Homework Assignment 3 - Coding Part Write-up
Networks and Markets

Omer Zohar Gil Aharoni Adam Tuby

August 9, 2024

Part 4: Implementing Matching Market Pricing

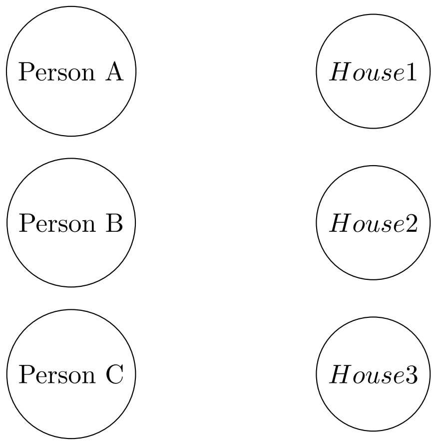
1 Question 7

(b) Consider the matching market example in Lecture 5 Page 7:

Values for Person A: $\{House1 : 4, House2 : 12, House3 : 5\}$

Values for Person B: $\{House1 : 7, House2 : 10, House3 : 9\}$

Values for Person C: $\{House1 : 7, House2 : 7, House3 : 10\}$



Formally, the matching market context is $\Gamma = (\{A, B, C\}, \{1, 2, 3\}, v)$, where v is the valuation function defined as follows:

$$v_A(1) = 4, v_A(2) = 12, v_A(3) = 5$$
$$v_B(1) = 7, v_B(2) = 10, v_B(3) = 9$$
$$v_C(1) = 7, v_C(2) = 7, v_C(3) = 10$$

We turn to run the algorithm of Theorem 8.8 to find a market equilibrium (p, M) to find the maximum social value, in order to validate out implementation's output. We begin by initializing the prices vector $\vec{p} \equiv 0$ to be the zero vector. We then proceed to run the algorithm, updating the prices vector until there is a perfect matching M in the induced preferred choice graph for $(\Gamma, \vec{p})$:

1. Observing the following *induced preferred-choice graph* from $(\Gamma, \vec{p})$:

Values for A: $\{1 : 4, 2 : 12, 3 : 5\}$

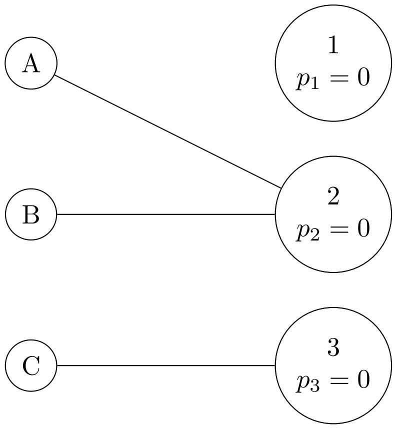
Utility for A: $\{1 : 4, 2 : 12, 3 : 5\}$

Values for B: $\{1 : 7, 2 : 10, 3 : 9\}$

Utility for B: $\{1 : 7, 2 : 10, 3 : 9\}$

Values for C: $\{1 : 7, 2 : 7, 3 : 10\}$

Utility for C: $\{1 : 7, 2 : 7, 3 : 10\}$



There obviously isn't a perfect matching as $S = \{A, B\}$ is a constricted set with $|N(S)| = |\{2\}| = 1 < 2 = |S|$ (which, by a theorem we've seen in class, implies that there isn't a perfect matching). Thus, we raise the prices for all items in $N(S)$ by 1, and update the prices vector $\vec{p}$ accordingly. The updated prices vector is $\vec{p} = (a : 0, b : 1, c : 0)$. Not all prices are greater than zero, so we don't perform a shift operation, and we proceed to the next iteration.

