Optimizing Social Welfare for Network Bargaining Games

Hubert Chan, Fei Chen, Li Ning
The University of Hong Kong

Presented by: Li Ning

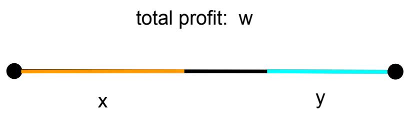
September 5, 2012

Splitting Profit

- There is a group of n users. They are connected by an undirected graph.

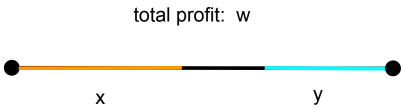
Splitting Profit

- There is a group of n users. They are connected by an undirected graph.
- For each edge, there is a potential contract to be made. The two participating users split the profit associated with the edge. Legal splitting requires $x + y \leq w$.



Splitting Profit

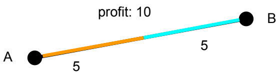
- There is a group of n users. They are connected by an undirected graph.
- For each edge, there is a potential contract to be made. The two participating users split the profit associated with the edge. Legal splitting requires $x + y \leq w$.

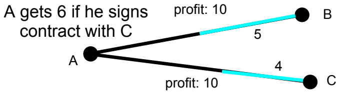


- Capacity: each user i can make at most b_i contracts.

Bargaining

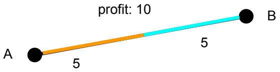
- When a user has many choices of contract to execute, it will bargain in order to get more benefit.

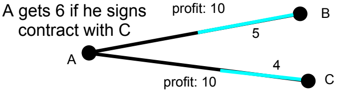




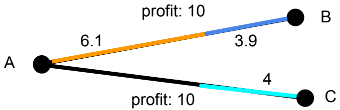
Bargaining

- When a user has many choices of contract to execute, it will bargain in order to get more benefit.





- Some user may compromise and change its demand.



Social welfare

Users are usually selfish and only aim at maximizing their own benefit.

We are also interested in the social welfare, which is the total profit of those contracts finally executed.

Fact: The social welfare is maximized if the contracts executed corresponds to a maximum b-matching of the given graph.

Outcome and outside option

- An *outcome* consists of the set of executed contracts, and how the profit in each contract is distributed among the two participating users.

Outcome and outside option

- An *outcome* consists of the set of executed contracts, and how the profit in each contract is distributed among the two participating users.
- The *outside option* of a node is the maximum profit the node can get from another node with whom there is no current contract.

Formal definition for outside option

For node i, let x_i denote its benefit from the current contract.

Formal definition for outside option

For node i, let x_i denote its benefit from the current contract.

- **Alternative option**: For two nodes i, j, between which there is no contract currently, the alternative option for i from j is

$$w_{ij} - x_j.$$

Formal definition for outside option

For node i, let x_i denote its benefit from the current contract.

- **Alternative option**: For two nodes i, j, between which there is no contract currently, the alternative option for i from j is

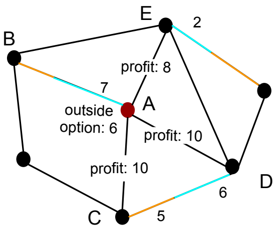
$$w_{ij} - x_j.$$

- **Outside option**: the outside option α_i of node i is defined to be

$$\alpha_i = \max_j w_{ij} - x_j,$$

where j is over all the nodes which have no contract with node i.

Example for outside option



Node A has neighbors B, C, D, and E. Given the current contract in the figure, A makes contract with B, and gets 7. Furthermore, node A has outside option 6, because

- C gets 5 currently. Then A can get 5 if he switch to make contract with C.
- D gets 6 currently. Then A can get 4 if he switch to make contract with D.
- E gets 2 currently. Then A can get 6 if he switch to make contract with E.

Stable and balanced outcomes

Desirable properties of outcomes:

- An outcome is *stable* if for every contract a user makes, the profit it gets is at least its outside option.

$$x_i \geq \alpha_i$$

Stable and balanced outcomes

Desirable properties of outcomes:

- An outcome is *stable* if for every contract a user makes, the profit it gets is at least its outside option.

