CO 456 Fall 2024 Lecture 16 (Wednesday November 6)

7 (Backtrack) Auction mechanism design

7.2 Myerson's lemma

Theorem 15. (Myerson's Lemma) For a single parameter environment, an allocation rule $x : B \to X$ is implementable if and only if x is monotone. Moreover, given a monotone allocation rule $x : B \to X$, there exists a unique payment rule $p : B \to \mathbb{R}^N$ such that (x, p) is DSIC, and $p_i(b) = 0$ whenever $b_i = 0$.

For the converse, we will prove the case where the allocation rule x is piecewise constant.

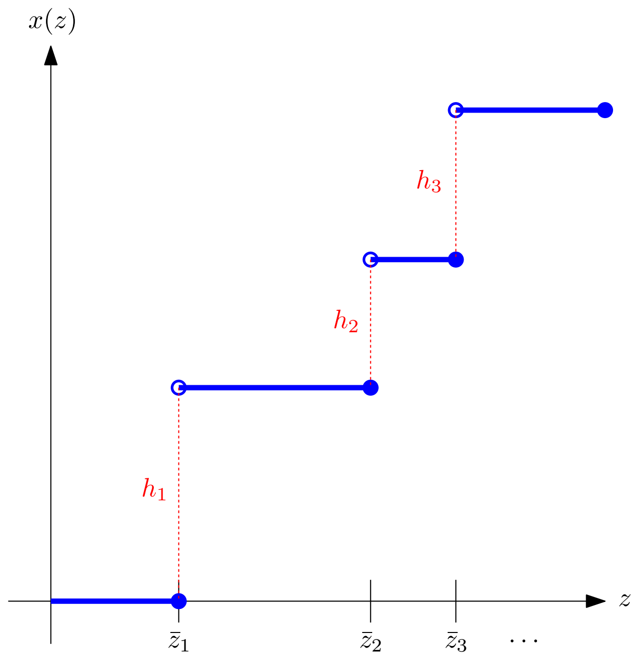
Approach. Assume x is monotone and (x, p) is DSIC. Derive p.

Then prove that using this particular p, (x, p) is indeed DSIC.

Monotone piecewise constant x. Jump points $\bar{z}_1 < \bar{z}_2 < \cdots < \bar{z}_q$.

x is constant in the intervals $[0, \bar{z}_1], (\bar{z}_1, \bar{z}_2], \ldots, (\bar{z}_{q-1}, \bar{z}_q], (\bar{z}_q, \infty)$.

Define h_j to be the jump at $\bar{z}_j$.



Lemma 15.1. If x is monotone and piecewise constant, and (x, p) is DSIC with $p_i(b) = 0$ whenever $b_i = 0$, then p is unique.

Proof. We first show that p is also piecewise constant.

Assume $y > z$. Suppose $z \in (\bar{z}_j, \bar{z}_{j+1})$ for some j.

We assume DSIC, so the payment sandwich applies.

$$z[x(y) - x(z)] \le p(y) - p(z) \le y[x(y) - x(z)].$$

Divide every term by $y - z$ to get

$$z \cdot \frac{x(y) - x(z)}{y - z} \leq \frac{p(y) - p(z)}{y - z} \leq y \cdot \frac{x(y) - x(z)}{y - z}.$$

Take the limit as $y \to z^+$ to get

$$\lim_{y \to z^+} \frac{x(y) - x(z)}{y - z} = x'(z) = 0 \quad \text{since } x \text{ is constant at } z.$$

Both ends of the payment sandwich approach 0. By Squeeze Theorem,

$$\lim_{y \to z^+} \frac{p(y) - p(z)}{y - z} = p'(z) = 0.$$

So p is piecewise constant with jumps at $\bar{z}_1, \ldots, \bar{z}_q$. What is this jump?

