
Remember: you may work in groups of up to three people, but must write up your solution entirely on your own. Collaboration is limited to discussing the problems – you may not look at, compare, reuse, etc. any text from anyone else in the class. Please include your list of collaborators on the first page of your submission. You may use the internet to look up formulas, definitions, etc., but may not simply look up the answers online.

Please include proofs with all of your answers, unless stated otherwise. Your solution must by typeset (*not* handwritten), and must be submitted by gradescope.

---

1 Bulow-Klemperer (50 points)

This problem considers a variation on the Bulow-Klemperer theorem. Consider selling $k \geq 1$ identical items (with at most one given to each bidder) to bidders with valuations drawn i.i.d. from F. Prove that for every $n \geq k$, the expected revenue of the Vickrey auction (with no reserve) with $n + k$ bidders is at least that of the revenue-optimal auction for F with n bidders. Recall that the Vickrey auction in this context would give the k items to the k highest bidders, and charge each one of them the $(k + 1)$st highest bid.

2 Unit-Demand Valuations (50 points)

Consider a combinatorial auction with n players and item set M with $m = |M|$. A player i has a *unit-demand* valuation if there exist parameters $v_i^1, v_i^2, \ldots, v_i^m \in \mathbb{R}_{\geq 0}$ (one parameter per item) such that $v_i(S) = \max_{j \in S} v_i^j$ for all $S \subseteq M$ (and $v_i(\emptyset) = 0$). In other words, the value of a bundle for player i is determined by the single most valuable element in that bundle (from the perspective of player i).

Give a mechanism for combinatorial auctions in which all players have unit-demand valuations which satisfies the following properties:

(a) It is incentive-compatible,

(b) It maximizes social welfare (i.e., maximizes $\sum_{i=1}^{n} v_i(S_i)$ where S_i is the bundle given to player i by the mechanism), and

(c) It runs in time polynomial in n and m.

Hint: Show that VCG can be implemented in polynomial time (in this setting) by reducing it to a well-known graph algorithm problem.