A Fast and Distributed Algorithm for Computing Fisher Equilibrium in Economies with Decreasing Marginal Returns

Lisa Fleischer Rohit Khandekar Amin Saberi

Abstract

We consider a problem of computing Fisher equilibrium in the market where aggregate demand is a convex function of prices. We show that an approximate market equilibrium can be computed using a natural tâtonnement-like process. Our algorithm iteratively updates the prices of the goods according to whether they are in excess demand or excess supply. It can be implemented in a distributed manner and computes an approximate equilibrium in linear *number of iterations.*

1 Introduction

In the market equilibrium problem, there is a set of infinitely divisible goods of fixed quantity and a set of agents. Each agent has a utility function that takes as argument an allocation of goods, and a budget. Given a set of prices for the goods, each agent seeks to maximize his or her utility, subject to spending at most its budget. If this results in demand equal to the supply for each good, then these prices are called equilibrium prices, and we say that the market "clears".

Walras formulated this problem and proposed that an appropriately defined process that increases prices of goods with excess demand and decreases prices of goods in excess supply should eventually lead to equilibrium prices [22]. This was given a formal expression via differential equations by Samuelson [21]. In 1950's Eisenberg and Gale gave a convex program that describes the allocation of goods to agents under equilibrium prices. The equilibrium prices can be calculated from the solution by . please fill this in

Recent attention to algorithmic techniques for finding equilibrium have led to a number of new methods for efficient calculation of equilibrium prices for several interesting special classes of utility functions [6, 10, 15, 16, 11, 19, 9, 3, 5, 24]. These algorithms are centralized in the sense that they require all the utility values of the buyers for its implementation. These algorithms also do not capture a central element that is arguably evident in Samuelson's differential expression: the dynamics of market behavior. mention these special classes here?

In this extended abstract, we give a decentralized algorithm to compute equilibrium prices when aggregate demands are a convex function of prices. This captures some natural markets where marginal demand decreases as prices increase. It includes the special cases of Leontief and Cobb-Douglass economies.

Our algorithm updates prices in a natural, iterative manner, adjusting a good's price upward when demand for that good under the current prices exceeds supply, and adjusting it downward when its supply exceeds demand. The algorithm is also fast: the number of iterations is independent of the number of agents, and almost linear in the number of goods. Do we want to make a direct comparison to the fastest known runtime for an algorithm for this class of problems? If we can, this is always useful.

2 Problem Definition

In this section, we define the market problem, discuss some assumptions we consider, and describe mathematical formulations of the problem.

In the market, there are n agents and m goods. There is one unit of supply of each good. Agent $i \in [n]$ has an initial endowment c_i. An *allocation* $x_i = (x_{i1}, \ldots, x_{im})$ is, for all $j \in [m]$, an assignment of x_{ij} units of good j to agent i. Given prices $\pi_j \geq 0$ for goods $j \in [m]$, agent i seeks an allocation that is within her budget: $\sum_j \pi_j x_{ij} \leq c_i$; and maximizes her utility $u_i(x_i)$. Without loss of generality, we assume $\sum_i c_i = 1$. A price vector π is called *market clearing* if the aggregate demand of each good equals its total supply. The market equilibrium problem is to compute market clearing prices.

We assume the following properties about the utility functions u_i.

- **Monotonicity.** We assume that for each agent $i \in [n]$, we have $u_i(x) \geq u_i(y)$ for $x \geq y$.

- **Convexity.** Although the bundle x_i that maximizes the utility of the agent i subject to her budget constraint may not, in general, be unique, we assume that the agents have a canonical way to break ties. Let $x_i(\pi)$ be the unique bundle agent i demands given the price vector π.

  $$x_i(\pi) := \text{argmax}\{u_i(x) \mid x \in \Re^m_+, \pi^T x \leq c_i\}$$

  Let $x(\pi) = \sum_i x_i(\pi)$ be the aggregate demand over all the agents and for goods $j \in [m]$, and $x^j(\pi)$ denote the jth component of $x(\pi)$, i.e., the aggregate demand of good j.

