

CS 580: Algorithmic Game Theory, Fall 2026

HW 1 Solutions

Problem 1: Competitive Equilibrium with Equal Incomes

Consider a market $\mathcal{M}$ with n agents and m divisible goods, where supply of good j is q_j and the valuation of agent i is defined by a monotonically non-decreasing concave function $V_i : \mathbb{R}_+^m \rightarrow \mathbb{R}_+$. A *competitive equilibrium with equal income (CEEI)* of such a market is a pair (X, p) where p is a price vector $(p_1, \dots, p_m)$, p_j is the price-per-unit of good j and $X = (X_1, \dots, X_n)$ is an allocation of goods to agents s.t.,

- **Optimal Bundle.** For each agent i ,

$$X_i \in \arg \max_{Y \geq 0: \sum_j p_j Y_j \leq 1} V_i(Y).$$

- **Demand equals Supply.** For each good j , $\sum_{i \in [n]} X_{ij} \leq q_j$, and whenever $p_j > 0$ we have $\sum_{i \in [n]} X_{ij} = q_j$.

Part A: Normalization to Unit Supply

Show that it is without loss of generality to assume that $q_j = 1$ for all goods j , that is, to assume that the supply of every good is one. Formally, come up with another market $\mathcal{M}' = ([n], [m], (q'_j)_{j \in [m]}, (V'_i)_{i \in [n]})$ such that $q'_j = 1, \forall j \in [m]$, and show that a CEEI of $\mathcal{M}'$ can be mapped to a CEEI of $\mathcal{M}$.

Proof Assume $q_j > 0$ for every good j . Define $\mathcal{M}'$ by setting $q'_j = 1$ and

$$V'_i(Y'_1, \dots, Y'_m) = V_i(q_1 Y'_1, \dots, q_m Y'_m) \quad \text{for every } Y' \in \mathbb{R}_+^m.$$

These valuations remain monotone and concave. Given a CEEI (X', p') of $\mathcal{M}'$, define:

- $X_{ij} = q_j X'_{ij}$ for every agent i and good j .
- $p_j = p'_j / q_j$ for every good j .

Feasibility and market clearing follow from

$$\sum_i X_{ij} = q_j \sum_i X'_{ij} \leq q_j,$$

with equality whenever $p_j > 0$, since then $p'_j > 0$.

For optimality, any bundle $Y \geq 0$ corresponds uniquely to the bundle Y' defined by $Y'_j = Y_j/q_j$. This correspondence preserves both expenditure and value:

$$\sum_j p_j Y_j = \sum_j p'_j Y'_j, \quad V_i(Y) = V'_i(Y').$$

Thus it maps affordable bundles bijectively between the two markets. Since X'_i is optimal in $\mathcal{M}'$, its image X_i is optimal in $\mathcal{M}$. Therefore (X, p) is a CEEI.

Part B: Polynomial-Time Computation of Event 2

Show that when the DPSV algorithm is executed to compute a CEEI of such a market, Event 2 of the algorithm can always be computed in polynomial time. That is, the value of α for which a new MBB edge appears between some agent $i \in A_D$ and some good $j \in G_F$, and the pair (i, j) , can be computed in polynomial time. Here, α is the constant by which the prices of the goods in G_D are scaled in one iteration of the algorithm. Set A_D consists of agents with MBB edges to goods in G_D before α starts increasing, and G_F is the set of goods with demand = supply, meaning if the market consisted only of the subset (A_F, G_F) , then the prices of goods in G_F and their allocation to A_F satisfy the requirements of a CEEI. (See Lecture 3 slides for further details)

Proof In the additive-valuation setting, write $V_i(Y) = \sum_j v_{ij} Y_j$, and let $p_j > 0$ denote the current prices. If $A_D = \emptyset$ or $G_F = \emptyset$, no such event is possible. Otherwise, for each $i \in A_D$, define

$$d_i = \max_{j \in G_D} \frac{v_{ij}}{p_j}, \quad f_i = \max_{j \in G_F} \frac{v_{ij}}{p_j}.$$

Before Event 2, no MBB edge connects A_D to G_F , so $d_i > f_i$. Scaling the prices in G_D by α changes these ratios to d_i/α and f_i , respectively. A frozen good therefore becomes MBB for agent i when

$$\alpha(i) = \frac{d_i}{f_i}.$$

Set $\alpha(i) = +\infty$ if $f_i = 0$, since no finite scaling creates such an edge for that agent.

The first new edge appears at

$$\alpha^* = \min_{i \in A_D} \alpha(i).$$

If $\alpha^* = +\infty$, Event 2 does not occur. Otherwise, choose

$$i^* \in \arg \min_{i \in A_D} \alpha(i), \quad j^* \in \arg \max_{j \in G_F} \frac{v_{i^*j}}{p_j}.$$

At α^* , the edge (i^*, j^*) becomes an MBB edge.

Computing these maxima and minima takes $O(|A_D|m)$ arithmetic operations. For rationally encoded valuations and prices, the divisions and comparisons have polynomial bit complexity. Thus both α^* and a corresponding pair (i^*, j^*) can be computed in polynomial time.

