COS 445 — PSet 4

Due online Monday, April 16th at 11:59 pm

Please reference the course infosheet for the complete homework/collaboration policy. Highlights below:

- You must write up your solutions by yourself, without any collaborators or external references.
- Unless otherwise stated, you may collaborate with other students and consult external references.
- You must list all collaborators and external references consulted and upload it as a separate file to Mechanical TA (do not include collaborators in your problem submission).
- Some problems will be marked as no collaboration problems. This is to make sure you have experience solving a problem start-to-finish by yourself in preparation for the midterms/final.
- Please upload your solutions for 1–3 as separate PDF files to Mechanical TA The name of your file will be visible to graders, so if you would like to remain anonymous to graders, give your PDF a generic name. Please upload your code and writeup for problem 4 to CS Dropbox.

Problem 1: Combinatorial Auctions (10 points, no collaboration)

In a combinatorial auction there are $m$ items for sale to $n$ buyers. Each buyer $i$ has some valuation function $v_i(\cdot)$ which takes as input a set $S$ of items and outputs that bidder’s value for that set (so $v_i(S) = 5$ means that bidder $i$ gets value 5 for receiving set $S$). These functions will always be monotone ($v_i(S \cup T) \geq v_i(S)$ for all $S, T$), and satisfy $v_i(\emptyset) = 0$.

Design an auction that is incentive compatible and maximizes the welfare. Note that each item can be awarded to at most a single bidder (or to no one, if desired).

Hint: The intention of this problem is for you to figure out how to instantiate the VCG mechanism for this setting. You may either use VCG for guidance and provide a complete proof that your mechanism is IC and maximizes welfare, or you may **prove** that your mechanism is in fact a special case of VCG (such a proof might be pretty short).

Problem 2: Some issues with Greedy (15 points)

Again consider a combinatorial auction. Consider the following mechanism for allocating items:

- Initialize $S_i = \emptyset$ (each bidder $i$ initially gets no items). Initialize $p_i = 0$ (each bidder initially pays 0).
- For $j = 1$ to $m$ (for each item in order)
  - Ask each bidder their marginal value for item $j$: $b_{ij}(S_i) = v_i(S_i \cup \{j\}) - v_i(S_i)$ (how much additional value would they get right now by adding $j$).
– Award item \( j \) to the bidder \( i \) who reports the largest value (breaking ties lexicographically), add to their payment the second-highest report. That is, if \( i \) reports the largest marginal value and \( i' \) reports the second-largest, Update \( S_i \) to \( S_i \cup \{j\} \), and \( p_i \) to \( p_i + b_{ij}(S'_i) \).

• Award bidder \( i \) the set \( S_i \) of items and charge them \( p_i \).

Part a (5 points)

Prove that if all valuation functions are additive (that is, \( v_i(S) = \sum_{j \in S} v_i(\{j\}) \), for all \( S \)), then it is an equilibrium for all bidders to truthfully report \( b_{ij}(S_i) = v_i(S_i \cup \{j\}) - v_i(S_i) \) in every round.

Part b (10 points)

Provide one example of valuation functions \( v_1(\cdot) \) and \( v_2(\cdot) \), such that it is not an equilibrium for both bidders to tell the truth. Specifically, prove that if bidder 2 tells the truth, then bidder 1 can do strictly better by lying.

Problem 3: Revenue Equivalence (25 points)

Consider a single-item auction with two bidders whose values are drawn from the equal-revenue curve \( ER \), \( (F(x) = 1 - 1/x \) for all \( x \geq 1 \), and \( f(x) = 1/x^2 \) for all \( x \geq 1 \)). Find a bidding strategy \( b(\cdot) \) that is a Bayes-Nash equilibrium: for all \( v_1 \), given that bidder 2 is going to draw a value \( v_2 \leftarrow ER \) and bid \( b(v_2) \), your expected utility from participating in the First-Price Auction is (weakly) maximized by bidding \( b(v_1) \). Recall that your utility is equal to \( v_1 - b \) if you win and bid \( b \), and zero otherwise.