  We assume that for each good $j \in [m]$, the function $x^j(\pi)$ is a *convex* function of $\pi \geq 0$, i.e., $x^j(\alpha\pi_1 + (1-\alpha)\pi_2) \leq \alpha x^j(\pi_1) + (1-\alpha)x^j(\pi_2)$ for all $\pi_1, \pi_2 \geq 0$ and $\alpha \in [0, 1]$.

Under the assumption of monotonicity, given any price vector π, there exists a demand bundle for each agent that maximizes her utility and *exhausts her budget*. Thus if the sum of prices $\sum_j \pi_j = 1$ equals the total endowment in the market and if the demand of each good is at most its supply, there exists demands which maximize the agents' utilities and clear the market. Thus the market equilibrium problem can be formulated as follows:

$$\begin{array}{c}\text{Find } \pi \in \Re^m_+ \text{ such that} \\ \sum_j \pi_j = 1 \\ x^j(\pi) \leq 1 \text{ for each } j \in [m]\end{array} \tag{1}$$

The convexity of the functions $x^j(\pi)$ implies that this mathematical program is convex.

If we want to revive our old theorem with regards to the Eisenberg-Gale program, this place (section 2) would be a good place to discuss the (general) Eisenberg-Gale program and explain its relation to the above program. If we can revive the old theorem under the assumption of concave or log-concave utilities, I suggest we do so. Even if we can't, we might want to describe briefly the EG program and its relation to ours.

3 A Distributed Algorithm to Compute Equilibrium Prices

In this section we give a simple, distributed, natural iterative algorithm to compute equilibrium prices, assuming a natural demand oracle.

Our techniques build on an extensive history of using and analyzing Lagrangian based methods for finding approximate solutions to highly structured linear and convex programs [1, 20, 17, 23, 18].

Our algorithm solves the program (1) by maintaining a price π_j for each good j (see Figure 1). These prices are, in fact, the dual variables associated with the constraints $x^j(\pi) \leq 1$. In step (2c), we let $\rho = \max_j x^j(\hat{\pi})$ denote the maximum demand of any item j in this iteration. The update in the prices is given in step (2d).

Theorem 3.1 *The algorithm in Figure 1, given any $\epsilon > 0$, computes $\tilde{\pi} \geq 0$ such that $\sum_j \tilde{\pi}_j = 1$ and $x^j(\tilde{\pi}) \leq 1+\epsilon$ by making $O(m\epsilon^{-2}\log m)$ calls to the following "demand" oracle: given $\pi \geq 0$ such that $\sum_j \pi_j = 1$, compute $x(\pi)$.*

The proof of Theorem 3.1 depends on a sequence of lemmas. Let $\tilde{\pi}$ denote the output of the algorithm and let $\tilde{x} = (\sum_{r=1}^N \frac{x_r}{\rho_r})/(\sum_{r=1}^N \frac{1}{\rho_r})$ where x_r and ρ_r are the values of x and ρ in rth iteration. From convexity of the functions $x^j(\pi)$, it is easy to see that $x^j(\tilde{\pi}) \leq \tilde{x}^j$. Thus in order to prove $x^j(\tilde{\pi}) \leq 1 + O(\epsilon)$, it is enough to prove that $\tilde{x}^j \leq 1 + O(\epsilon)$.

Lemma 3.2 *For each $j \in [m]$,*

$$\frac{\tilde{x}^j}{1+\epsilon} \leq \left(\frac{\ln \pi_j^N}{\epsilon \sum_{r=1}^N \frac{1}{\rho_r}}\right) \leq \tilde{x}^j$$

where π_j^N denotes the value of π_j after N rounds and ρ_r denotes the value of ρ in rth iteration.

