Social Networks and Online Markets

Homework 2

Due: 20/6/2021, 23:59

Instructions

You must hand in the homeworks electronically and before the due date and time.

The first homework has to be done by each **person individually**.

Handing in: You must hand in the homeworks by the due date and time by an email to `aris@diag.uniroma1.it` that will contain as attachment **(not links to some file-uploading server!)** a .zip or .pdf file with your answers.
After you submit, you will receive an acknowledgement email that your homework has been received and at what date and time. If you have not received an acknowledgement email within 2 days after you submit then contact Aris.

The solutions for the theoretical exercises must contain your answers either typed up or hand written clearly and scanned.

If you need any technical help, you can email George Birbas: `birbas@diag.uniroma1.it`.

For information about collaboration, and about being late check the web page.

Problem 1. Consider the following setting that we have also seen in the class: We have a set of players N with $|N| = n$, and a set of items M with $|M| = m$. Each player i, has an additive valuation function $v_i(\cdot)$ over the items. Our goal is to produce an allocation $\mathcal{A} = (A_1, \ldots, A_n)$ of the items to the players. Recall that an allocation is basically a partition of the set of items, that is, $\forall i,j \in \{1, 2, ..., n\}$ where, $i \neq j$, we have that $A_i \cap A_j = \emptyset$, and in addition $\cup_{\{1,2,\ldots,n\}} A_i = M$.

Consider the following mechanism: The players are ordered in an arbitrary way and the mechanism runs in rounds following this ordering. In each round a player, when his order comes, chooses his most desirable item among the remaining ones, that is, the first player gets his most desirable item, the second player gets his most desirable item among the ones that remain, and so on. So if we have n players, the mechanism runs as follows (where each agent, gets in his turn his most desirable item among the ones that remain), $1 \to 2 \to 3 \to ... \to n \to 1 \to 2 \to ...$ until we run out of items. As we discussed in the class, this mechanism is not truthful but has some fairness properties. Consider the following fairness concepts:

Definition 1 *An allocation $\mathcal{A} = (A_1, \ldots, A_n)$ is envy-free, if for every $i, j \in N$, $v_i(A_i) \geq v_i(A_j)$.*

Definition 2 *An allocation $\mathcal{A} = (A_1, \ldots, A_n)$ is an EFX (envy-free up to any good) allocation, if $v_i(A_i) \geq v_i(A_j \setminus \{g\})$ holds for every pair $i, j \in N$, with $A_j \neq \emptyset$, and for every $g \in A_j$.*

Definition 3 *An allocation $\mathcal{A} = (A_1, \ldots, A_n)$ is an EF1 (envy-free up to one good) allocation, if for every pair of agents $i, j \in N$, with $A_j \neq \emptyset$, there exists an item $g \in A_j$, such that $v_i(A_i) \geq v_i(A_j \setminus \{g\})$.*

Prove that:

1. This mechanism does not always produce envy-free allocations.

2. This mechanism does not always produce EFX allocations.

3. This mechanism always produces EF1 allocations.

Problem 2. Consider the following problem: There is a set of N players and one single digital good. Each agent has a value for the digital good that describes how much he wants it. Since the good is digital, it can be provided to any number of players. However, there is a cost for the provision of the good that is equal to 1 (this is the same regardless of the number of the players that get the good in the end).

The designer wants to design a mechanism that is truthful, and also covers the cost created due to the provision of the good. So he thinks as follows: Let the players declare their values, and then set the price of the good to $p = \frac{1}{n}$. If the value of **every** player is more than $\frac{1}{n}$, then everyone will get the item and the procedure terminates. However, if there are some players, say k in number, that have value less than $\frac{1}{n}$, then they are removed from the game and they do not get the digital good. If something like this happen, the price of the digital good is updated to $p = \frac{1}{n-k}$, and the same procedure is followed again with the new price and the new set of players (the set of players that remain). This goes on, until either we end up with a set of players where everyone has a value higher than the current price (this will be the set of the winners), or we end up with an empty set of players.