Let $z = \bar{z}_j$ for some j. From the payment sandwich, take the limit as $y \to z^+$ to get

$$\lim_{y \to z^+} y(x(y) - x(z)) = \lim_{y \to z^+} z(x(y) - x(z)) = z \cdot h_j.$$

By Squeeze Theorem,

$$\lim_{y \to z^+} p(y) - p(z) = z \cdot h_j = \bar{z}_j h_j.$$

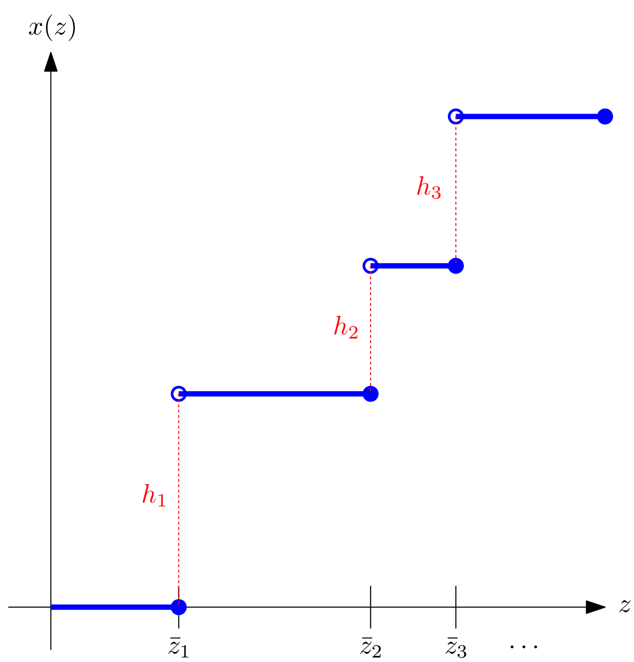
So the payment jump at $\bar{z}_j$ is $\bar{z}_j h_j$.

By assumption, $p(0) = 0$, so this uniquely defines p: If j is the largest index where $z \geq \bar{z}_j$, then

$$p(z) = p_i(z, b_{-i}) = \sum_{k=1}^{j} \bar{z}_k h_k.$$

□

Illustration.



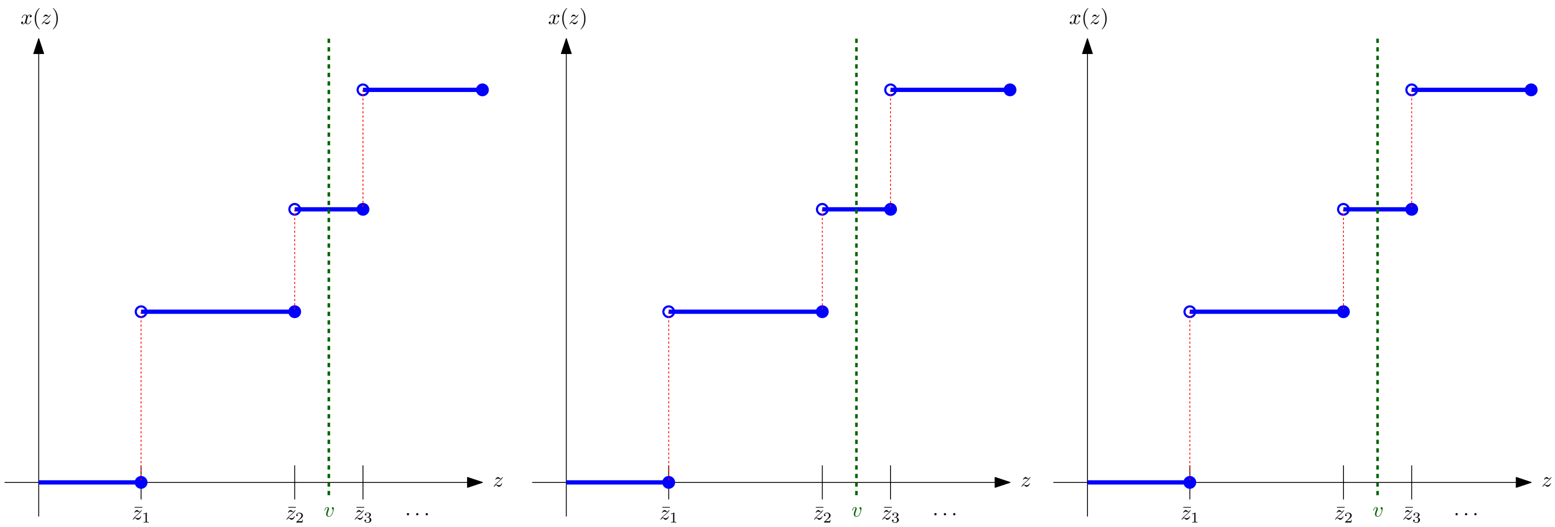
Graphical interpretation of p. The payment is the area to the left of z in the graph of x. This can be generalized to any monotone allocation rule.

