Dynamic Market Equilibrium and Convex Optimization

July 25, 2010

This is a note I took from ASI 2010: Theory and Applications on Algorithmic Game Theory on dynamic market equilibrium and convex optimization.

1 The Arrow-Debreu Competitive Economy Equilibrium Problem

1.1 Background

Walras-Arrow-Debreu competitive market equilibrium:

- Each of a population of n traders has an initial endowment of *divisible goods* and a non-decreasing *utility function* on all m types of good.
- They trad/exchange according to *market prices*, that is, every one/exchange according to market prices, that is, every one is able to sell the *entire* initial good endowment and then uses the revenue to buy a *bundle of goods* such that its utility function is maximized.
- Wether or not equilibrium *prices* could be set every good such that this is possible? An affirmative answer was given by *Arrow and Debreu* in 1954, "Existence of an Equilibrium for a Competitive Economy," Econometrica 22, who showed that such equilibrium would exist if the utility functions were *concave* under mild conditions.

1.2 Computational market equilibrium principles

Let:

1. $p_i \geq 0$ be the price for good i, $i = 1, \cdots, m$
2. $x_{ij} \geq 0$ be the amount of good i purchased by trader j

Then, $x_{ij}, p_i, i = 1, \cdots m, j = 1, \cdots, n$, is a market equilibrium if and only if it meets following *economic principles*:

1.2.1 Principle I: Individual Rationality

For prices p_i, $i = 1, \cdots, m$ and $\mathbf{x}_{\cdot}$, $\mathbf{x_j} = (x_{1j}, \cdots, x_{nj})$ is a maximal solution for the jth trader:

$$\begin{array}{rl} maximize_{x_j} & u^j(x_j, \bar{x}_j) \\ s.t. & \sum_i p_i x_{ij} \leq \sum_i p_i w_{ij}, \\ & x_{ij} \geq 0, \forall i. \end{array}$$

where $u^j(\cdot)$ is the utility function of trade j concave in its won decision variable x_j, and *externalities* $\bar{x}_j$ represent the purchasing variables of the rest of traders.

1.2.2 Principle II: Physical Constraint

The total *purchase volume* for good i should not exceed its available physical supply:

$$\sum_j x_{ij} \leq b_i := \sum_j w_{ij}, \forall i,$$
or,
$$\sum_j \mathbf{x}_j \leq \mathbf{b} := \sum_j \mathbf{w_j}.$$

1.2.3 Principle III: Walras' Law: Market "Fairness"

For every good i,

$$\sum_j x_{ij} < b_i \Rightarrow p_i = 0, \forall i,$$
or,
$$p_i(b_i - \sum_j x_{ij}) = 0, \forall i \Rightarrow \mathbf{p}^T(\mathbf{b} - \sum_j \mathbf{x}_j) = 0;$$

so that good i is a "free" good, and this is the only way to *clear* the market.

1.3 Fisher's Exchange Market Model

- The traders are divided into two categories: m (good) producers and n buyers, where each buyer is equipped with a *utility function* on all goods and a money budget w_j.
- The trade/exchange according to *market prices*, that is, every producer is able to sell its good and every buyer is able to use the budget to buy a *bundle of goods* such that its utility function is maximized.
- Whether or not equilibrium *prices* could be set for every good such that this is possible? Fisher's model is a special case of Walras' model when money is also considered a good so that Arrow and Debreu's result applies.

2 Utility functions and complexity classes of the equilibrium problem

2.1 Degree-One Homogeneous Utility Functions

Cobb-Douglass Utility : $u^j(\mathbf{x}_j) = \Pi_i x_{ij}^{a_{ij}}$,
where a_{ij} is a given *utility exponent* with $\sum_i a_{ij} = 1$ and $a_{ij} \geq 0$ for all i, j.

Linear Utility : $u^j(\mathbf{x}_j) = \sum_i a_{ij} x_{ij}$,
where a_{ij} is a given *utility coefficient* with $a_{ij} \geq 0$ for all i, j.

Leontief Utility (piece-wise lenear): $u^j((x)_j = \min_i\{\frac{x_{ij}}{a_{ij}}\}$,
where a_{ij} is a given utility factor with $a_{ij} \geq 0$ for all i, j, where $\frac{*}{0} := \infty$.