2. Observing the following *induced preferred-choice graph* from $(\Gamma, \vec{p})$:

Values for A: $\{1 : 4, 2 : 12, 3 : 5\}$

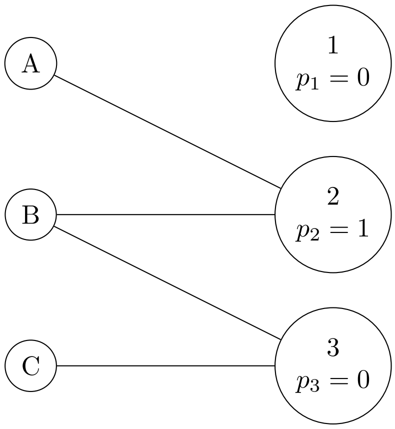
Utility for A: $\{1 : 4, 2 : 11, 3 : 5\}$

Values for B: $\{1 : 7, 2 : 10, 3 : 9\}$

Utility for B: $\{1 : 7, 2 : 9, 3 : 9\}$

Values for C: $\{1 : 7, 2 : 7, 3 : 10\}$

Utility for C: $\{1 : 7, 2 : 6, 3 : 10\}$



There obviously isn't a perfect matching as $S = \{A, B, C\}$ is a constricted set with $|N(S)| = |\{2, 3\}| = 2 < 3 = |S|$ (which, by a theorem we've seen in class, implies that there isn't a perfect matching). Thus, we raise the prices for all items in $N(S)$ by 1, and update the prices vector $\vec{p}$ accordingly. The updated prices vector is $\vec{p} = (a : 0, b : 2, c : 1)$. Not all prices are greater than zero, so we don't perform a shift operation, and we proceed to the next iteration.

3. Observing the following *induced preferred-choice graph* from $(\Gamma, \vec{p})$:

Values for A: $\{1 : 4, 2 : 12, 3 : 5\}$

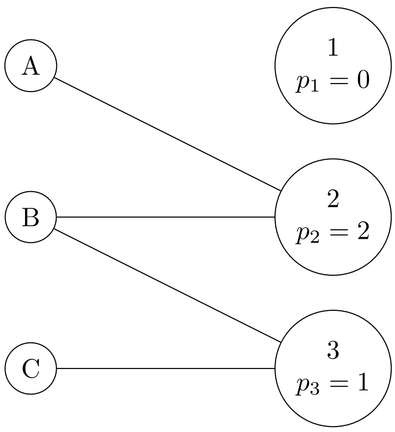
Utility for A: $\{1 : 4, 2 : 10, 3 : 4\}$

Values for B: $\{1 : 7, 2 : 10, 3 : 9\}$

Utility for B: $\{1 : 7, 2 : 8, 3 : 8\}$

Values for C: $\{1 : 7, 2 : 7, 3 : 10\}$

Utility for C: $\{1 : 7, 2 : 5, 3 : 9\}$



Similar to the previous iteration, we raise the prices for $\{2, 3\}$, and update the prices vector

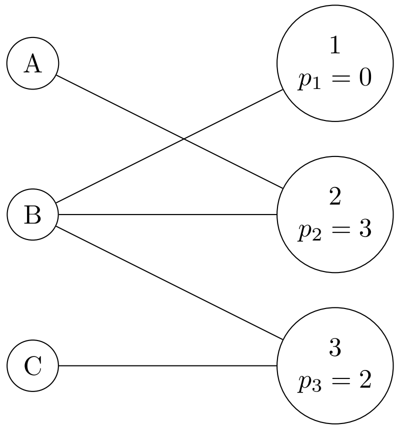
$\vec{p}$ accordingly. The updated prices vector is $\vec{p} = (a : 0, b : 3, c : 2)$. Not all prices are greater than zero, so we don't perform a shift operation, and we proceed to the next iteration.