Lemma 15.1 only says that if (x, p) is DSIC, then there is only one possible choice of p. We still need to prove that (x, p) is DSIC.

Lemma 15.2. The payment rule p from Lemma 15.1 is DSIC.

Proof by picture.

(1) Bidding their valuation. (2) Underbid. (3) Overbid.

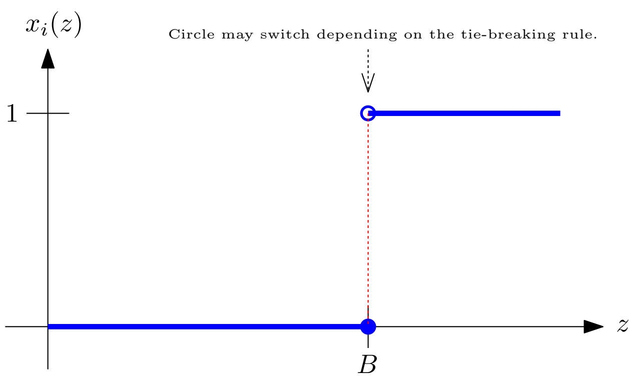


Utilities for (2) and (3) are at most the utility of (1). Utility of (1) is non-negative. Hence DSIC. $\square$

7.3 Applying Myerson's Lemma

Single item auction. Let x be the allocation rule that gives the item to the highest bidder. This is monotone.

Consider a player i and bids $b_{-i} \in B_{-i}$. The allocation for i has a jump point, at $B = \max_{j \neq i} b_j$.

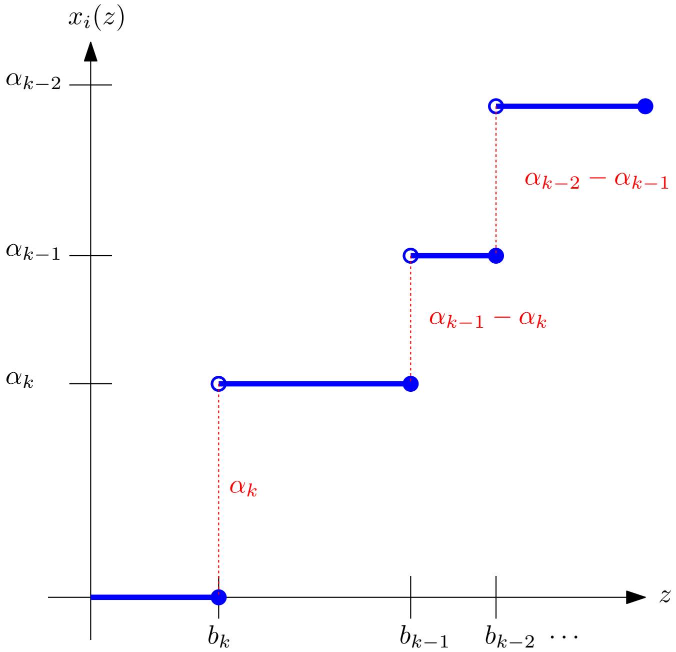


When $b_i > B$, player i wins. Payment is $B \cdot 1 = B$. This is the second-price payment rule, it is unique for x.

