3/29/22: Knapsack Auctions

Motivating example: selling Super Bowl ads

Setup:

- we're selling $W \in \mathbb{R}^+$ "stuff" (total ad time)
- bidders $[n]$ (companies that want to buy time for ad)
- bidder i:
  - wants w_i "stuff" (public) (time for their ad)
  - gets value v_i (private) if receive $\geq w_i$ "stuff" (value for getting enough time for their ad)
  - Quasilinear utility: $u_i(b) = v_i - p$ if receiving $\geq w_i$ "stuff", charged price p.
- Given bids $b \in \mathbb{R}^n_{\geq 0}$, divide up "stuff" among bidders.

- Single-Parameter Environment:
  - $X = \{x \in \{0,1\}^n : \sum_{i=1}^n w_i x_i \leq W\}$
  - Surplus of $x \in X$ is $\sum_{i=1}^n v_i x_i$

- Myerson's applies: just need to find monotone allocation rule

Surplus-Maximizing rule:

$$x(b) = \operatorname*{argmax}_{x \in X} \left(\sum_{i=1}^n b_i x_i \right)$$

Thm: surplus-maximizing allocation is monotone

Pf: Let $i \in [n]$, other bids b_{-i}.

$$y > z \geq 0$$

Shorthand notation:

$$x = x(b_{-i}, z)$$
$$x' = x(b_{-i}, y)$$

WTS: $x_i \leq x'_i$ (i gets at least as much stuff by bidding y as by bidding z)

$$\sum_{j \neq i} x'_j b_j + x'_i z \leq \sum_{j \neq i} x_j b_j + x_i z$$ (x surplus-maximizing allocation for (b_{-i}, z))

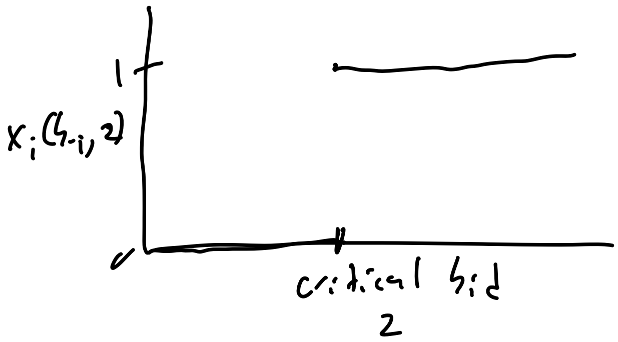
$$\sum_{j \neq i} x_j b_j + x_i y \leq \sum_{j \neq i} x'_j b_j + x'_i y$$ (x' surplus-maximizing allocation for (b_{-i}, y))

$$\Rightarrow \cancel{\sum_{j \neq i} (x'_j b_j + x_j b_j)} + x'_i z + x_i y \leq \cancel{\sum_{j \neq i} (x_j b_j + x'_j b_j)} + x_i z + x'_i y$$ (add inequalities)

$$\Rightarrow x'_i z + x_i y \leq x_i z + x'_i y$$

$$\Rightarrow x_i (y - z) \leq x'_i (y - z)$$

$$\Rightarrow x_i \leq x'_i$$ $(y - z \geq 0)$



price = critical bid

Note: Monotonicity proof didn't use anything about knapsack!

$\Rightarrow$ In SPE, surplus-maximizing allocation is monotone!

Computational Efficiency: Can we compute surplus-maximizing allocation?

Knapsack problem:

Given: - knapsack of size C

- n items, item i has size s_i, value a_i

Find $S \subseteq [n]$ which fit in knapsack, max value:

$$\max \sum_{i \in S} a_i \quad \text{subject to} \quad \sum_{i \in S} s_i \leq C$$

NP-hard!

Reduce to finding surplus-maximizing allocation of a knapsack auction:

$W = C, \quad w_i = s_i, \quad v_i = a_i$

Algorithmic Mechanism Design: relax surplus-maximization requirement to get computational efficiency

By Myerson: monotone approximation algorithm for surplus-maximization?

