A Fast and Simple Algorithm for Computing Market Equilibria

Lisa Fleischer[1], Rahul Garg[2], Sanjiv Kapoor[3], Rohit Khandekar[2], and Amin Saberi[4]

[1] Dartmouth College. lkf@cs.dartmouth.edu.
[2] IBM T.J. Watson Research Center. {grahul,rohitk}@us.ibm.com.
[3] Illinois Institute of Technology. kapoor@iit.edu.
[4] Stanford University. saberi@stanford.edu.

Abstract. We give a new mathematical formulation of market equilibria using an *indirect utility function*: the function of prices and income that gives the maximum utility achievable. The formulation is a *convex program* and can be solved when the indirect utility function is convex in prices. We illustrate that many economies including

- Homogeneous utilities of degree $\alpha \in [0, 1]$ in Fisher economies — this includes Linear, Leontief, Cobb-Douglas
- *Resource allocation utilities* like multi-commodity flows

satisfy this condition and can be efficiently solved.

Further, we give a natural and decentralized price-adjusting algorithm in these economies. Our algorithm, mimics the natural tâtonnement dynamics for the markets as suggested by Walras: it iteratively adjusts a good's price upward when the demand for that good under current prices exceeds its supply; and downward when its supply exceeds its demand. The algorithm computes an approximate equilibrium in a number of iterations that is independent of the number of traders and is almost *linear* in the number of goods. Interestingly, our algorithm applies to certain classes of utility functions that are not *weak gross substitutes*.

1 Introduction

The market equilibrium model, common in economics, is that of a market with m traders and n goods, where the traders are endowed with money or/and goods and wish to optimize their utilities. Market equilibrium is defined by a price and an allocation such that no trader has any incentive to trade and there is no excess demand of any good. While the problem was originally formulated by Walras [29] in 1874, the existence of such an equilibrium was established by Arrow and Debreu [1] in 1954 using a fixed-point argument.

The result of Arrow and Debreu does not give much insight into the dynamics of the market. How does market find the equilibrium prices? What is the complexity of finding these prices? Interested in answering the first question, economists focused on decentralized dynamics that converge to equilibrium. Most notably, Samuelson [26] formalized Walras' idea of tâtonnement as a set of differential equations relating the adjustment of the price with excess demand. Later, Arrow et al. and Nikaido and Uzawa [2, 23] showed that in markets with gross substitute property, the process proposed by Samuelson converges to an equilibrium. The number of iterations of such a process depends on the utility functions of the traders.

In computer science literature, the focus has been on designing polynomial-time algorithms for several special cases using techniques such as primal-dual, auctions algorithms and convex programming [9, 7, 17, 13, 15, 10, 30]. The surveys of Vazirani [28] and Codenotti and Varadarajan [4] discuss these results. These algorithms (with the notable exception of [8][5]) are typically centralized.

This paper attempts to combine the advantages of the both approaches for a restricted class of markets. We present a fast and relatively natural algorithm for computing approximate equilibrium prices. The number of iterations required by our algorithm to converge to approximate equilibrium prices is almost linear in the number of goods and is independent of the number of traders. Another desirable feature of our algorithm is its distributed nature: it does not need to gather the information on utility functions and endowments of the traders in a central place to compute the prices. It only offers the sellers a procedure for updating the prices based on the difference of demand and supply of their good that converges to market equilibria. In fact, except a normalization variable, the only information passed between buyer and seller of a good is the current price of the goods and the demand corresponding to the current price.

From an algorithm design perspective, our procedure is different from primal-dual or auction algorithms in the sense that the prices (dual variables) do not approach the equilibrium from below. The process may underestimate or overshoot the equilibrium prices several times before it converges. In that sense, our algorithm is closest to the results of [12, 24]. The analysis uses a new convex program for characterizing equilibria. For that reason the class of markets for which we can analyze our procedure is slightly more restricted than the class of markets comprising weakly gross substitute goods. At the same time, it include resource allocation markets, which are in fact not gross-substitute markets.

In particular, our algorithm applies to the market model for network congestion control as a part of a larger class of resource allocation markets [18]. For the case of multiple sources and sinks, the problem of determining, or discovering, equilibrium prices using a tâtonnement or combinatorial process, appears rather challenging, especially since there are no known combinatorial polynomial time algorithms for solving the feasibility of multi-commodity flows in networks. Fortunately, approximate solutions are tractable as we illustrate in this paper.

1.1 Results

The new convex program. We give a new formulation of the market equilibrium problem using *indirect utility function*. An indirect utility function $\widetilde{u}$ of price

[5] The result of [8] has a running time that is independent of both number of traders and number of goods, but is dependent on some other market parameters. For example, when all traders share linear utilities, the procedure in [8] may not converge.