2.2 Computational Complexity Classes

Utility/Model	Fisher's Model	Arrow-Debreu Model
Cobb-Douglas	Strongly Polynomial	Strongly Polynomial
Linear	Polynomial	Polynomial
Leontief	Polynomial[1]	NP-Hard/PPAD

2.3 An Aggregate Social Utility Maximization

Consider the convex optimization problem

$$\begin{aligned} maximize \sum_j w_j \log(u_j) & \\ s.t. \quad A\mathbf{u} &\leq \mathbf{b}, \\ \mathbf{u} &\geq 0 \end{aligned}$$

Theorem Any optimal solution and its lagrange multiplier of the aggregate problem are an equilibrium point.

2.4 Equilibrium vs. Quasi-equilibrium

2.5 Characterization of the Leontief Economy Equilibrium

Leontief Equilibrium Problem.

Linear Complementary Problem (LCP).

Every Complementary solution induces an equilibrium.

Theorem of LCP Equivalence Every non-trivial complementary solution to the LCP induces an equilibrium utility vector whose support is a subset of the original support.

2.6 Hardness Results

- It's NP-Hard to decide whether or not it has a true equilibrium (Codenotti et al. 2005). In addition, the following problems are NP-Hard:

  1. Is there more than one equilibrium?
  2. Is there an equilibrium where at least k goods are positively priced?
  3. Is there an equilibrium where at most k goods are positively priced?

- Computing an exact equilibrium is PPAD hard (Daskalakis, Papadimitirou... 2005)
- Computing an approximate equilibrium is also PPAD hard.

3 The Sequential/Dynamic Economy Equilibrium Problem

3.1 The Sequential market/Economy

Recall the static social Leontief economy equilibrium problem in a Fisher market:

$$\begin{aligned} maximize \quad & \sum_j w_j \log(u_j) = \log(\Pi_j u_j^{w_j}) \\ & s.t. \quad A\mathbf{u} \leq \mathbf{b}, \\ & \qquad u \geq \mathbf{0}, \end{aligned} \tag{1}$$

In sequential/dynamic market we only know $\mathbf{b}$ and n in advance:

- The economy information/data (constraint matrix and budget) are revealed trader by trader (column by column) *sequentially*.
- An *irrevocable* good allocation must be made by the market as soon as a trader arrives without observing or knowing the future data.

3.2 Aim: Pricing the Sequential Market

Could the market maker *pricing* the goods sequentially such that the market approximately clears at the very end, and the social objective value is near the static *optimal one*?

3.3 Observations of Sequential Market Equilibria

- The maximal value for fixed $\mathbf{b}$, $OPT(A, \mathbf{w})$, of the static problem is an upper bound on the objective value of the sequential problem.

- If the static market clearing prices $\mathbf{p}$ are known, then

$$u_t = \frac{w_t}{a_t^T \mathbf{p}}$$

so that there would be no difference between the static and sequential markets.

- But the market maker does not have full information/data, nor even any possible stochastic distribution of the input.

- Thus, the sequential algorithm needs to be data-driven and learning-based, in particular, learning-while-doing.

Model Assumptions:

1. The columns $\mathbf{a}_t$, together with w_t, arrive in a random order. That is, the set of columns together with their budgets can be adversarily picked at the start, but, after they are chosen, each permutation δ has the same chance to happen.

2. The total number of columns n is known as a priori.

3.4 Performance Ratio

An algorithm/mechanism quality is evaluated on the expected performance over all the permutations comparing to the static optimal value (by competitive ratio).

3.5 Physical Bound and Results

Theorem For any fixed $\epsilon > 0$, there is an algorithm that is $1 - O(\epsilon)$ competitive for the sequential market on all inputs if either

$$\frac{b_i}{max_t\{a_{it}\}} \geq \Omega\Big(\frac{m \log log(nw_{max}/w_{min}\epsilon)}{\epsilon^2}\Big), \forall i. \tag{2}$$

It illustrates that if the amount of physical goods is sufficiently large and the budget is more or less evenly distributed amon traders, then the sequential market can be equally efficient as the static market with full information.

