

Economics and Computation (Spring 2026)

Assignment #3

Due: 3/24/2026 11:59pm ET

Problem 1: The VCG Mechanism

[20 points] Under the VCG Mechanism, the allocation rule maximizes (utilitarian) social welfare. Consider, instead, a *weighted social welfare* objective, defined as $\max_{x \in A} \sum_{i \in N} \alpha_i v_i(x)$, for given multipliers $\alpha_i \geq 0$, $i = 1, \dots, n$.

For an allocation rule optimizing weighted social welfare, design a payment rule such that the mechanism is strategyproof.

Solution:

Proposed Payment Rule: In line with the VCG Mechanism we have studied in class, we want to charge agent i with the externality they impose on others. First, we follow the allocation rule and choose $x^*(v)$ that maximizes the weighted social welfare:

$$x^*(v) \in \arg \max_{x \in A} \sum_{i \in N} \alpha_i v_i(x)$$

For the payment method, we have two cases depending on whether α_i is 0.

If $\alpha_i \neq 0$, we define the payment method for agent i to be

$$p_i(v) = h_i(v_{-i}) - \sum_{j \neq i} \frac{\alpha_j}{\alpha_i} v_j(x^*(v))$$

where h is an arbitrary function that does not depend on v_i .

If $\alpha_i = 0$, then we define the payment rule for player i as $p_i(\mathbf{v}) = 0$.

Strategyproofness: We want to show that no misreport will yield a higher utility for agent i . For $\alpha_i \neq 0$, we have

$$\begin{aligned} u_i(v) &= v_i(x^*(v)) - p_i(v) \\ &= v_i(x^*(v)) - h_i(v_{-i}) + \sum_{j \neq i} \frac{\alpha_j}{\alpha_i} v_j(x^*(v)) \end{aligned}$$

We have designed a function that is simply the maximized weighted social welfare with $\frac{1}{\alpha_i}$. Here, we notice when an agent i reports $\hat{v}_i \neq v_i$, it can only impact the utility through the x^* chosen from the objective function.

$$\begin{aligned} \alpha_i \left[v_i(x^*(v)) + \sum_{j \neq i} \frac{\alpha_j}{\alpha_i} v_j(x^*(v)) \right] &= \alpha_i v_i(x^*(v)) + \sum_{j \neq i} \alpha_j v_j(x^*(v)) \\ &= \sum_{i \in N} \alpha_i v_i(x^*(v)) \end{aligned}$$

If agent i overstates or understates \hat{v}_i , then they risk selecting an outcome that lowers the utility compared to the maximizing choice x^* .

For $\alpha_i = 0$, we have

$$f \in \operatorname{argmax}_{x \in A} \sum_{j \in N} \alpha_j v_j(x) = \operatorname{argmax}_{x \in A} \sum_{\substack{j \in N \\ j \neq i}} \alpha_j v_j(x).$$

This means even if player i deviates from their true valuation, f would not change i.e., $f(\mathbf{v}) = f(v'_i, \mathbf{v}_{-i})$ for all value preference v_i . Thus, we get that

$$u_i(f(\mathbf{v})) = v_i(f(\mathbf{v})) - p_i(\mathbf{v}) = v_i(f(v'_i, \mathbf{v}_{-i})) = u_i(f(v'_i, \mathbf{v}_{-i})),$$

as desired.

Problem 2: Strategyproof approximation algorithms

[20 points] In Lecture 9, we discussed the greedy mechanism for single-minded bidders. Let us modify the mechanism by tweaking the allocation rule: instead of ordering bids by decreasing w_i , sort the bids by decreasing $w_i/\sqrt{|T_i|}$. It turns out that this gives a \sqrt{m} -approximation.

Prove that the modified greedy algorithm is strategyproof.

Guidance: Adapt the lemma on Slide 7 by showing that the critical value of i is $\min\{w'_i : w'_i/\sqrt{|T_i|} \geq \max_{j \in N'_i(T_i)} w_j/\sqrt{|T_j|}\}$. Next, adapt the proof on Slide 8.