$\pi \in \Re^n_+$ and budget (or income) $e \in \Re_+$ gives the maximum utility achievable under those prices and budget as follows:

$$\widetilde{u}(\pi, e) = \max\{u(x) \mid x \in \Re^n_+, \pi \cdot x \leq e\}$$

where u is the utility function defined on allocation of goods. Although indirect utility functions have been extensively used in Economics to study the behavior of aggregate demand [20, 27], here we use them to formulate and solve the market equilibrium problem. Our formulation becomes a convex program if the indirect utility functions are convex on a suitably defined set of prices and income. This enables polynomial-time computation of (approximate) market equilibrium using standard convex programming techniques.

We show that, in the Fisher setting, the indirect utility functions are convex if the utility functions are homogeneous of degree 1. Such utility functions include linear, Leontief, Cobb-Douglas, CES, resource allocation markets. If the utility function u is increasing in all its components, then a necessary and sufficient condition for convexity of the corresponding indirect utility function is (see Proposition 2.4 in [25]): $-\frac{x \cdot \partial^2 u(x) x}{\partial u(x) x} \leq 2$ for all x. Surprisingly, this condition has the same form as those for monotone utilities [7]. They turn out to be a special case of monotone utilities for which market equilibrium can be computed using ellipsoid method [7]. However, note that polynomial time convergent tâtonnement processes are not known for monotone utilities.

The algorithm. A natural approach to computing the equilibrium price (as originally envisaged by Walras) is an iterative algorithm termed as *tâtonnement* process where the prices of goods are updated locally as a function of excess demand. Stability of these processes have been studied extensively in the literature [2, 21] (see [22] for a survey). It has been shown that if the utility functions satisfy the *weak gross substitute* (WGS) property then the continuous process is stable and converges to market equilibrium. Polynomial-time convergence of such a process was only recently established in exchange economies with WGS utilities by the works of Codenotti et al. [7].

Our formulation enables us to design efficient processes similar to tâtonnement that converge close to a market equilibrium in polynomial time whenever the indirect utility functions of traders are convex. This partially answers the question raised in [18, 19, 7] on convergence of tâtonnement processes for a class of utility functions that do not satisfy WGS, for example, Leontief and resource allocation utilities. In order to obtain a $(1 + \epsilon)$ (weak) approximate market equilibrium, our process requires every trader to perform at most $O(\epsilon^{-2} n \log n)$ computations of its demand. For multi-commodity flow resource allocation market, for example, the demand oracle is the shortest-path computation under the given edge-lengths (prices). Thus our algorithm needs $\widetilde{O}(kn)$ shortest path computations for a market with k commodities and n edges. This contrasts against the algorithm of [18] for single-source multi-sink markets that needs $O(k^2)$ max-flow computations. We point out, however, that the algorithm of [18] computes an exact equilibrium while we compute only an approximate equilibrium.

Organization The rest of the paper is organized as follows. In Section 2, we define the market equilibrium problem and formulate a mathematical program using indirect utility functions. We also outline a convex programming technique for solving this formulation if the indirect utility functions are convex. Section 3 then presents the prominent cases where we consider several utilities in Fisher economy under which the indirect utility functions turn out to be convex. In Section 4, we present our algorithm for computing approximate market equilibria assuming convexity of indirect utility functions. Section 5 concludes with some open directions.

2 An Alternate Formulation using Indirect Utility Functions

We first describe the exchange market model. Let us consider m economic agents who represent traders of n goods. Let $\Re^n_+$ (resp. $\Re^n_{++}$) denote the subset of $\Re^n$ where the coordinates are non-negative (resp. strictly positive). The jth coordinate will stand for good j. Each trader i ($i = 1, \ldots, m$) is associated with

- a non-empty convex set $\mathcal{K}_i \subseteq \Re^n$ which is the set of all "feasible" allocations that trader i may receive (in many cases, $\mathcal{K}_i = \Re^n_+$),
- a *concave* utility function $u_i : \mathcal{K}_i \to \Re_+$ which represents her preferences for the different bundles of goods, and
- an initial endowment of goods $w_i = (w_{i1}, \ldots, w_{in})^\top \in \mathcal{K}_i$.

At given prices $\pi \in \Re^n_+$, the trader i sells her endowment, and gets the bundle of goods $x_i = (x_{i1}, \ldots, x_{in})^\top \in \mathcal{K}_i$ which maximizes $u_i(x)$ subject to budget constraint[6] $\pi \cdot x \leq \pi \cdot w_i$. A market equilibrium is a price vector $\pi \in \Re^n_+$ and bundles $x_i \in \mathcal{K}_i$ such that: $x_i \in \text{argmax}\{u_i(x) \mid x \in \mathcal{K}_i, \pi \cdot x \leq \pi \cdot w_i\}$ for all i, and $\sum_i x_i \leq \sum_i w_i$. The above described market model is called an *exchange economy*.

