CS364A: Exercise Set #4

Due by the beginning of class on Wednesday, October 23, 2013

Instructions:

(1) Turn in your solutions to all of the following exercises directly to one of the TAs (Kostas or Okke). Please type your solutions if possible and feel free to use the LaTeX template provided on the course home page. Email your solutions to `cs364a-aut1314-submissions@cs.stanford.edu`. If you prefer to hand-write your solutions, you can give it to one of the TAs in person at the start of the lecture.

(2) Your solutions will be graded on a "check/minus/zero" system, with "check" meaning satisfactory and "minus" meaning needs improvement.

(3) Solve these exercises and write up your solutions on your own. You may, however, discuss the exercises verbally at a high level with other students. You may also consult any books, papers, and Internet resources that you wish. And of course, you are encouraged to contact the course staff (via Piazza or office hours) to clarify the questions and the course material.

(4) No late assignments will be accepted.

Lecture 7 Exercises

Exercise 28

Recall that raising Yahoo!'s reserve prices was particularly effective for valuable keywords (typical valuations-per-click well above the old reserve price of $.10) that had few bidders (at most 6, say). Give at least two examples of keywords that you think might have these properties, and explain your reasoning in 1-2 sentences.

Exercise 29

Prove that the payment $p_i(\mathbf{b})$ made by a bidder i in the VCG mechanism is at least 0 and at most $b_i(\omega^*)$, where ω^* is the outcome chosen by the mechanism.

Exercise 30

Consider a combinatorial auction where you know a priori that every bidder is *unit demand*. This means that the valuation of a bidder i can be described by m private parameters (one per good) $v_{i1}, \ldots, v_{im}$, and its valuation for an arbitrary set S of goods is defined as $\max_{j \in S} v_{ij}$. Prove that the VCG mechanism can be implemented in polynomial time for unit-demand bidders. [Cf., Problem 12(d) on Problem Set #2.]

Exercise 31

Consider two goods (A and B) and three bidders. Bidder #1 has valuation 1 for A and B together (i.e., $v_1(AB) = 1$) and 0 otherwise. Bidder #2 has valuation 1 for A (i.e., $v_2(AB) = v_2(A) = 1$) and 0 otherwise. Bidder #3 has valuation 1 for B and 0 otherwise. Compute the VCG allocation and payments when only the first two bidders are present. Do the same when all three bidders are present.

Can adding an extra bidder ever decrease the revenue of the Vickrey (single-item) auction? Give a brief explanation.

Exercise 32

Show that the following is possible: two bidders in a combinatorial auction are given no goods by the VCG mechanism if they bid truthfully, yet both can achieve positive utility if they both submit suitable false bids.

Can this ever happen in the Vickrey auction? Give a brief explanation.

Exercise 33

Consider a combinatorial auction in a which a bidder can submit multiple bids under different names, unbeknownst to the mechanism. The allocation and payment of a bidder is the union and sum of the allocations and payments, respectively assigned to all of its pseudonyms. Show that the following is possible: a bidder in a combinatorial auction can earn higher utility from the VCG mechanism by submitting multiple bids than by bidding truthfully.

Can this ever happen in the Vickrey auction? Give a brief explanation.

Lecture 8 Exercises

Exercise 34

Recall the reverse auction setting from lecture. B denotes the set of bidders. There is a set $\mathcal{F} \subseteq 2^B$ of feasible sets that is upward closed (i.e., supersets of feasible sets are again feasible).

- Initialize $S := B$.
- While there is a bidder $i \in S$ such that $S \setminus \{i\}$ is feasible:
  - (*) Delete some such bidder from S.
- Return S.

Suppose we implement the step (*) using a scoring rule, which assigns a number to each bidder i. At each iteration, the bidder with the largest score (whose deletion does not destroy feasibility of S) gets deleted. The score assigned to a bidder can depend on i, i's bid, the bids of other bidders that have already been deleted, the feasible set $\mathcal{F}$, and the history of what happened in previous iterations. (Note a score is *not* allowed to depend on the value of the bids of other bidders that have not yet been deleted.)

Assume that the scoring rule is increasing — holding everything fixed except for b_i, i's score is increasing in its bid b_i. Then, show that the allocation rule above is monotone: for every i and $\mathbf{b}_{-i}$, if i wins with bid b_i and $b'_i < b_i$, then i also wins with bid b'_i.