4. Observing the following *induced preferred-choice graph* from $(\Gamma, \vec{p})$:

Values for A: $\{1 : 4, 2 : 12, 3 : 5\}$
Utility for A: $\{1 : 4, 2 : 9, 3 : 3\}$

Values for B: $\{1 : 7, 2 : 10, 3 : 9\}$
Utility for B: $\{1 : 7, 2 : 7, 3 : 7\}$

Values for C: $\{1 : 7, 2 : 7, 3 : 10\}$
Utility for C: $\{1 : 7, 2 : 4, 3 : 8\}$



And there is a perfect matching in the induced preferred choice graph, which is $M = \{\{A, 2\}, \{B, 1\}, \{C, 3\}\}$. Thus, the market equilibrium is $(\vec{p}, M) = ((1 : 0, 2 : 3, 3 : 2), \{\{A, 2\}, \{B, 1\}, \{C, 3\}\})$, and we are done.

We found the market equilibrium to be $(\vec{p}, M) = ((1 : 0, 2 : 3, 3 : 2), \{\{A, 2\}, \{B, 1\}, \{C, 3\}\})$. The maximum social value is therefore $v(A, 2) + v(B, 1) + v(C, 3) = 12 + 7 + 10 = 29$.

Our algorithm found exactly this market equilibrium.

2 Question 8

(a) In this part we analyze how the prices output by the VCG mechanism compare with the ones output by the algorithm of Theorem 8.8 (finding a market equilibrium (p, M)). The following are the examples we analyze and their corresponding results for each mechanism:

1. Example 1:

Values for b_1: $\{i_1 : 3, i_2 : 0, i_3 : 15\}$

Values for b_2: $\{i_1 : 11, i_2 : 12, i_3 : 8\}$

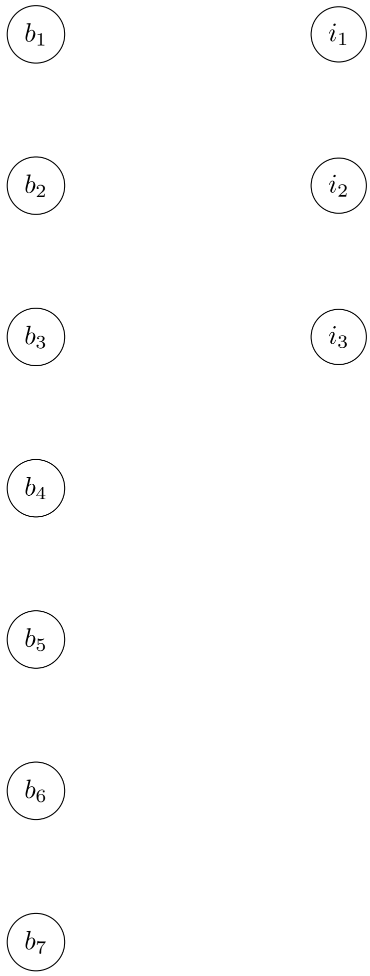
Values for b_3: $\{i_1 : 14, i_2 : 13, i_3 : 18\}$

Values for b_4: $\{i_1 : 3, i_2 : 18, i_3 : 5\}$

Values for b_5: $\{i_1 : 15, i_2 : 9, i_3 : 1\}$

Values for b_6: $\{i_1 : 14, i_2 : 17, i_3 : 9\}$

Values for b_7: $\{i_1 : 14, i_2 : 17, i_3 : 13\}$



And we observe that the prices output by the VCG mechanism and the algorithm of Theorem 8.8 are the same (the matching is also the same because we used the same algorithm to compute the socially optimal state as part of the VCG mechanism)