Solution: Critical value of bidder in greedy single-minded auctions. Consider Figure 1. Let N_i be the set of winners if i is removed. Let N'_i be the conflict set of i . First, observe that if $j \geq 0$ bids are allocated before bid i then these are the same bids, and allocated in the same order, as the first j bids allocated in N_i . Thus, a bid with value $\hat{w}_i = w_{crit,i}(T_i)$ wins because any bids allocated before i are in $N_i \setminus N'_i$, and thus don't block i . Also, if $\hat{w}_i < w_{crit,i}(T_i)$, then i is not allocated because at least one of the bidders in N'_i is allocated before i , and blocks i from being allocated.

Strategyproofness of greedy single-minded auctions for a monotone score function.

Consider monotone score function $\sigma(T_i, w_i) = w_i/\sqrt{|T_i|}$. For any such T_i , the allocation is monotone weakly increasing in report w_i : if allocated for report w'_i then i is allocated for any $w''_i > w'_i$ since the score of the bid weakly increases (score function is monotone) and so no new bids can be allocated before i at w''_i than w'_i (and i continues to be scored ahead of any bids in $N'_i(T_i)$). From this, and following the above lemma, bidder i is allocated bundle T_i at price $w_{crit,i}(T_i)$ if and only if $\hat{w}_i \geq w_{crit,i}(T_i)$. Since $w_{crit,i}(T_i)$ is self-report independent, bidding $\hat{w}_i = w_i$ is optimal.



Figure 1: From Slide 7 of Mechanism Design 2 lecture. Given that the target bundles includes the pineapple and the pear, the 3rd and 5th bundles from the right are in the conflict set, and i needs to score ahead of both of them to be allocated.

Problem 3: Cake cutting

[30 points] In class we discussed the Even-Paz Algorithm, which guarantees a proportional allocation of the cake with $O(n \log n)$ queries in the Robertson-Webb Model. However, in the analysis we made the simplifying assumption that $n = 2^k$ for some $k \in \mathbb{N}$.

Generalize the algorithm to an arbitrary number of players n , and prove that your generalized algorithm is proportional and that it requires $O(n \log n)$ queries.

Solution: Algorithm (divide and conquer):

If $n = 1$: the single player takes the whole cake.

If $n = 2k$ for some $k \in \mathbb{N}^+$: every player draws a vertical line on the cake that cuts it in half according to their valuation function. Cut the cake anywhere between the k th and $(k + 1)$ th lines inclusive. The players who drew the k leftmost lines recurse on the left piece and the players who drew the k rightmost lines recurse on the right piece.

If $n = 2k + 1$ for some $k \in \mathbb{N}^+$: every player draws a vertical line such that, according to their valuation function, the left side of the line is worth $\frac{k}{2k+1}$ of the total cake and the right side of the line is worth $\frac{k+1}{2k+1}$ of the total cake. Cut the cake on the median line. The players who drew the k leftmost lines recurse on the left piece and the players who drew the $k + 1$ rightmost lines recurse on the right piece.

Proof of proportionality: This algorithm is proportional: for any piece C of cake and for any number n of players, each player will walk away with a slice S_i that they feel is worth at least $\frac{1}{n}$ of the value of C . (Equivalently, $\forall i, n \cdot V_i(S_i) \geq V_i(C)$). The proof is by induction.

Base case ($n = 1$): The player walks away with all of the remaining cake. $1 \cdot V_i(C) \geq V_i(C)$.

Induction hypothesis: for all $k \leq n$, this algorithm always achieves a proportional allocation among $(k - 1)$ players.

Induction step ($n = 2k$): Note that after the cake is cut, the k players who recurse on the left piece (call it L) value it at least half as much as they value C . By the IH, each player i will end up with a piece that they think is worth at least $\frac{V_i(L)}{k}$. But since we just showed that $V_i(L) \geq \frac{V_i(C)}{2}$ for these players, this value is at least $\frac{V_i(C)}{n}$. So proportionality is achieved for these k players. An identical argument applies to the other k players who recurse on the right piece.

