Mechanism Design (Contd.)

Last lec: Single parameter (single stuff)
agent $i$ has value $v_i/\text{unit-stuff}$
bids $b_i$

$\bar{b} = (b_1, \ldots, b_n)\xrightarrow{\text{Mechanism}} x_i(\bar{b}) = v_i$

$U_i(\bar{b}) = v_i - x_i(\bar{b}) - p_i(\bar{b})$

Myerson's Lemma:
1. $(x, \pi)$ is DSIC $\iff x(\cdot)$ is monotone
   - $\forall b_i, v_i, b_j, x_i(b_i, b_i)$ is monotone.
2. There is a unique $\pi$ s.t. $(x, \pi)$ is DSIC
3. $\pi$ has an explicit formula.

Proof:

Claim 1: $(x, \pi)$ is DSIC $\iff x(\cdot)$ is monotone.

Claim 2: If $(x, \pi)$ DSIC, then $\pi$ has an explicit formula.

Fix $i$. Fix $b_i$.

$x(b_i) = x_i(b_i, b_i)$

$\pi(b_i) = \pi_i(b_i, b_i)$
\[ p(b_i) = p_i(b_i, b_{-i}) \]
\[ p(b_i) = \frac{1}{b_i} \sum_{k=1}^{d} x_k \text{d} k \]
\[ p(b_i) = \int_0^{x(b_i)} d x \]

Claim 3: Such \((x, p)\) is DSIC

\[ v_i = v_i(x(b_i) - p_i) \]

True

Awesome Auctions:
1. DSIC
2. s.w. maximizing
3. Efficient implementation

\[ \delta \text{fixed } x(\cdot) = \arg\max_{x \in X} \sum_{i \in N} \delta_i x_i \]
Monotone? YES! (ex.)
Monotone? YES! (ex.)

\text{(Key)}
\\Rightarrow \text{Fires } p() \text{ (Myerson's Lemma)}

\text{Awesome Auction} \rightarrow \text{Reduces}

Design "Algorithm" to find s.w. maximizing allocation.
\\Rightarrow \text{Efficient (poly-time)?}

E.g.: TV Ad. Auction.
\underbrace{\text{W}}_{\text{viewer}} \to \text{commercial break.}
\underbrace{\text{N}}_{\text{set of advertisers}}:
\begin{align*}
i \in N: & 1. \text{ad } \& \text{ length } w_i \text{ secs.} \\
& 2. \text{value } v_i \text{ (private)} \\
& 3. \text{bid } b_i
\end{align*}

\Rightarrow \text{set of allocations}
\hspace{1cm} X = \left\{ x \in \{0, 1\}^N \mid \sum_{i \in N} x_i w_i \leq W \right\}

\Rightarrow \text{s.w. max allocation}
\hspace{1cm} x() = \arg \max_{x \in X} \sum_{i \in N} x_i b_i \leftarrow \text{Knapsack. NP-hard.}
Greedy Algorithm

1. \[ \frac{b_1}{w_1} > \frac{b_2}{w_2} > \ldots > \frac{b_n}{w_n} \]
allocate in this order until space allows.

Suppose allocate \( b_1, \ldots, k \)

2. \( \arg\max_i b_i < \)

Find \( k \) such that \( \sum_{i=1}^{k} b_i > b_{i^*} \) for all \( i^* \).

Else allocate only \( i^* \).

Then: Greedy s.w. \( \geq \frac{1}{2} \) max. s.w.

Ps.: (exe).

Claim: Greedy allocation rule is monotone!

\[ \frac{1}{2} \text{Awesome auction:} \quad \text{DSJC} \]

\[ - \text{max s.w.} \]

- Efficiently implementable

(1+ε)-approx. algo for knapsack

\( x^* \) is not monotone!

Can we make it monotone?

Many other settings give rise to NP-hard opt problems
Many often setting give rise to N-simple or...

> auctions facilities
> auction random time
> Spectrum auction.

\[ \alpha \approx \text{approx also for } \max \text{ s.w.} \]

\[ \beta \approx \text{awesome auction} \]

DSIC mechanism?

\[ \downarrow \]

YES, it corresponding \( x(c) \) is monotone.

But \( x(c) \) need not be monotone.

Can be made monotone if \( X \) is downward closed.

\[ \text{if } x \in X \text{ and } y \leq x, y \in X. \]

(dawik et al.)

\( x \times \times x \times \times \)

General setting: Multi parameter.

e.g. auction & k-heterogeneous items

(painting, laptop, phone, pen, ...)

\[ R: \text{ set of possible auction} \]

\[ \sqrt{\text{single-item}} \]

\[ R = \{ i \in \text{moss} / i \in N \} \]
\[ \begin{align*}
\rightarrow \mathcal{R}: \text{set of possible outcomes} \\
\rightarrow \text{For every agent } i \in \mathcal{N} \\
\quad & 1. V_i : \mathcal{R} \rightarrow \mathbb{R} \text{ (private)} \\
\quad & 2. b_i : \mathcal{R} \rightarrow \mathbb{R} \text{ (bid)}.
\end{align*} \]

Goal: Design a DSIC Mechanism.

\[ \text{Vickery-Clark-Groves (VCG) Mechanism:} \]

\[ \text{[only DSIC Mechanism]} \]

1. \( w^* \in \arg \min_{w \in \mathcal{R}} \sum_{i \in \mathcal{N}} b_i(w) \)

\[ \rightarrow \text{“pain” = “loss of value”} \]

2. \( V_i, p_i = \text{“externality” agent } i \text{ causes to } \mathcal{R} \text{ s.w.o.} \)

\[ \text{by everyone else by participating in the auction:} \]

\[ \min_{\omega} \sum_{k=1}^{n} b_k(\omega) - \sum_{k=1}^{n} b_k(\omega^*) \]
\[ \text{independent } \delta_{bi} \]

\[ h_i(b_i) \]

Thm: VCG is DSIC.

Ps:

\[ u_i = v_i(w^*) - p_i \]

\[ = v_i(w^*) - \left( \frac{h_i(b_i)}{k+1} - \sum_{k+1}^\infty b_k(w^*) \right) \]

\[ = \left[ v_i(w^*) + \sum_{k+1}^\infty b_k(w^*) \right] - h_i(b_i) \]

\[ i \text{ wants to maximize } h_i(b_i) \]

Though exp: suppose are allow i to pick the outcome.

\[ i \text{ pick: any } w \text{ with } v_i(w) + \sum_{k+1}^\infty b_k(w) \]

\[ \text{auctioneer picking: any } w \text{ with } b_i(w) + \sum_{k+1}^\infty b_k(w) \]

\[ i \text{ achieves the goal by bidding } b_i = v_i \text{ truthfully} \]
VCG is DSIC

Issues:
1. Implementation efficient?
2. Representation of bids
   10 items
3. Revenue
4. Signaling (indirect implementation)

Q: Is VCG same as Myerson for
   single parameter?
   e.g. single item, k-identical items, sponsored search?