Lecture 11

Wednesday, 28 September 2022 10:57 AM

Equilibrium Computation in Potential Games

Recall: NCG: $(G=(V,E), s, t, N, (c_e)_{e\in E})$, $n = |N|$

s is a PNE iff s is a local minimizer of

$$\phi(s) = \sum_e \sum_{i=1}^{n_e(s)} c_e(i)$$

Algorithm for computing equilibria, assuming c_e is increasing for all $e \in E$ $(c_e(i+1) \geq c_e(i))$.

Construct graph $\hat{G} = (V, \hat{E})$, where each edge $e \in E$ is replaced by n copies, $e^1, \dots, e^n$

Each edge has a fixed cost $\hat{c}$: $\hat{c}_{e^k} = c_e(k)$

Each edge has unit capacity : $u_e = 1 \;\forall e \in \hat{E}$.

In the graph $\hat{G}$, with edge costs $\hat{c}$, find a min-cost s-t flow of demand n.

$$\min \sum_{e\in\hat{E}} c_e \cdot f_e$$

f is an s-t flow of demand n in $\hat{G}$

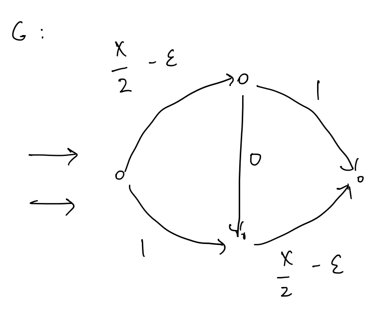
$f_e \leq 1 \quad \forall e \in \hat{E}$

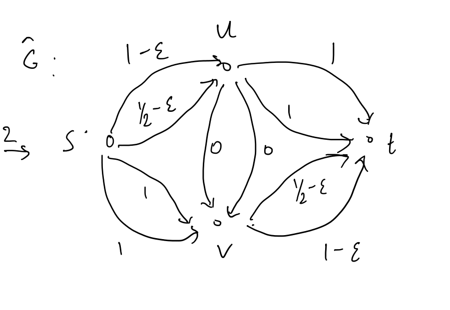
Since the flow polytope is integral, $\exists$ a min-cost flow f s.t. $f_e \in \{0,1\} \;\forall e \in \hat{E}$.

Thus, f consists of n disjoint paths $s_1, \dots, s_n$

Claim: $s = (s_1, \dots, s_n)$ is a global minimizer for the potential fn. for the NCG

(prove yourself)





Now suppose we generalize NCGs, to all each player to have its own source s_i & destination t_i

Then an NCG $\Gamma = (G=(V,E), \{(s_i,t_i)\}_{i\in[n]}, \{c_e\}_{e\in E})$

And $\forall$ players i, $\mathcal{S}_i = \{p : p \text{ is an } s_i - t_i \text{ path in } G.\}$

Claim: If all players have the same source $(s_i = s \;\forall i)$, or all players have the same sink $(t_i = t \;\forall i)$, an equilibrium can be found efficiently

(prove yourself)

What if players have diff sources, destinations?

General Congestion Games

$$(R, N, \{\mathcal{S}_i\}_{i\in N}, \{c_e\}_{e\in R})$$

where $\mathcal{S}_i \subseteq 2^R$, $c_e : [n] \to \mathbb{R}_+$

(give Braess example)

This is also a potential game. Can equilibrium be found in poly-time?

Max-Cut Game

Players correspond to vertices in an undirected, weighted graph G.

Each player has 2 strategies : L or R.

Given a strategy s_i for each player,

$$c_i(s) = \sum_{\substack{e=\{i,j\}:\\ s_i = s_j}} w_e \qquad \Big(\text{or } u_i(s) = \sum_{\substack{e=\{i,j\}:\\ s_i \neq s_j}} w_e\Big)$$

This is a potential game.

Can check $\phi(s) = \sum_{\substack{e=\{i,j\}\\ s_i\neq s_j}} w_e$ is a potential fn.

Clearly if $w_e = 1 \;\forall e \in E$, a PNE can be computed in poly time. But what about the general case?