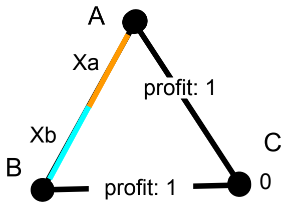
$$x_i \geq \alpha_i$$

- An outcome is *balanced* if, in addition to stability, for every contract made, after each participating user gets its outside option, the surplus is divided equally.

$$x_i - \alpha_i = x_j - \alpha_j$$

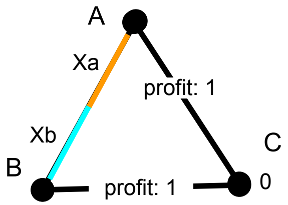
Stable outcome does not always exist

Since stability is a desirable property, the stable outcome does not always exist. For example, for a triangle case with all edges' weight to be 1, there is no stable solution.



Stable outcome does not always exist

Since stability is a desirable property, the stable outcome does not always exist. For example, for a triangle case with all edges' weight to be 1, there is no stable solution.



- Without loss of generality, we assume finally A and B make the contract, and they get x_a and x_b respectively. Clearly, $x_a + x_b \leq 1$.
- Node C gets 0. Hence both A and B have outside option 1.
- Since at least one of x_a and x_b is less than 1, it follows that at least one of A and B is not stable.

Stable and Integrality Gap

A stable outcome exists *iff* the LP relaxation for maximum matching problem on a given graph has integrality gap 1.

Related work includes:

- "The cooperation game theory foundations of network bargaining games", MohammadHossein Bateni, MohammadTaghi Hajiaghayi, Nicole Immorlica, and Hamid Mahini, *ICALP 2010*.
- "Fast convergence of natural bargaining dynamics in exchange networks", Yashodhan Kanoria, Mohsen Bayati, Christian Borgs, Jennifer T. Chayes, and Andrea Montanari, *SODA 2011*.

Our Motivation

We consider the integrality gap condition as a limitation in practice. Hence, in this work, we consider an alternative model for network bargaining games, for which

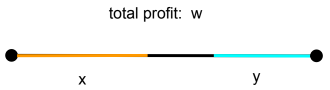
- different notions of equilibrium are investigated;
- the existence of the equilibrium does not require the integrality gap condition.

Our Model

- Edges take over the role of "rational" players from the nodes. Each edge $e = \{i, j\} \in E$ corresponds to an *agent*.

Our Model

- Edges take over the role of "rational" players from the nodes. Each edge $e = \{i, j\} \in E$ corresponds to an *agent*.
- **Action set**: for agent $e = \{i, j\}$, its action set is all the possible split of the edge weight, i.e., $\{(x, y) : x + y \leq w_{ij}, x \geq 0, y \geq 0\}$.



Remark: It is possible that $x + y < w_{ij}$. The remaining weight will be lost.

Selection Process

- **Selection process**: based on a configuration m of actions made by all the agents, a selection process (which can be randomized) is run, and it will choose a b-matching M.

Selection Process

- **Selection process**: based on a configuration m of actions made by all the agents, a selection process (which can be randomized) is run, and it will choose a b-matching M.
- **Payoff**: the payoff $u_e(m)$ of the agent on edge e is the probability that e is in the returned matching M.

Selection Process

- **Selection process**: based on a configuration m of actions made by all the agents, a selection process (which can be randomized) is run, and it will choose a b-matching M.
- **Payoff**: the payoff $u_e(m)$ of the agent on edge e is the probability that e is in the returned matching M.
- Since the payoff function is defined, then Nash Equilibrium is also well defined. In this paper, we are interested in the social welfare $S(m) = \sum_e u_e(m) \cdot w_e$, which is the expected weight of the maximum matching returned, given an action configuration m.

More about the Selection Process

- All players (agents) make their actions, and the payoff is determined by the selection process.

More about the Selection Process

- All players (agents) make their actions, and the payoff is determined by the selection process.
- In the selection process, the node users will be considered. We assume the node users will exhibit two characteristics: *greed* and *spite*.

Greedy Users

For a node user i with capacity b_i, it will definitely want an offer that is strictly better than its $(b_i + 1)$-st best offer.

If the proposal made by an edge agent fulfills this requirement for both participating node users, then such edge will be definitely selected.

We call the resulting payoff function *greedy*.

Spiteful Users

For a node user i with capacity b_i, it will definitely reject an offer that is strictly worse than its b_i-th best offer.

If the proposal made by an edge agent fulfills this requirement for at least one participating node user, then such edge will be definitely rejected.

We call the resulting payoff function *spiteful*.

Ambiguous Edges

There are possibly ambiguous edges, which are neither definitely selected nor definitely rejected.

For example, if a node has equal b_i-th and $(b_i + 1)$-st best offers, then this node will not definitely select and not definitely reject each of them.