1. Initialize $\pi_j = 1$ for $1 \le j \le m$.
2. Repeat $N = \frac{m}{\epsilon} \log_{1+\epsilon} m$ times:
   (a) Declare prices $\hat{\pi}_j = \pi_j / \sum_k \pi_k$ for goods j.
   (b) Each agent i demands the bundle $x_i(\hat{\pi})$.
   (c) Compute the aggregate demand $x = x(\hat{\pi}) = \sum_i x_i(\hat{\pi})$ and let $\rho = \max_j x^j$ where x^j is the aggregate demand of good j.
   (d) Update for each good j:
   $$\pi_j \leftarrow \pi_j \left(1 + \frac{\epsilon}{\rho} x^j\right).$$
3. Output $\tilde{\pi} = (\sum_{r=1}^N \frac{\hat{\pi}^r}{\rho_r}) / (\sum_{r=1}^N \frac{1}{\rho_r})$ where $\hat{\pi}^r$ and ρ_r are the values of $\hat{\pi}$ and ρ in rth iteration.

Figure 1. Algorithm for the convex program (1)

Proof. Let x_r denote the value of x in rth iteration. Since $0 \le \frac{\epsilon}{\rho_r} x_r^j \le \epsilon$, using $\exp(\epsilon/(1+\epsilon)) \le 1+\epsilon \le \exp(\epsilon)$, we have

$$\exp\left(\frac{\epsilon x_r^j}{(1+\epsilon)\rho_r}\right) \le \left(1 + \frac{\epsilon x_r^j}{\rho_r}\right) \le \exp\left(\frac{\epsilon x_r^j}{\rho_r}\right).$$

Multiplying these inequalities for $1 \le r \le N$, we get

$$\exp\left(\frac{\epsilon}{1+\epsilon}\sum_{r=1}^N \frac{x_r^j}{\rho_r}\right) \le \prod_{r=1}^N \left(1 + \frac{\epsilon x_r^j}{\rho_r}\right) \le \exp\left(\epsilon \sum_{r=1}^N \frac{x_r^j}{\rho_r}\right).$$

Since $\pi_j^N = \prod_{r=1}^N \left(1 + \frac{\epsilon x_r^j}{\rho_r}\right)$, we conclude

$$\frac{1}{1+\epsilon}\sum_{r=1}^N \frac{(Ax_r)_j}{\rho_r} \le \frac{\ln \pi_j^N}{\epsilon} \le \sum_{r=1}^N \frac{x_r^j}{\rho_r}.$$

Dividing the above inequalities by $\sum_r \frac{1}{\rho_r}$ proves the lemma. ∎

Lemma 3.3

$$\frac{\ln m}{\epsilon^2 \sum_r \frac{1}{\rho_r}} \le \max_j \tilde{x}^j \le (1+\epsilon)\left(1 + \frac{\ln m}{\epsilon \sum_r \frac{1}{\rho_r}}\right).$$

Proof. From the definition of ρ in step (2c) and update in step (2d), we get that, in each iteration, at least one π_j increases by a factor of $(1+\epsilon)$. Thus after $N = \frac{m}{\epsilon}\log_{1+\epsilon} m$ iterations, there exists $k \in [m]$ such that π_k increases by a factor of $(1+\epsilon)$ at least $\epsilon^{-1}\log_{1+\epsilon} m$ times and hence $\pi_k^N \ge m^{1/\epsilon}$, i.e., $\ln \pi_k^N \ge \frac{\ln m}{\epsilon}$. Therefore we have

$$\begin{aligned} \max_j \tilde{x}^j &\ge \tilde{x}^k \\ &\ge \frac{\ln \pi_k^N}{\epsilon \sum_{r=1}^N \frac{1}{\rho_r}} \quad \text{(from Lemma 3.2)} \\ &\ge \frac{\ln m}{\epsilon^2 \sum_r \frac{1}{\rho_r}}. \end{aligned}$$

Let π^r denote the value of π after rth iteration and let $\hat{\pi}^r$ denote the normalized vector given by $\hat{\pi}^r_j = \pi^r_j / \sum_k \pi^r_k$. Note that

$$\begin{aligned}\sum_j \pi^r_j &= \sum_j \pi^{r-1}_j + \frac{\epsilon}{\rho_r} \sum_j \pi^{r-1}_j x^j_r \\ &= \left(\sum_j \pi^{r-1}_j\right)\left(1 + \frac{\epsilon}{\rho_r}\sum_j \hat{\pi}^{r-1}_j x^j_r\right) \\ &\leq \left(\sum_j \pi^{r-1}_j\right)\left(1 + \frac{\epsilon}{\rho_r}\right).\end{aligned}$$