Induction step ($n = 2k + 1$): Note that after the cake is cut, the k players who recurse on the left piece (call it L) value it at least $\frac{k}{2k+1}$ as much as they value C . By the IH, each player i will end up with a piece that they think is worth at least

$$\frac{V_i(L)}{k} \geq \frac{V_i(C)}{2k+1} = \frac{V_i(C)}{n}.$$

A structurally identical argument applies to the $k + 1$ players who recurse on the right piece: by the IH, they each end up with a piece that they think is worth at least

$$\frac{V_i(R)}{k+1} \geq \frac{V_i(C)}{2k+1} = \frac{V_i(C)}{n}.$$

So proportionality is achieved for everybody.

Proof of runtime: The algorithm makes $O(n)$ cuts at each level of recursion. And there are $O(\log n)$ levels of recursion, because at each step, at least $1/3$ of the players are removed, so the depth of the recursion tree is at most $\log_{3/2} n$.

Problem 4: Rent division

[30 points] In class we stated two lemmas in the context of the rent division with quasi-linear utilities. The second lemma can actually be strengthened as follows:

Lemma: If (π, \mathbf{p}) is an EF solution and σ is a welfare-maximizing assignment, then for all $i \in N$,
 $v_{i\pi(i)} - p_{\pi(i)} = v_{i\sigma(i)} - p_{\sigma(i)}$.

Using this lemma (without proof), describe a polynomial-time algorithm for computing the maximin solution and prove its correctness.

Guidance: Follow the algorithm sketch on Slide 19 of Lecture 11 but replace the system of linear inequalities with an appropriate linear program. You may assume that a linear program can be solved in polynomial time. Note that a linear program can maximize the minimum of linear functions, similarly to our linear program for computing a maximin strategy in a two-player zero-sum game.

Solution: We first find the matching of players to rooms of maximum weight, where the weight of an edge between a player and a room is the utility that the player has for the room. This can be done in polynomial time, and the resulting allocation σ will be a welfare-maximizing assignment. We then compute the price vector $\mathbf{p} = (p_1, p_2, \dots, p_n)$ by solving the following linear program:

$$\begin{array}{ll}
\text{maximize} & t \\
\text{subject to} & t \leq v_{i\sigma(i)} - p_{\sigma(i)} \quad \text{for all } i \in N \\
& v_{i\sigma(i)} - p_{\sigma(i)} \geq v_{ij} - p_j \quad \text{for all } i, j \in N \\
& \sum_{j=1}^n p_j = R
\end{array}$$

Our computed solution (σ, \mathbf{p}) is clearly EF since we enforced this in the constraints, so all that remains is to show that it maximizes the minimum utility of any player.

Observe that, given a price vector, the optimal t is the minimum utility of any agent under assignment σ according to those prices. Therefore, since we are maximizing t , the optimal price vector \mathbf{p} that we compute maximizes the minimum utility of any agent under assignment σ and prices satisfying EF. Suppose the true Maximin solution is (σ^*, \mathbf{p}^*) . Since, by the definition of the Maximin solution, (σ^*, \mathbf{p}^*) is EF, the lemma implies that, for all $i \in N$,

$$v_{i\sigma(i)} - p_{\sigma(i)}^* = v_{i\sigma^*(i)} - p_{\sigma^*(i)}^*.$$

This means that \mathbf{p}^* is a feasible solution to the LP, as (σ, \mathbf{p}^*) must clearly be EF by the lemma. Therefore,

$$\min_i v_{i\sigma(i)} - p_{\sigma(i)} \geq \min_i v_{i\sigma(i)} - p_{\sigma(i)}^* = \min_i v_{i\sigma^*(i)} - p_{\sigma^*(i)}^*,$$

where the inequality follows from the fact that \mathbf{p} maximized the LP and \mathbf{p}^* is feasible, and the equality follows from the lemma. In other words, (σ, \mathbf{p}) achieves at least as high a minimum utility as the optimal minimum utility (subject to EF), so it must also have an optimal minimum utility (subject to EF). Thus, (σ, \mathbf{p}) is a Minimax solution.