Algorithmic Game Theory November 25, 2013

Lecture 6

Lecturer: Michal Feldman Scribe: Moshe Sulamy, Yehonatan Ginzberg, Alon Ardenboim

1 Single Item Auction

There's a single seller who wants to sell a single item. There are n bidders that bid for the item.

Our model: Each agent $i \in \{1, \ldots, n\}$ has a private value v_i which stands for how much the agent is willing to pay for the item. The value is private in the sense that it is known only to the agent and not to the seller, or to the other agents. The utility of agent i is quasi-linear:

$$u_i = \begin{cases} v_i - p & \text{if } i \text{ wins} \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

where p is the payment incurred by agent i. We assume: an agent isn't required to pay if she doesn't receive any items. The auction is a ***sealed bid auction***, meaning that it satisfies the following conditions:

1. Every agent puts her bid b_i in a sealed envelope and hands it to the seller.
2. The seller collects the envelopes and decides:
   (a) Who gets the product (**allocation rule**).
   (b) How much the winner pays.

1.1 First Price Auction

The seller gives the item to highest bidder. The bidder who received the item pays her bid. This auction is a very complicated auction. First of all, the agents have a hard time to devise a strategy. It isn't at all clear which strategy would maximize a bidder's utility. Bidding your value would be a non-optimal strategy, because no matter what the outcome, each agent's utility would be 0 (if she wins, her utility is $v_i - p = v_i - v_i = 0$). Not only that, but the auction is also hard to analyze from the perspective of the auction designer. This auction is not truthful, and the agents' strategies aren't clear, therefore the outcome of the auction isn't clear.

1.2 Second Price Auction

Once again, the highest bidder gets the item. She pays the value of the second highest bid. This auction is used in many real market settings, for instance, eBay's auctions are carried this way (the auction is dynamic, but the outcome of the auction is identical to the second price auction).

Theorem 1 *For each agent, the dominant strategy is truth-telling ($b_i = v_i$ for each $i \in \{1, \ldots, n\}$).*

Proof: We fix agent i, its value v_i, and all the other agents' bids $\mathbf{b}_{-i}$. Let $\bar{b} = \max_{j \neq i} b_j$ be the highest bid of an agent which is not agent i. If agent i wins her utility is $v_i - \bar{b}$, otherwise it is 0. We need to consider two cases:

1. $v_i < \bar{b}$: The best thing agent i can do is lose and have a 0 utility. She'll do just that if she bids ($b_i = v_i$).
2. $v_i \geq \bar{b}$: The highest utility agent i can get is $v_i - \bar{b}$. Again, truth-telling will get her that.

■

This auction is called ***Vickery auction***. Another nice property of Vickery auctions is the following:

Theorem 2 *In a Vickery auction every (rational) agent has a non-negative utility.*

Proof: If an agent doesn't get the item, her utility is 0. If an agent gets the item, her utility is $v_i - \bar{b} > 0$. ■

This property is often called ***individual rationality*** (IR) or ***voluntary participation***. Another desired property of this auction is that the agent that gets the item is the one who values it the most. The outcome of the auction therefore maximizes social welfare. We have shown the 3 following important properties of Vickery auctions:

1. The mechanism is truthful.
2. The mechanism maximizes the social welfare.
3. All of the computation done by the mechanism takes polynomial time.

2 Myerson's lemma

In many cases, we approach the problem of devising a mechanism in two steps:

1. Given that the agents are truth-telling, find the allocation that maximizes the social welfare.
2. Given the allocation in step 1, find payments that will induce truth-telling.

From now on, we'll talk about single parameter environments. Each player i has a value v_i which stands for her value per unit of the product. We use X to denote the set of feasible allocations. $x \in X$ is an n-dimensional vector $\{x_1, \dots, x_n\}$, where x_i stands for how many units of the product agent i gets.

Examples:

1. Single item auction: $X = \{x \in \{0,1\}^n : \sum_{i=1}^n x_i \leq 1\}$.
2. k item auction: $X = \{x \in \mathbb{N}^n : \sum_{i=1}^n x_i \leq k\}$.