3.6 Linear Optimization Market

For simplicity, we consider a static linear optimization market

$$\begin{aligned} maximize_x \quad & \sum_{t=1}^{n} \pi_t x_t \\ s.t. \quad & \sum_{t=1}^{n} a_{it} x_t \leq b_i, \, \forall i = 1, \cdots, m \\ & x_t \in [0,1] or \{0,1\}, \, \forall t = 1, \cdots, n \end{aligned}$$

Each order t requests up to one unit of a bundle of m goods, and is willing to pay π_t for it. A technical assumption:

$$\pi_t \text{ either positive or negative, and } \mathbf{a}_t = (a_{it})_{i=1}^m \in [-1, 1]^m.$$

3.7 Related Prior Online Problems

- Revenue Management Problems
- k-secretary Problem
- Worst-case analysis for online *packing and routing* problems
- Online Adwords Problem

4 Computing Sequential Economy Equilibria

4.1 One-Time Learning Algorithm

We start with a simple one-time learning algorithm.

Key Observation: The dual optimal solution of the offline problem can determine a primal (approximate) optimal solution. therefore, we hope that by solving a *partial* linear program up to ϵn arrived orders, we can get a dual solution $\bar{p}$ that is close enough to get a near optimal solution for pricing the future orders. The algorithm is as follows:

1. Set $x_t = 0$ for all $t \leq \epsilon n$
2. solve the ϵ part of the problem:

$$\begin{aligned} maximize_x \quad & \sum_{t=1}^{\epsilon n} \pi_t x_t \\ s.t. \quad \sum_{t=1}^{\epsilon n} a_{it} x_t & \leq (1-\epsilon)\epsilon b_i, \quad i = 1, \cdots, m \\ 0 \leq x_t & \leq 1, \quad t = 1, \cdots, \epsilon n \end{aligned}$$

   Get the optimal dual solution $\bar{p}$

3. Determine the future allocation $x_t(\bar{p})$ as:

$$x_t(\bar{p}) = \begin{cases} 0 & if\, \pi_t \leq \bar{p}^T \mathbf{a}_t \\ 1 & if\, \pi_t > \bar{p}^T \mathbf{a}_t \end{cases} \tag{3}$$

   If $a_{it} x_t(\bar{p}) \leq b_i - \sum^{t-1} j = 1 a_{ij} x_j$, set $x_t = x_t(\bar{p})$; otherwise, set $x_t = 0$.

4.2 Dynamic Price Updating Algorithm

4.2.1 Problem

- The one-time learning algorithm is simple, but the condition required on the size of B is strong.
- The one-time learning algorithm only computes the price once. Therefore potential improvement might be made by updating the price dynamically during the process.
- However, we have to determine how frequently we should update the prices

In the dynamic price updating algorithm, we update the price at time $\epsilon n, 2\epsilon n, 4\epsilon n \cdots$ At time $\ell = \{\epsilon n, 2\epsilon n, \cdots\}$, the price is the *optimal dual solution* to the following linear program:

$$\begin{aligned} maximize_x \quad & \sum_{t=1}^{\ell} \pi_t x_t \\ s.t. \quad \sum_{t=1}^{\ell} a_{it} x_t \leq (1 - h_\ell)\frac{\ell}{n} b_i, \quad & i = 1, \cdots, m \\ 0 \leq x_t \leq 1, \quad & t = 1, \cdots, \ell \end{aligned}$$

where $h_\ell = \epsilon\sqrt{\frac{n}{\ell}}$

And this price is used to determine the allocation for the next immediate period.

4.2.2 Algorithm

In this algorithm, we update the price $\log_2(1/\epsilon)$ times during the entire time horizon.

The numbers h_ℓ play an important role in improving the condition on B in our main theorem. It basically balances the probability that the inventory ever gets violated and the lost of revenue due to the factor $1 - h_\ell$.

This scheme is an analogy of the "doubling trick" in the machine learning community.

Note that h_ℓ is chosen as ϵ in the one-time learning case. However, in the dynamic learning case, h_ℓ is gradually reduced from $\sqrt{\epsilon}$ at the every beginning to ϵ at the very end.

Choosing large h_ℓ (more conservative) at the beginning periods and smaller h_ℓ (more risk neutral) at the later periods, we can now control the loss of revenue by an ϵ order while the required size of B can be weakened by an ϵ factor.

5 Open Problems for Dynamic Markets

- Derive a necessary and sufficient condition for near-optimality? Improve the condition on B by removing m or $\log(n/\epsilon$ or both?

- I Improve the condition on B assuming that $(\pi_t, (a)_t)$ are drawn independently from a distribution?
- Does the price vector converges or has a concentration?
- Derive near-optimal sequential algorithms for other utility functions?
- Derive near-optimal sequential algorithms for general demand functions?
- Iterative algorithms for sequential markets.