601.436/636 Algorithmic Game Theory **Lecturer:** Michael Dinitz
Topic: Single-Parameter Environments: Myerson's Lemma **Date:** 4/2/20
Scribe: Michael Dinitz

17.1 Introduction

Last class we showed that for a single-item sealed-bid auction, the second-price auction was "awesome": it was incentive-compatible (bidding their true valuation is a dominant strategy for all players), it maximized social welfare, and was efficiently computable. Then we talked about sponsored search auctions, where we wanted to auction off ad slots with varying click-through rates. These were different from single-item auctions, since we had multiple heterogeneous items, but were similar in that each bidder had a single private parameter v_i. We asked whether we could design awesome auctions for sponsored search, and talked about a two-stage design process: first figure out the allocation which maximizes social welfare if we assume that players bid truthfully, and then figure out prices to make this incentive-compatible so that they will bid truthfully. The first part of this was pretty easy (at least for sponsored search), but the second part was the tricky thing.

Today we're going to generalize to a setting called *single-parameter environments* which includes both single-item and sponsored search, and show how to solve the second part of this design process using *Myerson's Lemma*.

17.2 Setting

17.2.1 Single-Parameter Environments

- Bidders/players $[n]$
- Bidder i has private valuation v_i: think of this as "value per unit of stuff".
- There is a *feasible set* $X \subseteq \mathbb{R}^n$. For each $\vec{x} \in X$ and $i \in [n]$, think of x_i as "amount of stuff bidder i gets in allocation $\vec{x}$". Note that X doesn't (in this definition) have to be "nice" in any way.
- The social surplus of $\vec{x} \in X$ is $\langle \vec{v}, \vec{x} \rangle = \sum_{i=1}^n v_i x_i$, so to maximize social surplus we want to choose the allocation $x \in X$ maximizing this. We will often be trying to maximize this, but not necessarily all the time.

It's not too hard to see that both the single-item and the sponsored search setting are examples of single-item environments. For single-item auctions, $X = \{\vec{x} \in \{0,1\}^n : \sum_{i=1}^n x_i = 1\}$. For sponsored search it's a little more complicated, but recall that the k slots had click through rates $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_k$. Let $\alpha_0 = 0$. Then we can set X to be the set of vectors where every entry is either a click through rate of a slot or zero, and each slot appears at most once. Formally, this is

$$X = \{\vec{x} \in \mathbb{R}^n : \exists f : [n] \to \{0\} \cup [k] \text{ where } x_i = \alpha_{f(i)} \ \forall i \in [n] \text{ and } f(i) = f(j) \implies f(i) = 0 \ \forall i \neq j\}.$$

Then any $\vec{x} \in X$ does indeed result in social surplus of $\sum_{i=1}^{n} v_i x_i$.

17.2.2 Sealed-Bid Auction

We talked about this quite a bit last time, but let's formalize how a sealed-bid auction works. There are three steps:

1. Collect bids $\vec{b} = (b_1, b_2, \dots, b_n) \in \mathbb{R}^n$ from the bidders (bidder i tells us b_i)
2. Choose a feasible allocation $\mathbf{x}(\vec{b}) \in X$. Here $\mathbf{x}$ is called the *allocation rule*, and we'll abuse notation a little bit and set $x_i(\vec{b}) = \mathbf{x}(\vec{b})_i$. This notation might be a little confusing, since here $\mathbf{x}$ is a *function* from bids to feasible allocations and earlier it was an allocation. I'm using this notation to be consistent with Roughgarden's book.
3. Choose payments $\mathbf{p}(\vec{b}) \in \mathbb{R}^n$. We will similarly let $p_i(\vec{b}) = \mathbf{p}(\vec{b})_i$.

We'll focus on quasilinear utilities, so $u_i(\vec{b}) = v_i x_i(\vec{b}) - p_i(\vec{b})$. We're also going to focus on payment rules where $p_i(\vec{b}) \in [0, b_i \cdot x_i(\vec{b})]$. So payments cannot be negative (the auctioneer is not allowed to pay the bidder), and they can't be more than the value of a truthful bidder (so truthful bidders will never have negative utility).

