b_v = 2$ for all $v \in A$ but $b_v = 2$ for at most k vertices of B, then the nucleolus of the b-matching game in G is polynomial-time computable.
- If $b \equiv 2$, then the nucleolus of the non-simple b-matching game on G is polynomial-time computable.

History & Related Work

1-Matching Games

- 2003; Kern, Paulusma: Posed the question of computing the nucleolus as an open problem.

History & Related Work

1-Matching Games

- 2003; Kern, Paulusma: Posed the question of computing the nucleolus as an open problem.
- 2008; Deng, Fang: Conjectured this problem to be NP-hard.

History & Related Work

1-Matching Games

- 2003; Kern, Paulusma: Posed the question of computing the nucleolus as an open problem.
- 2008; Deng, Fang: Conjectured this problem to be NP-hard.
- 2018; Könemann, Pashkovich, Toth: The nucleolus is computable in polynomial time.

History & Related Work

b-Matching Games

- 2010; Bateni et al: Polynomial-time algorithm to compute the nucleolus in bipartite graphs when one side of the bipartition is restricted to $b_v = 1$.

History & Related Work

b-Matching Games

- 2010; Bateni et al: Polynomial-time algorithm to compute the nucleolus in bipartite graphs when one side of the bipartition is restricted to $b_v = 1$.
- 2018; Biro et al: Testing core membership in bipartite graphs is NP-hard if $b \equiv 3$ and $w \equiv 1$.

History & Related Work

b-Matching Games

- 2010; Bateni et al: Polynomial-time algorithm to compute the nucleolus in bipartite graphs when one side of the bipartition is restricted to $b_v = 1$.
- 2018; Biro et al: Testing core membership in bipartite graphs is NP-hard if $b \equiv 3$ and $w \equiv 1$.
- 2019; Biro et al: Testing core non-emptiness, and thus computing the nucleolus, is NP-hard when $b \leq 3$ and $w \equiv 1$.

History & Related Work

b-Matching Games

- 2010; Bateni et al: Polynomial-time algorithm to compute the nucleolus in bipartite graphs when one side of the bipartition is restricted to $b_v = 1$.
- 2018; Biro et al: Testing core membership in bipartite graphs is NP-hard if $b \equiv 3$ and $w \equiv 1$.
- 2019; Biro et al: Testing core non-emptiness, and thus computing the nucleolus, is NP-hard when $b \leq 3$ and $w \equiv 1$.
  - Proof uses gadget graph with many odd cycles.

History & Related Work

b-Matching Games

- 2010; Bateni et al: Polynomial-time algorithm to compute the nucleolus in bipartite graphs when one side of the bipartition is restricted to $b_v = 1$.
- 2018; Biro et al: Testing core membership in bipartite graphs is NP-hard if $b \equiv 3$ and $w \equiv 1$.
- 2019; Biro et al: Testing core non-emptiness, and thus computing the nucleolus, is NP-hard when $b \leq 3$ and $w \equiv 1$.
  - Proof uses gadget graph with many odd cycles.
  - Supports plausible conjecture that nucleolus is polynomial-time computable for bipartite graphs.

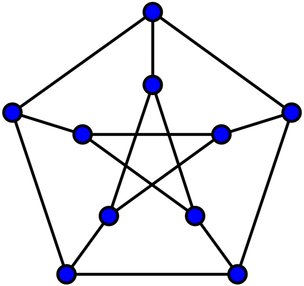
History & Related Work

b-Matching Games

- 2010; Bateni et al: Polynomial-time algorithm to compute the nucleolus in bipartite graphs when one side of the bipartition is restricted to $b_v = 1$.
- 2018; Biro et al: Testing core membership in bipartite graphs is NP-hard if $b \equiv 3$ and $w \equiv 1$.
- 2019; Biro et al: Testing core non-emptiness, and thus computing the nucleolus, is NP-hard when $b \leq 3$ and $w \equiv 1$.
  - Proof uses gadget graph with many odd cycles.
  - Supports plausible conjecture that nucleolus is polynomial-time computable for bipartite graphs.
  - Surprisingly, our work answers this in the negative.

