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 $\alpha_i \geq 0$, $i = 1, \ldots, n$.

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

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.

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.

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 $(\pi, \mathbf{p})$ 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.