2. Example 2:

Values for b_1: $\{i_1 : 12, i_2 : 14, i_3 : 16, i_4 : 8, i_5 : 6, i_6 : 17\}$

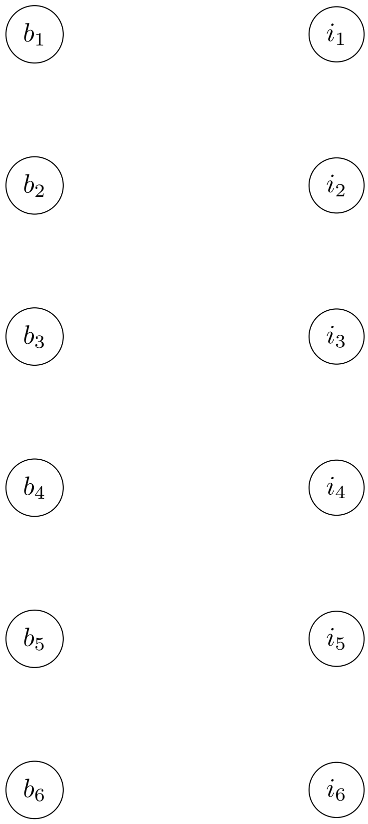
Values for b_2: $\{i_1 : 11, i_2 : 7, i_3 : 9, i_4 : 19, i_5 : 1, i_6 : 11\}$

Values for b_3: $\{i_1 : 18, i_2 : 13, i_3 : 17, i_4 : 17, i_5 : 2, i_6 : 16\}$

Values for b_4: $\{i_1 : 15, i_2 : 0, i_3 : 4, i_4 : 1, i_5 : 15, i_6 : 15\}$

Values for b_5: $\{i_1 : 7, i_2 : 8, i_3 : 5, i_4 : 12, i_5 : 18, i_6 : 13\}$

Values for b_6: $\{i_1 : 7, i_2 : 19, i_3 : 8, i_4 : 12, i_5 : 4, i_6 : 1\}$



And we observe that the prices output by the VCG mechanism and the algorithm of Theorem 8.8 are the same (the matching is also the same because we used the same algorithm to compute the socially optimal state as part of the VCG mechanism)

3. Example 3:

Values for b_1: $\{i_1 : 8, i_2 : 11, i_3 : 0, i_4 : 3, i_5 : 6, i_6 : 7\}$

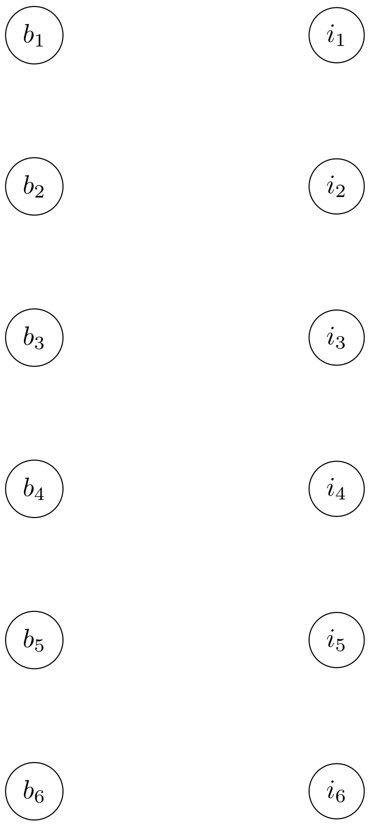
Values for b_2: $\{i_1 : 19, i_2 : 14, i_3 : 15, i_4 : 14, i_5 : 14, i_6 : 16\}$

Values for b_3: $\{i_1 : 17, i_2 : 19, i_3 : 19, i_4 : 13, i_5 : 8, i_6 : 17\}$

Values for b_4: $\{i_1 : 2, i_2 : 15, i_3 : 1, i_4 : 18, i_5 : 11, i_6 : 10\}$

Values for b_5: $\{i_1 : 8, i_2 : 9, i_3 : 7, i_4 : 15, i_5 : 6, i_6 : 10\}$

Values for b_6: $\{i_1 : 12, i_2 : 15, i_3 : 15, i_4 : 8, i_5 : 2, i_6 : 1\}$



And we observe that the prices output by the VCG mechanism and the algorithm of Theorem 8.8 are the same (the matching is also the same because we used the same algorithm to compute the socially optimal state as part of the VCG mechanism)

4. Example 4:

Values for b_1: $\{i_1 : 5, i_2 : 3, i_3 : 0, i_4 : 7, i_5 : 10, i_6 : 5, i_7 : 17, i_8 : 6, i_9 : 18, i_{10} : 8\}$

