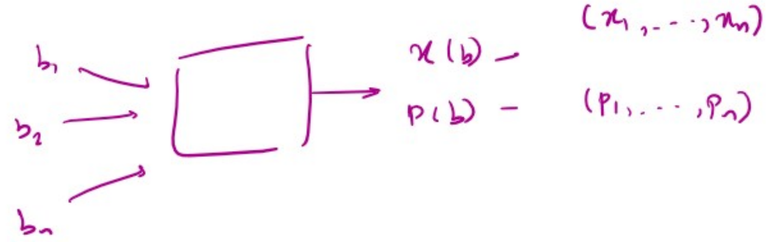
Auction Design (contd.)

Thursday, November 2, 2023 10:57 AM



Awesome Auctions.

1) DSIC – (x, p)

2) Social welfare/surplus maximization.
   $\sum_i b_i x_i$

3) Polynomial time implementable.

→ Single parameter setting → v_i → value per unit stuff.

$b_1, b_2, \ldots, b_n$

- Assume true values
- Ensure (2) & (3)
- Design payments so that (1) is satisfied.

→ Knapsack Auctions

- commercial → W secs.
- n bidders.

  advertisement → w_i secs

  v_i

$(x_1, x_2, \ldots, x_n)$ → 0/1 vectors

$(b_1, b_2, \ldots, b_n)$

$$\left.\begin{aligned} \max \quad & \sum_{i \in [n]} b_i x_i \\ \text{s.t.} \quad & \sum w_i x_i \le W \\ & x_i \in \{1, 0\} \end{aligned}\right\} \rightarrow \text{NP-hard.}$$

$x_i \in \{1, 0\}$

→ Algorithm 1:

W : 120

HP, Dell, Ice cream, . .

v_i

w_i – 10sec 20sec.

I/p: $b_1, b_2, \dots, b_n$

$w_1, w_2, \dots, w_n$

W.

1) Reorder the agents:

$$\frac{b_1}{w_1} \geq \frac{b_2}{w_2} \geq \dots \geq \frac{b_n}{w_n}. \rightarrow$$

2) Pick agents greedily in above order till $\sum_{i=1}^{k} w_i \leq W$ is violated. (say k advertisers → are picked)

3) If $\sum_{i=1}^{k} b_i \leq \max_{i \in [n]} b_i \rightarrow$

$\hookrightarrow i^*$

then $x_{i^*} = 1$

else

$x_1 = x_2 = \dots = x_k = 1$

4) Return x.

Claim1: Allocation returned by Algorithm 1 is monotone (exe).

Claim 2: Allocation returned by Algorithm 1 is $\frac{1}{2}$-approximation to maximum social welfare.

Proof: $\underbrace{\frac{b_1}{w_1} \geq \frac{b_2}{w_2} \geq \dots \frac{b_k}{w_k}}$ $\geq \frac{b_n}{w_n}$

$(W - \sum^{k} w_i) \;|$

$$\left(W - \sum_{i=1}^{k} w_i\right)$$

$$\to \quad \sum_{i=1}^{k} b_i + \left(W - \sum w_i\right) \cdot \frac{b_{k+1}}{w_{k+1}}$$

$$\max\left\{\sum_{i=1}^{k} b_i, b_{k+1}\right\}$$

$$\geq \frac{1}{2}\left(\sum_{i=1}^{k} b_i + b_{k+1}\right)$$

By Claim 1 + Myerson.

$\Downarrow$

(P) $\exists$ (x, p) in DSIC

Knapsack: FPTAS

$(1-\varepsilon)$ in poly in input, $1/\varepsilon$

If FPTAS's allocation was monotone

$\Downarrow$

DSIC, poly time, $(1-\varepsilon)$ SW maximizing.

$\hookrightarrow$ not monotone

But can be tweaked to be monotone!!!

Q: Given an algorithm that outputs α-approx^n to social welfare, can we tweak the algorithm to find monotone allocation that is an α-approx.

Chawla et al.

when feasible allocations are downward closed, this is true.

$(x_1, x_2, \ldots, x_n) \in F$

$(y_1, y_2, \ldots, y_n)$ s.t. $y \leq x$

$\in F$

$(u_1, u_2, \ldots, u_n)$

↳ EF

→ Revealation principle

DSIC

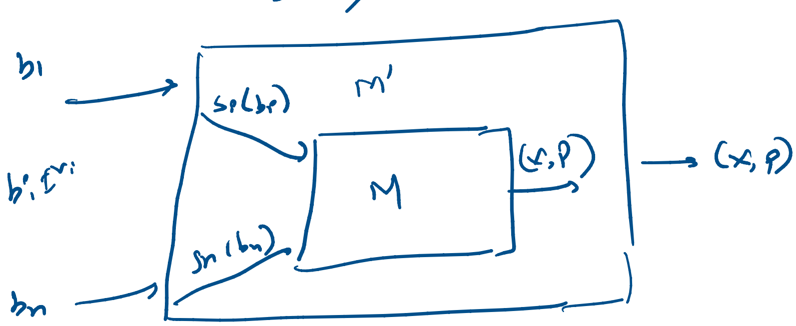
1) ∃ a dominant strategy

2) Truthful bidding is the dominant strategy

Claim: If ∃ mechanism M s.t. ① is true then

∃ a mechanism M' s.t. DSIC

Proof: For agent i, in M.



$S_i(V_i)$

Multi Parameter Setting.

e.g. Auction of k heterogeneous items

(laptop, painting, pen ...)

Ω : Set of all possible outcomes.

| single item setting.
Ω : {i | i wins}.

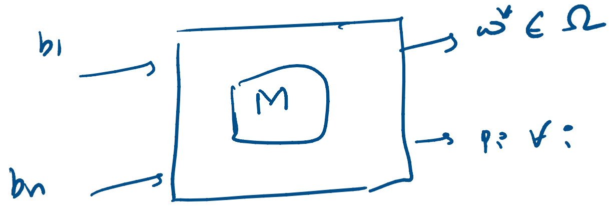
– n agents.

For every agent $i \in N$

$\triangleright v_i : \Omega \to \mathbb{R}$ (Private)

2) $b_i : \Omega \to \mathbb{R}$

Goal: Design a DSIC mechanism.



$U_i := v_i(\omega^*) - p_i$

Vickrey-Clark-Grove [VCG Mechanism].

[only DSIC mechanism]

1) $\omega^* \in \arg\max_{\omega \in \Omega} \sum_{i \in [n]} b_i(\omega)$

2) $\forall i \quad p_i = \max_{\omega \in \Omega} \sum_{j \in [n], j \neq i} b_j(\omega) - \sum_{j \in [n], j \neq i} b_j(\omega^*)$