

**These notes may contain errors, and are not intended for use as a primary source.** I have included references at the end of the notes for those who wish detail surrounding skipped steps.

## 1 Logistics

Course webpage is now live. Find it through my homepage. Please sign up on Piazza!

Lectures will pretty soon start requiring background on Linear Programming and LP duality. I included some references on the course webpage in case you need to brush up on this. I will also reference something called the “Maximal-in-Range” mechanisms and claim that they are truthful, without proof. I also included links on the course page, but this is also posed as an exercise from Lecture 1.

**DEFINITION 1 (MAXIMAL-IN-RANGE)** *An algorithm is maximal-in-range if there exists a subset  $\mathcal{T}$  of partitions of items, and for all  $v_1(\cdot), \dots, v_n(\cdot)$ , the algorithm selects a partition in  $\arg \max_{(S_1, \dots, S_n) \in \mathcal{T}} \{\sum_i v_i(S_i)\}$ .*

## 2 An $O(\sqrt{m})$ -approximation for subadditive buyers

Consider an auction with  $n$  buyers,  $v_1(\cdot), \dots, v_n(\cdot)$ , and  $m$  items,  $M = \{1 \dots m\}$ . First, we'll provide an  $O(\sqrt{m})$ -approximation under the assumption that valuation functions are subadditive, due to [?]. Recall that a valuation function is subadditive if  $v(X \cup Y) \leq v(X) + v(Y)$  for all  $X, Y$ . Note that this implies that  $v(X) \leq \sum_{i \in X} v(\{i\})$ .

Consider the following algorithm: query each buyer  $i$  for  $v_i(M)$ , and  $v_i(\{j\})$  for all items  $j$ . Then, do the following:

- Let  $i^* := \arg \max_i \{v_i(M)\}$ .
- Create a bipartite graph with bidders on the left and items on the right. Put an edge between node  $i$  and  $j$  of weight  $v_i(\{j\})$  for all  $i, j$ . Find the max-weight matching in this graph, and let  $j(i)$  denote the item matched to bidder  $i$  in this matching (or null if no item is given to bidder  $i$ ).
- If  $v_{i^*}(M) \geq \sum_i v_i(j(i))$ , then give all items to bidder  $i^*$ . Else, give each bidder  $i$  the item  $j(i)$ .

**THEOREM 1 ([?])**

*The algorithm above guarantees an  $O(\sqrt{m})$ -approximation whenever all valuations are subadditive.*

PROOF: Let  $S_1, \dots, S_n$  denote the welfare-maximizing partition. Further relabel the bidders so that  $|S_1| \geq \dots \geq |S_n|$ . Let  $i'$  denote the largest  $i$  such that  $|S_i| > \sqrt{m}$  (or  $n+1$  if no such  $i$  exists). There are two cases to consider.

First, maybe  $\sum_{i \leq i'} v_i(S_i) \geq OPT/2$ . Note that  $i' \leq \sqrt{m}$  (as each  $i \leq i'$  receives at least  $\sqrt{m}$  items). Therefore, there exists an  $i \leq i'$  such that  $v_i(S_i) \geq OPT/(2\sqrt{m})$ , as there always must be a valuation as least as large as the average. Therefore,  $v_{i^*}(M) \geq v_i(M) \geq v_i(S_i) \geq OPT/(2\sqrt{m})$ , and our algorithm achieves at least  $v_{i^*}(M)$ , so we get our approximation.

Next, maybe  $\sum_{i \leq i'} v_i(S_i) < OPT/2$ . Then  $\sum_{i > i'} v_i(S_i) \geq OPT/2$ . Now we wish to invoke subadditivity: we know that  $v_i(S_i) \leq \sum_{j \in S_i} v_i(\{j\})$ . Therefore, for all  $i$ , there certainly exists a  $j \in S_i$  such that  $v_i(\{j\}) \geq v_i(S_i)/|S_i|$ . For  $i > i'$ , this further means there exists a  $j \in S_i$  such that  $v_i(\{j\}) \geq v_i(S_i)/\sqrt{m}$ . Therefore, as the  $S_i$ s are disjoint, there exists a matching of bidders to items guaranteeing welfare  $\geq \sum_{i > i'} v_i(S_i)/\sqrt{m} \geq OPT/(2\sqrt{m})$ .  $\square$