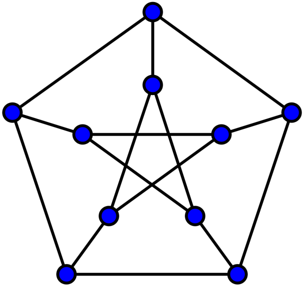
Hardness Proof Overview

- Cubic Subgraph Problem: Given a graph $G = (N, E)$, does it contain a subgraph where each vertex has degree 3?



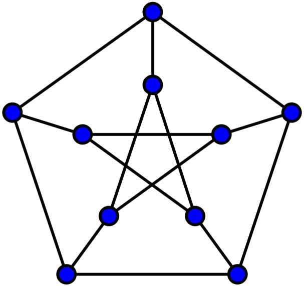
Hardness Proof Overview

- Cubic Subgraph Problem: Given a graph $G = (N, E)$, does it contain a subgraph where each vertex has degree 3?
- 1984; Plesnik: Cubic subgraph is NP-hard even in bipartite planar graphs of maximum degree 4.



Hardness Proof Overview

- Cubic Subgraph Problem: Given a graph $G = (N, E)$, does it contain a subgraph where each vertex has degree 3?
- 1984; Plesnik: Cubic subgraph is NP-hard even in bipartite planar graphs of maximum degree 4.
- Two From Cubic Subgraph Problem: Given a graph $G = (N, E)$, does it contain a subgraph where every vertex has degree 3 *except for two vertices of degree 2*?



Hardness Proof Overview

Theorem [Könemann, Toth, Zhou '21]

Two From Cubic Subgraph is NP-hard even in bipartite graphs of maximum degree 7.

- Builds on Plesnik's proof.

Hardness Proof Overview

Theorem [Könemann, Toth, Zhou '21]

Two From Cubic Subgraph is NP-hard even in bipartite graphs of maximum degree 7.

- Builds on Plesnik's proof.
- Requires significant innovation in the gadget graph.

Hardness Proof Overview

Theorem [Könemann, Toth, Zhou '21]

Two From Cubic Subgraph is NP-hard even in bipartite graphs of maximum degree 7.

- Builds on Plesnik's proof.
- Requires significant innovation in the gadget graph.
- Relies on a piece of graph theory of individual interest.

Hardness Proof Overview

Theorem [Könemann, Toth, Zhou '21]

Two From Cubic Subgraph is NP-hard even in bipartite graphs of maximum degree 7.

- Builds on Plesnik's proof.
- Requires significant innovation in the gadget graph.
- Relies on a piece of graph theory of individual interest.
  - Let X be a regular subgraph of some graph G.

Hardness Proof Overview

Theorem [**Könemann, Toth, Zhou '21**]

Two From Cubic Subgraph is NP-hard even in bipartite graphs of maximum degree 7.

- Builds on Plesnik's proof.
- Requires significant innovation in the gadget graph.
- Relies on a piece of graph theory of individual interest.
  - Let X be a regular subgraph of some graph G.
  - Let Y be a highly vertex-connected subgraph of G.

Hardness Proof Overview

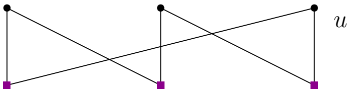
Theorem [Könemann, Toth, Zhou '21]

Two From Cubic Subgraph is NP-hard even in bipartite graphs of maximum degree 7.

- Builds on Plesnik's proof.
- Requires significant innovation in the gadget graph.
- Relies on a piece of graph theory of individual interest.
  - Let X be a regular subgraph of some graph G.
  - Let Y be a highly vertex-connected subgraph of G.
  - "Either $V(Y) \subseteq V(X)$ or $V(Y) \cap V(X) = \varnothing$".

