Lecture 14

Monday, 17 October 2022 9:19 AM

PoA: Smooth Games & Load Balancing Games

Recall:

Theorem: In congestion games w/ affine cost fns, PoA $\leq 5/2$

Proof:

$$\text{Cost}(s) = \sum_{i=1}^{n} c_i(s) \leq \sum_{i=1}^{n} c_i(s_i^*, s_{-i}) \leq \frac{5}{3}\text{cost}(s^*) + \frac{1}{3}\text{cost}(s)$$

s is an eq.

(i) affine delay fns.
(ii) $x(y+1) \leq \frac{5}{3}x^2 + \frac{1}{3}y^2$
(holds for any s, s^*)

Theorem: In monotone valid utility games, PoA ≤ 2

Proof:

$$V(o) - V(s) \leq V(o \cup s) - V(s) \leq \sum_{i=1}^{n} V(o^i \cup s) - V(o^{i-1} \cup s)$$

monotonicity (where $o^i = \{o_1, o_2, \dots, o_i\}$, $o^0 = \phi$)

$$\leq \sum_{i=1}^{n} u_i(o_i, s_{-i}) \leq \sum_{i=1}^{n} u_i(s) = V(s)$$

(i) submodularity, (ii) contribution $\leq$ utility (holds for any o, s)

s is equilibrium

We will now generalize these techniques for bounding the PoA.

Defn: A cost-minimization game is (λ, μ)-smooth if for all strategy profiles s, s^*

$$\sum_{i=1}^{n} c_i(s_i^*, s_{-i}) \leq \lambda\, \text{cost}(s^*) + \mu\, \text{cost}(s)$$
$$(\lambda > 0,\ \mu < 1)$$

thus, congestion games are $(\frac{5}{3}, \frac{1}{3})$-smooth

Defn: A utility maximizing game is (λ, μ)-smooth if for all strategy profiles s, s^*,

$$\sum_{i=1}^{n} u_i(s_i^*, s_{-i}) \geq \lambda V(s^*) - \mu V(s)$$
$$(\lambda > 0,\ \mu \geq 0)$$

Then monotone VU games are $(1,1)$-smooth

Theorem: The PoA for a (λ, μ)-smooth cost minimization game is at most $\lambda/(1-\mu)$
The PoA for a (λ, μ)-smooth utility maximization game is at most $\lambda/(1+\mu)$

(can be easily seen)

Looking at the proof in this way allows us to generalize these bounds.

Currently, we've shown these bounds only for PNE. However, we can show that the same bounds extend to the (possibly) much larger class of CCE:

Recall CCE:

$$\mathbb{E}_{s\sim\sigma}[c_i(s)] \leq \mathbb{E}_{s\sim\sigma}[c_i(s_i', s_{-i})] \quad \forall i, s_i' \in S_i$$

Theorem: For a (λ, μ)-smooth cost-minimization game, the PoA for CCE is at most $\lambda/(1-\mu)$

Proof: Let σ be a CCE, s^* is an optimal (min cost) profile.

Then
$$\mathbb{E}_{s\sim\sigma}[\text{cost}(s)] = \sum_{i=1}^{n} \mathbb{E}_{s\sim\sigma}[c_i(s)]$$
$$\leq \sum_{i=1}^{n} \mathbb{E}_{s\sim\sigma}[c_i(s_i^*, s_{-i})]$$
$$\leq \mathbb{E}_{s\sim\sigma}\left[\sum_{i=1}^{n} c_i(s_i^*, s_{-i})\right]$$
$$\leq \mathbb{E}_{s\sim\sigma}[\lambda\, \text{cost}(s^*) - \mu\, \text{cost}(s)]$$

uses that ineq. holds for arbit s

$$= \lambda\, \text{cost}(s^*) - \mu\, \mathbb{E}_{s\sim\sigma}[\text{cost}(s)]$$

Thus, $\mathbb{E}_{s\sim\sigma}[\text{cost}(s)] \leq \frac{\lambda}{1+\mu}\text{cost}(s^*)$ ∎

Further, the bounds on PoA (approximately) hold even if we go beyond exact equilibria.