Thm: There is an FPTAS (fully polynomial-time approximation scheme) for knapsack: $\forall \varepsilon > 0$,

- Approximation ratio $\frac{ALG}{OPT} \geq 1 - \varepsilon$
- Running time $\text{poly}(n, \frac{1}{\varepsilon})$

But not monotone!

Goal: monotone approximation algorithm for knapsack

Today: monotone $\frac{1}{2}$-approximation

Digression: Can arbitrary α-approximations be turned monotone?

WLOG, $w_i \leq W \quad \forall i \in [n]$

Sort and reindex so

$$\frac{b_1}{w_1} \geq \frac{b_2}{w_2} \geq \cdots \geq \frac{b_n}{w_n}$$

order by "bang for the buck"

Accept bids in this order until one doesn't fit:

$$i^* = \text{largest } i \text{ s.t. } \sum_{j=1}^{i} w_j \leq W$$

Solution 1:

$$x_i = \begin{cases} 1 & \text{if } i \leq i^* \\ 0 & \text{otherwise} \end{cases}$$

give stuff to best i^* bidders

Solution 2: Let $i' = \operatorname{argmax}_{i \in [n]} b_i$ (break ties arbitrarily)

$$x'_i = \begin{cases} 1 & \text{if } i = i' \\ 0 & \text{otherwise} \end{cases}$$

Return x if $\sum_{i=1}^{n} b_i x_i \geq \sum_{i=1}^{n} b_i x'_i$, else return x'

<u>Claim</u>: Both x, x' feasible allocations

<u>Pf</u>: ✓

Thm: This allocation rule is ½-approximation

Pf sketch:

If $\sum_{i=1}^{i^*} b_i \geq \frac{1}{2} \cdot OPT$, done ✓

So suppose $\sum_{i=1}^{i^*} b_i < \frac{1}{2} \cdot OPT$

Claim: $\sum_{i=1}^{i^*+1} b_i \geq OPT$

$\Rightarrow OPT \leq \sum_{i=1}^{i^*} b_i + b_{i^*+1} < \frac{1}{2} \cdot OPT + b_{i^*+1}$

$\Rightarrow b_{i^*+1} > \frac{1}{2} \cdot OPT$

$\Rightarrow \operatorname{argmax}_{i \in [n]} b_i > \frac{1}{2} \cdot OPT$

Thm: Allocation rule is monotone

Pf: Fix $i \in [n]$, other bids b_{-i}, $y > z \geq 0$

If $x_i(b_{-i}, z) = 0 \Rightarrow x_i(b_{-i}, z) \leq x_i(b_{-i}, y)$

If $x_i(b_{-i}, z) = 1$:

If returned solution 2, z highest bid

$\Rightarrow y$ still highest bid

$\Rightarrow x_i(b_{-i}, y) = 1$

If returned solution 1, $i \leq i^*$ when $b_i = z$

$\Rightarrow i$ still $\leq i^*$ when $b_i = y$

$\Rightarrow x_i(b_{-i}, y) = 1$ ✓

<u>Revelation Principle</u>:

Our definition of incentive-compatible:

1) Every bidder has a dominant strategy
2) That dominant strategy is <u>direct revelation</u>: tell private info to mechanism

Maybe it helps to not require 2?

Thm: For every mechanism $M=(x,p)$ in which every bidder has a dominant strategy, there is an equivalent direct-revelation mechanism $M'=(x',p')$.

Pf: For each $i \in [n]$, let $s_i(v_i)$ be dominant strategy for i with private info v_i.

Given bid vector $b=(b_1,b_2,\dots,b_n)$, let

$$s(b)=(s_1(b_1),s_2(b_2),\dots,s_n(b_n))$$

Set
$$x'(b)=x(s(b))$$
$$p'(b)=p(s(b))$$

$\Rightarrow$ bidding b_i in M' same as bidding $s(b_i)$ in M

$\Rightarrow$ with private info v_i, bidding v_i is dominant strategy