CS 618, Algorithmic Game Theory
First Problem Set

Costa Huang

April 24, 2019

Due on May 7, 2019 (by 6:00 pm). Feel free to contact me with *any* questions.

1. (10 pts) Consider a single-item auction with at least three bidders. Prove that awarding the item to the highest bidder, at a price equal to the third-highest bid, yields an auction that is not truthful. Then, prove that awarding the item to the second-highest bidder at a price equal to the highest bid is not truthful either.

(1) Assume the first auction is truthful. Consider the following example:

$$\begin{array}{lll} v_1 = 5 & v_2 = 7 & v_3 = 10 \\ b_1 = 5 & b_2 = x & b_3 = 10 \end{array}$$

where v_i and b_i is the evaluations and bids of the item, respectively. Consider what agent 2 might bid. Note that being truthful and let bid $b_2 = 7$ isn't going to win the item, but agent 2 could lie. Let $b_2 = 12$. Then according to the rules of the game, agent 2 pays the third highest bid, which is $b_1 = 5$. As a result, agent 2 successfully lied to gain a positive utility of $u_2 = v_2 - p_2 = 7 - 5 = 2$. Agent 2 is better off lying. A contradiction to the assumption that the first auction is truthful. Therefore the first auction is not truthful.

(2) Now, assume the second auction is truthful. Consider the following example:

$$\begin{array}{lll} v_1 = 5 & v_2 = 7 & v_3 = 10 \\ b_1 = 5 & b_2 = x & b_3 = 10 \end{array}$$

Consider what agent 2 might bid. Being truthful and let $b_2 = 7$ will win the item and pay the highest bid $b_3 = 10$, which results in $u_2 = v_2 - p_2 = 7 - 10 = -3$. So, agent 2 would lie, say to set $b_2 = 0$, and therefore would not get the item, resulting in $u_2 = 0$. Agent 2 is better off lying. A contradiction to the assumption that the second auction is truthful. Therefore the second auction is not truthful.

2. (15 pts) Prove that for every false bid $b_i \neq v_i$ by a bidder in a single-item second-price auction, there exist bids b_{-i} by the other bidders such that i's utility when bidding b_i is strictly less than when bidding v_i.

Let agent i be chosen arbitrarily. Suppose agent i made a false bid $b_i \neq v_i$. There are two cases.

(1) $b_i > v_i$: In this case, randomly select an agent j and let b_j be a real number such that $v_i < b_j < b_i$. Then set $b_{-i,-j} = 0$ (this notation meaning set the bid for everyone else other than agent i, j to be 0). By construction, agent i wins the auction and pays the second price b_j. But since $v_i < b_j$,

$$u_i = v_i - p_i = v_i - b_j < 0$$

If agent i had bid $b_i = v_i$, however, agent j would have won the auction, and thus when bidding truthfully, agent i has $u_i^{'} = 0 > u_i$

(2) $b_i < v_i$: In this case, randomly select an agent j and let b_j be a real number such that $b_i < b_j < v_i$. Then set $b_{-i,-j} = 0$. By construction, agent i loses the auction and therefore $u_i = 0$. Had agent i bid truthfully ($b_i = v_i$), he/she would have won the auction and pay the second price of b_j. But since $v_i > b_j$,

$$u_i^{'} = v_i - p_i = v_i - b_j > 0 > u_i = 0$$

In both cases when $b_i \neq v_i$, we constructed b_{-i} such that the utility u_i when $b_i \neq v_i$ is strictly less than $u_i^{'}$ when $b_i = v_i$.

3. (15 pts) Suppose a subset S of the bidders in a second-price single-item auction decide to collude, meaning that they submit their bids in a coordinated way to maximize the sum of their utilities. Assume that bidders outside of S bid truthfully. State and prove necessary and sufficient conditions on the set S such that the bidders of S can increase their combined utility via non-truthful bidding.

Let T the the set of truthful bidders. Take the highest two bidders i and j in S, where $b_i < b_j$. The necessary and sufficient conditions to increase the combined utility of S is that $v_i > b_i \geq b_{-i,-j}$, which means b_i has to be higher than all the bids from T.

proof. The condition is sufficient because agent j would win the auction and pay the second price of b_i. Notice $v_i - b_i = \epsilon > 0$ and $b_i = v_i - \epsilon$. Then the collective utility of S would be $u = u_j - b_i = u_j - (v_i - \epsilon) = u_j - v_i + \epsilon$, which is strictly greater than the utility $u' = u_j - b_i = u_j - v_i$ had S bid truthfully.

Then, notice saying the conditions are necessary means $v_i > b_i$ and $b_i \geq b_{-i,-j}$ must be true at the same time. Suppose they are not.

(1) If $v_i > b_i$ is not necessary, then consider the situation where $v_i = b_i$. In this case, agent j would still win the auction, but since i is being truthful, no real benefit of collusion is obtained. In the case where $v_i < b_i$, let $b_i - v_i = \epsilon$. Then the collective utility of S would be $u = u_j - b_i = u_j - (v_i + \epsilon) = u_j - v_i - \epsilon$, which is strictly less than the utility $u' = u_j - b_i = u_j - v_i$ had S bid truthfully.