OBSERVATION 1

*The above algorithm is maximal-in-range. Therefore, it can be turned into a truthful auction.*

PROOF: Observe that the algorithm considers every allocation that is either a matching (gives each bidder at most one item) or awards all items to the same bidder. The best allocation from this collection is selected, therefore it is maximal-in-range.  $\square$

### 3 A (nearly) matching lower bound

Perhaps surprisingly, the algorithm described above is tight (even without considering incentives) when one can only use subexponentially-many value queries, even for the more restrictive class of XOS valuation functions, as all XOS valuation functions are also subadditive. Recall that an additive valuation function is defined so  $v(S) = \sum_{i \in S} v(i)$ , and an XOS valuation function is defined so  $v(T) = \max_{v_i} \{v_i(T)\}$ , where the  $v_i$ s are some set of additive valuation functions. Consider the following construction of valuation functions for  $n = \sqrt{m}$  bidders.

1. First, define  $a_S(T) := |S \cap T|$ . Observe that  $a_S(\cdot)$  is additive, for all  $S$ .
2. Also define  $a(T) := (1 + \delta)|T|/m^{1/2-\delta}$ . Observe that  $a(\cdot)$  is additive.
3. Define  $v(T) := \max_{S, |S| \leq (1+\delta)m^{2\delta}} \{a_S(T)\}$ . Observe that  $v(\cdot)$  is XOS. Note  $v(T) = \min\{|T|, (1 + \delta)m^{2\delta}\}$ , but we defined it to make it clear that it is XOS.
4. Define  $w(T) := \max\{v(T), a(T)\}$ . Observe that  $w(\cdot)$  is XOS.
5. Next, sample uniformly at random partition of  $M$  into  $\sqrt{m}$  sets,  $S_1, \dots, S_n$ .
6. Finally, define  $u_i(T) := \max\{w(T), a_{S_i}(T)\}$ .

Let's first go over the main idea of the construction: If all bidders have (identical) valuations equal to  $w(\cdot)$ , then the welfare of any allocation is upper bounded by  $(1 + \delta)m^{1/2+2\delta} + (1 + \delta)m^{1/2+\delta}$ . To see this, observe  $\sum_i \max\{v(S_i), a(S_i)\} \leq \sum_i v(S_i) + a(S_i) = \sum_i v(S_i) + \sum_i a(S_i)$ . Note in the right sum each item contributes  $(1 + \delta)/m^{1/2-\delta}$ , so the total is bounded by  $m \cdot (1 + \delta)/m^{1/2-\delta}$ . In the left sum, each term is bounded by  $(1 + \delta)m^{2\delta}$ , so even if all  $n = \sqrt{m}$  bidders get this much value, it contributes at most  $(1 + \delta)m^{1/2+2\delta}$ . The point here is that if all bidders have valuation  $w(\cdot)$ , then the welfare is not very good.

On the other hand, say the valuations are instead  $u_1, \dots, u_n$ . Then the partition  $S_1, \dots, S_n$  has  $\sum_i u_i(S_i) = m$ , and the welfare is very good. So the plan is just to show that it requires exponentially many value queries to determine whether a given valuation is either  $w(\cdot)$  or  $u_i(\cdot)$ . This means no polytime approximation algorithm could do better when run with  $u_i(\cdot)$ s than when run on  $w(\cdot)$ s, but as the gap between the optimal solutions is  $O(\sqrt{m})$ , no algorithm can get better than an  $O(\sqrt{m})$  approximation.

PROPOSITION 1 ([?])

For any  $\delta > 0$ , any (randomized) algorithm that, when randomly given the valuation function  $w(\cdot)$  or  $u_i(\cdot)$ , correctly identifies the function with probability at least  $1/2 + Ce^{-\Omega(m^{2\delta})}$  (over the randomness of drawing  $T_i$ , and any randomness in the algorithm) requires at least  $C$  value queries.

In particular,  $C = e^{m^\delta}$  and let  $\delta$  be a constant  $> 0$  to conclude that exponentially many value queries are necessary to beat a  $m^{1/2-\delta}$  approximation.

To prove the proposition, we will need a slight extension of the Chernoff bound from independent to negatively correlated random variables, which I will not prove here.