Defn: For a cost-minimization game, s is an ε-PNE if $\forall i, s_i' \in S_i$,

$$C_i(s) \leq (1+\varepsilon)\, C_i(s_i', s_{-i})$$

Theorem: If s is an ε-PNE for a (λ, μ)-smooth cost-minimization game, and $\varepsilon < \frac{1}{\mu} - 1$, then:

$$C(s) \leq \frac{\lambda}{\frac{1}{1+\varepsilon} - \mu} C(s^*)$$

(prove yourself)

Load-Balancing Games

Given:
- n agents / jobs, job i has weight w_i
- m machines, machine j has rate r_j
- $S_i = [m]$ $\forall i \in N$ (each job chooses a machine)

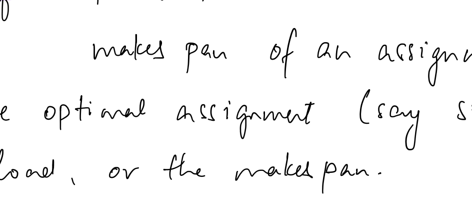
define: $tw_j(s) = \sum_{i: s_i = j} w_i$ (total weight on j)

$l_j(s) = tw_j(s) / r_j$ (load on machine j)

this is the time taken by machine j to complete all jobs assigned to it.
assume: machines do proportional time-sharing, hence all jobs on machine j complete at time $l_j(s)$
- $c_i(s) = l_{s_i}(s)$ (the completion time for machine s_i)
- $\text{cost}(s) = \max_i c_i(s) = \max_j l_j(s)$ (egalitarian objective)

Defn.: the maximum load over machines is called the makespan of an assignment.
The optimal assignment (say s^*) minimizes the maximum load, or the makespan.

Example: $w_1 = w_2 = 1$, $w_3 = 2$
$r_1 = r_2 = 1$, $r_3 = 2$



s^* assigns w_i to r_i, OPT $= 1$
worst eq. s assigns w_1, w_2 to r_3, w_3 to r_2
then $l_2(s) = 2$

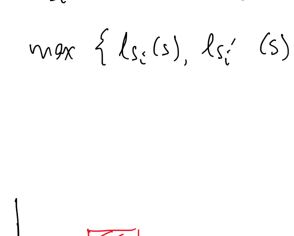
hence PoA ≥ 2

Theorem: Every load-balancing game has a PNE

Proof: Fix a strategy profile s, job i. Say i has a better strategy, say s_i'. Let $s' = (s_i', s_{-i})$

Then: $l_{s_i}(s) > l_{s_i'}(s') > l_{s_i'}(s)$

Hence $\max\{l_{s_i}(s), l_{s_i'}(s)\} = l_{s_i}(s)$



$\max\{l_{s_i}(s), l_{s_i'}(s)\}$
$= l_{s_i}(s)$
$> \max\{l_{s_i}(s'), l_{s_i'}(s')\}$

We want to say that the maximum load (makespan) strictly decreases, whenever an agent reduces cost... but that may not be true (however, makespan will not increase).

Instead, consider the vectors of sorted loads (for simplicity, assume all loads distinct):

Let $\pi: [m] \to [m]$, $\sigma: [m] \to [m]$ be s.t.

$l_{\pi(1)}(s) > l_{\pi(2)}(s) > \dots > l_{\pi(m)}(s)$

and

$l_{\sigma(1)}(s') > l_{\sigma(2)}(s') > \dots > l_{\sigma(m)}(s')$

$\text{sort}(s) := (l_{\pi(1)}(s), l_{\pi(2)}(s), \dots, l_{\pi(m)}(s))$
$\text{sort}(s') := (l_{\sigma(1)}(s), l_{\sigma(2)}(s), \dots, l_{\sigma(m)}(s))$

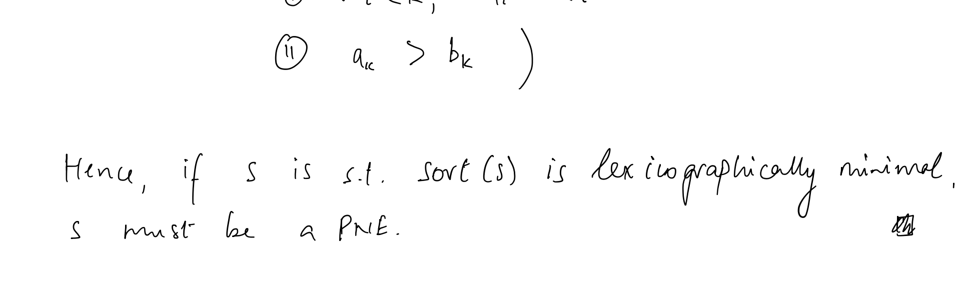
Then in the lexicographic ordering, $\text{sort}(s) > \text{sort}(s')$.
(prove yourself)

(For 2 vectors $a = (a_1, \dots, a_m)$, $b = (b_1, \dots, b_m)$,
$a > b$ if $\exists k \leq m$ s.t.:
(i) $\forall i < k$, $a_i = b_i$
(ii) $a_k > b_k$)