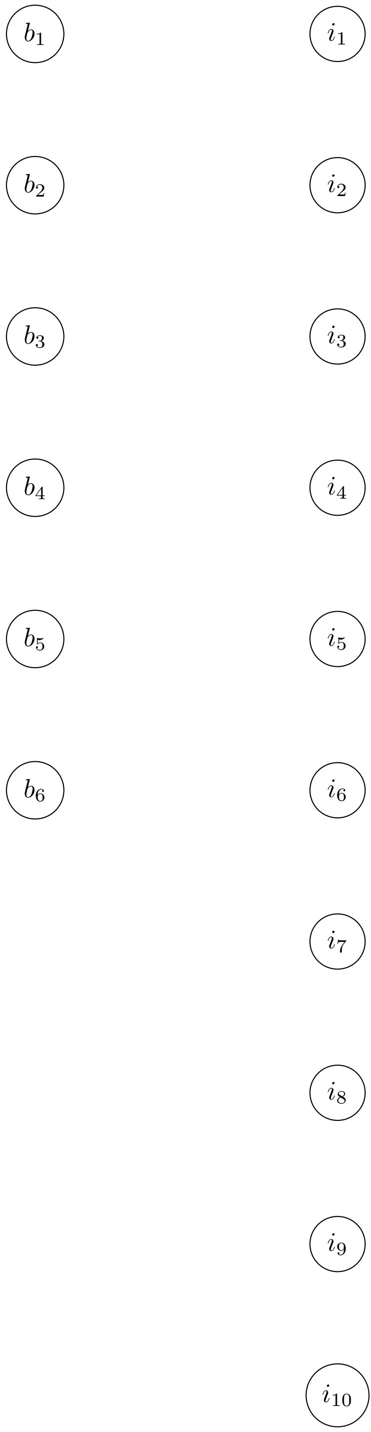
Values for b_2: $\{i_1 : 5, i_2 : 4, i_3 : 6, i_4 : 9, i_5 : 15, i_6 : 9, i_7 : 17, i_8 : 2, i_9 : 10, i_{10} : 14\}$

Values for b_3: $\{i_1 : 10, i_2 : 11, i_3 : 10, i_4 : 6, i_5 : 4, i_6 : 10, i_7 : 16, i_8 : 11, i_9 : 10, i_{10} : 6\}$

Values for b_4: $\{i_1 : 2, i_2 : 19, i_3 : 4, i_4 : 12, i_5 : 5, i_6 : 8, i_7 : 12, i_8 : 0, i_9 : 11, i_{10} : 11\}$

Values for b_5: $\{i_1 : 18, i_2 : 7, i_3 : 15, i_4 : 11, i_5 : 7, i_6 : 4, i_7 : 2, i_8 : 9, i_9 : 9, i_{10} : 8\}$

Values for b_6: $\{i_1 : 5, i_2 : 2, i_3 : 2, i_4 : 5, i_5 : 1, i_6 : 12, i_7 : 13, i_8 : 18, i_9 : 8, i_{10} : 1\}$



And we observe that the prices output by the VCG mechanism and the algorithm of Theorem 8.8 are the same (the matching is also the same because we used the same algorithm to compute the socially optimal state as part of the VCG mechanism)

5. Example 5:

Values for b_1: $\{i_1 : 15, i_2 : 3, i_3 : 1, i_4 : 5\}$

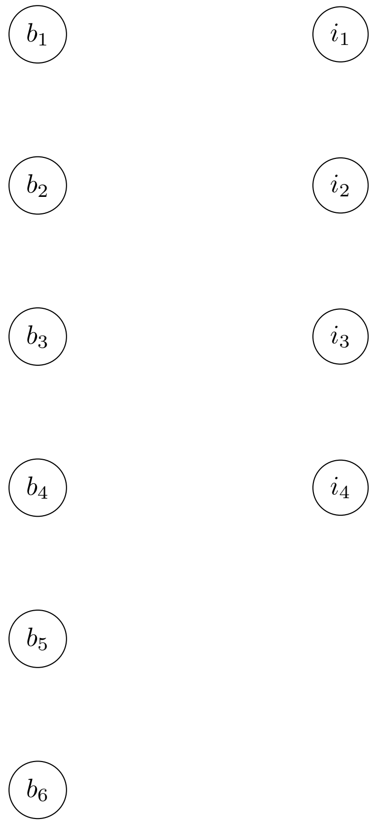
Values for b_2: $\{i_1 : 11, i_2 : 11, i_3 : 16, i_4 : 5\}$