THEOREM 2

Generalized Chernoff Bound : Let  $X = \sum_{i=0}^n X_i$ , where  $X_i$  are random variables  $\in \{0, 1\}$ . Let  $S = \{X_i : i \in [m]\}$ . If  $\forall Z \subseteq S / \{X_i\}$ ,  $Pr[X_i = 1 | X_j = 1 \forall X_j \in Z] \leq Pr[X_i = 1]$  (note equality holds for independent RVs), then  $Pr[X \geq (1 + \delta)E[X]] \leq e^{-\delta E[X]/3}$  for  $\delta \geq 1$ .

PROOF: We'll prove this by showing that, with probability at least  $1 - Ce^{-m^{2\delta}}$ , any set of  $C$  value queries return exactly the same value when made on  $w(\cdot)$  or  $u_i(\cdot)$ . We will show no matter what (randomized) procedure we use to (adaptively) pick our queries, given that they are answered according to  $w(\cdot)$ , with probability at least  $1 - Ce^{-m^{2\delta}}$ ,  $S_i$  is chosen so that  $u_i(\cdot)$  answers them identically.

To do this, we'll simply show that for any set  $T$ , over the randomness in selecting  $S_i$ , the probability that  $u_i(T) = w(T)$  is high, and then take a union bound over all  $C$  queries. Let's start with a few observations. First observe that as  $u_i(T) = \max\{w(T), |T \cap S_i|\}$ ,  $u_i(T) \neq w(T)$  only when  $|T \cap S_i| > w(T)$ . Now observe for all  $T$ , the probability a given element  $i \in T$  is also in  $S_i$  is exactly  $1/\sqrt{m}$ , so  $E[|T \cap S_i|] = |T|/\sqrt{m}$ .

Let's first consider sets  $T$  of size at most  $m^{1/2+\delta}$ . In this range,  $v(T) \geq a(T)$ , as for  $|T| = m^{1/2+\delta}$ ,  $a(T) = v(T)$ , but  $v(T)$  grows more quickly for small  $T$ . So in this case  $w(T) = \min\{|T|, (1 + \delta)m^{2\delta}\}$ . In order to have  $u_i(T) \neq w(T)$ , we would need to have  $|T \cap S_i| > \min\{|T|, (1 + \delta)m^{2\delta}\}$ . For  $|T| \leq (1 + \delta)m^{2\delta}$  this is clearly impossible. For  $|T| > (1 + \delta)m^{2\delta}$  observe that every element of  $T$  is in  $S_i$  with probability  $1/\sqrt{m}$ , independently. Therefore, the expected number of elements in  $T \cap S_i$  is  $|T|/\sqrt{m} \leq m^\delta$ .

As  $\Pr[i \in S_i | j_1 \in S_i \dots j_k \in S_i, j_z \neq i, \forall z] = (\sqrt{m} - k)/(m - k) \leq 1/\sqrt{m}$ , the condition for our generalized Chernoff holds. If we let  $\delta' = m^{2\delta}/\mu - 1$ , we conclude  $\Pr[|T \cap S_i| > (1 + \delta')\mu] \leq \Pr[|T \cap S_i| > (1 + \delta)m^{2\delta}] \leq e^{1/3*(-m^{-2\delta}-\mu)} = e^{-\Omega(m^{2\delta})}$ .

Now consider any set of size at least  $m^{1/2+\delta}$ . Then  $w(T) = (1 + \delta)|T|/m^{1/2-\delta}$ . In order to have  $u_i(T) \neq w(T)$ , we must have  $|T \cap S_i| > (1 + \delta)|T|/m^{1/2-\delta}$ . Again, note that the expected size of  $|T \cap S_i|$  is  $|T|/\sqrt{m}$ , and again we would need deviation by a multiplicative factor of  $m^\delta$ . By the Chernoff bound, the probability that this occurs is at most  $e^{-m^{1/2+2\delta}} = e^{-\Omega(m^{2\delta})}$ .