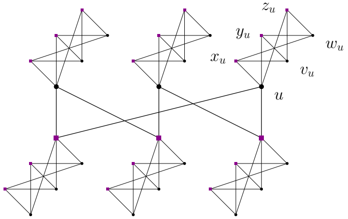
Hardness Proof Overview

- Let $G = (N, E)$ be bipartite instance of two from cubic subgraph.



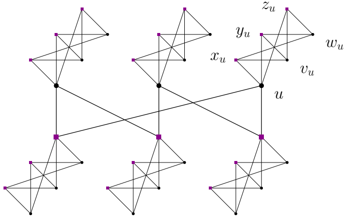
Hardness Proof Overview

- Let $G = (N, E)$ be bipartite instance of two from cubic subgraph.
- Create gadget graph $G^* = (N^*, E^*)$ by "adding a $K_{3,3}$" to every vertex.



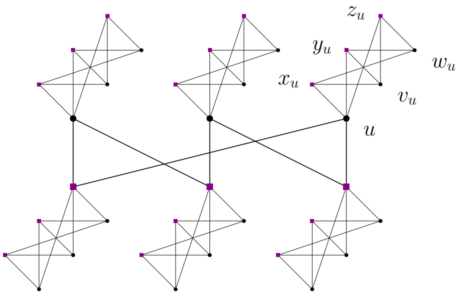
Hardness Proof Overview

- Let $G = (N, E)$ be bipartite instance of two from cubic subgraph.
- Create gadget graph $G^* = (N^*, E^*)$ by "adding a $K_{3,3}$" to every vertex.
- The nucleolus of the unweighted 3-matching game on G^* is "some specific allocation" if and only if G does not contain a two from cubic subgraph.



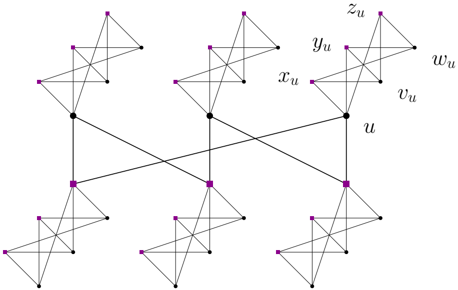
Gadget Graph

- The maximum cardinality 3-matching on G^* has size $\frac{3}{2}|N^*|$.



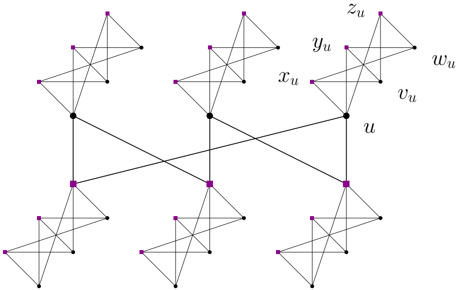
Gadget Graph

- The maximum cardinality 3-matching on G^* has size $\frac{3}{2}|N^*|$.
- G^* remains bipartite, thus the core is non-empty.



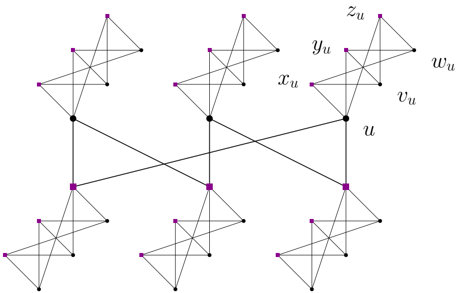
Gadget Graph

- The maximum cardinality 3-matching on G^* has size $\frac{3}{2}|N^*|$.
- G^* remains bipartite, thus the core is non-empty.
- Biro et al. used gadget for hardness of core separation.