A mechanism M takes as it's input the agents' bids $\mathbf{b} = (b_1, \dots, b_n)$ and outputs an allocation vector $x \in X$ and a payment vector $p = (p_1, \dots, p_n)$. The utility of player i is:

$$u_i(\mathbf{b}) = v_i \cdot x_i(\mathbf{b}) - p_i(\mathbf{b}),$$

where $p_i(\mathbf{b}) \in [0, b_i \cdot x_i(\mathbf{b})]$ (meaning that the mechanism is IR and the agents don't receive money from the mechanism).

Definition 3 *An allocation rule x is **implementable** if there exist a payment vector p such that (x, p) is a truthful mechanism.*

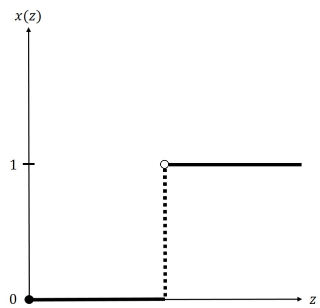
Definition 4 *An allocation rule x is **monotone** if for every bidder i and every set of bids $\mathbf{b}_{-i}$, $x_i(z, \mathbf{b}_{-i})$ is non-decreasing in z.*

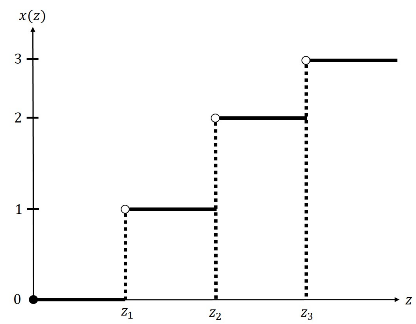
Lemma 5 (Myerson's lemma (81')) *In every single-parameter environment it holds that:*

1. *An allocation rule x is implementable $\iff$ x is monotone.*
2. *If x is monotone, then there exists a **unique** payment p s.t. (x, p) is truthful (assuming that if $b_i = 0$ then $p_i = 0$).*

Proof: Fix an agent i and all the other agents' bids b_{-i}. Let x be an allocation rule, we shall simplify notation by marking: $x(z) := x_i(z, b_{-i})$ and $p(z) := p_i(z, b_{-i})$ the allocation and payment of agent i for bid z.

Example 1 *Graphs showing the jumps in allocation rule $x(z)$ for single item auction and multiple item auction*





Back to the proof, we'll start the by assuming that there exists a payment p s.t. (x, p) is truthful. Let $0 \leq z < y$, we can say that:

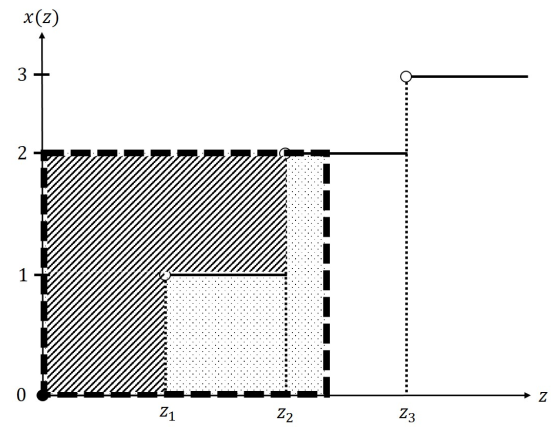
1. If the agent's value is z then: $z \cdot x(z) - p(z) \geq z \cdot x(y) - p(y)$
2. Iif the agent's value is y then: $y \cdot x(y) - p(y) \geq y \cdot x(z) - p(z)$

Utilizing the equations, lets look at the *jump* (difference) in payment between the values z and y:

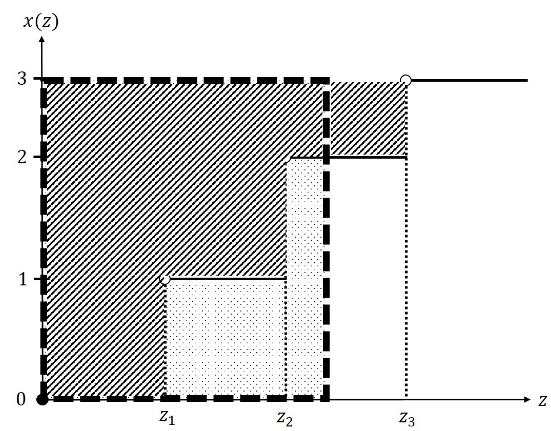
$$z \cdot x(z) - z \cdot x(y) \leq p(y) - p(z) \leq y \cdot x(y) - y \cdot x(z)$$