A Protocol $\mathcal{P}$ for Edge Agents

- We consider a protocol $\mathcal{P}$, which is adapted from the protocol proposed by Kanoria et al. [KBB+11]. Edge agents communicate under protocol $\mathcal{P}$ to make their final actions.

A Protocol $\mathcal{P}$ for Edge Agents

- We consider a protocol $\mathcal{P}$, which is adapted from the protocol proposed by Kanoria et al. [KBB+11]. Edge agents communicate under protocol $\mathcal{P}$ to make their final actions.
- In round t,

A Protocol $\mathcal{P}$ for Edge Agents

- We consider a protocol $\mathcal{P}$, which is adapted from the protocol proposed by Kanoria et al. [KBB+11]. Edge agents communicate under protocol $\mathcal{P}$ to make their final actions.
- In round t,
  - edge agent $\{i, j\}$ maintains an action $(m^{(t)}_{j \to i}, m^{(t)}_{i \to j})$;

A Protocol $\mathcal{P}$ for Edge Agents

- We consider a protocol $\mathcal{P}$, which is adapted from the protocol proposed by Kanoria et al. [KBB+11]. Edge agents communicate under protocol $\mathcal{P}$ to make their final actions.
- In round t,
  - edge agent $\{i, j\}$ maintains an action $(m^{(t)}_{j \to i}, m^{(t)}_{i \to j})$;
  - for each node user i, all edges incident to i send their $m^{(t)}_{\cdot \to i}$ values to others.

Rule $\mathcal{T}$

Outside option:

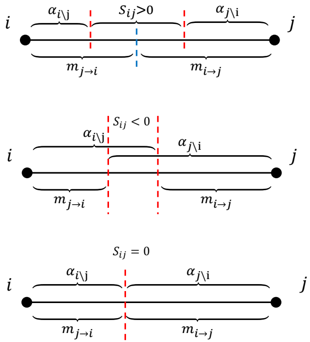
$$\alpha_{i\backslash j}^{(t)} := \max_{k \in N(i)\backslash\{j\}}^{(b_i)} m_{k\to i}^{(t-1)}$$
$$\alpha_{j\backslash i}^{(t)} := \max_{k \in N(j)\backslash\{i\}}^{(b_j)} m_{k\to j}^{(t-1)}$$

Surplus:

$$S_{ij} := w_{ij} - \alpha_{i\backslash j}^{(t)} - \alpha_{j\backslash i}^{(t)}$$

Update:

$$m_{j\to i}^{(t)} := (w_{ij} - \alpha_{j\backslash i}^{(t)})_+ - \frac{(S_{ij})_+}{2}$$
$$m_{i\to j}^{(t)} := (w_{ij} - \alpha_{i\backslash j}^{(t)})_+ - \frac{(S_{ij})_+}{2}$$



We can treat one step running of $\mathcal{T}$ as a function on m.

Update Function

- Given an action configuration m, we define the update function T as

$$T(m) := \frac{1}{2}m + \frac{1}{2}\mathcal{T}(m).$$

Update Function

- Given an action configuration m, we define the update function T as

$$T(m) := \frac{1}{2}m + \frac{1}{2}\mathcal{T}(m).$$

- A fixed point of T is an action configuration m, such that $m = \mathcal{P}(m)$.

Update Function

- Given an action configuration m, we define the update function T as

$$T(m) := \frac{1}{2}m + \frac{1}{2}\mathcal{T}(m).$$

- A fixed point of T is an action configuration m, such that $m = \mathcal{P}(m)$.
- **Update protocal**: If we define the update protocol as $\mathcal{P} : m^{(t)} = T(m^{(t-1)})$. Then $\mathcal{P}$ will always converge to a fixed point of T, which can be represented by a configuration of actions made by edge agents.

Our Results:
Pure Nash Equilibrium always exists if the node users are greedy and spiteful

Theorem:
Every fixed point of $\mathcal{P}$ is a Nash equilibrium whenever node users are greedy and spiteful, **regardless of their decisions on ambiguous edges**.

To achieve good social welfare by maximizing matching among ambiguous edges

- Hence, it is possible to achieve good social welfare, by selecting a maximum b'-matching among the ambiguous edges, where b' is remaining node capacities (recall that greedy edges are definitely selected and occupy some capacities).

Our Results: Price of Stability

- Run protocol $\mathcal{P}$ on edge agents.