Note that the allocation rule and the payment rule completely define the mechanism, so we'll say that a mechanism is a pair $(\mathbf{x}, \mathbf{p})$. They are what we are trying to design. Moreover, once the functions $\mathbf{x}$ and $\mathbf{p}$ have been fixed, this is now a well-defined game (albeit one with an infinite set of pure strategies): the strategies for each player are their bids, and we have a utility function for each player from strategy profiles to real numbers. But, like last time, we won't really think of this as a game.

17.3 Myerson's Lemma

Now let's formalize what we're trying to do, before we actually do it.

Definition 17.3.1. *An allocation rule* $\mathbf{x}$ *is* implementable *if there is a payment rule* $\mathbf{p}$ *such that* $(\mathbf{x}, \mathbf{p})$ *is incentive-compatible.*

So the question we ended with last time is: for sponsored search, is the greedy allocation implementable?

For the single-item setting, we proved last time that allocating to the highest bidder is implementable (the payment rule is to charge the second-highest bid). What about other allocation rules? For example, is "give the item to the second-highest bidder" implementable?

We're going to do more than just prove that the greedy allocation for sponsored search is implementable, or that "give to the second highest" in the single item setting is not. We're going to precisely characterize which allocation rules are implementable and what payment rule to use.

Definition 17.3.2. *An allocation rule* $\mathbf{x}$ *is* monotone *if for all* $i \in [n]$ *and other bids* b_{-i}*, the allocation* $x_i(b_{-i}, z)$ *is nondecreasing in* z*.*

In other words, for every player, increasing their bid cannot result in them getting less stuff. So, for example, in the single item setting the "give tot he highest bidder rule" is monotone, but "give to the second highest bidder" is not monotone.

The main theorem we're going to prove today is the following, which we'll state in three parts, and which is known as *Myerson's Lemma.*

Theorem 17.3.3 (Myerson '81)**.**
1. *An allocation rule* $\mathbf{x}$ *is implementable* if and only if $\mathbf{x}$ *is monotone.*
2. *If* $\mathbf{x}$ *is monotone, then there is a* unique *payment rule* $\mathbf{p}$ *such that* $(\mathbf{x},\mathbf{p})$ *is incentive-compatible (assuming* $b_i = 0 \implies p_i(b) = 0$*).*
3. *The unique payment rule from the previous part is given by an explicit formula.*

This theorem means that if we figure out the optimal allocation for surplus maximization (part 1 of our design process), we then just need to check whether it's monotone! If it's not, then there is no incentive-compatible mechanism that maximizes social surplus. If there is, then we can write down a formula for the prices to make it incentive-compatible.

17.3.1 Proof of Myerson's Lemma

Let's focus on part 1 of the theorem – parts 2 and 3 will follow naturally.

Implementable $\implies$ Monotone. Suppose that $\mathbf{x}$ is implementable. Then there exists $\mathbf{p}$ such that $(\mathbf{x},\mathbf{p})$ is incentive-compatible. Let $i \in [n]$, and let b_{-i} be a set of other bids. To simplify notation, let $x(z) = x_i(b_{-i}, z)$ be the allocation to i as a function of its bid (when all others are fixed) and let $p(z) = p_i(b_{-i}, z)$ be the price that i is charged. So we want to show that x is nondecreasing.

Let $y \geq z \geq 0$. It could be the case that $v_i = z$. If this is the case, then since the mechanism is incentive-compatible player i cannot do any better by bidding y (overbidding). Thus

$$\begin{aligned} & z \cdot x(z) - p(z) \geq z \cdot x(y) - p(y) \\ \implies & p(y) - p(z) \geq z \cdot x(y) - z \cdot x(z) = z(x(y) - x(z)) \end{aligned}$$

Similarly, it could be the case that $v_i = y$. If this is the case, then since the mechanism is incentive-compatible bidder i cannot do better by bidding z (underbidding). Thus