Sponsored search auction. k slots with CTR $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_k$. Let x be the allocation rule that assigns α_j to the j-th highest bidder. This is monotone.

Consider a player i and bids $b_{-i} \in B_{-i}$. Assume $b_1 \geq b_2 \geq \cdots \geq b_k$ in b_{-i}.

There are jump points at $b_k, b_{k-1}, \ldots$. Player i gets α_k if $b_k < z \leq b_{k-1}$, α_{k-1} if $b_{k-1} < z \leq b_{k-2}$, $\ldots$. Each jump is $\alpha_{j-1} - \alpha_j$.



The payment is

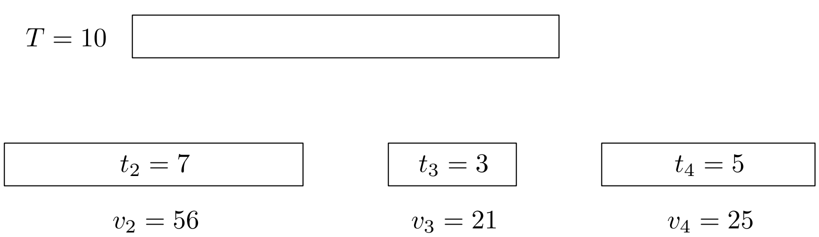
$$p_i(z) = \sum_{\{j \,:\, b_j < z\}} b_j(\alpha_j - \alpha_{j+1}).$$

7.4 Auctions that are almost ideal

Recall: An ideal auction is (1) DSIC; (2) welfare-maximizing; and (3) efficient.

Example. We are managing advertising slots for a TV station. We have T seconds to fill. There are potential advertisers N. Advertiser i has an ad of length $t_i \leq T$ and gets a value of v_i. Run an auction to determine which ads to air.

One instance.



Feasible allocations: $X = \{x \in \{0,1\}^N \ : \ \sum_{i \in N} t_i x_i \leq T\}$.

Social welfare: $\sum_{i \in N} v_i x_i$

To maximize social welfare, we need to find $\max\{v^T x \ : \ x \in X\}$.

Recall. We first assume players bid truthfully and find an allocation that is welfare maximizing. Myerson's Lemma gives a payment rule that is DSIC.

Problem. Maximizing social welfare is equivalent to the knapsack problem. This is NP-hard.

We cannot fulfill (2) and (3) simultaneously.

Solution. Relax (2) or (3). We cannot give up (3) in practice, so we need to give up (2).

Instead of finding a welfare-maximizing allocation, we find a close approximation that is fast to run.

7.5 Approximating maximum welfare

Starting idea. We want to fill each unit of space with as much value as possible. Calculate value density of the items, and sort them in non-increasing order $\frac{v_1}{t_1} \geq \frac{v_2}{t_2} \geq \cdots \geq \frac{v_n}{t_n}$.

Example. $T = 10$

i	1	2	3	4
t_i	2	7	3	5
v_i	20	56	21	25
v_i/t_i	10	8	7	5

First try. Pick items $1, 2, \ldots$ until we cannot put the next one in the knapsack.

In the example above, pick 1 and 2. $t_1 + t_2 = 9 \leq 10$, no room for 3. Total value 76.

This can be very bad. Say $v_1 = 1, t_1 = 1$ and $v_2 = T - 1, t_2 = T$. 1 has higher density than 2. If we pick 1, then we do not have room for 2. We get a value of 1. Obvious optimal solution is to pick 2 instead.

Better idea. Suppose we stop at item i. Check the value of item $i+1$. If $v_{i+1} > v_1 + \cdots + v_i$, then $i+1$ is the only winner.

Approximation algorithm. Assume $N = \{1, \ldots, n\}$ with $\frac{v_1}{t_1} \geq \frac{v_2}{t_2} \geq \cdots \geq \frac{v_n}{t_n}$.