Gadget Graph

- The maximum cardinality 3-matching on G^* has size $\frac{3}{2}|N^*|$.
- G^* remains bipartite, thus the core is non-empty.
- Biro et al. used gadget for hardness of core separation.
  - "some allocation" resides in the core of game on G^* if and only if G has no cubic subgraph.



The Reduction

Theorem [**Könemann, Toth, Zhou '21**]

$x \equiv \frac{3}{2}$ is the nucleolus of the 3-matching game on G^* if and only if G does not contain a two from cubic subgraph.

The Reduction

- Let (x^*, ϵ_k) be an optimal solution to each LP_k of Kopelowitz scheme.

$$
\begin{aligned}
&\max \epsilon && LP_k \\
&x(S) = \nu(S) + \epsilon_r && \forall S \in \mathcal{J}_r, 0 \le r \le k-1 \\
&x(S) \ge \nu(S) + \epsilon && \forall S \in \mathcal{J} \setminus \bigcup_{r=0}^{k-1} \mathcal{J}_r
\end{aligned}
$$

The Reduction

- Let (x^*, ϵ_k) be an optimal solution to each LP_k of Kopelowitz scheme.
- If there is no two from cubic subgraph, $\epsilon_1 = 0, \epsilon_2 = \frac{3}{2}$, and $x^* \equiv \frac{3}{2}$ is the unique optimal solution to LP_2.

$$\begin{aligned} &\max \epsilon && LP_k \\ &x(S) = \nu(S) + \epsilon_r && \forall S \in \mathcal{J}_r, 0 \leq r \leq k-1 \\ &x(S) \geq \nu(S) + \epsilon && \forall S \in \mathcal{J} \setminus \bigcup_{r=0}^{k-1} \mathcal{J}_r \end{aligned}$$

The Reduction

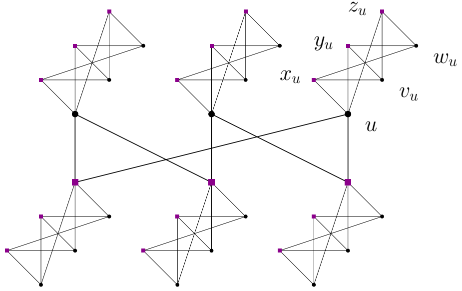
- Let (x^*, ϵ_k) be an optimal solution to each LP_k of Kopelowitz scheme.
- If there is no two from cubic subgraph, $\epsilon_1 = 0, \epsilon_2 = \frac{3}{2}$, and $x^* \equiv \frac{3}{2}$ is the unique optimal solution to LP_2.
- If there is a two from cubic subgraph, $\epsilon_1 = 0$ and $x \equiv \frac{3}{2}$ is not optimal in LP_2.

$$
\begin{array}{ll}
\max \epsilon & LP_k \\
x(S) = \nu(S) + \epsilon_r & \forall S \in \mathcal{J}_r, 0 \leq r \leq k-1 \\
x(S) \geq \nu(S) + \epsilon & \forall S \in \mathcal{J} \setminus \bigcup_{r=0}^{k-1} \mathcal{J}_r
\end{array}
$$

Case I: No Two From Cubic Subgraph

$\epsilon_1 = 0$

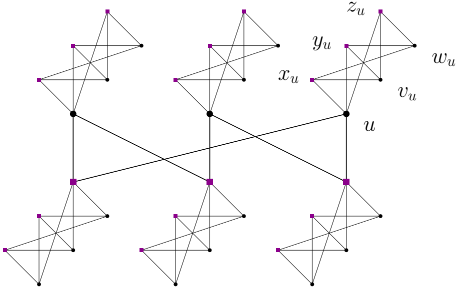
- Core is non-empty for bipartite graphs, so $\epsilon_1 \geq 0$.