The following parts will guide you through a proof using Revenue Equivalence. You should complete all parts and not provide an alternative proof.

Part a (5 points)

What is the expected revenue of the second-price auction when two bidders with values independently drawn from equal-revenue curves bid their true value?

Part b (5 points)

In the second-price auction, what is the expected payment made by bidder one, conditioned on bidding \( v_1 \), and that bidder two truthfully reports \( v_2 \leftarrow ER \)?

Note that we are not conditioning on bidder 1 winning. To be extra formal, let \( P^{SPA}_1(v_1) \) denote the random variable that is \( v_2 \) if \( v_1 > v_2 \), and 0 otherwise. What is \( \mathbb{E}_{v_2 \leftarrow ER}[P^{SPA}_1(v_1)] \)?

Part c (10 points)

For a given bidding strategy \( b(\cdot) \), define \( P^{FPA}_1(v_1, b) \) to be the random variable that is \( b(v_1) \) if \( v_1 > v_2 \), and 0 otherwise. Find a bidding strategy \( b(\cdot) \) such that:

• \( b(\cdot) \) is strictly monotone increasing on \([1, \infty) \) \( (b(v) > b(v') \iff v > v') \). That is, bidder 1 will win exactly when \( v_1 > v_2 \) if both bidders use strategy \( b(\cdot) \).
• For all $v_1 \in [1, \infty)$, $E_{v_2 \sim ER}[P^FPA_1(v_1, b)] = E_{v_2 \sim ER}[P^{SPA}_1(v_1)]$. That is, the expected payment made by bidder 1, conditioned on $v_1$ is the same in both auctions.

Part d (5 points)

Prove that the strategy you found in Part c is a Bayes-Nash Equilibrium of the first-price auction for two bidders with values drawn from the equal-revenue curve. You will receive partial credit for correctly setting up the necessary equations and verifying them with an online solver.

**Hint:** Proving this inevitably will require taking derivatives, but there is an approach that avoids intense calculations.

**Problem 4: Start your Startup (15 points)**

In this programming task, your team will play the role of a no-name tech startup in Silicon Valley, desperate to implement your grand vision of a toothbrush sharing economy (or whatever else your startup does). Of course, the most fashionable way to spread the message nowadays is to pay for clicks: participate in auctions for search engine ad slots. The setup is as follows.

Each day $t$, there will be a new keyword up for auction. On that day (and no earlier), your marketing research team will tell you that you should value a click for this keyword at $v_t$ dollars, which you can think of as a real number drawn from an equal revenue curve. More clearly, $v_t$ will always be at least 1, and at most 1000. $v_t$ will be $> x$ with probability $1/x$ for all $x \in [1, 1000)$, and equal to 1000 with probability $1/1000$. All values are drawn independently for each team and independently for each day.

There are 10 ad positions up for auction. The first three have click-through rates of 50%, 35%, and 30%. The next seven have identical rates of 15%. Your value for an ad slot with click-through rate $c$ is $c \cdot v_t$. The auctioneer runs a generalized second-price (GSP) auction:

- Each startup submits a sealed bid $b_i \geq 0$.
- The top 10 bidders get the top 10 ad slots in order, with ties broken randomly.
- The winner of the top slot pays the second-highest bid per click, the winner of the second-highest slot pays the third-highest bid per click, and so on. Of course, teams which do not win a slot pay nothing. For instance, if you win the top slot and the second slot bid 1 then you would pay 0.5.

This process is repeated for $T = 10000$ days. Initially, each team has a budget of $B = 5000$ dollars, and payments are deducted from this budget. Of course, a team is not allowed to bid more than its current balance. (Any bid that does so will be set to 0, and incur a deduction.) Your objective is to maximize the sum of values obtained over all days, plus your remaining budget. Formally, if on day $t$ you win a slot with click-through rate $c_t$ and pay $p_t$, your payoff at the end of the game is $5000 + \sum_t v_t \cdot c_t - p_t$. Note that your revenues from winning ads are not delivered until after the last auction, so you may not bid those revenues. That is, $\sum_t p_t \leq 5000$.