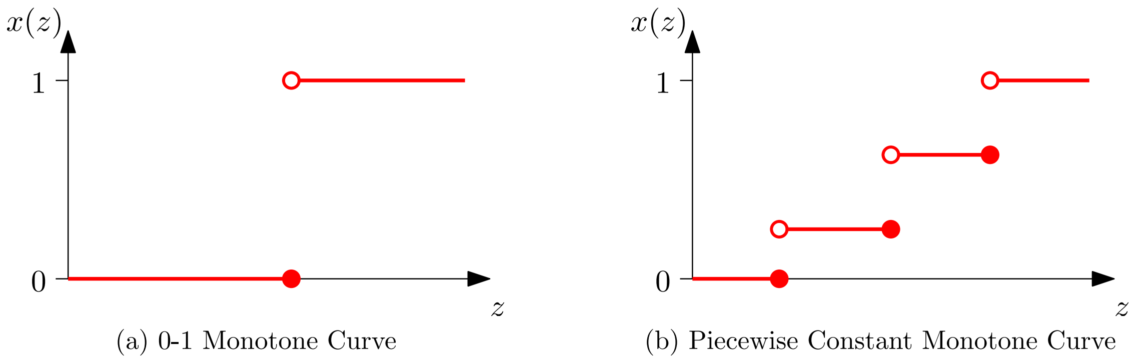
$$\begin{aligned} & y \cdot x(y) - p(y) \geq y \cdot x(z) - p(z) \\ \implies & p(y) - p(z) \leq y \cdot x(y) - y \cdot x(z) = y(x(y) - x(z)) \end{aligned}$$

So we have lower and upper bounds on $p(y) - p(z)$. These bounds are called the "payment difference sandwich" bounds, since they sandwich the payment difference. We will also use them later, when we're proving the other direction. Combining them gives

$$\begin{aligned}
&z(x(y) - x(z)) \le y(x(y) - x(z)) \\
\implies &(y - z)(x(y) - x(z)) \ge 0 \\
\implies &x(y) - x(z) \ge 0 && \text{(since } y \ge z) \\
\implies &x(y) \ge x(z)
\end{aligned}$$

Thus $\mathbf{x}$ is monotone.

Monotone $\implies$ Implementable. We want to show that there is some $\mathbf{p}$ so that $(\mathbf{x}, \mathbf{p})$ is incentive-compatible. Let $i \in [n]$. and fix some other bids b_{-i}. Let $x(z) = \mathbf{x}(b_{-i}, z)$, and let's try to design $p_i(z) = \mathbf{p}(b_{-i}, z)$ so that the best thing for player i to do is set $b_i = v_i$. Let's focus on a specific case, since it's a little easier and involves all of the main ideas (just avoid some of the annoying math): the case when $\mathbf{x}$ is monotone piecewise constant. In other words, x (the allocation function for player i when the other bids have been fixed to b_{-i} is flat except for a constant number of breakpoints that "jump" to the next flat part (see Figure 17.3.1).



(a) 0-1 Monotone Curve (b) Piecewise Constant Monotone Curve

Figure 17.3.1: Monotone piecewise constant

Note that both the single-item allocation function (give to the highest bidder) and the sponsored search greedy allocation function have this form, so this is a particularly interesting special case.

So we need to figure out what $p(z)$ should be. What constraints do we know are true for p? Since we want $(\mathbf{x}, \mathbf{p})$ to be incentive-compatible, the payment difference sandwich inequalities must be satisfied by p, i.e.,

$$z(x(y) - x(z)) \le p(y) - p(z) \le y(x(y) - x(z))$$

for $y \ge z$. So fix z, and let y tend to z from above. If z is not a breakpoint, then $x(y) - x(z)$ tends to 0, so $p(y) - p(z)$ must also tend to 0. Thus for parts of $\mathbb{R}_{\ge 0}$ where x is flat, p must also be flat.

What about when z is a breakpoint? Suppose that there is a jump of magnitude h at z, i.e., $\lim_{y \to z^+} x(y) = x(z) + h$ (where this limit notation means the one-sided limit from above). Then the payment difference inequalities imply that $\lim_{y \to z^+} p(y) = p(z) + zh$. In other words, at a breakpoint, the price function jumps by the value of the breakpoint times the amount that the allocation jumps.