Case I: No Two From Cubic Subgraph

$\epsilon_1 = 0$

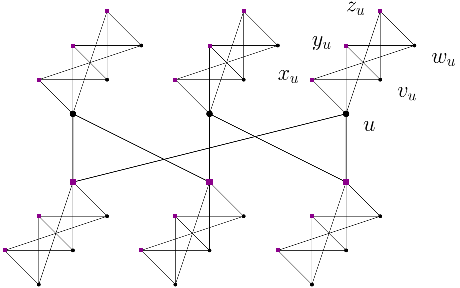
- Core is non-empty for bipartite graphs, so $\epsilon_1 \geq 0$.
- $\epsilon_1 \leq x^*(u, v_u, w_u, x_u, y_u, z_u) - \nu(K_{3,3}) = 0$.



Case I: No Two From Cubic Subgraph

$\epsilon_1 = 0$

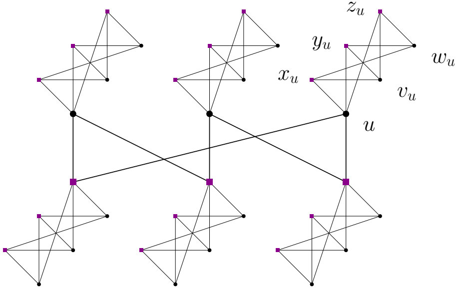
- Core is non-empty for bipartite graphs, so $\epsilon_1 \geq 0$.
- $\epsilon_1 \leq x^*(u, v_u, w_u, x_u, y_u, z_u) - \nu(K_{3,3}) = 0.$
  - $\sum_{u \in N} e(K_{3,3}, x^*) = e(N^*, x^*) = 0.$



Case I: No Two From Cubic Subgraph

$\epsilon_1 = 0$

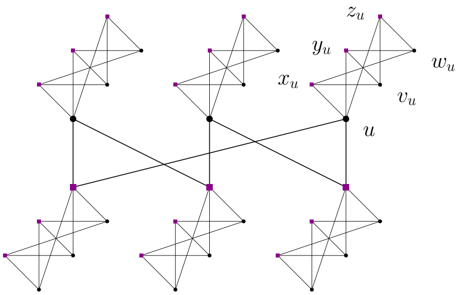
- Core is non-empty for bipartite graphs, so $\epsilon_1 \geq 0$.
- $\epsilon_1 \leq x^*(u, v_u, w_u, x_u, y_u, z_u) - \nu(K_{3,3}) = 0$.
  - $\sum_{u \in N} e(K_{3,3}, x^*) = e(N^*, x^*) = 0$.
- The only coalitions fixed in LP_1 are the union $K_{3,3}$ gadgets.



Case I: No Two From Cubic Subgraph

$\epsilon_2 = \frac{3}{2}$

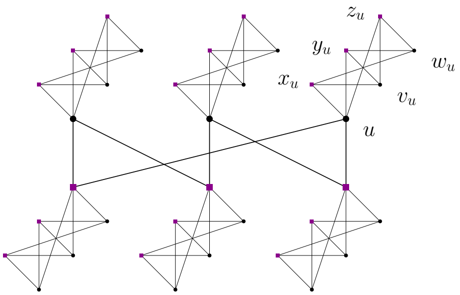
- Since $x(N^*) = \frac{3}{2}|N^*|$, $\epsilon_2 \leq \min_{v \in N^*} x^*(v) \leq \frac{3}{2}$.



Case I: No Two From Cubic Subgraph

$\epsilon_2 = \frac{3}{2}$

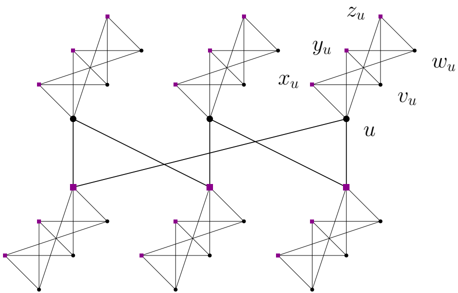
- Since $x(N^*) = \frac{3}{2}|N^*|$, $\epsilon_2 \leq \min_{v \in N^*} x^*(v) \leq \frac{3}{2}$.
- Minimum excess coalitions not fixed in LP_1 contain the singletons and so $\epsilon_2 = \frac{3}{2}$.