So for any set, the probability over the randomness of  $S_i$  that  $w(T) \neq u_i(T)$  is at most  $e^{-\Omega(m^{2\delta})}$ . Taking a union bound over the sets queried (conditioned on getting answers consistent with  $w(\cdot)$ ) gives the proposition.  $\square$

## 4 Gross Substitutes: Connecting Value and Demand Queries

Next, we'll transition to Gross Substitutes, and built towards something called a *Walrasian Equilibria* (defined next lecture) and an exact algorithm. This lecture, we'll start the building blocks. All of the presentation below is taken from a survey of Paes Leme [?]. Recall the definition of gross substitutes:

**DEFINITION 2 (GROSS SUBSTITUTES)** *A valuation function  $v(\cdot)$  is gross substitutes if for all price vectors  $p, q$  with  $p_j \geq q_j$  for all items  $j$ , if  $S$  is demanded by  $v(\cdot)$  on  $q$ , and  $A$  is the subset of  $S$  satisfying  $p_j = q_j$ , then there exists a set  $T \supseteq A$  that is demanded on  $p$ .*

It turns out that this is equivalent to the following two conditions:

**DEFINITION 3 (SINGLE IMPROVEMENT)** *A valuation function satisfies the single-improvement property if for all price vectors  $p$ ,  $S$  is demanded by  $v(\cdot)$  on  $p$  if and only if  $S$  is weakly preferred to all  $T$  that can be obtained by adding, removing, or exchanging one element. That is, if we let  $u(S) = v(S) - \sum_{j \in S} p_j$ :*

$$v(S) \geq \max_{i \notin S, j \in S} \{\max\{u(S \cup \{j\} \setminus \{i\}), u(S \setminus \{i\}), u(S \cup \{j\})\}\}$$

**DEFINITION 4 (MATROIDAL MAP)** *A valuation function is a matroidal map if for all price vectors  $v$ , the demand of  $v(\cdot)$  can be found via the following greedy algorithm: Initialize  $S = \emptyset$ . Then iteratively let  $j := \arg \max_i \{v(S \cup \{i\}) - v(S) - p_i\}$ . If  $v(S \cup \{j\}) - v(S) > p_j$ , update  $S$  to  $S \cup \{j\}$ . Else, return  $S$ .*

### THEOREM 3

*The three conditions GS, SI, MM are equivalent.*

We won't prove these in this class since it's a bit tangential. But clearly presenting any of these proofs would be an appropriate course project. But we will conclude the following useful corollary:

### COROLLARY 2

*A demand query for any GS valuation can be implemented using polynomially-many value queries.*

This isn't obviously useful yet, but it will be in the next lecture when we design algorithms for GS valuations using demand queries. To build towards this algorithm, we'll introduce the concept of a *Walrasian Equilibrium*. This is extremely useful in its own right, but we won't focus on it as much in this course (but it again can be a valid source to present for the class project).

**DEFINITION 5 (WALRASIAN EQUILIBRIUM)** *For  $n$  bidders with valuation functions  $v_1(\cdot), \dots, v_n(\cdot)$ , a Walrasian Equilibrium is a price vector  $p$  such that there exists a partition of  $M$  into  $S_1, \dots, S_n$  such that bidder  $i$  demands  $S_i$  on  $p$  for all  $i$ . Note that this must be a partition: every item is demanded exactly once.*

**THEOREM 4 ([?])**

*When all  $v_i(\cdot)$  are GS, a Walrasian Equilibrium exists.*

**PROOF:** Consider the following algorithm:

1. Initialize  $p_j = 0$  for all items  $j$ . Initialize  $\delta > 0$  to some constant. Initialize  $S_1 = M$ ,  $S_i = \emptyset$  for all  $i > 1$ . Initialize the current player  $i$  to 2.
2. Repeat the following:
  - (a) Ask the current player  $i$  their demand at the price vector  $p'$ , where  $p'_j = p_j$  for all  $j \in S_i$ , and  $p'_j = p_j + \delta$  for all  $j \notin S_i$ . Tie-break in favor of sets that contain  $S_i$  (if one exists). Call this set  $T$ . Update  $S_i := T$ . For all  $i' \neq i$ , update  $S_{i'} := S_{i'} \setminus T$ .
  - (b) Update  $i := i + 1 \pmod{n}$ .
  - (c) Update  $p_j := p'_j$  for all  $j \in T$ .

We first wish to claim that at all times during the algorithm, all items are allocated. To see this, first observe that initially all items are allocated (to bidder 1). So we just need to make sure that no item goes unallocated. Observe that the only time an item is removed from  $S_i$  is either when it is stolen by another player (in which case it remains allocated), or thrown away by player  $i$  in favor of a better set that doesn't contain  $S_i$ . So we just need to make sure that the latter never happens (i.e. there is always a set player  $i$  wants that contains  $S_i$ ).