- Find i such that it is the largest index with $t_1 + \cdots + t_i \leq T$.
- If $v_1 + \cdots + v_i \geq v_{i+1}$, then $\{1, \ldots, i\}$ are the winners. Otherwise, $\{i+1\}$ is the only winner.

Suppose APX is the value of the winners produced by the approximation algorithm, and OPT is the actual optimal value. How good is APX?

Theorem 15.3. $APX \geq \frac{1}{2} OPT$.

Proof. OPT is the optimal solution of the integer program (IP): $\max\{v^T x \ : \ t^T x \leq T, x_i \in \{0,1\} \ \forall i\}$.

Its LP relaxation is (LP): $\max\{v^T x \ : \ t^T x \leq T, 0 \leq x_i \leq 1 \ \forall i\}$. (Solutions can be fractional.)

Let i be the largest index with $t_1 + \cdots + t_i \leq T$. Consider the solution x where

$$x_1 = \cdots = x_i = 1, \quad x_{i+1} = \frac{T - t_1 - \cdots - t_i}{t_{i+1}}, \quad x_{i+2} = \cdots = x_n = 0.$$

(We are filling the remaining space of the knapsack with a fraction of item $i+1$.)

As we have completely filled in the knapsack with the highest-density items, x is optimal. (Alternative: Prove this using duality.)

Suppose optimal value of (LP) is v^* (achieved by x). Since (LP) is a relaxation of (IP), $OPT \leq v^*$. We want to prove that $APX \geq v^*/2$.

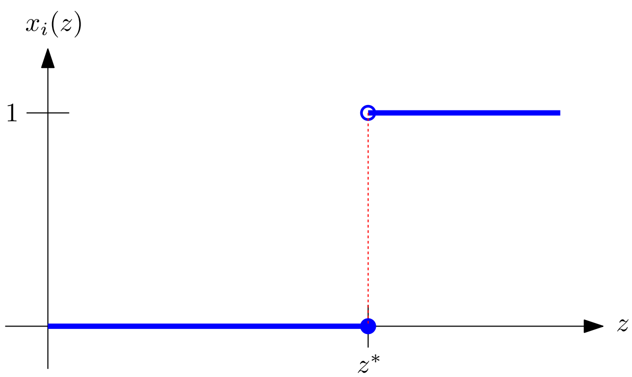
Let $S = \{1, \ldots, i\}$. We see that $v^* \leq v_1 + \cdots + v_i + v_{i+1} = v(S) + v_{i+1}$. Two cases:

- If $v(S) \geq v_{i+1}$, then the algorithm picks S, so $APX = v(S)$. Then
$$v^* \leq v(S) + v_{i+1} \leq v(S) + v(S) = 2 \cdot APX.$$
- If $v(S) < v_{i+1}$, then the algorithm picks $\{i+1\}$, so $APX = v_{i+1}$. Then
$$v^* \leq v(S) + v_{i+1} \leq v_{i+1} + v_{i+1} = 2 \cdot APX.$$

So $APX \geq \frac{v^*}{2} \geq \frac{1}{2} OPT$. □

Exercise? The allocation rule derived from the approximation algorithm is monotone.

Myerson's Lemma gives the payment rule. A player's allocation has one jump point at some point z^*. The payment is z^*.



How to find z^*? Figure out the point at which the approximation algorithm will switch between picking the item and not picking the item.

Example. $T = 10$

i	1	2	3	4
t_i	2	7	3	5
v_i	20	56	21	25
v_i/t_i	10	8	7	5

The algorithm picks 1 and 2. What is the payment for player 2? Lower v_2 until the algorithm does not pick v_2.

As long as $v_2/t_2 \geq 7$, the algorithm will pick 2. So player 2 can go down to 49 and still be picked.

What happens if $v_2 < 49$? Say $v_2/t_2 \geq 5$. The algorithm picks 1 and 3 with combined value 41.

If $v_2 > 41$, then the algorithm will still pick 2.

If $v_2 \leq 41$, then the algorithm will stick with 1 and 3.

The jump point is at 41, so the payment is 41.