Assuming that $p(0) = 0$ (for normalization, which part 2 of Myerson's Lemma states explicitly), this now gives a simple formula for p:

$$p(b_i) = \sum_{j=1}^{\ell} z_j \cdot (\text{jump in } x \text{ at } z_j)$$

where $z_1, z_2, \ldots, z_\ell$ are the breakpoints of x in $[0, b_i)$. So now, assuming that $\curvearrowleft$ is monotone piecewise constant, we can explicitly compute $p(\vec{b})$: to compute the price $p_i(\vec{b})$ charged to player i, we just look at the allocation function of player i (where all other players have their bids fixed to b_{-i}), check where the breakpoints are that are below b_i, and compute the price accordingly.

It's not too hard to extend this to general monotone $\mathbf{x}$, but I won't do the details here. Basically, if x is differentiable (rather than piecewise constant) then we can look at the payment sandwich inequalities and again take the limit as $y \to z$ to get that $p'(z) = z \cdot x'(z)$, which when combined with the normalization $p(0) = 0$ yields that

$$p(b_i) = \int_0^{b_i} z \cdot \frac{d}{dz} x(z)\ dz$$

This can then be extended to non-differentiable monotone functions using some calculus tricks. So we have an explicit formula for the prices, so as long as we can (numerically) compute these integrals and derivatives then we can compute prices (part 3 of the theorem).

It's important to note that, because of the payment sandwich inequalities and the normalization, the price function that we designed is the only possible price function that could possible be incentive compatible (part 2 of the theorem). So if we can prove that $(\mathbf{x}, \mathbf{p})$ is in fact incentive compatible, we'll be done.

I'm not going to give the end of this proof, that $(\mathbf{x}, \mathbf{p})$ is incentive compatible, in general. Let's just do a proof by picture (Figure 17.3.2, stolen from Roughgarden's book) for the monotone piecewise constant case.

The first column is the truthful bidding case, the second is the overbidding case, and the third is the underbidding case. The first row is the value to player i, the second is the payment from player i, and the third is the utility of player i. As you can see, the best thing for player i to do is bid truthfully.

17.4 Examples: single-item and sponsored search

17.4.1 Single-item

Let's re-think about the single-item case now that we have the power of Myerson's lemma. Clearly the social surplus-maximizing allocation rule is $\mathbf{x}(\vec{b})$ is the vector with a 1 in the index with maximum bid and a 0 everywhere else. Stated differently, if we fix player i and other bids b_{-i}, the

