CS 5854: Networks, Crowds, and Markets
Homework 3

Instructor: Rafael Pass   TAs: Cody Freitag, Drishti Wali

Assigned: October 31, 2019   Due: November 21, 2019, 11:59 pm

Homework Policies and Guidelines:

Submission: Your homework solutions must be **typed** and submitted as a single .pdf file in CMS. You must additionally submit a single .zip file containing all relevant files specified in the assignment for all coding problems. Template .tex and .py files will be provided containing an outline for your submission.

Late Days: Each student may use four "late days" in total as desired throughout the semester, each of which grants a 24-hour extension to an assignment's due date. Late work beyond this limit will still be accepted and graded until grades have been released for that assignment, but (unless discussed in advance with the TA and/or instructor) will have a negative impact on final grades.

Collaborations: You may work in groups of up to **4** students. Every member of the group must list all other collaborators at the top of their assignment. *(Note: the maximum size of a connected component of groups must be 4. If A,B,C, and D work together, it must not be the case that A and E work together.)*

Your submitted answers, explanations, and discussion for all written questions in the pdf must be your own, individual, unique solutions. You *may* share and even submit the same code and graphs with your collaborators. Additionally, you may make use of published material (papers, github, wikipedia, etc.), provided that you acknowledge and specifically cite all sources used. It is considered a violation of academic integrity to submit a problem solution that you are unable to explain orally to a member of the course staff.

To make this further clear, you are encouraged to discuss problems and examples and work on code in a group, but you must then separately write up the solution yourself. You may not copy any collaborator or external resource even if you cite them. It is up to the TAs and graders to determine whether or not two submissions (or a submission and an external source) are too similar to be considered "your own words." Furthermore, you may receive a 0 for any problem that the TAs and graders determine is not your own work even if cited.

How to receive credit: You must **justify all answers** to receive credit unless specified otherwise. We will do our best to make clear the level of justification we expect for each problem. For coding questions, please turn in complete, executable code for each part of the question that asks for an implementation, and include a .txt file containing any required outputs if not already included in the main .pdf file. *Using Python is strongly preferred.* If you want to use another language, you must first consult with the TAs. We will not grade code based on style, but we may mark down code if we are unable to understand what it is doing. You may use standard libraries to implement data structures such as graphs, but, unless otherwise specified, you may not use pre-existing implementations of any algorithms without express permission from the TAs. (If a problem asks you to implement X and you use a package that implements X for you, you will not get credit.)

Part 1: Matching Markets and Exchange Networks

1. Consider two sellers, a and b, each offering a distinct house for sale, and a set of two buyers, x and y. x has a value of 2 for a's house and 4 for b's house, while y has a value of 3 for a's house and 6 for b's house, summarized in the following table.

Buyer	a's house	b's house
value for x	2	4
value for y	3	6

   (a) Suppose that a charges a price of 0 for her house, and b charges a price of 1. Is this set of prices market-clearing? As part of your answer, say what the preferred choice graph is with this given set of prices, and use this to justify your answer.

   (b) If the above prices are not market-clearing, compute a set of market-clearing prices p with an associated matching M. Explain how you found them and why they are market-clearing (you don't necessarily need to use Theorem 8.8 in the notes for this, but it could be helpful).

   (c) For the market equilibrium computed above (corresponding to a set of market-clearing prices), compute the social value of the buyers and the social welfare of the buyers and sellers. Briefly explain why these are the maximum possible in this matching market context.

2. Suppose now we have a set of three sellers a, b, and c, and a three buyers labeled, x, y, and z. The valuations of the buyers are summarized in the following table.

Buyer	a's house	b's house	c's house
value for x	5	7	1
value for y	2	3	1
value for z	5	4	4

   (a) Using the algorithm from Theorem 8.8 in the notes, compute a market equilibrium (consisting of a matching and set of prices). Show your work.

   (b) Recall that in chapter 9, we showed that every matching market context can be encoded as an exchange network, and furthermore, exchange networks are "asymmetric" in the sense that the buyers and sellers have the same role. As a result, we can flip the script such that the sellers are now the "buyers" and the buyers are now the "sellers." The corresponding matching market context now has the following set of values.