(2) if $b_i \geq b_{-i,-j}$ is not necessary, then if $\max b_{-i,-j} > b_j$, S would not win the auction because their collusion has no impact on the fact that the truthful agent with $\max b_{-i,-j}$ will win the auction. Then if $\max b_{-i,-j} < b_j$, agent j would pay the second price of $\max b_{-i,-j}$, and the collusion within S again would not change the utility of S, which is $b_j - \max b_{-i,-j}$.

By proof of contradiction, we have shown the the stated conditions above are necessary.

4. (20 pts) Consider the following extension of the sponsored search setting described in class. Each bidder i now has a publicly known quality β_i, in addition to a private valuation v_i per click. As usual, each slot j has a CTR α_j, and $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_k$. We assume that if bidder i is placed in slot j,

then the probability of a click is $\beta_i\alpha_j$. Thus bidder i derives value $v_i\beta_i\alpha_j$ from the j-th slot. Describe a truthful and welfare-maximizing auction for this generalized sponsored search setting. Explicitly discuss how you assign bidders to slots, how you define the payments, and formally argue why the resulting mechanism is truthful and welfare maximizing.

To maximize social welfare, we propose the allocation rule:

$$x(b) = \arg\max_{x\in X} \sum_{i=1}^{n} b_i x_i \beta_i$$

Select agent i to be arbitrary. We want to show the allocation rule is nondecreasing. Namely,

$$x_i(z', b_{-i}) - x_i(z, b_{-i}) \geq 0$$

for her bid z, b_{-i} by other agents, and any bid $z' > z$. Let J be the index of agents that were chosen to give nonzero allocation based on $x(b)$ and that $x_t > x_i$ for $t \in J$. Let $j = \min J$. There are two cases.

(1) $z'\beta_i > b_j\beta_j$. Then

$$\begin{aligned} z'x_j\beta_i + \sum_{t\in J-\{j\}} b_t x_t \beta_t - \sum_{t\in J} b_t x_t \beta_t &= z'x_j\beta_i - b_j x_j \beta_j \\ &\propto z'\beta_i - b_j\beta_j > 0 \end{aligned}$$

This implies

$$z'x_j\beta_i + \sum_{t\in J-\{j\}} b_t x_t \beta_t > \sum_{t\in J} b_t x_t$$

This means z' would yield a better value for the $x(b)$, and therefore x_j would be chosen for agent i. This means

$$x_i(z', b_{-i}) - x_i(z, b_{-i}) = x_j - x_i > 0$$

(2) $z'\beta_i \leq b_j\beta_j$. Then the value of $x(b)$ is not affected by z', and agent i's allocation is unchanged.

$$x_i(z', b_{-i}) - x_i(z, b_{-i}) = x_i - x_i = 0$$

Therefore the allocation rule $x(b)$ is monotone and therefore we obtain the payment rule p for which (x, p) is DSIC by Myersons lemma. Since $(x, p$ is DSIC, the auction is truthful; therefore outcome of the auction is always

$$\begin{aligned} x(b) &= \arg\max_{x\in X} \sum_{i=1}^{n} b_i x_i \beta_i \\ &= \arg\max_{x\in X} \sum_{i=1}^{n} v_i x_i \beta_i \end{aligned}$$

which is welfalre maximizing.

5. (15 pts) In a setting with two bidders whose values are drawn uniformly from the interval $[0, 1]$, what is the revenue-optimal auction, and what is its expected revenue?

We first claim that the uniform distribution at interval $[0, 1]$ is a regular ditribution:

proof. $\phi(z) = z - \frac{1-z}{1} = 2z - 1$. Then we want to show $\phi(z') - \phi(z) \geq 0$ for $z' > z$. Notice that

$$\begin{aligned}\phi(z') - \phi(z) &= 2z' - 1 - 2z + 1 \\ &= 2z' - 2z \\ &\propto z' - z > 0\end{aligned}$$

This shows the virtual valuation function is not only non-decreasing but also strictly increasing. Hence the uniform distribution at interval $[0, 1]$ is a regular distribution.

By the lecture 4 slides on the theorem "virtual welfare maximizers are optimal", an agent gets the item if his virtual value is nonnegative. So the reserve price should be

$$\phi^{-1}(0) = \frac{1}{2}$$

Therefore in this setting, a second price auction with a reversed pprice of $1/2$ is optimal with two bidders whose values are drawn uniformly from the interval $[0, 1]$. The expected revenue is $5/12$, which is also given in the slide.

6. (15 pts) Consider a k-unit auction with n unit-demand bidders (k identical units of the same item and each bidder can get at most one of these units). Describe the feasible set X for this single-parameter environment. Then assume that each of the bidders' values are drawn i.i.d. from a regular distribution F, and describe a revenue-optimal auction. Which of the following does the reserve price of this auction depend on: k, n, or F?

