# CMSC 858F: Algorithmic Game Theory
# Fall 2010
# Market Clearing with Applications

**Instructor:** Mohammad T. Hajiaghayi
**Scribe:** Rajesh Chitnis

September 15, 2010

## 1 Overview

We will look at MARKET CLEARING or MARKET EQUILIBRIUM prices. We will consider two models with linear utilities : FISHER model and ARROW-DEBREU model. See [1] and chapter 5 of [2] for more details. Finally we see an application of the FISHER model to wireless networks.
I have also used as reference the scribe notes for this same lecture when this course was taught by Prof. Hajiaghayi at Rutgers in Spring 2009.

## 2 FISHER model with linear utilities

We have the following setting :

1. A set $B$ of $m$ BUYERS with budgets $B_1, B_2, \ldots, B_m$
2. A set $G$ of $n$ GOODS for sale with quantities $q_1, q_2, q_n$
3. Utility of buyer $i$ for good $j$ is $u_{ij}$. $\{u_{ij} \in \mathbb{Q}\}$
4. Amount of good $j$ bought by buyer $i$ is $x_{ij}$. $\{x_{ij} \in \mathbb{Q}\}$
5. Each buyer has linear utility, that is $\forall\ i \in B$

$$u_i = \sum_{j \in G} u_{ij} x_{ij}$$

# 3 Formal definition of MARKET EQULIBRIUM

A market equilibrium (or market clearing)is a price vector $P = (p_1, p_2, \ldots, p_n)$ such that

1. Utility of each buyer $u_i$ is maximized within his budget. That is, $\forall\ i \in B$, we have $\sum_{j \in G} p_j x_{ij} \leq B_i$
2. Demand and suppy for each good $j$ are equal. That is, $\forall\ j \in G$, we have $\sum_{i \in B} x_{ij} = q_j$

Note that all the budgets are also spent in this situation because if some buyer $i$ had leftover budget then he could buy some good $j$ and then for that good $j$ we will have $\sum_{i \in B} x_{ij} \neq q_j$

Further we can also assume that $q_j = 1\ \forall j \in G$ as all equations are linear and we can scale by appropriate factors.

**Theorem 1** *Under the mild assumption that each good has atleast one potential buyer (i.e.* $\forall\ j \in G\ \exists\ i \in B$ *such that* $u_{ij} > 0$*), market clearing/equilibrium price always exists.*

# 4 ARROW-DEBREU model with linear utilities

It is also called as WALRASIAN model or EXCHANGE model.
We have the following setting :

1. A set B of $m$ AGENTS
2. A set $G$ of $n$ GOODS for sale
3. Agent $i$ comes to the market with an initial endowment of goods given by $e_i = (e_{i1}, e_{i2}, \ldots, e_{in})$
4. Utility of buyer $i$ for good $j$ is $u_{ij}$. $\{u_{ij} \in \mathbb{Q}\}$
5. Amount of good $j$ bought by buyer $i$ is $x_{ij}$ $\{x_{ij} \in \mathbb{Q}\}$
6. Each buyer has linear utility, that is $\forall\ i \in B$

$$u_i = \sum_{j \in G} u_{ij} x_{ij}$$

Again, we can assume as in the FISHER model that amount of each good present is 1 as all equations are linear and we can scale appropriately. That is, $\forall\ j \in G$ we have $\sum_{i \in B} e_{ij} = 1$

Here each agent can buy and sell unlike in the FISHER model. We want a price vector $P = (p_1, p_2, \ldots, p_n)$ so that if each agent sells his initial endowment at these prices and buys his optimal bundle, then the market clears. [no deficiency or surplus of any good].

# 5 FISHER model is special case of ARROW-DEBREU model

In the following matrix, the rows correspond to agents and the columns correspond to goods.

$$\text{Endowment Matrix} = \begin{pmatrix} 0 & 0 & \cdots & 0 & B_1 \\ 0 & 0 & \cdots & 0 & B_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & B_m \\ q_1 & q_2 & \cdots & q_n & 0 \end{pmatrix}$$

A given instance of FISHER model with linear utilities, $m$ buyers and $n$ goods reduces to a ARROW-DEBREU model with linear utilities, $(m+1)$ buyers and $(n+1)$ goods as follows :

1. We add MONEY as the $(n+1)^{th}$ good
2. We add ONLY-SELLER as the $(m+1)^{th}$ agent in addition to the initial buyers
3. For each of the first $m$ agents, their only initial endowment is money and that too equal to their budget. They also have zero utility for money and positive utility for all the other goods.
4. The $(m+1)^{th}$ agent has as initial endowment quantity $q_j$ of good $j$ for all $j \in G$. He has positive utility for money and zero utility for all other goods.