You strategy should implement two functions. The first function will take as input a value $v_t$ (your value per-click for that day) and return your bid for that day. The second will take an array $W_i$, where $W_i$ denotes the winning bid for slot $i$ on day $t - 1$, and an integer $i^*$, which indicates whether your strategy won an auction the previous day. That is, the startup that won slot 1 on day $t - 1$ bid $W_1$ per click. You will not know the ID of the startup that won each slot. If $i^* \geq 0$, then you won slot $i^*$ and bid $W_{i^*}$. When $i^* \geq 0$ you will also be told $p_{i^*}$, your payment. You should track the day $t$ and your remaining budget, and whatever information about past payments you like.
Part a (5 points)

Prove that if $T = 1$ (i.e. there was only a single round), overbidding would be a dominated strategy. Very briefly note why it might potentially make sense to overbid when $T = 10000$.\(^1\)

Part b (5 points)

We saw in class that GSP is not truthful, so it may make sense to lie. If the assignment was nearly identical, except we replaced GSP every round with VCG (which we showed in class to be truthful), is it now an equilibrium for everyone to bid truthfully every round? Give a brief justification of why or why not.

Part c (5 points) Programming Component Specifications and Writeup

You will implement the Bidder interface provided in Bidder.java, which requires the following methods, as documented in Bidder.java.

```java
// Return your bid for the current day
// Called once per day before the auction
public double getBid(double dailyValue);

// Let you know if you won, and how much the winners paid
// Called once per day after the auction
public void addResults(List<Double> bids,
                        int myBid, double myPayment);
```

We've also provided a sample strategy, Bidder_truthful.java, and the usual construction of testing code: AuctioneerBase.java and a makefile to run it against varied strategies.

Your file must follow the naming convention Bidder_netID1_netID2.java, where netID1 and netID2 are the Net IDs of the submitters. If you work alone, use only one name; if we approved you working in a triple, use all three. All advice from previous homeworks applies again to this homework, especially that you should consider the performance of your strategy against the strategies and distribution thereof which you expect your peers to play.

Your writeup should provide an overview of the main ideas in your code (remember that we also have your code — so you don’t need to provide pseudocode or a step-by-step description of your algorithm), and justify why you think it will perform well, in addition to concrete answers to parts a and b.

Penalties may be given for code which does not compile, throws exceptions, or violates assertions. Remember to test your code with settings besides the default settings of our testing environment. Extra credit may be awarded for reporting substantive bugs in our testing code.

---

\(^1\)To be clear, the course staff is suggesting neither that you overbid, nor that you don’t overbid. Just that you think about it.
Extra Credit: Walrasian Equilibria

In a combinatorial auction (same definition as Problem 1), a Walrasian Equilibrium is a price for each item \( \vec{p} \) such that:

- Each buyer \( i \) selects to purchase a set \( B_i \in \arg \max_S \{ v_i(S) - \sum_{j \in S} p_j \} \).
- The sets \( B_i \) are disjoint, and \( \bigcup_i B_i = [m] \).

Prove that a Walrasian equilibrium exists for \( v_1, \ldots, v_n \) if and only if the optimum of the LP relaxation below (called the configuration LP) is achieved at an integral point (i.e. where each \( x_{i,S} \in \{0, 1\} \)).

\[
\max \sum_i \sum_S v_i(S) \cdot x_{i,S} \\
\forall i, \sum_S x_{i,S} = 1 \\
\forall j, \sum_{S \ni j} \sum_i x_{i,S} \leq 1
\]

(1)

Finally, provide an example of two valuation functions \( v_1, v_2 \) over two items where a Walrasian equilibrium doesn’t exist.