If a player is a winner (gets the digital good) then his utility is $u_i = v_i - p$, while if he is removed from the game (he does not get the item) his utility is $u_i = 0$. Finally, the social welfare in this setting is defined as $SW = \sum_w v_i - 1$, i.e., the sum of the values of the winners minus the cost. If there are no winners at all, then the social welfare is 0, since no player gets the item and thus, there is no cost from the provision of the good.

Prove that:

1. This mechanism always covers the cost of the digital good i.e., the sum of the payments is at least 1.

2. This mechanism is indeed truthful.

3. The Price of Anarchy of this mechanism is infinite. *Hint: Use the fact that the mechanism is truthful, and create an instance where while every agent reports the truth, this leads to an outcome with zero social welfare. The instance must be designed in such a way so that the optimal social welfare is strictly positive.*

Problem 3. Consider n workers that are available for executing k jobs. Each worker executes the list of jobs assigned to her serially and without any breaks and all workers start working at the same time in parallel. Each worker i specifies the time t_{ij} needed to execute the job j and if a worker is assigned jobs in a set S, then the total time required for the worker to complete the jobs is $\sum_{j \in S} t_{ij}$. The goal is to assign the jobs to to workers so as to minimize the completion time of the last job, i.e. for the full set of jobs to be executed as quickly as possible.

The workers however are lazy and do not really want to work, so they may want to misreport their specified times t_{ij} needed to complete the jobs. For this reason, we can use payments to incentivize workers to tell us the truth, i.e. each worker receives a monetary compensation.

1. Consider the following version of the VCG mechanism, run independently for every job j: Assign the job to the worker i that can do it as quickly as possible, i.e. $\arg\min_i t_{ij}$ and pay him the processing time of the second fastest worker for this particular job. Show that this mechanism is an n-approximation to the optimal outcome.

2. Prove that no truthful mechanism can achieve an approximation ratio better than 2. (*Hint: Use the following property that follows from weak monotonicity without proof: If a worker i decreases all the reported processing times t_{ij} for jobs that she receives and increases all the reported processing times for jobs she does not receive, then she gets the same allocation of jobs.*)

Problem 4. Consider the following function

$$\Phi(\alpha) = \sum_{r \in R} \sum_{i=1}^{\#(r,\alpha)} c_r(i),$$

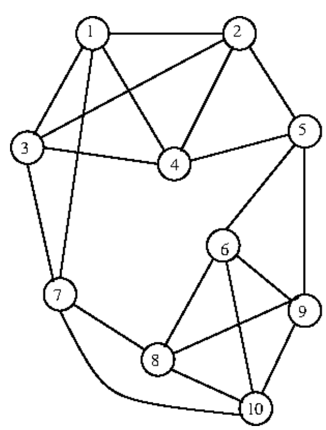
where α is a strategy profile $(\alpha_1, \ldots, \alpha_n)$, $r \in R$ is a resource, c_r is the cost function of resource r and $\#(r,\alpha)$ is the number of players using resource r in profile α. This is a potential function and it is called *Rosenthal's potential*.

Now consider a congestion game for which we have the following guarantee: The cost functions c_r are such that no resource is ever used by more than λ players. Use Rosenthal's potential to show that the Price of Stability of such a game is at most λ.

Problem 5. Consider the game given by the following payoff matrix. Reduce this game to a 1×1 game by iteratively removing strictly dominated strategies.

	C1	C2	C3	C4	C5
R1	4,-1	3,0	-3,1	-1,4	-2,0
R2	-1,1	2,2	2,3	-1,0	2,5
R3	2,1	-1,1	0,4	4,-1	0,2
R4	1,6	-3,0	-1,4	1,1	-1,4
R5	0,0	1,4	-3,1	-2,3	-1,-1

Problem 6. We are given the following graph, $G = (V,E)$.



1. Find the densest subgraph using the greedy algorithm we saw in class.

2. Find a minimum cut.

3. Demonstrate (by calculating λ_2, $\phi(G)$, etc.) that Cheeger's inequalities hold for this graph.

4. Find the cut that satisfies Part 3 (and show that it does).

You may use a computer to compute eigenvalues and eigenvectors, but you must specify in your solution how (what program and what code/commands you used).