(note that, for $w_e = 1$, finding a global max-cut is NP-hard)

The Class PLS (Polynomial Local Search)

A family of problems $\mathcal{F}$ is in PLS if there exist poly-time algorithms for the following:

(i) an algorithm that takes as input an instance I of $\mathcal{F}$, and returns a solution s,

(ii) an algorithm that takes as input an instance I of $\mathcal{F}$ and a solution s of I, and returns the cost of s,

(iii) an algorithm that takes as input instance I & soln s, and returns all neighbouring solns. of s

Note that, given I and s, using (ii) & (iii) we can in poly-time find a neighbouring soln. of lower cost or determine s is locally minimal.

The problem is to find a locally minimal cost solution given an instance I of $\mathcal{F}$.

The max-cut game, NCGs, general congestion games, are all in PLS.

PLS-reductions:

$F_1 \leq_{PLS} F_2$ if $\exists f_1, f_2$ that run in poly-time, s.t.:

f_1 maps instances of F_1 to instances of F_2

f_2 maps solutions of F_2 to solutions of F_1, s.t. if s is a local optimum in F_2, then $f_2(s)$ is a local optimum in F_1

Theorem: The Max-cut game is PLS-complete (SPY '88)

Theorem: General Congestion Games are PLS-complete.

Proof: Let $\Gamma = (G, w)$ be an instance of a max-cut game.

Players: Players in MCG = Players in GCG

Resources: For each edge e in MCG, there are 2 resources, e_L & e_R in GCG

(so $R = \bigcup_{e\in E} \{e_L, e_R\}$)

Strategies: For each player i, $\mathcal{S}_i = \{\bigcup_{e\in E}\{e_L\}, \bigcup_{e\in E}\{e_R\}\}$

(thus, if $e = \{i,j\} \in E$, then the resources e_L, e_R can be used by exactly 2 players i & j in the GCG)

Costs: $c_{e_L}(1) = c_{e_R}(1) = 0$

$c_{e_L}(2) = c_{e_R}(2) = w_e$

This defines a GCG.

Now consider a strategy profile $(s = (s_1, \dots, s_n))$ in the GCG.

then $c_i(s) = \sum_{\substack{j:\{i,j\}\in E\\ s_j = s_i}} w_{ij}$

and hence a local minimum in the GCG gives a local minimum in the MC game.

Pareto Optimality, social welfare, & Price of Anarchy

Given a game Γ, a strategy profile s is said to be Pareto-optimal if $\nexists\, s'$ s.t.:

$\forall i, \quad u_i(s') \geq u_i(s)$

$\exists i \quad u_i(s') > u_i(s)$

Equivalently, for all s', either $\forall i \;\; u_i(s) = u_i(s')$, or $\exists i : u_i(s') < u_i(s)$.

For economists, efficiency $\Leftrightarrow$ Pareto-optimality.

Say that $s >_P s'$ if $\forall i,\; u_i(s) \geq u_i(s')$ and $\exists i : u_i(s) > u_i(s')$

Note that $>_P$ induces a partial order over the set of strategy profiles.

To induce a total order, we can think of a global utility fn., or a social welfare fn:

$\sum_i u_i(s) \to$ utilitarian social welfare

$\prod_i u_i(s) \to$ (for utilities) Nash social welfare

$\min_i u_i(s) \to$ egalitarian social welfare

...

This allows us to quantitatively see how "good"/"bad" an equilibrium is, compare 2 equilibria, etc

Typically in computer science, by social welfare we mean the utilitarian social welfare $\sum_i u_i(s)$.

For a game Γ. We can ask how bad is the social welfare at equilibrium compared to the best possible social welfare.

This ratio is known as the Price of Anarchy:

$$PoA(\Gamma) = \frac{\max\{\sum_i u_i(s) : s \text{ is a pure strategy profile}\}}{\min\{\sum_i u_i(s) : s \text{ is a PNE}\}}$$

(assume for now we stick to games with a PNE)

Thus, PoA compares social welfare at the worst PNE, to the best possible social welfare.

We could ask, what about the best PNE?

This is called the Price of Stability:

$$PoS(\Gamma) = \frac{\max\{\sum_i u_i(s) : s \text{ is a pure strategy profile}\}}{\max\{\sum_i u_i(s) : s \text{ is pure NE}\}}$$