# 6 Proof of Theorem 1

We assume that the $u_{ij}$ and the $B_i$ variables are rational. As all equations are linear, with appropriate scaling, we can assume that they are (large) integers.

We formulate the Fisher model with a convex program and then solve it using the Eisenberg-Gale approach.

## 6.1 Eisenberg-Gale Convex program

We can represent the problem as the following convex program : (why this represents the problem accurately is not clear and that is actually the hard part of the problem)

$$max \quad \prod_{i \in B} u_i^{B_i}$$

where

- $u_i = \sum_{j \in G} u_{ij} x_{ij} \qquad \forall \ i \in B$
- $\sum_{i \in B} x_{ij} \le 1 \qquad \forall \ j \in G$
- $x_{ij} \ge 0 \qquad \forall \ i \in B \ , \ j \in G$

We can instead solve the equivalent problem of maximizing the logarithm of the above function. So our convex program becomes :

$$max \quad \sum_{i \in B} B_i.log(u_i)$$

## 6.2 General result about convex programs

Let $f$ be a given convex function. Then we can solve in polynomial time the problem of minimizing $f$ over a convex body.

## 6.3 Formulating our problem in convex programming setting

The function $f(x) = log(x)$ is concave as its double derivative is negative. Hence, $f(x) = -log(x)$ is convex. Also sum of convex functions is convex. Thus we have the following :

$$max \quad \sum_{i \in B} B_i.log(u_i) \quad \Rightarrow \quad min \quad \sum_{i \in B} -B_i.log(u_i)$$

where the right hand side above is in the standard form of convex programming. We solve it in polynomial time using the known methods for convex programming.

## 6.4 Karush-Kuhn-Tucker (KKT) conditions for Convex Programs

The KKT conditions for convex programs are similar to the primal-dual conditions for linear programming. Corresponding to the constraint $\sum_{i \in B} x_{ij} \le 1$, we have a variable $p_j$. Our convex program has the following conditions :

$$\forall \ j \in G, \ p_j \ge 0 \tag{1}$$

$$\forall\, j \in G,\ p_j > 0 \Rightarrow \sum_{i \in B} x_{ij} = 1 \tag{2}$$

$$\forall\, i \in B,\ \forall\, j \in G,\quad \frac{u_{ij}}{p_j} \leq \frac{\sum_{j \in G} u_{ij} x_{ij}}{B_i} \tag{3}$$

$$\forall\, i \in B,\ \forall\, j \in G,\quad x_{ij} > 0 \Rightarrow \frac{u_{ij}}{p_j} = \frac{\sum_{j \in G} u_{ij} x_{ij}}{B_i} \tag{4}$$

Note that $\frac{u_{ij}}{p_j}$ is the utility for buyer $i$ per unit price of item $j$ and $\frac{\sum_{j \in G} u_{ij} x_{ij}}{B_i}$ is the utility per budget for buyer $i$.

## 6.5 Proof that the vector given by KKT conditions satisfies market clearing and vice versa

Suppose we have market equilibrium. Then we claim that the price vector $P = (p_1, p_2, \ldots, p_n)$ satisfy the above 4 equations. Equations (1) and (2) are obvious. Equation (3) just states that utility for buyer $i$ per unit price of good $j$ must be less than or equal to his utility per unit budget otherwise he would have bought more of good $j$. Equation (4) states that buyer $i$ purchasing good $j$ implies that his utility per unit price of good $j$ must be maximum possible i.e. equal to his utility per unit budget.

Now, suppose we have a price vector $P = (p_1, p_2, \ldots, p_n)$ which satisfies above four conditions. We claim that this price vector gives us a market equilibrium. We had an assumption that every good has a potential buyer. That is, $\forall\, j \in G,\ \exists\, i \in B$ such that $u_{ij} > 0$. By equation (3), $\forall\, j \in G,\ \exists\, i \in B$ such that $p_j \geq \frac{B_i u_{ij}}{\sum_{j \in G} u_{ij} x_{ij}} > 0$. Thus, by equation (2), $\forall\, j \in G$ we have that $\sum_{i \in B} x_{ij} = 1$ i.e. all of good $j$ is sold. Now we claim that the following equation holds true for all $i \in B$ :

$$\frac{B_i u_{ij}}{\sum_{j \in G} u_{ij} x_{ij}} x_{ij} = p_j x_{ij} \tag{5}$$

If $x_{ij} = 0$, then the equation hold trivially otherwise we are through by equation (4). Summing equation (5) over all $j \in G$, we have $B_i = \sum_{j \in G} p_{ij} x_{ij}$ i.e. all budgets are used up. Thus, there is market clearing.