The last inequality follows from the fact that $x_r = x(\hat{\pi}^{r-1})$ and that $\sum_j \pi_j x^j(\pi) \leq 1$ for any π such that $\sum_j \pi_j = 1$. Therefore we have

$$\sum_j \pi^N_j \leq \left(\sum_j \pi^0_j\right)\prod_{r=1}^{N}\left(1 + \frac{\epsilon}{\rho_r}\right) \leq m \cdot \exp\left(\epsilon \sum_r \frac{1}{\rho_r}\right).$$

Thus

$$\ln \sum_j \pi^N_j \leq \ln m + \epsilon \sum_r \frac{1}{\rho_r}. \tag{2}$$

Therefore

$$\begin{aligned}\frac{\max_j \tilde{x}^j}{1+\epsilon} &\leq \frac{\ln \max_j \pi^N_j}{\epsilon \sum_r \frac{1}{\rho_r}} \quad \text{(from Lemma 3.2)} \\ &\leq \frac{\ln \sum_j \pi^N_j}{\epsilon \sum_r \frac{1}{\rho_r}} \\ &\leq \frac{\ln m + \epsilon \sum_r \frac{1}{\rho_r}}{\epsilon \sum_r \frac{1}{\rho_r}} \quad \text{(from (2))} \\ &\leq 1 + \frac{\ln m}{\epsilon \sum_r \frac{1}{\rho_r}}.\end{aligned}$$

Thus the proof is complete. ∎

Now we are ready to prove our main theorem.
Proof. [of Theorem 3.1] From Lemma 3.3, it is evident that

$$\begin{aligned}(1 - \epsilon(1+\epsilon)) \max_j \tilde{x}^j &= \max_j \tilde{x}^j - \epsilon(1+\epsilon) \max_j \tilde{x}^j \\ &\leq \max_j \tilde{x}^j - (1+\epsilon)\frac{\ln m}{\epsilon \sum_r \frac{1}{\rho_r}} \\ &\leq (1+\epsilon)\left(1 + \frac{\ln m}{\epsilon \sum_r \frac{1}{\rho_r}}\right) - (1+\epsilon)\frac{\ln m}{\epsilon \sum_r \frac{1}{\rho_r}} \\ &\leq 1 + \epsilon.\end{aligned}$$

Thus we have

$$\max_j \tilde{x}^j \leq \frac{1+\epsilon}{1 - \epsilon(1+\epsilon)} \leq 1 + O(\epsilon)$$

as desired. Now using convexity of functions $x^j(\pi)$, we conclude that $\max_j x^j(\tilde{\pi}) \leq \max_j \tilde{x}^j \leq 1 + O(\epsilon)$. Note that the number of oracle calls in the algorithm is $N = \frac{m}{\epsilon} \log_{1+\epsilon} m = O(m\epsilon^{-2} \log m)$. ∎

4 Examples

There are several well-known utility functions for which the assumptions of monotonicity and convexity of the aggregate demand hold. For example,

- Leontief utility functions
$$u(x_1, \ldots, x_m) = \min_{j:\alpha_j>0} \alpha_j x_j$$
for some constants $\alpha_j \geq 0$. Given prices π_j, the bundle that maximizes the Leontief utility satisfies $x^j = \frac{c_i}{\alpha_j \sum_{k:\alpha_k>0} \pi_k/\alpha_k}$ for all j such that $\alpha_j > 0$. It turns out that this is a convex function of the price vector π.

- Cobb-Douglas utility functions
$$u(x_1, \ldots, x_m) = \prod_j x_j^{\alpha_j}$$
for some constants $\alpha_j \geq 0$. Without loss of generality, we can assume that $\sum_j \alpha_j = 1$. Given prices $(\pi_1, \ldots, \pi_m) \geq 0$ and a unit budget, the bundle $x = (x_1, \ldots, x_m)$ that maximizes the utility is given by $x^j = \alpha_j/\pi_j$ for each j. Such an x^j is also a convex function of the prices π.