The feasible set X is the set $X = \{x_1, x_2, ..., x_n\}$ where each x_i is a binary variable with 1 means agent i is allocated an item and 0 means he doesn't get an item. Furthermore, the allocation should not exceed the amount of units available. Namely,

$$0 \leq \sum_{i=1}^{n} x_i \leq k$$

The revenue-optimal auction is the one given by the virtual welfare maximizer. The allocation rule is

$$x(b) = \arg\max_{x \in X} \sum_{i=1}^{n} \phi_i(v) x_i$$

Since F is regular, the above (welfare-maximizing) allocation rule is monotone (Exercise 5.5 of the book by Roughgarden), the payment rule is given by Myersons lemma. and we have to set the reserve price to be

$$r = \min \phi_i^{-1}(0), \quad i \in N$$

Where N is the indexing set for the n unit-demand bidders. As shown, the reserved price depennds on F and n.

A further explanation on the payment is that agent i will pay the minimum b_i to ensure that $x_i = 1$.

7. (10 pts) Compute the allocation and payments of the VCG auction in the following instance that comprises a set of 4 agents $N = \{1, 2, 3, 4\}$ and a set of 4 indivisible items $M = \{a, b, c, d\}$ (the items

are indivisible, so each item can be assigned to at most one agent).

- $v_1(\emptyset) = 0$, $v_1(\{a\}) = 1$, $v_1(\{b\}) = 2$, $v_1(\{c\}) = 2$, $v_1(\{d\}) = 4$, $v_1(\{a, b, c, d\}) = 11$, and $v_1(S) = 5$ for all other $S \subseteq M$
- $v_2(\emptyset) = 0$, $v_2(\{a\}) = 1$, $v_2(\{b\}) = 1$, $v_2(\{c\}) = 1$, $v_2(\{d\}) = 1$, $v_2(\{a, b\}) = 5$, and $v_2(S) = 4$ for all other $S \subseteq M$
- $v_3(\emptyset) = 0$, $v_3(\{a\}) = 1$, $v_3(\{b\}) = 2$, $v_3(\{c\}) = 4$, $v_3(\{d\}) = 1$, $v_3(\{b, c\}) = 7$, and $v_3(S) = 3$ for all other $S \subseteq M$
- $v_4(\emptyset) = 0$, $v_4(\{a\}) = 1$, $v_4(\{b\}) = 1$, $v_4(\{c\}) = 1$, $v_4(\{d\}) = 3$, and $v_4(S) = 3$ for all other $S \subseteq M$

The allocation is as follow

a	b	c	d
0.00	0.00	0.00	1.00
1.00	1.00	0.00	0.00
0.00	0.00	1.00	0.00
0.00	0.00	0.00	0.00

So agent 1 gets item d, agent 2 gets item a, b, and agent 3 gets item c, and the maximum welfare is 13.

8. (Extra credit: 10 pts) Implement the VCG auction and run it on randomly generated instances with n agents and m indivisible items. Each agent's value for a bundle of k items is generated as $k \times rand(0, 1)$, where $rand(0, 1)$ is a uniformly at random generated value in the interval $[0, 1]$. Use a different random value for each bundle and each agent. Using these instances, count how long it takes for your algorithm to compute the allocation and the prices for all combinations of values of $n \in \{2, 4, 5, 6\}$ and values of $m \in \{4, 6, 8, 10, 12, 14, 16, 18, 20\}$. Submit the amount of time for each pair of n, m values, as well as the code that you wrote and used.

This creates a combinatorial explosion... My computer is really slow so I limited the parameter to $n = \{2, 3, 4, 5\}$ and $m = \{2, 3, 4, 5\}$ And the result is as follow.

```
n=2, m=2 takes  0.0
max_welfare=1.3312242222569104
n=2, m=3 takes  0.0009970664978027344
max_welfare=2.7844974545881804
n=2, m=4 takes  0.0020003318786621094
max_welfare=3.665547313516595
n=2, m=5 takes  0.010007143020629883
max_welfare=4.054170687185614
n=3, m=2 takes  0.0009942054748535156
max_welfare=1.8688589297525555
n=3, m=3 takes  0.005998134613037109
max_welfare=2.766439811779276
n=3, m=4 takes  0.0370023250579834
max_welfare=3.8442529091092354
n=3, m=5 takes  0.2969982624053955
max_welfare=4.623379111905488
n=4, m=2 takes  0.0020041465759277344
```

```
max_welfare=1.5407801180276726
n=4, m=3 takes  0.0359959602355957
max_welfare=2.670930432197369
n=4, m=4 takes  0.5850002765655518
max_welfare=3.799785650658025
n=4, m=5 takes  9.365022420883179
max_welfare=4.641199203480946
n=5, m=2 takes  0.009000301361083984
max_welfare=1.8431651218222895
n=5, m=3 takes  0.2859947681427002
max_welfare=2.8172396867212677
n=5, m=4 takes  9.679998874664307
max_welfare=3.756075725239705
```

And $n = 5, m = 5$ is still running after 10 mins.