## 6.6 Summary

We formulate our problem as a convex program which we solve in polynomial time using known methods. We use KKT conditions to get another convex program with same solution and we show that this satisfies market clearing conditions thus implying that our initial convex program also finds the solution to market clearing.

In the linear case of FISHER model with each good having a potential buyer, we have

1. Market Clearing always exists
2. The set of equilibrium allocations is convex
3. Equlibrium utilities and prices are unique (The assignments may not be unique though)

Also, see [4] and [3] for more details and new approaches.

# 7 Application to distributed load balancing in wireless networks

See [5] for the journal paper. The slides shown in class are available at the course webpage

## 7.1 Overview of the problem

We have wireless devices (referred to as clients or users) who demand connection to access points (AP). The AP's have limited capacities and variable transmission power i.e. they are capable of operating at various levels. Devices would like to connect to the AP that offers them the strongest signal. We want to design a way to efficiently assign clients to AP's so that there is no over-loading of AP's and maximum number of clients are assigned thus satisfying maximum demand.
The simple heuristic of a client connecting to the AP with lightest load among those with reasonable signals fails because this will require co-operation from clients and AP's which is not possible in practise.
The proposed cell breathing heuristic employs ideas from game theory. We always want a client to associate himself with the AP providing strongest signal. The crucial point is that here an AP changes its communication radius depending on its load.

## 7.2 Relation to Market Clearing

Our situation is analagous to the FISHER model with linear utilities as demonstrated by the table below

| **Market Equilibrium** | **Load Balancing** |
|---|---|
| Seller | AP |
| Buyer | Wireless Client |
| Goods | Network Connectivity |
| Supply | Capacity of AP |
| Price | Power of AP |
| Utility | Strength of Signal received |
| Market Clearance | Either all clients are served or all AP's are saturated |

However, the analogousness is only inspirational. There is no known reduction from FISHER model to our problem but we try and use ideas from known algorithms for FISHER model. In fact, there are some key differences between our model and the FISHER setting as we observe below

| **Market Equilibrium** | **Load Balancing** |
|---|---|
| Demand is dependent on price | Demand is independent of price (power) |
| Demand is splittable | Demand is not splittable |
| Can be solved in poly time | Continous power levels case can be solved in poly time but discrete case is APX-hard |
| Equilibrium clears both sides and can be computed | Equilibrium clears either client side or AP side and may not exist |

## 7.3 Known approaches

Three approaches for Market Equilibrium :

1. Convex Programming based : Eisenberg and Gale (1957)
2. Primal-Dual based : Devanur, Papadimitriou,Saberi and Vazirani (2004)
3. Auction based : Garg and Kapoor (2003)

Three approaches for Load Balancing :

1. Linear Programming : Minimum weight complete matching
2. Primal-Dual : Uses properties of bipartite matching
3. Auction : Useful in dynamically changing situations

We shall see the linear programming approach in the following subsection

## 7.4 Linear Programming based approach

We assume that if an AP $j$ is transmitting at power level $P_j$ then a client $i$ at distance $d_{ij}$ receives signal whose strength is given by

$$P_{ij} = \frac{aP_j}{(d_{ij})^c}$$

where $a$ and $c$ are constants capturing various models of power attenuation.

We create a complete bipartite graph with clients on one side and AP's on other. Conceptually, on the AP side, each AP is repeated as many times as its capacity. We put weight between client $i$ and AP $j$ as follows

$$w_{ij} = c.\ln(d_{ij}) - \ln(a) = -\ln(\frac{P_{ij}}{P_j})$$

Now, find the minimum weight complete matching. Power assignments are the dual variables.

**Theorem 2** *Minimum weight matching is supported by a power assignment to the AP's.*

We have the following two cases for the primal program :

1. **Solution can satisfy all clients**
2. **Solution can saturate all AP's**

We consider the case of satisfying all clients i.e. the complete matching covers all clients. Let $C$ be the client set and let $A$ be the set of access points. The linear program is as follows

$$\min \sum_{i \in C, j \in A} w_{ij} x_{ij}$$

subject to

- $\forall\ i \in C, \quad \sum_{j \in A} x_{ij} = 1$
- $\forall\ j \in A, \quad \sum_{i \in C} x_{ij} \leq C_j$
- $\forall\ j \in A,\ i \in C, \quad x_{ij} \geq 0$