The new form of the equations yields that $z[x(y) - x(z)] \leq y[x(y) - x(z)]$. After applying simple arithmetic we have $(z - y)[x(y) - x(z)] \leq 0$. We chose z and y such that $(z - y) < 0$, therefore an allocation rule x is monotone if it's implementable

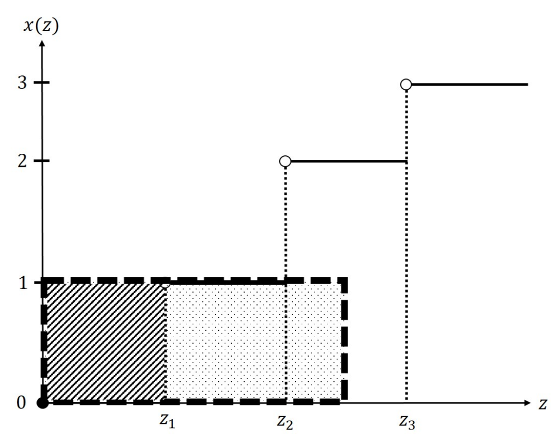
We will now assume that the allocation rule, x, is monotone, we prove with graphs that it yields a truthful mechanism (x, p).



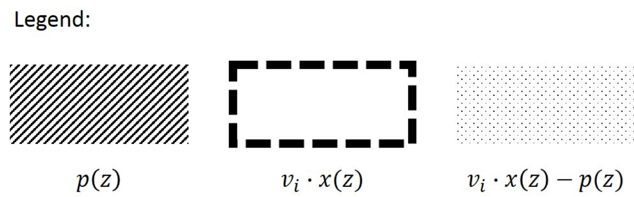
True value



More than the true value



Less then the true value



From the graphs it is clear that telling the true value is the dominant strategy.

Up till now we've seen that monotone iff (x, p) is truthful, lets assume a monotone x and look again at $z \cdot x(z) - z \cdot x(y) \leq p(y) - p(z) \leq y \cdot x(y) - y \cdot x(z)$, the limit when $y \downarrow z$ is:

1. If z is not a *jump* point, we'll get that $p(y) = p(z)$.
2. On the other hand, if there is a jump in z of height h we'll get that $p(y) - p(z) = z \cdot h$

From this observation we can conclude that there is only one way to calculate the payment:

$$p(b_i, b_{-i}) = \sum_{j=1}^{l} z_j \cdot h_j$$

Where $z_1, ..., z_l \in [0, b_i]$ are the jump points of the allocation function, x, and $h_1, ..., h_l$ are the jump heights. ■

Observation 6 *It is not a coincidence that Second Price Auction is truthful - Its allocation is monotone and it fulfills the unique payment.*

3 Knapsack Auction

We'll now look at another auction, called the ***knapsack auction***.

3.1 Our Model

In the ***knapsack auction***, each agent i has a ***weight*** w_i, which is public knowledge, and a private value v_i. The seller has a ***capacity*** W.

Let X be the group of ***feasible allocation vectors***. For each $x \in X$, where $x = (x_1, \ldots, x_n)$, $x_i \in \{0, 1\}$, it must hold that:

$$\sum_{i=1}^{n} w_i x_i \leq W$$

Example 2 *We can think of a TV show commercial break example for a knapsack auction, where the seller has an allocated time-slot for commercials of length W, which is his capacity. Each agent i is an advertiser wanting to air his commercial of length w_i, the agent's weight, during the break, with v_i being the maximum cost he is willing to pay for airing the commercial.*

Claim 7 *Any auction of k identical items is a private case of the knapsack auction.*

Proof: Set weight $w_i = 1$ for each agent i, and capacity $W = k$. By defining a knapsack auction with these parameters we can easily see that it is identical to an auction of k identical items. ■

3.2 Mechanism Design

We'll try to design a knapsack auction that meets our 3 demands:

1. Truthfulness.
2. Maximizing social welfare.
3. Can be calculated in polynomial time.

We use our 2-step method. Assume agents are truthful, and calculate the allocation:

$$\vec{x}(\vec{b}) = \underset{x}{argmax} \sum_{i=1}^{n} b_i x_i$$

This allocation rule is monotone and thus, according to Myerson's lemma, it will allocate payments p s.t. (x, p) is truthful.