Seller	if x buys	if y buys	if z buys
value possible for a	5	2	5
value possible for b	7	3	4
value possible for c	1	1	4

   You can think of the value of a seller being the maximum value it can charge for a home. Now the prices for the buyer correspond to discounts on the house (increasing value for the buyers).

Using the algorithm from Theorem 8.8 in the notes, compute a market equilibrium (consisting of a matching and set of "prices") for this new matching market context. Show your work.

(c) Interpret the matching and "prices" computed in part (b) as a matching and actual prices charged when selling the houses. How do these values compare to those computed in part (a)? Give a 1-3 sentence intuitive explanation of why the values might be different. In particular, explain what the consequence is that the algorithm of Theorem 8.8 always has a "free" item in the resulting market equilibrium.

3. (a) We observed (Claim 9.5) that an exchange network consisting of a cyclic graph of 3 nodes has no stable outcome. Does this generalize to every cyclic graph? If not, can we characterize for which values of n the n-node cyclic graph has a stable outcome? Justify your answers.

(b) Show by constructing an example that an exchange network that contains a 3-node cycle (but doesn't necessarily entirely consist of one) can still have a stable outcome.

Part 2: Auctions and Mechanism Design

4. Consider an auction where a seller wants to sell one unit of a good to a group of bidders. The seller runs a sealed-bid, second-price auction. Your firm will bid in the auction, but it does not know for sure how many other bidders will participate in the auction. There will be either two or three other bidders in addition to your firm. All bidders have independent, private values for the good. Your firm's value for the good is c. What bid should your firm submit, and how does it depend on the number of other bidders who show up? Give a brief (1-3 sentence) explanation for your answer.

5. Consider a second-price, sealed-bid auction with two bidders who have independent, private values v_i which are either 1 or 3. For each bidder, suppose the probabilities of v_i being 1 or 3 are independent (from other bidders) and both $1/2$. (If there is a tie at a bid of x for the highest bid the winner is selected at random from among the highest bidders and the price is x.)

(a) What is the seller's expected revenue from the auction? Justify your answer.

(b) Assume that we add a third bidder that behaves identically to the first two (whose value v_i is also chosen independently of those for the first two). What is the seller's expected revenue? Justify your answer.

(c) Explain the trend you noticed between parts (a) and (b)—that is, why changing the number of bidders affects (or doesn't affect) the seller's expected revenue.

* **Bonus Question 1.** Let's say we define a "third-price auction" in the same manner that we defined the first-price and second-price auctions. Is this mechanism DST, NT, or neither? Does it DST-implement, NT-implement, Nash-implement, or fail to implement social welfare maximization? Justify your answers, by proof or by counterexample.

Part 3: Sponsored Search

6. Suppose a search engine has two ad slots that it can sell. Slot a has a clickthrough rate of 10 and slot b has a clickthrough rate of 5. There are three advertisers who are interested in these slots. Advertiser x values clicks at 3 per click, advertiser y values clicks at 2 per click, and advertiser z values clicks at 1 per click.

   Compute the socially optimal allocation and the VCG prices for it (using the Clarke pivot rule). Show your work in full, and explain what your answer means in the sponsored search context.

Part 4: Implementing Matching Market Pricing

The goal of this exercise is to implement an algorithm for finding market-clearing and VCG prices in a bipartite matching market. Include your code in matching_market.py.

7. Recall the procedure constructed in Theorem 8.8 of the notes to find a market equilibrium in a matching market.

   (a) The first step of this procedure involves either finding a perfect matching or a constricted set. Recall that this can be done using maximum flow. Now, using your maximum-flow implementation from assignment 2, implement an algorithm that finds either a perfect matching M or a constricted set S in a bipartite graph.

   (b) Now, given a bipartite matching frame with n players, n items, and values of each player for each item, implement the full procedure to find a market equilibrium.

   (c) Submit your code along with its output on the matching market frame in figure 8.3 as well as 3 other small (10-20 node) test examples of your choice. Include the input and outputs in a file called p7.txt.

8. Now, given a matching market frame, we will implement VCG pricing in this frame according to the results of Theorem 10.8 in the notes.

   (a) In order to do this, we must find the socially optimal outcome. Briefly justify that the outcome that your algorithm from the previous question finds is socially optimal (this can be done by simply stating 1-2 theorems from the notes).