Our Results: Price of Stability

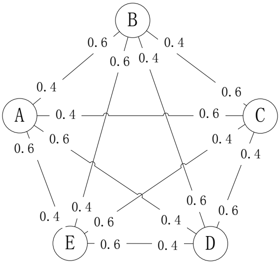
- Run protocol $\mathcal{P}$ on edge agents.
- If node users collaboratively find a maximum weighted b'-matching on the set of ambiguous edges, then the social welfare is at least $\frac{2}{3}S^*$, where S^* is the maximum social welfare.
  - The factor $\frac{2}{3}$ is due to the integrality gap of max matching LP.

Our Results: Price of Stability

- Run protocol $\mathcal{P}$ on edge agents.
- If node users collaboratively find a maximum weighted b'-matching on the set of ambiguous edges, then the social welfare is at least $\frac{2}{3}S^*$, where S^* is the maximum social welfare.
  - The factor $\frac{2}{3}$ is due to the integrality gap of max matching LP.
- There are distributed protocols to find maximum weighted matchings, achieving $(1+c)$-approximation for $c > 0$ in poly-logarithmic time.
  - In such a case, the price of stability is $1.5(1+c)$.

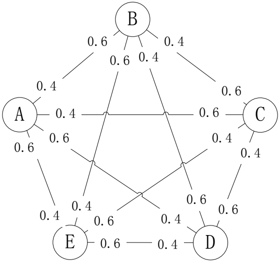
Our Results: Price of Anarchy

- Protocol $\mathcal{P}$ returns a Nash equilibrium and results in good social welfare if node users perform a collaborative effort on ambiguous edges. In general, the social welfare from a Nash equilibrium can be as low as zero.



Our Results: Price of Anarchy

- Protocol $\mathcal{P}$ returns a Nash equilibrium and results in good social welfare if node users perform a collaborative effort on ambiguous edges. In general, the social welfare from a Nash equilibrium can be as low as zero.



- Price of Anarchy is ∞.

Our Results: Rate of Convergence

- Although protocol $\mathcal{P}$ always converges to some fixed point, it is possible that $\mathcal{P}$ will never exactly reach that point.

Our Results: Rate of Convergence

- Although protocol $\mathcal{P}$ always converges to some fixed point, it is possible that $\mathcal{P}$ will never exactly reach that point.
- We can stop $\mathcal{P}$ in polynomial time and still guarantee good social welfare, if we relax the notions of greed and spite:

Our Results:
Rate of Convergence

- Although protocol $\mathcal{P}$ always converges to some fixed point, it is possible that $\mathcal{P}$ will never exactly reach that point.
- We can stop $\mathcal{P}$ in polynomial time and still guarantee good social welfare, if we relax the notions of greed and spite:
  - ϵ-greedy: a user definitely accepts an offer strictly greater than its $(b_i + 1)$-st best offer plus ϵ;

Our Results: Rate of Convergence

- Although protocol $\mathcal{P}$ always converges to some fixed point, it is possible that $\mathcal{P}$ will never exactly reach that point.
- We can stop $\mathcal{P}$ in polynomial time and still guarantee good social welfare, if we relax the notions of greed and spite:
  - ϵ-greedy: a user definitely accepts an offer strictly greater than its $(b_i + 1)$-st best offer plus ϵ;
  - ϵ-spiteful: a user definitely rejects an offer strictly less than its (b_i)-th best offer minus ϵ.

Our Results: Rate of Convergence

- Although protocol $\mathcal{P}$ always converges to some fixed point, it is possible that $\mathcal{P}$ will never exactly reach that point.
- We can stop $\mathcal{P}$ in polynomial time and still guarantee good social welfare, if we relax the notions of greed and spite:
  - ϵ-greedy: a user definitely accepts an offer strictly greater than its $(b_i + 1)$-st best offer plus ϵ;
  - ϵ-spiteful: a user definitely rejects an offer strictly less than its (b_i)-th best offer minus ϵ.
- There exists $T = \Theta(\frac{W^2 n^4}{\epsilon^2})$, where W is the maximum edge weight and n is the number of node users, such that the configuration returned by $\mathcal{P}$ after T rounds can result in social welfare at least $\frac{2}{3}S^*$ given that node users are ϵ-greedy and ϵ-spiteful.

Conclusions

- If the topology of the given graph and the edge weights naturally indicate that
  - certain edges should (value 1 in max matching LP solution) be selected,
  - while some should be rejected (value 0 in max matching LP solution),

  then, our framework of greed and spite can detect these edges.
- However, our framework does not specify how to select the ambiguous edges naturally and efficiently at the same time.

Thank you!