To see this, observe that the price of item  $j$  only increases when it is stolen by another bidder. So let's consider two consecutive steps (i.e. steps  $n$  rounds apart) where bidder  $i$  updates their set  $S_i$ . Let  $p$  denote the price vector the first time they select  $T_1$  to be the new  $S_i$ , and  $p'$  the price vector the second time. By the reasoning in the first sentence of this paragraph, if  $S_i$  denotes bidder  $i$ 's allocation at the start of the second step, we must have  $p'_j = p_j$  for all  $j \in S_i \subseteq T_1$ . Also, we have  $p'_j \geq p_j$  for all  $j$  since the prices only increase. Therefore, because  $T_1$  was demanded at prices  $p$ , and  $p'$  only increases prices from  $p$ , we may invoke the definition of GS to claim that there exists a set  $T_2 \supseteq S_i$  that bidder  $i$  demands at  $p'$ . As such, no bidder will ever throw away items, and every item remains allocated through the entire procedure.

Now, we want to claim that the procedure always terminates, and terminates in an almost-Walrasian Equilibrium for any  $\delta$ . First, observe that if the procedure makes a full lap of all  $n$  bidders without changing the allocation, then it will never change again.

Next, observe that every time the allocation changes, at least one price increases by  $\delta$ . As  $v_i(M) < \infty$  for all  $i$ , there can only be finitely many price increases (as each price increase also corresponds to a bidder demanding that item, and no one will demand an item priced above  $\max_i \{v_i(M)\}$ ). So the algorithm terminates.

It is also clear that when the algorithm terminates, we have an almost-Walrasian Equilibrium: at the price vector  $p'$  (which increases prices not in  $S_i$  by  $\delta$ ), bidder  $i$  demands the set  $S_i$ . But  $p'$  is different for each bidder  $i$ , so there's no sense in which we can say that  $p'$  is a Walrasian Equilibrium. However, if we consider the price vector  $p$ , we know that:

$$v_i(S_i) - \sum_{j \in T} p_j \geq v_i(S_i) - \sum_{j \in T} p'_j \geq v_i(T) - \sum_{j \in T} p'_j \geq v_i(T) - \sum_{j \in T} p_j - m\delta,$$

for any set  $T$ . Therefore,  $p$  is in some sense an  $m\delta$ -approximate Walrasian Equilibrium: if we are allowed to violate the preferences by up to  $m\delta$ , then we can pick a partition of the items where every bidder demands their set (up to  $m\delta$ ) at price vector  $p$ .

Now consider taking an infinite sequence of  $\delta$  approaching 0 such as  $\delta = 1, 1/2, 1/3, \dots$ . As this is an infinite sequence, there is some partition  $(S_1, \dots, S_n)$  that repeats in the sequence infinitely often. Now restrict attention to price vectors that are output along with this partition: all such vectors must lie in  $[0, \max_i \{v_i(M)\}]^m$ . So we have an infinite sequence of points in a compact set, and therefore this sequence must have a limit point  $p$ . We now claim that  $p$  must be a Walrasian equilibrium.

Indeed, it is now the case that for *all*  $\varepsilon > 0$ , there exists an infinitely long subsequence of  $1, 1/2, \dots$  such that the price vectors output when we run the algorithm on all  $\delta$  in this sequence are within  $\varepsilon$  in (say)  $\ell_1$  distance from  $p$ . In particular, this means that for all  $\varepsilon > 0$ , there exists a  $\delta < \varepsilon$  and a price vector  $q$  such that  $|p - q|_1 \leq \varepsilon$  and  $q$  is an  $m\varepsilon$ -Walrasian equilibrium. This implies that the following inequalities hold for all  $\varepsilon > 0$ :

$$v_i(S_i) - \sum_{j \in S_i} p_j \geq v_i(S_i) - \sum_{j \in T} q_j - m\varepsilon \geq v_i(T) - \sum_{j \in T} q_j - 2m\varepsilon \geq v_i(T) - \sum_{j \in T} p_j - 3m\varepsilon.$$

Or in other words,  $v_i(S_i) - \sum_{j \in S_i} p_j \geq v_i(T) - \sum_{j \in T} p_j$ , and  $p$  is a Walrasian Equilibrium.

□

## 5 Possible Project Topics

1. Present any of the equivalences between properties GS, SI, MM.
2. Present other results in [?] not covered in the course.
3. Present the  $1 - 1/e$  approximation using value queries for submodular functions [?], perhaps along with the matching lower bound [?].