   (b) Finally, implement the VCG mechanism and Clarke pivot rule to construct an algorithm that produces a positive set of VCG prices in any matching market frame.

   (c) Submit your code along with its output on the matching market frame in figure 8.3 as well as 3 other small (10-20 node) test examples of your choice. Include the input and outputs in a file called p8.txt.

9. Now simulate your VCG pricing algorithm in the following context.

   (a) First, construct a graph of 20 buyers and 20 items. Assume that "item" i is actually a bundle of i identical goods. Now assign each player a random value per good (say, from 1 to 50; ties can be allowed). You shouldn't need to do any additional work on your algorithm to do this, instead just set each buyer's value per bundle appropriately. How should we set these values?

(b) Having set the values accordingly, run your VCG pricing algorithm and turn in the results in a file called p9.txt. Explain why the results you obtained make sense in the context above.

* **Bonus Question 2.** (Please note that each part of this question is intended to be harder than the previous parts; each part will be graded separately, and you are not required to do the whole thing.)

(a) Implement an algorithm for GSP pricing in a matching-market context. Compare your VCG prices from the previous question to the GSP prices when run in the same contexts (with the same randomness). Also, try to find and characterize some contexts where VCG and GSP prices are similar, and where they are wildly different. Include a file called bonus-2a.txt displaying input and output on a few examples.

(b) GSP isn't truthful, so interesting things might happen if we run BRD on a GSP matching market auction (where a player's "strategy" is their valuation report). Implement an algorithm that picks a random starting point and runs BRD to attempt to find a GSP equilibrium state. What happens? Does BRD converge; if so, how quickly? Include a file called bonus-2b.txt displaying input and output on a few examples.

(c) *(Very difficult.)* Either prove that BRD will converge in a GSP context, or disprove it by counterexample.

Part 5: Exchange Networks for Uber

We will construct a simplified market scenario for a ridesharing app like Uber. Our world will consist of an $n \times n$ grid (think of 100×100 as a test example), and there will be two types of participants, riders and drivers.

- A rider R is specified by a current location $(x_0, y_0) \in [n] \times [n]$, a desired destination $(x_1, y_1) \in [n] \times [n]$, and a value for reaching that destination.

- A driver D is specified by a current location $(x_0, y_0) \in [n] \times [n]$.

We define the cost of a matching between a rider R and a driver D, $c(R, D)$, to be the distance from the driver to the rider and then to the destination (measured via manhattan distance, so the distance from $(0, 0)$ to $(5, 2)$ is 7).

10. Encode the above example as an exchange network and implement it in 'uber.py.' Namely, define a graph $G = (V, E)$ where the vertices are the riders and drivers and there is an edge between every rider and driver. What value should we associate with each edge in the graph? Justify your answer.

11. Using your algorithm for matching markets above, implement a procedure (in uber.py) that computes a stable outcome in this context. Note that there may not be the same number of riders and drivers.

(a) Construct at least two test examples with at least 5 riders and drivers. Discuss what the stable outcome is for your chosen examples. Explain what your results say about how much the riders are charged and how much the drivers are profiting.

(b) Implement a procedure that on a 100×100 grid (1) generates r riders (located randomly in the grid) each with a value of 100, d drivers (also located randomly in the grid) and (2) computes a stable outcome given this context. Run your procedure many times when $r = d$ (say, 10 each), r is much less than d (say, 5 vs. 20), and when r is much greater than d (say, 20 vs. 5). Discuss your results in detail. Specifically, discuss prices and profits for different ranges of r and d.

12. With ridesharing apps, drivers often have preferences for where they want to drive. For example, they know they are more likely to get a high value ride from the airport. Describe in a few sentences how you you could modify the existing setup to include driver's preferences.

* **Bonus question 3** Suppose that the city implements a public transportation system. The cost to take the public transportation system is $a + b \cdot dist$(initial location, destintation), where a is a fixed base fare and b is some constant factor multiplier. Furthermore, anyone can take this public transportation system from any location.

(a) Implement a procedure that includes the public transportation option in when computing the stable outcome above. Does a stable outcome always exist in this case?

(b) Choose different values for a and b and discuss how it affects the prices charged to riders/ profits of drivers.

(c) (Open ended: variable points awarded based on quality of solution.) Extend your context to include other additional factors that you might find interesting. Summarize what you did and what results you found in your writeup.