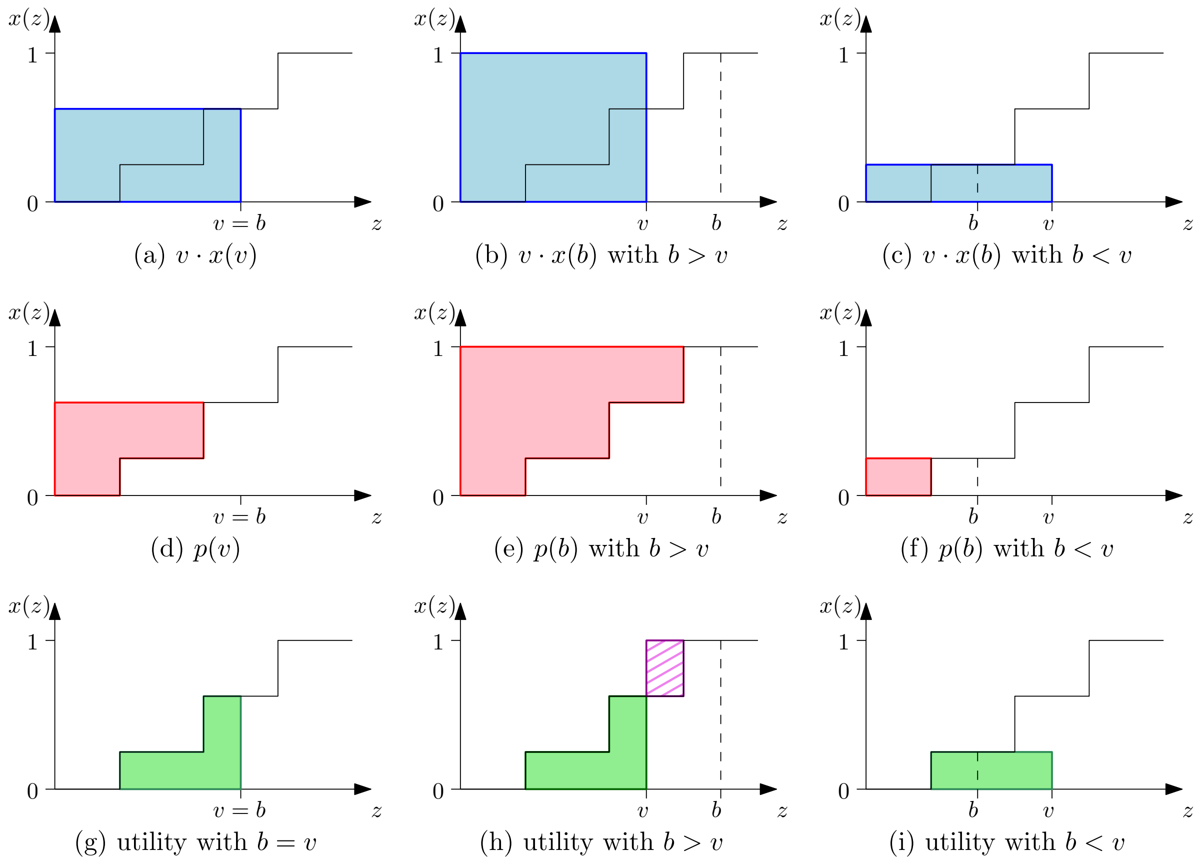


Figure 17.3.2: Proof by picture that $(\mathbf{x}, \mathbf{p})$ is incentive-compatible

allocation function for player i is (if we don't worry about ties).

$$x(b_{-i}, z) = \begin{cases} 0 & \text{if } z \le \max_{j \ne i} b_j \\ 1 & \text{if } z > \max_{j \ne i} b_j \end{cases}$$

This is clearly monotone for every i and b_{-i}, so the overall allocation function $\mathbf{x}$ is monotone and so by Myerson's lemma there is a pricing function that makes the whole thing incentive compatible. Looking at the formula for the piecewise constant case (which this is), we see that the price charged to player i (as a function of their bid z when the other bids are b_{-i}) is

$$p(b_{-i}, z) = \begin{cases} 0 & \text{if } z \le \max_{j \ne i} b_j \\ \max_{j \ne i} b_j \cdot 1 & \text{if } z > \max_{j \ne i} b_j \end{cases}$$

This is the second-price auction!

17.4.2 Sponsored search

Fix a player i and all other bids b_{-i}. Without loss of generality, rename the players so that $b_1 \geq b_2 \geq \cdots \geq b_{n-1}$, and $i = n$. Recall that $\alpha_1 \geq \alpha_2 \cdots \geq \alpha_k$ were the click through rates of the k slots. Then the allocation function for player n is

$$x(b_{-n}, z) = \alpha_j \quad \text{if } b_j < z < b_{j-1}$$

Note that the allocation jumps from α_{j+1} to α_j once the player has the j'th highest bid, i.e., when $z = b_j$. Thus we get that the payment for the player who is the the j'th largest bidder is

$$p_j(\vec{b}) = \sum_{\ell=j}^{k} b_{\ell+1}(\alpha_\ell - \alpha_{\ell+1})$$

since this is the sum over all breakpoints before b_j of the value of the breakpoint times the jump in the allocation.

Generalized Second Price. We know from Myerson's Lemma that this payment scheme is the *only* payment scheme that makes the surplus-maximizing allocation incentive-compatible. But if you didn't know Myerson's Lemma, and only knew about second-price auctions, you might naturally use a different auction: the j'th highest bidder pays the $(j+1)$st highest bid. This is known as the *Generalized Second Price* auction, and we know from Myerson that it *is not* incentive compatible. This is what is actually used in practice by many search companies who didn't have a mechanism design team when they started, and since it is not incentive compatible it is much trickier to reason about. See the problems in Roughgarden's book for some interesting equilibrium-based analysis of GSP.