Values for b_3: $\{i_1 : 9, i_2 : 15, i_3 : 13, i_4 : 17\}$

Values for b_4: $\{i_1 : 15, i_2 : 11, i_3 : 10, i_4 : 16\}$

Values for b_5: $\{i_1 : 19, i_2 : 0, i_3 : 12, i_4 : 7\}$

Values for b_6: $\{i_1 : 17, i_2 : 16, i_3 : 13, i_4 : 9\}$



And we observe that the prices output by the VCG mechanism and the algorithm of Theorem 8.8 are the same (the matching is also the same because we used the same algorithm to compute the socially optimal state as part of the VCG mechanism)

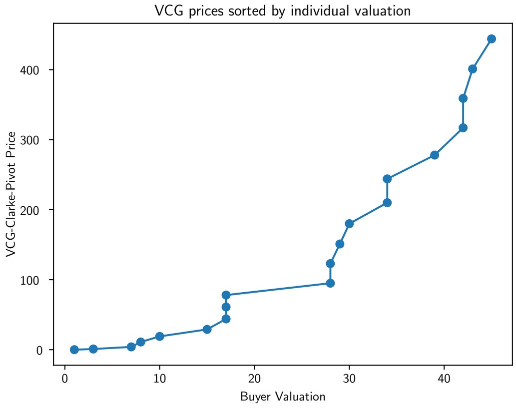
That is, in all examples we analyzed, the prices output by the VCG mechanism and the algorithm of Theorem 8.8 were the same, and the matching was also the same because we used the same algorithm to compute the socially optimal state as part of the VCG mechanism. We analyzed far more examples besides the ones presented here, and the results were consistent across all of them—the prices output by the VCG mechanism and the algorithm of Theorem 8.8 were the same (and the matching was also the same because we used the same algorithm to compute the socially optimal state as part of the VCG mechanism).

3 Bonus Question 2

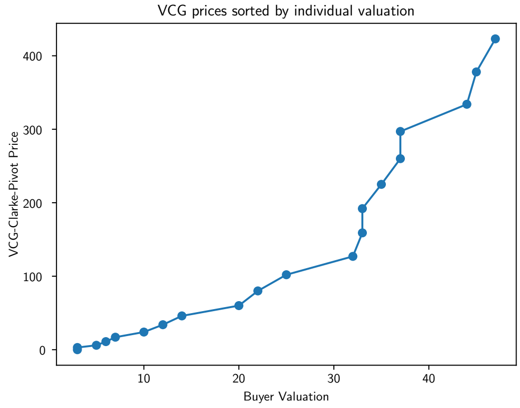
(a) We structure a *markets-for-bundles* context of identical goods as a simple matching market context, where each bundle B_j's value for bidder b_i is the product of the value of b_i for the good and the amount of goods in the bundle, c_i. That is, $v_i(B_j) = c_j \cdot t_i$, where t_i is the value of b_i for the singular good, and c_j is the amount of goods in bundle B_j. We then run the VCG algorithm we implemented in the previous part on a few randomized examples of such a markets-for-bundles context, where there are $n = m = 20$ bundles and bidders, and the values t_i are randomized between 1 and 50, and where $c_j = j$ ($j \in \{1, 2, \ldots, 20\}$).

Figure 1 summarizes the results of the VCG algorithm on 4 such randomized examples of markets-for-bundles contexts (the results remained the same for other examples we ran). The x-axis represents the individual valuation t_i of the bidders for the singular good, and the y-axis represents the VCG price for the bidder—commonly referred to as the *externalities* of the bidder on the market.