Case I: No Two From Cubic Subgraph

$\epsilon_2 = \frac{3}{2}$

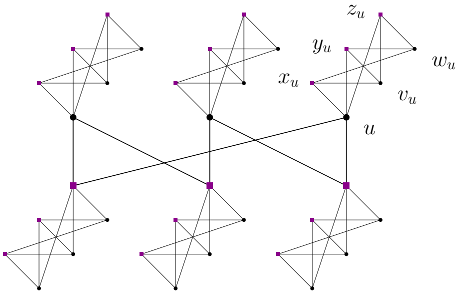
- Since $x(N^*) = \frac{3}{2}|N^*|$, $\epsilon_2 \leq \min_{v \in N^*} x^*(v) \leq \frac{3}{2}$.
- Minimum excess coalitions not fixed in LP_1 contain the singletons and so $\epsilon_2 = \frac{3}{2}$.
  - Uses fact that G does not contain a two from cubic subgraph.



Case II: Contains Two From Cubic Subgraph

Converse when G does contain a two from cubic subgraph is similar.

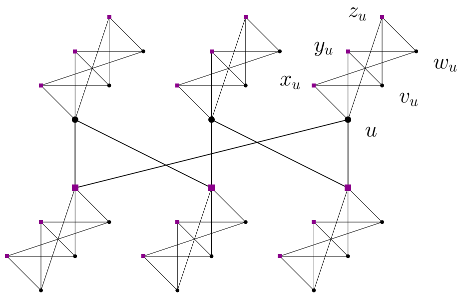
- $\epsilon_1 = 0$



Case II: Contains Two From Cubic Subgraph

Converse when G does contain a two from cubic subgraph is similar.

- $\epsilon_1 = 0$
- Construct allocation which is feasible in LP_2 with strictly greater objective than $x \equiv \frac{3}{2}$.



Positive Results

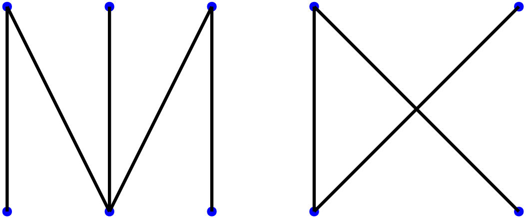
Theorem [**Könemann, Toth, Zhou '21**]

Let $G = (N, E), w$, and $b \leq 2$ be an instance of b-matching. Suppose G has bipartition $N = A \cup B$. Let $k \geq 0$ be a universal constant.

- Suppose $b_v = 2$ for all $v \in A$ but $b_v = 2$ for at most k vertices of B, then the nucleolus of the b-matching game in G is polynomial-time computable.
- If $b \equiv 2$, then the nucleolus of the non-simple b-matching game on G is polynomial-time computable.

Positive Results

- Prune constraints from Kopelowitz scheme which are “not necessary”.



