Capacitated Network Bargaining Games: Stability and Structure

Lucy Verberk

Eindhoven University of Technology

Joint work with Laura Sanità

Capacitated Network Bargaining Games

In *capacitated network bargaining games* we are given a triple (G, w, c). The vertices of the graph represent players, the edges represent potential deals between players and the edge weights represent the values of the deals. Each player v can enter in at most c_v deals. For each deal, the involved players have to decide how to split the value of their deal. Hence, an *outcome* is naturally associated with a c-matching M, and a vector $a \in \mathbb{R}^{2E}_{\geq 0}$ that satisfies $a_{uv} + a_{vu} = w_{uv}$ if $uv \in M$, and $a_{uv} = a_{vu} = 0$ otherwise. An outcome is *stable* if no pair of players has an incentive to break the current outcome to enter in a deal with each other.

LP Characterization

A key property of (capacitated) NBG is that instances admitting a stable outcome have a very nice LP characterization, as shown by [6, 2]: given an instance (G, w, c), there exists a stable outcome for the corresponding game on G if and only if the value of a maximum-weight c-matching $\nu^c(G)$ equals the value of a maximum-weight *fractional* c-matching $\nu^c_f(G)$, defined as

$$\nu^c_f(G) := \max\{w^\top x : x(\delta(v)) \leq c_v \ \forall v \in V, 0 \leq x \leq 1\}.$$

In other words, instances admitting stable outcomes are the ones for which the LP relaxation of the maximum-weight c-matching problem has an optimal integral solution. A graph G for which $\nu^c(G) = \nu^c_f(G)$ is called *stable*.

The Stabilizer Problem

The stabilizer problem is motivated by the fact that not all graphs are stable. The goal is to minimally modify a graph as to ensure a stable outcome. A natural way to modify a graph is by reducing the capacity of vertices (players), or removing edges (blocking deals).

Capacity-stabilizer problem: given an instance (G, w, c), find a minimum-cardinality multi-set S of vertices, such that if you reduce the capacity of all vertices in S by one, you obtain a stable graph.

Edge-stabilizer problem: given an instance (G, w, c), find a minimum-cardinality set $F \subseteq E$, such that $G \setminus F$ is stable.

In unit-capacity ($c = 1$) graphs, the capacity-stabilizer problem is polynomial-time solvable [1, 5, 7], while the edge-stabilizer problem is NP-hard, and even hard-to-approximate with a constant factor, but admits an $O(\Delta)$-approximation algorithm [4, 3, 7].

Structure

Extreme point solutions of $\nu^c_f(G)$, or *basic* fractional c-matchings, satisfy $x_e \in \{0, \frac{1}{2}, 1\}$ for all edges $e \in E$, and the edges with $x_e = \frac{1}{2}$ induce vertex-disjoint odd cycles with saturated vertices. To stabilize an instance, we hence want to get rid of these odd cycles.

References

[1] Sara Ahmadian, Hamideh Hosseinzadeh, and Laura Sanità. Stabilizing network bargaining games by blocking players. *Mathematical Programming*, 172:249–275, 2018.

[2] MohammadHossein Bateni, MohammadTaghi Hajiaghayi, Nicole Immorlica, and Hamid Mahini. The cooperative game theory foundations of network bargaining games, 2010.

[3] Adrian Bock, Karthekeyan Chandrasekaran, Jochen Könemann, Britta Peis, and Laura Sanità. Finding small stabilizers for unstable graphs. *Mathematical Programming*, 154:173–196, 2015.

[4] Corinna Gottschalk. Personal communication, 2018.

[5] Takehiro Ito, Naonori Kakimura, Naoyuki Kamiyama, Yusuke Kobayashi, and Yoshio Okamoto. Efficient stabilization of cooperative matching games. *Theoretical Computer Science*, 677:69–82, 2017.

[6] Jon Kleinberg and Éva Tardos. Balanced outcomes in social exchange networks. In *Proceedings of the 40th STOC*, pages 295–304, 2008.

[7] Zhuan Khye Koh and Laura Sanità. Stabilizing weighted graphs. *Mathematics of Operations Research*, 45(4):1318–1341, 2020.

Capacity-Stabilizer

Theorem 1

The capacity-stabilizer problem is polynomial-time solvable.

The main idea of the algorithm is this: compute a basic maximum-weight fractional c-matching, and for each odd cycle induced by $x_e = \frac{1}{2}$ edges, choose one vertex and reduce its capacity by one.

Key properties. Our algorithm preserves the total value that the players can get up to a factor that is asymptotically best possible, and it reduces the capacity of each vertex by at most one. This has a nice network bargaining interpretation: there is always an optimal and at the same time *fair* way to stabilize instances, as no player will have its capacity dramatically reduced compared to others.

Edge-Stabilizer

Theorem 2

The edge-stabilizer problem admits an $O(\Delta)$-approximation algorithm.

Based on the unit-capacity approximation algorithm, we use the capacity-stabilizer algorithm, but instead of reducing the capacity of the vertices, we remove all edges incident with those vertices, except the edges e with $x_e = 1$.

Key properties. In unit-weight ($w = 1$) instances, our algorithm does not decrease the total value that the players can get.

Polyhedral Tools: Circuits!

The stabilizer results for unit-capacity instances mainly used combinatorial techniques, we instead rely on (new) polyhedral arguments. Decreasing the capacity of a vertex or removing an edge, are operations that correspond with translating inequalities of the LP that describes $\nu^c_f(G)$. We prove the following general theorem.

Theorem 3

Let $\mathcal{P}$ be any polytope, $a^\top x \leq b$ be an inequality of the description of $\mathcal{P}$, and $\delta \in \mathbb{R}_{>0}$. Let $\overline{x}$ be an optimal solution of the LP $\max\{c^\top x : x \in \mathcal{P}, a^\top x \leq b - \delta\}$, and assume that $\overline{x}$ is a non-optimal vertex of the LP $\max\{c^\top x : x \in \mathcal{P}\}$. Furthermore, assume that there is no vertex $\widetilde{x}$ of $\mathcal{P}$ satisfying $b - \delta < a^\top \widetilde{x} < b$. Then it is possible to move to an optimal solution of $\max\{c^\top x : x \in \mathcal{P}\}$ from $\overline{x}$ in one step over the edges of $\mathcal{P}$ (i.e., there is an optimal vertex of $\mathcal{P}$ adjacent to $\overline{x}$).

Exploiting circuits. Moving along an edge of the fractional c-matching polytope corresponds to taking a circuit direction of this polytope, which has a well-known graphical interpretation. Since circuits have minimal support, moving along it cannot increase too much the number of odd cycles in the support of a basic fractional c-matching. Exploiting this we can show that the minimum number of odd cycles is a lower bound on the amount of capacity that needs to be reduced/number of edges that need to be removed.

Open Problems

- Stabilize by removing vertices, when the capacity is bounded by 2? (Poly-time when $c = 1$, APX-hard when $c \leq 3$.)
- Stabilize by removing edges and simultaneously preserving the weight of a maximum-weight c-matching. Approximation algorithm?