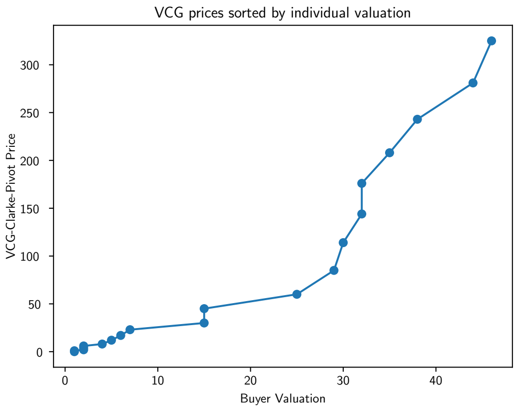
We observe a clear trend, where the VCG prices—i.e. *externalities* of bidders, are increasing with the valuation of the bidders for the singular good (post tie-breaks of valuations), which is expected given the structure of the markets-for-bundles context.



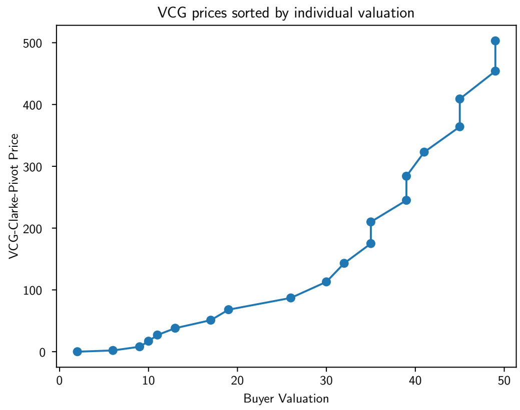
(a) Example 1



(b) Example 2



(c) Example 3



(d) Example 4

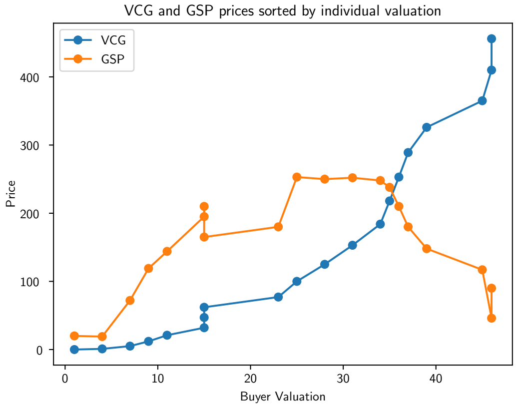
Figure 1: VCG prices for bidders in markets-for-bundles contexts

As we've seen in class, the socially optimal assignment in a *markets-for-bundles* of identical goods, is that the larger bundles are assigned to the higher valuation of a singular good. Thus, as the valuation of the singular good increases, the bidder b_i (w.l.o.g. the bidders are sorted by decreasing valuation of the singular goods) is assigned a larger bundle B_i (w.l.o.g. the bundles are sorted by decreasing sizes c_i). Without said bidder b_i, the bundles assignment shifts for the lower-bidding bidders. That is, $\forall j > i \,.\, b_j$ is changed to be assigned B_{j-1}. This sets the externality to be $\sum_{j=i+1}^{n}(c_{j-1} - c_j) \cdot t_j$ for bidder b_i. In our case, this is $\sum_{j=i+1}^{n} t_j$. In either case, it is easy to see that as the valuation of the singular good increases (i.e., i grows), the externality of the bidder (i.e., of b_i) should increase for similarly distributed bundle sizes and similarly distributed valuations of the singular good (as in our randomization).