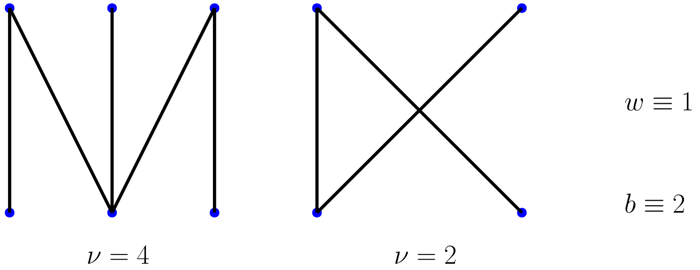
$\nu = 4$ $\nu = 2$

$w \equiv 1$

$b \equiv 2$

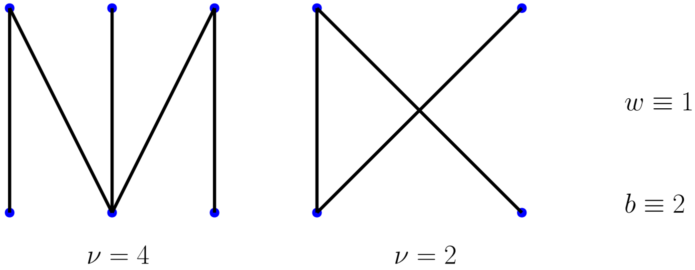
Positive Results

- Prune constraints from Kopelowitz scheme which are “not necessary”.
- If the remaining constraints are polynomial-sized, the nucleolus can be computed in polynomial time.



Positive Results

- Prune constraints from Kopelowitz scheme which are "not necessary".
- If the remaining constraints are polynomial-sized, the nucleolus can be computed in polynomial time.
- If core is non-empty and there is some maximum b-matching of $G[S]$ that is disconnected, S can be omitted.



Simple b-Matching Games

- $b_v = 2$ for all $v \in A$ but $b_v = 2$ for at most k vertices of B.

Simple b-Matching Games

- $b_v = 2$ for all $v \in A$ but $b_v = 2$ for at most k vertices of B.
    - Extension of the work from Bateni et al.

Simple b-Matching Games

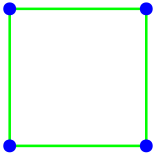
- $b_v = 2$ for all $v \in A$ but $b_v = 2$ for at most k vertices of B.
  - Extension of the work from Bateni et al.
- Then the largest connected component in a b-matching has cardinality at most $2k + 3$.

Simple b-Matching Games

- $b_v = 2$ for all $v \in A$ but $b_v = 2$ for at most k vertices of B.
  - Extension of the work from Bateni et al.
- Then the largest connected component in a b-matching has cardinality at most $2k + 3$.
- Run Kopelowitz scheme with $O\left(|N|^{2k+3}\right)$ constraints.

Non-simple b-Matching Games

- Suppose $b \equiv 2$.

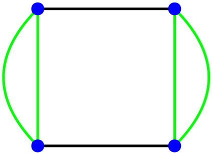


$w \equiv 1$

$b \equiv 2$

Non-simple b-Matching Games

- Suppose $b \equiv 2$.
- There is a maximum non-simple b-matching consisting of only parallel edges.

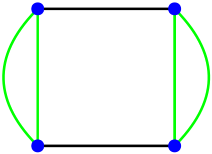


$w \equiv 1$

$b \equiv 2$

Non-simple b-Matching Games

- Suppose $b \equiv 2$.
- There is a maximum non-simple b-matching consisting of only parallel edges.
- Run Kopelowitz scheme with $O\left(|N|^2\right)$ constraints.



$w \equiv 1$

$b \equiv 2$

Conclusion

- Computing the nucleolus for simple bipartite b-matching games when $b \leq 3$ is NP-hard.

Conclusion

- Computing the nucleolus for simple bipartite b-matching games when $b \leq 3$ is NP-hard.
- When $b \leq 2$, there are polynomial-time algorithms which compute the nucleolus for special cases.

Conclusion

- Computing the nucleolus for simple bipartite b-matching games when $b \leq 3$ is NP-hard.
- When $b \leq 2$, there are polynomial-time algorithms which compute the nucleolus for special cases.
- Can we compute the nucleolus for b-matching games in general graphs when $b \leq 2$ in polynomial time?

Conclusion

- Computing the nucleolus for simple bipartite b-matching games when $b \leq 3$ is NP-hard.
- When $b \leq 2$, there are polynomial-time algorithms which compute the nucleolus for special cases.
- Can we compute the nucleolus for b-matching games in general graphs when $b \leq 2$ in polynomial time?
- Is there a combinatorial algorithm to compute the nucleolus for b-matching games?

Thanks!