However, the allocation problem is known to be NP-hard, which means that no polynomial-time algorithm to solve the problem is known. Thus, we can't calculate the maximal social welfare in polynomial time.

3.3 Relaxing the Requirements

Let's go back and examine our requirements:

Truthfulness means that, in a dominant strategy, each agent's best payoff is achieved by telling the truth (i.e., $b_i = v_i$ for each agent i). Perhaps this requirement is too strong? Weakening this requirement won't help - we still won't be able to calculate the maximal social welfare in polynomial time.

Thus, we'll weaken our second requirement, and keep requirements 1 and 3 the same. In order to weaken the 2nd requirement (maximize social welfare), we need to look at approximation algorithms. This is the core of algorithmic mechanism design.

Algorithm Mechanism Design deals with the question - how little can we weaken our 2nd requirement, such that requirements 1 and 3 still hold?

3.4 Knapsack Approximation Algorithm:

Going back to the knapsack auction problem, we'll look at an algorithm which gives 2-approximation to the problem:

We receive the bids of all agents, $b_1, \ldots, b_n$, and know the weights of the agents $w_1, \ldots, w_n$.

The algorithm:

- Sort and rename all agent, s.t. $\frac{b_1}{w_1} \geq \frac{b_2}{w_2} \geq \ldots \geq \frac{b_n}{w_n}$.
- Go over all agents in this order.
- Allocate agents in a greedy fashion.

This algorithm has a problem in the case that some agent has a very large weight and very large value, which we'd prefer to allocate even though he's not allocated by our algorithm.

Example 3 *Imagine a capacity of $W = 1000$, with 2 agents - one having the same weight and value $w_1 = b_1 = 1$, and the second having $w_2 = 1000$, $b_2 = 999$. Allocating the second agent alone is a much better solution than the allocation produced by the algorithm - which will only allocate the first agent.*

Thus, the addition to our algorithm is:

- Return the maximum among the found allocation, and the allocation of the single agent with the highest value.

Claim 8 *This algorithm gives 2-approximation to the problem*

Proof: We'll look at the fractional version of the problem, where we're allowed to allocate fractions of agents - taking up that fraction of the capacity, and gaining that fraction of their value.

Claim 9 *The greedy algorithm is optimal in the fractional version.*

Proof: The fractional version can be looked at the following way - the seller has W items, and for each agent i, w_i is the maximal number of item the agent can obtain. Formulating the problem this way, $\frac{v_i}{w_i}$ stands for the value per item of agent i. Now it is clear that allocating the items to the agents with the highest value per item maximizes the social welfare. ■

Assume k agents are fully allocated by the algorithm, and agent $k+1$ is partially allocated. The non-fractional algorithm then returns one of 2 possible results:

1. $\sum_{i=1}^{k} v_i$
2. v_{k+1}

We expand on that to get the following:

$SW \geq max\{\sum_{i=1}^{k} v_i, v_{k+1}\}$	the solution can't be better than the maximal social welfare
$max\{\sum_{i=1}^{k} v_i, v_{k+1}\} \geq \frac{1}{2}\sum_{i=1}^{k+1} v_i$	half of sum of two values smaller than their max
$\frac{1}{2}\sum_{i=1}^{k+1} v_i \geq \frac{1}{2} \underset{fractional}{OPT}$	according to proof
$\frac{1}{2} \underset{fractional}{OPT} \geq \frac{1}{2}\underset{int}{OPT}$	int optimal can't be better than fractional optimal

From these we see that the solution output by the algorithm is at least half of the optimal solution, thus it is a 2-approximation to the problem. ■