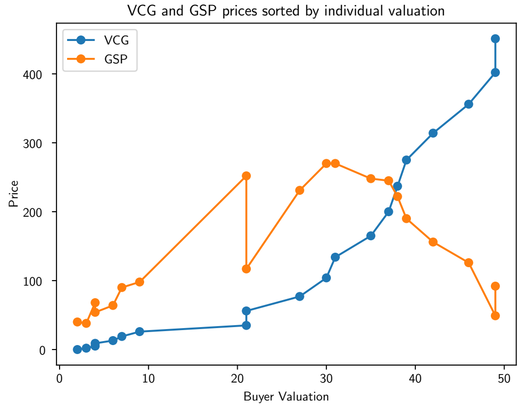
(b) We implemented a GSP pricing mechanism in a *matching-market* context. The following are comparisons of the prices output by the GSP mechanism and the VCG mechanism over different contexts.

Randomized Matching-Market Contexts: We use the randomization scheme described in the previous part to generate randomized matching-market contexts, where there are $n = m = 20$ bidders and goods, and the values t_i are randomized between 1 and 50. We then run the GSP pricing mechanism we implemented on a few such randomized examples of matching-market contexts, and compare the prices output by the GSP mechanism to the VCG prices.

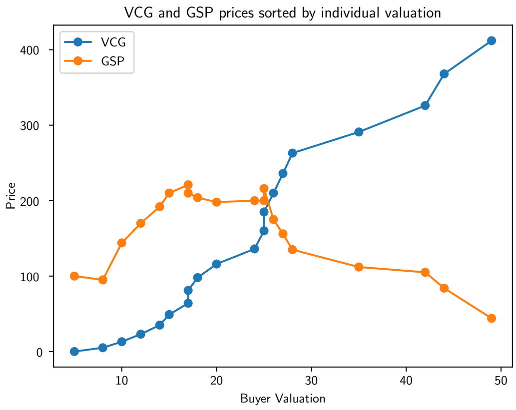
Figure 2 summarizes the results of the GSP pricing mechanism on 4 such randomized examples of matching-market contexts (the results remained the same for other examples we ran), compared to the VCG prices. The x-axis represents the individual valuation t_i of the bidders for the singular good, and the y-axis represents the GSP price for the bidder in orange and the VCG price for the bidder in blue.



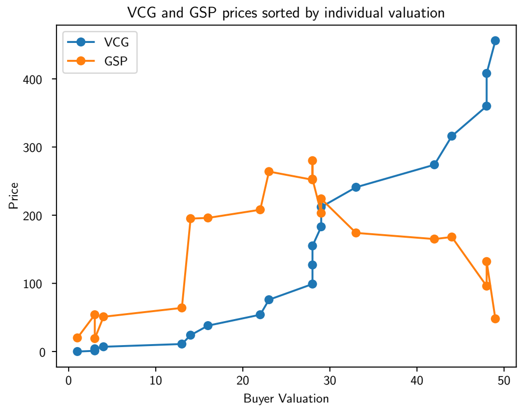
(a) Example 1



(b) Example 2



(c) Example 3



(d) Example 4

Figure 2: GSP and VCG prices for bidders in matching-market contexts

We observe a clear trend, that the GSP prices mimic a *downward-opening* parabola, where the prices are highest for bidders of intermediate valuation for the singular good, and decrease as the valuation of the singular good increases or decreases. This is expected, as the GSP pricing mechanism is designed to maximize the revenue of the seller, and thus the prices are set to be the highest for the bidders that are most likely to win the good, and decrease as the valuation of the singular good increases or decreases.

(c)

(d)

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 $\ell \times \ell$ grid, and there will be two types of participants, riders and drivers.

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

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, i.e., L^1 distance).

1 Question 9

We are tasked with encoding the above as an exchange network. Namely, defining a graph $G = (V, E)$ where the vertices are the riders and rivers and there is en edge between every rider and driver.

Denoting the value for rider R reaching his destination as v_R, and reminded that the value of an edge in an exchange network must be a natural number, we define the values associated with each edge in the graph in the following way:

$$v(\overrightarrow{RD}) = \begin{cases} v_R - c(R, D) & \text{if } v_R \geq c(R, D) \\ 0 & \text{otherwise} \end{cases}$$

2 Question 10

(a)

(b)

3 Question 11

4 Bonus Question 3

(a)

(b)

References