The dual variables corresponding to the first 2 constraints are $\lambda_i$ and $\pi_j$ respectively. We write the dual program which goes as follows

$$max \quad \sum_{i \in C} \lambda_i + \sum_{j \in A} C_j \pi_j$$

subject to

- $\forall\ j \in A, \quad \pi_j \geq 0$
- $\forall\ j \in A,\ i \in C, \quad \lambda_i + \pi_j \leq w_{ij}$

We choose $P_j = e^{\pi_j}$. We will now use the complementary slackness conditions to show that the minimum weight complete matching is supported by the above chosen power levels. By dual feasibility, we have

$$-\lambda_i \geq \pi_j - w_{ij} = ln(P_j) - c.ln(d_{ij}) + ln(a) = ln\Big(\frac{a.P_j}{(d_{ij})^c}\Big) \tag{6}$$

By complementary slackness we have

$$x_{ij} = 1 \Rightarrow -\lambda_i = ln\Big(\frac{a.P_j}{(d_{ij})^c}\Big) \tag{7}$$

Equations (6) and (7) imply that each client is connected to the AP from which it receives strongest signal strength.

Now we consider the case when the complete matching saturates all the AP's. The linear program is as follows :

$$min \quad \sum_{i \in C, j \in A} w_{ij} x_{ij}$$

subject to

- $\forall\ i \in C, \quad \sum_{j \in A} x_{ij} \leq 1$
- $\forall\ j \in A, \quad \sum_{i \in C} x_{ij} = C_j$
- $\forall\ j \in A,\ i \in C, \quad x_{ij} \geq 0$

The rest of the proof is similar to the previous case.

## 7.5 The case of Unsplittable Demand

$$min \quad \sum_{i \in C, j \in A} w_{ij} x_{ij}$$

subject to

- $\forall\ i \in C, \quad \sum_{j \in A} x_{ij} = 1$
- $\forall\ j \in A, \quad \sum_{i \in C} D_i x_{ij} \leq C_j$
- $\forall\ j \in A,\ i \in C, \quad x_{ij} \geq 0$

The integer program is APX-hard in general due to knapsack. However under the realistic assumption that number of clients is much larger than the number of AP's , we can obtain a nice approximation heuristic.

**Theorem 3** *Number of $x_{ij}$ which are integral is atleast (number of clients) - (number of AP's)*

So most clients are served unsplittably. We do not serve those clients which were served splittably. This algorithm is almost optimal.

### 7.6 Discrete Power Levels

In real life, AP's only have fixed number of discrete power levels. In such case , equilibrium may not exist. In fact, it is NP-hard to test whether equilibrium in such cases. However, if every client has a deterministic tie-breaking rule then we can compute the equilibrium if it exists under the given tie-breaking rule.

We first start with maximum power levels for each AP. Now, for every overloaded AP, we reduce its power by one notch. If an equilibrium exists then it can be computed in time $O(mk)$ where $m$ is number of AP's and $k$ is the number of power levels. The proof follows by induction. Let $P_j$ be equilibrium power level for $j^{th}$ AP. We can show inductively that when AP $j$ reaches power level $P_j$ then it will never get overloaded again (we use the deterministic tie-breaking rule here).

### 7.7 Conclusion

The theory of market equilibrium is a good way of synchronizing independent enitities to do distributed load balancing

## 8 Homework Assignment

Suppose you want to sell your bike, car, house, etc., and you want to sell online (e.g., craigslist). What would your strategy be? What is the difference in putting your listing on craigslist vs. putting it on the classifieds of a magazine? Will you search for other bikes? What are the assumptions? How much would you sell for? Do you want to put it below/above the average price, and how much above or below?

# References

[1] Hajiaghayi, Mohammad T. *Algorithmic Game Theory: Lecture 3– Handwritten notes*. 15 September 2010. University of Maryland.

[2] Nisan, Noam et al. *Algorithmic Game Theory*. New York: Cambridge University Press, 2007.

[3] Kamal Jain, *A Polynomial Time Algorithm for Computing the Arrow-Debreu Market Equilibrium for Linear Utilities*, FOCS 2004

[4] Nikhil Devanur, Christos H. Papadimitriou, Amin Saberi and Vijay V. Vazirani, *Market equilibrium via a primal-dual-type algorithm*, J. ACM , 55(5), 2008

[5] P. Bahl, M. T. Hajiaghayi, K. Jain, V. S. Mirrokni, L. Qiu, and A. Saberi. *Cell Breathing in Wireless LANs: Algorithms and Evaluation*, IEEE Transactions on Mobile Computing, 6(2) : 164-178, 2007.