References

[1] Daniel Bienstock. *Potential function algorithms for approximately solving linear programs: theory and practice*. International Series in Operations Research and Management Science. Kluwer Academic Publisher, 2002.

[2] Y. Bartal, J. W. Byers, D. Raz, Fast, Distributed Approximation Algorithms for Positive Linear Programming with Applications to Flow Control. SIAM J. Comput. 33(6): 1261-1279 (2004).

[3] B. Codenotti, B. McCune, and K. Varadarajan, Market Equilibrium via the Excess Demand Function, STOC 2005.

[4] B. Codenotti, S. Pemmaraju and K. Varadarajan, On the Polynomial Time Computation of Equilibria for Certain Exchange Economies, SODA 2005.

[5] B. Codenotti, S. Pemmaraju and K. Varadarajan, Algorithms Column: The Computation of Market Equilibria, ACM SIGACT News 35(4) December 2004.

[6] N. R. Devanur, C. H. Papadimitriou, A. Saberi, V. V. Vazirani, Market Equilibrium via a Primal-Dual-Type Algorithm. FOCS 2002, pp. 389-395. (Full version with revisions available on line, 2003.)

[7] E. Eisenberg, Aggregation of Utility Functions. Management Sciences, Vol. 7 (4), 337–350 (1961).

[8] D. Gale, The Linear Exchange Model, Journal of Mathematical Economics 3, pp. 205-209 (1976).

[9] K. Jain, A polynomial time algorithm for computing the Arrow-Debreu market equilibrium for linear utilities, Discussion Paper, Microsoft Lab., Seattle, WA (2003). FOCS 2004.

[10] K. Jain, M. Mahdian, and A. Saberi, Approximating Market Equilibria, Proc. APPROX 2003.

[11] K. Jain, V. V. Vazirani and Y. Ye, Market Equilibria for Homothetic, Quasi- Concave Utilities and Economies of Scale in Production, SODA 2005.

[12] F. P. Kelly. Models for a self–managed Internet. In Proceedings of Royal Society Meeting, December 1999.

[13] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan, Rate control for communication networks: shadow prices, proportional fairness and stability, Journal of the Operational Research Society, 1998, vol. 49, no. 3, pp. 237-252(16).

[14] L. Fleischer. Approximating fractional multicommodity flow independent of the number of commodities. *SIAM J. Discrete Math.*, 13:505–520, 2000.

[15] R. Garg and S. Kapoor, Auction Algorithms for Market Equilibrium. In *Proc. STOC*, 2004.

[16] R. Garg, S. Kapoor, and V. V. Vazirani, An auction-based market equilibrium algorithm for the separable gross substitutability case, APPROX 2004.

[17] N. Garg and J. Könemann. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. In *Proceedings, IEEE Symposium on Foundations of Computer Science*, pages 300–309, 1998.

[18] R. Khandekar. *Lagrangian relaxation based algorithms for convex programming problems*. PhD thesis, Indian Institute of Technology Delhi, 2004.

[19] E. I. Nenakov and M. E. Primak. One algorithm for finding solutions of the Arrow-Debreu model, Kibernetica 3, 127–128 (1983).

[20] S. Plotkin, D. Shmoys, and E. Tardos. Fast approximation algorithms for fractional packing and covering problems. *Math. Oper. Res.*, 20:257–301, 1995.

[21] P. Samuelson. *Foundations of Economic Analysis*. Harvard University Press, 1947.

[22] L. Walras. *Eléments d' Economie Politique Pure*. Corbaz, Lausanne, 1874. (Translated as: *Elements of Pure Economics*. Homewood, IL: Irwin, 1954.).

[23] N. Young. Sequential and parallel algorithms for mixed packing and covering. In *Proceedings, IEEE Symposium on Foundations of Computer Science*, pages 538–546, 2001.

[24] Y. Ye. “A path to the Arrow-Debreu competitive market equilibrium.” Working Paper, Department of Management Science and Engineering, Stanford University, Stanford, CA 94305, 2004.