COMP6207 Algorithmic Game Theory 2022/2023

Coursework 1

1. [*25 points*] Suppose the auctioneer is selling two items A and B. There are three bidders 1, 2 and 3. Each bidder can buy none, one, or both of the items. Each bidder is asked to report her valuation function, i.e., $v_i = (v_i(\varnothing), v_i(A), v_i(B), v_i(A, B))$. Given the following instance:

   Bidder 1 reports $v_1 = (0, 10, 3, 13)$.
   Bidder 2 reports $v_2 = (0, 2, 8, 12)$.
   Bidder 3 reports $v_3 = (0, 3, 5, 14)$.

   (a) Compute the VCG allocation (i.e. choice) and payments for each bidder. Provide explanations as well as the final answers. [8 points]

   (b) Discuss why bidder 3 might have an objection to the outcome. [5 points]

   (c) Now assume that bidder 1 has decided not to participate in the auction after all. Compute the VCG allocation and payments for each bidder in the scenario when only bidders 2 and 3 participate in the auction (yet again, provide explanations as well as the final answers). [6 points]

   (d) Can adding an extra bidder ever decrease the revenue of the Vickery (single-item) auction? Provide a brief explanation. [6 points]

2. [*10 points*] The VCG mechanism does not violate the Myerson-Satterthwaite Theorem because it is not budget balanced for general quasilinear preferences. But this seems like an easy enough problem to solve: we can just evenly redistribute any money that was collected by the mechanism (or, tax all agents equally if the net payment to the agents was positive). Below is a proposed, budget-balanced version of the VCG mechanism. It converts what was p_i into a temporary variable t_i. Then the new payments p_i contains an equal redistribution of the sum of the original payments.

   The *Budget-Balanced VCG Mechanism* is a direct mechanism (χ, p) where

   - $\chi(\hat{v}) = \text{argmax}_x \sum_i \hat{v}_i(x)$
   - $t_i(\hat{v}) = \sum_{j \neq i} \hat{v}_j(\chi(\hat{v}_{-i})) - \sum_{j \neq i} \hat{v}_j(\chi(\hat{v}))$
   - $p_i(\hat{v}) = t_i(\hat{v}) - \frac{1}{n} \sum_i t_i(\hat{v})$

   Prove that this mechanism is not dominant-strategy truthful, without using or relying on the characterisation of DS truthful mechanisms or any impossibility results. One option is to prove this directly. Alternatively, you can give two valuation functions for an agent i (one that gives her true valuation v_i and an alternative one v_i') and a set of declare valuation functions for as many other agents other than i that you need, and show that if all other agents declare these valuations, the utility for agent i is higher if she declares v_i' instead of v_i. (Hint: follow the alternative suggestion).

3. [*15 points*] Suppose you have an object that each of n agents desire, but which you do not value. Assume that each agent i values it at v_i, with v_i's drawn independently and randomly from some positive real line interval, say $[0, 10^{100}]$. Although you do not desire the object and also do not care about the actual values for the v_i's, you need to compute $\sqrt{v_i}$ for each i.

   Unfortunately, you face two problems. First, agents are not inclined to just reveal to you anything about their v_i's. Second, your computer is costly to operate. It costs you 1 unit to determine the greater of two values, 2 units to perform any basic arithmetic operation $(+, -, \times, /)$, and anything more complicated (such as $\sqrt{x}$) costs 20.

   (a) How much would it cost to compute $\sqrt{v_i}$ for each i using a straightforward VCG mechanism? Provide an explanation, as well as the final answer. [5 points]

   (b) Your answer above gives an upper bound on the cost of computing the square roots of the valuations. Design a direct mechanism with a dominant strategy equilibrium (that is every participant has a dominant strategy, no matter what her private valuation is) that will allow you to compute all $\sqrt{v_i}$ at *minimal cost.* Assume that agents can do computations for free.

   - Make sure you specify all the components of the mechanism (even if they seem obvious to you): players, actions, outcomes, mappings from actions to outcomes. Specify the algorithm that you will use to implement those mappings.
   - Explain why your mechanism has a dominant strategy equilibrium.
   - Give your mechanism's total computation cost (or a reasonable upper bound on it). Provide an explanation, as well as the final answer.

   You do not need to prove that your mechanism achieves minimal cost. (Hint: think about the revelation principle). [10 points]

4. [*20 points*] In the Stable Matching problem (SM) every agent finds all candidates acceptable. Consider a setting where the mechanism designer knows this. So the only way for an agent to misreport is to permute his or her preferences. Prove that, even with this restriction, no stable matching mechanism exists for which truth-telling is a dominant strategy for every agent.

5. [*10 points*] There are four graduate students looking for jobs and three companies hiring new employees. Each agent has a strict preference ordering over her/its acceptable candidates.

$s_1 : c_1 > c_3 > c_2$	$c_1 : s_1 > s_3$
$s_2 : c_2 > c_1$	$c_2 : s_2 > s_1 > s_4$
$s_3 : c_1 > c_3$	$c_3 : s_4 > s_1 > s_3$
$s_4 : c_2 > c_3 > c_1$	

Suppose that companies c_1 and c_3 are each hiring one employee (i.e. $q_1 = q_3 = 1$) and that company c_2 is hiring two employees (i.e. $q_2 = 2$). To find a stable matching, we are going to run an extension of Gale-Shapley to this setting. For each of the questions below, I recommend using the concurrent version in which all eligible agents propose at the same time.

(a) Find the stable matching when graduates propose (i.e. the graduate-oriented Gale-Shapley is executed). Show the intermediate steps as well as the matching returned.

(b) Find the stable matching when companies propose (i.e. the company-oriented Gale-Shapley is executed). Show the intermediate steps as well as the matching returned.

6. [*20 point*] There are eight students $A, B, \ldots, H$ and four copies of the same project. Students need to work in pairs on the project. Each student has a strict preference ordering over all other students.

$A : B > E > D > F > G > H > C$

$B : C > F > A > G > H > E > D$

$C : D > G > B > H > E > F > A$

$D : A > H > C > E > F > G > B$

$E : F > A > H > B > C > D > G$

$F : G > B > E > C > D > A > H$

$G : H > C > F > D > A > B > E$

$H : E > D > G > A > B > C > F$

Using Irvings' algorithm for Stable Roommates problem (SR), investigate whether this instance admits a stable matching. Provide the phase-1 table (no need to describe each step of phase-1 algorithm). Describe your execution of phase-2 algorithm, outlining each exposed rotation you find along the way. Provide a stable matching if this instance admits any.