Problem 2: Prop1+PO Allocation

Design a polynomial-time algorithm to compute a Prop1+PO allocation. The algorithm should find a Prop1+PO allocation of m indivisible items among n agents, each with an additive valuation function, namely $v_i(S) = \sum_{j \in S} v_{ij}$ for any $S \subseteq [m]$

Example Construction

The example construction we present is based on work from *Barman and Krishnamurthy, On the Proximity of Markets with Integral Equilibria (AAAI 2019)*

Assume the additive values v_{ij} are nonnegative rationals. We use the polynomial-time algorithm for computing a linear Fisher-market equilibrium (DPSV) as a subroutine.

Prop1+PO Allocation Algorithm.

Preprocessing. Temporarily remove agents who value every good at zero and goods valued at zero by everyone. Write I and J for the remaining agents and goods, and let $k = |I| \leq n$. If $k = 0$, any allocation works. Otherwise, apply the following algorithm, then restore the removed agents with empty bundles and distribute the removed goods arbitrarily. These goods affect no agent's value.

Main Algorithm.

1. **Compute a fractional equilibrium.** Treat the goods in J as divisible, with one unit of each good, and give each agent in I budget 1. Compute a CEEI (x, p) . All prices are positive, and every agent spends its full budget. Define $\alpha_i = \max_{j \in J} v_{ij}/p_j > 0$. The equilibrium conditions give

$$\sum_{i \in I} x_{ij} = 1, \quad \sum_{j \in J} p_j x_{ij} = 1, \quad x_{ij} > 0 \implies v_{ij} = \alpha_i p_j. \quad (1)$$

The last condition says that agents buy only maximum-bang-per-buck (MBB) goods.

2. **Make the spending graph a forest.** Form the bipartite graph of agents and goods, with an edge (i, j) of weight $b_{ij} = p_j x_{ij}$ whenever $x_{ij} > 0$. While a cycle remains, mark its edges alternately $+$ and $-$, and let ε be the smallest weight on a $-$ edge. Increase all $+$ weights by ε and decrease all $-$ weights by ε , deleting edges that reach zero.

At each vertex, one incident weight increases and one decreases, so every agent's total expenditure and every good's total revenue are unchanged. No new edge is introduced, and at least one edge is removed. Thus the process ends in a forest. Set $x_{ij} = b_{ij}/p_j$ using the final weights. All conditions in (1) still hold, since all surviving edges are MBB edges.

3. **Round the forest.** Root each tree at any agent and give each good entirely to its parent agent. Every good has exactly one parent, so this is an integral allocation of J . Let X denote the final allocation after restoring the removed agents and goods.

Proof of Prop1. First, the fractional allocation after Step 2 is proportional. Indeed, market clearing and budget exhaustion imply $\sum_{j \in J} p_j = k$. For every active agent i , the MBB conditions give

$$u_i := \sum_{j \in J} v_{ij} x_{ij} = \alpha_i, \quad v_i(M) = \sum_{j \in J} v_{ij} \leq \alpha_i \sum_{j \in J} p_j = k \alpha_i.$$

Hence $u_i \geq v_i(M)/k \geq v_i(M)/n$.

If i is a root, it receives every good in its fractional support, so $v_i(X_i) \geq u_i$. Otherwise, let g_i be its parent good. Agent i receives every good in its fractional support except g_i , which is assigned to another agent. Since each fractional share is at most 1,

$$v_i(X_i \cup \{g_i\}) \geq \sum_{j \in J} v_{ij} x_{ij} = u_i \geq \frac{v_i(M)}{n}.$$

Thus adding at most one good makes every active agent proportional. Removed agents have identically zero value, so they also satisfy Prop1.

Proof of Pareto Optimality. Give each active agent the positive weight $1/\alpha_i$. For every good, $v_{ij}/\alpha_i \leq p_j$, with equality whenever $j \in X_i \cap J$, because rounding assigns goods only along MBB edges. Therefore, for any other allocation Y of the original goods,

$$\sum_{i \in I} \frac{v_i(Y_i)}{\alpha_i} \leq \sum_{j \in J} p_j = \sum_{i \in I} \frac{v_i(X_i)}{\alpha_i}.$$

The inequality holds because each good is allocated at most once. A Pareto improvement would strictly increase this weighted sum: all weights are positive, and a removed agent cannot strictly improve. This is a contradiction, proving PO. Notice that this argument does *not* require the rounded bundles to respect the original budgets.

Justification of Polynomial Runtime. DPSV computes the equilibrium in polynomial time. The spending graph has at most nm edges. Each cycle cancellation removes an edge, and finding and updating a cycle takes polynomial time. Rooting the forest and assigning goods also take polynomial time. The arithmetic has polynomial bit complexity: if D is the product of the denominators of the initial expenditures, then $\log D$ is polynomial in the input length, and every later expenditure remains an integer multiple of $1/D$ in $[0, 1]$. Thus the entire algorithm is polynomial-time and returns a Prop1 + PO allocation. $\square$