Algorithmic Game Theory Due: May 30, 2018

Homework 2

Lecturer: Michal Feldman *Assistant: Kineret Segal*

1 Gross substitute valuations

Problem 1:

In this question we prove the following claim: a demand query for gross substitute valuation can be computed in $poly(m)$ time, where m is the number of items.

Fix some valuation function $v : 2^m \to \mathbb{R}_{\geq 0}$ over a set of items $[m]$ and item prices $\mathbf{p} := (p_1, ..., p_m)$. Define:

Definition 1 *for all $i \in [m]$ and $S \subseteq [m]$ the marginal value of i given S is*

$$v(i|S) := v(\{i\} \cup S) - v(S)$$

Definition 2 *demand set $D_v(\boldsymbol{p}) := argmax_{S \subseteq [m]}\{v(S) - \sum_{j \in S} p_j\}$.*

Definition 3 *Greedy algorithm:*

1. $S = \emptyset$, $M = [m]$
2. *While(true)*
   (a) *let $j \in argmax_M\{v(j|S) - p_j\}$*
   (b) *if $v(j|S) \leq p_j$, halt and return S.*
   (c) *else: $S := S \cup \{j\}$, $M := M \backslash \{j\}$.*

Definition 4 *greedy ordering of $[m]$ is the order in which the greedy algorithm chooses items, assuming that we force it to run for all m iterations and not only until the marginal value of the items have dropped below their prices.*

Definition 5 *v is called greedy solvable if, for every price vector $\boldsymbol{q}$ the greedy algorithm outputs a member of the demand set $D_v(\boldsymbol{q})$.*

1. Let v be a gross substitute valuation function. Let $[m] := \{1, ..., m\}$ denote the greedy ordering of items. Prove that for $t := 1, ..., m$, the set $S_t := \{1, ..., t\}$ maximizes $v(T) - \sum_{j \in T} p_j$ over all $T \subseteq [m]$ s.t. $|T| = t$.
   For the proof you can use (1) the fact that if v is gross substitute then also for any price vector $\mathbf{p} := (p_1, ..., p_m)$ $v(S) - \sum_{j \in S} p_j$ is gross substitute; and (2) the following lemma: for every gross substitute valuation v the following holds: $\forall S \subseteq [m]$ and $\forall T \subseteq [m] \backslash S$ with $|T| \geq 2$, and every $i \in T$:
   $$v(i|S) + v(T \backslash \{i\}|S) \leq \max_{j \in T, j \neq i} \{v(j|S) + v(T \backslash \{j\}|S)\}$$

2. Reminder: a valuation function is called *sub-modular* if $v(i|S) \geq v(i|T)$ for every $i \notin T$ and sets S, T s.t. $S \subseteq T \subseteq [m]$. Prove that every gross substitute function is also sub modular.

3. Use 1+2 to show that if v is gross substitute then v is greedy solvable; then deduce the claim.

Problem 2: Consider a setting of n agents with n corresponding gross substitute valuation functions $v_i : 2^m \to \mathbb{R}_{\geq 0}$ and m items. Given value $r \in \mathbb{R}_{\geq 0}$ s.t. $r := \max_{\{S_i\}_i \text{ allocation}} \sum_i v_i(S_i)$ show that there is an algorithm that finds an allocation that maximizes the welfare and runs in $poly(m, n)$ time. Hint: consider the linear program that was defined in class for the welfare maximization problem.

2 Walrasian Equilibrium

Problem 3 a: Recall that a matching market is a market in which there are m items and n agents, each with a unit demand valuation. This setting can be represented by a bipartite graph G where the left vertices are the agents, the right vertices are the items, and the weight of each edge (i, j) is $v_i(\{j\})$. Given a price vector $\mathbf{p} = (p_1, ..., p_m)$ the demand set of an agent with valuation v is defined as $\{S \mid \forall T \subseteq [m]\ v(S) - \mathbf{p}(S) \geq v(T) - \mathbf{p}(T)\}$, where for any subset $S \subseteq [m]$ we define $p(S) := \sum_{i \in S} p_i$. In this question we prove the following theorem:

Theorem 6 *In a matching market, if OPT (i.e. maximum weighted matching) is unique, then there are walrasian prices s.t. for any agent, her demand set under these prices contains only one set. These prices can be computed in poly-time.*

In order to find the prices, we construct, based on G, an induces directed complete graph $G_I = (V_I, E_I)$ on m vertices (which represent the items) in the following way: let M be the maximum matching M in the original graph G. For each pair of items j, j', let the weight of the edge (j, j') in G_I be $v_i(\{j\}) - v_i(\{j'\})$, where i is the agent that was matched by M to j.

1. Show that G_I doesn't contain any negative weight cycles.
2. Show that G_I doesn't contain any cycles of weight zero.

   Now, change G_I as follows: add a vertex s to V_I, with a directed edge of weight 0 from s to each vertex in V_I.
3. Use the new graph G_I in order to compute walrasian prices for the market. Prove your answer (hint: consider shortest paths from s to each vertex of V_I).
4. Change the weights on the graph in a clever way s.t. computing the prices exactly as before yields a walrasian prices s.t. the demand set of each agent contains only one set (hint: use the fact that there are no zero weight cycles).

Problem 3 b: Show that the assumption on the uniqueness of the optimum in the previous question is necessary; i.e., show that in a matching market, when there is no unique optimum, for any static prices there is an arrival order of the agents s.t. if each agent takes an item which maximize her utility the welfare can't be more than $\frac{2}{3}$ of OPT in the worst case (hint: it is enough to consider a setting of 3 agents and 3 items).

3 Posted Prices Mechanism

Problem 4: Consider a setting of m items and n agents with general valuation functions $v_i : 2^m \to \mathbb{R}_{\geq 0}$. Show that there is a partition of the items into bundles and pricing of these bundles s.t. if the agents come in an arbitrary order and each takes a bundle that maximizes her utility (breaking ties arbitrary), then the SW is at least half of the maximum social welfare.

1. Fix an optimal solution $\{S_1, ..., S_n\}$. Find prices $\mathbf{p}$ for these bundles s.t. for any arrival order, if x_i is the allocation of agent i and $\mathbb{I}(S_i \text{ is sold})$ is the indicator variable which equals 1 when bundle S_i was acquired by some agent and 0 otherwise, the following holds:
$$\forall i \in [n] \;\; u_i(x_i, \mathbf{p}) + \mathbb{I}(S_i \text{ is sold}) \cdot \mathbf{p}(S_i) \geq \frac{v_i(S_i)}{2}$$
2. Conclude the stated argument.