Hence, if s is s.t. $\text{sort}(s)$ is lexicographically minimal, s must be a PNE. ∎

Theorem: The PoA of load-balancing games is $O(\log m / \log\log m)$

Proof: Order machines by rates (decreasing):



Fix an eq. s

Let $\rho = \left\lfloor \frac{\text{cost}(s)}{\text{OPT}} \right\rfloor$

For $t = 0..\rho-1$, L_t is the marimal prefix of $1 \dots m$ s.t. $\forall j \in L_t$, $l_j(s) \geq t \cdot \text{OPT}$

Note
(i) $L_1 \supseteq L_2 \supseteq L_3 \supseteq \dots \supseteq L_{\rho-1}$
(ii) $L_0 = [m]$
(iii) $\forall i$ $\frac{w_i}{r_1} \leq \text{OPT}$

Claim 1: Machine $1 \in L_{\rho-1}$ (i.e., $l_j(s)/r_1 \geq (\rho-1)\,\text{OPT}$)
(hence, $|L_{\rho-1}| \geq 1$)

Claim 2: $|L_t| \geq (t+1)\,|L_{t+1}|$

The proof of the theorem follows, since:

$$m = |L_0| \geq 1 \cdot |L_1| \geq 1 \cdot 2\, |L_2|$$
$$\geq \dots \geq (\rho-1)!\, |L_{\rho-1}| \geq (\rho-1)!$$

Hence $m \geq \rho$ (for $\rho \geq 4$)
& $\ln m \geq \ln(\rho-1)! \geq (\rho-1)\ln(\rho-1)$
$\Rightarrow \rho - 1 \leq \frac{\ln m}{\ln(\rho-1)}$, or $\rho \leq O(\ln m / \ln\ln m)$ ∎

Proof of Claim 1: Say machine $1 \notin L_{\rho-1}$
$\Rightarrow l_1(s) < (\rho-1)\,\text{OPT}$
But $\exists i: c_i(s) = l_{s_i}(s) \geq \rho\,\text{OPT}$
Suppose agent i switches to machine 1.
Let $s' = (s_i = 1, s_{-i})$
Then $c_1(s') = l_1(s) + \frac{w_i}{r_1} < (\rho-1)\,\text{OPT} + \text{OPT}$
$= \rho\,\text{OPT}$

Hence, if m/c $1 \notin L_{\rho-1}$, then s is not an eq. ∎

Proof of Claim 2:

Subclaim: For an agent i, if $s_i \in L_{t+1}$ then $s_i^* \in L_t$

Proof: If $L_t = (1 \dots m)$, then clearly the subclaim holds.
Else, say $L_t = (1 \dots q-1)$. Then:
$$l_q(s) + \frac{w_i}{r_q} \geq l_{s_i}(s)$$
and since $s_i \in L_{t+1}$, $l_{s_i}(s) \geq (t+1) \cdot \text{OPT}$
But $l_q(s) < t \cdot \text{OPT}$. Hence $\frac{w_i}{r_q} > \text{OPT}$.
Then in s^*, i must be assigned to a faster machine, i.e., $s_i^* \in \{1 \dots q-1\}$.
Hence $s_i^* \in L_t$ ∎

Proof of Claim 2: Consider the total wt. of all jobs assigned to machines in L_{t+1}:

$$\sum_{i: s_i \in L_{t+1}} w_i \geq \sum_{j \in L_{t+1}} r_j (t+1)\,\text{OPT}$$

By subclaim, all of these jobs must be assigned to machines in L_t in OPT.

Hence $\sum_{i: s_i \in L_{t+1}} w_i \leq \sum_{j \in L_t} r_j \cdot \text{OPT}$

Thus, $\text{OPT} \sum_{j \in L_t} r_j \geq \text{OPT} \sum_{j \in L_{t+1}} r_j\, (t+1)$

$$\Rightarrow \sum_{j \in L_t \setminus L_{t+1}} r_j \geq t \sum_{j \in L_{t+1}} r_j$$

now, again say $L_{t+1} = (1 \dots q-1)$, $L_t = (1 \dots q, q+1, \dots)$
Then $\forall j \in L_{t+1}$, $r_j \geq r_q$
$\forall j \in L_t \setminus L_{t+1}$, $r_j \leq r_q$
Thus $r_q\, |L_t \setminus L_{t+1}| \geq t\, r_q\, |L_{t+1}|$
$\Rightarrow |L_t| \geq (t+1)\,|L_{t+1}|$ ∎