We make the following standard assumption on the utility functions:

Assumption 1 *For $\pi \in \Re^n_+$, any $x_i \in$ argmax$\{u_i(x) \mid x \in \mathcal{K}_i, \pi \cdot x \leq \pi \cdot w_i\}$ satisfies $\pi \cdot x_i = \pi \cdot w_i$.*

We now define a notion of indirect utility function induced by a utility function.

Definition 2 (Indirect utility function) *For trader i, the* indirect utility function $\widetilde{u}_i : \Re^n_+ \times \Re_+ \to \Re_+$ *gives the maximum utility achievable at given price and income:*

$$\widetilde{u}_i(\pi, e) = \max\{u_i(x) \mid x \in \mathcal{K}_i, \pi \cdot x \leq e\}.$$

The following theorem characterizes the set of all equilibria.

[6] For two vectors x and y, we use $x \cdot y$ to denote their inner product.

Theorem 3. *The following program gives precisely the set of all market equilibria in the exchange economy.*

$$\begin{array}{c} \sum_i x_i \leq \sum_i w_i \\ \widetilde{u}_i(\pi, \pi \cdot w_i) \leq u(x_i) \quad \textit{for all } i \\ \pi \in \Re^n_+ \\ x_i \in \mathcal{K}_i \quad \textit{for all } i. \end{array} \tag{1}$$

Proof. From the definition, it follows that a market equilibrium satisfies the above inequalities. Now for converse, consider a solution $(\pi, x_1, \ldots, x_m)$ of the above program. From the second constraint and Assumption 1, it follows that $\pi \cdot x_i \geq \pi \cdot w_i$ for all i. Furthermore from the first constraint, it follows that $\sum_i \pi \cdot x_i \leq \sum_i \pi \cdot w_i$. This implies that $\pi \cdot x_i = \pi \cdot w_i$ for all i and hence the solution $(\pi, x_1, \ldots, x_m)$ is a market equilibrium.

Note that the program (1) is convex when, for all i, the function $\widetilde{u}_i(\pi, \pi \cdot w_i)$ is a convex function of $\pi \in \Re^n_+$ and the utility function u_i is concave. Unfortunately, for many interesting utility functions u_i, the corresponding indirect utility function $\widetilde{u}_i$ is *not* convex. It turns out, however, that in *many* cases (as illustrated later in the paper), if we restrict the prices π to a carefully chosen convex set $\Pi \subseteq \Re^n_+$ that is guaranteed to contain an equilibrium price, the function $\widetilde{u}_i$ becomes convex in π. Therefore the program (1) reduces to the following convex program.

$$\begin{array}{c} \sum_i x_i \leq \sum_i w_i \\ \widetilde{u}_i(\pi, \pi \cdot w_i) \leq u(x_i) \quad \text{for all } i \\ \pi \in \Pi \\ x_i \in \mathcal{K}_i \quad \text{for all } i. \end{array} \tag{2}$$

In order to solve the above convex program using an ellipsoid algorithm, the convex set Π needs to be given in terms of a membership oracle.

Solving Program (2). Assuming that the convex sets Π and $\mathcal{K}_i$ are bounded and full dimensional,[7] the convex program (2) can be solved to an arbitrary degree of precision by an ellipsoid-like algorithm using the evaluation oracle for the functions u_i and $\widetilde{u}_i$ and membership oracles for Π and $\mathcal{K}_i$. We omit details here and refer the reader to Theorem 4.3.13 in [16].

3 Convexity of the Indirect Utility Functions

In this section, we give a class of Fisher economies in which the indirect utility function $\widetilde{u}_i$ is convex in π over a set Π. The Fisher economy is a special case of the exchange economy when $\mathcal{K}_i = \Re^n_+$ and the endowments w_i of the traders are *proportional*, i.e.,

$$w_i = \lambda_i w$$

---

[7] The economies considered in this paper have unbounded Π and $\mathcal{K}_i$ in their description. However one can usually obtain bounds on the largest value that an allocation or a price can take. Moreover the cases that Π is not full dimensional can be handled using standard projection techniques.

where $w \in \Re^n_{++}$ and $\lambda_1, \ldots, \lambda_m \in \Re_{++}$. In this case we let $\Pi = \{\pi \in \Re^n_+ \mid \pi \cdot w = 1\}$. Thus under any prices $\pi \in \Pi$, the income of trader i is fixed at λ_i.

We now quote a theorem of K.-H. Quah [25] which gives *necessary* and *sufficient* conditions on the utility functions u_i under which the indirect utility functions $\widetilde{u}_i(\pi, \lambda_i)$ are convex in $\pi \in \Pi$. We drop the subscript i to simplify the notation.

Proposition 1 (K.-H. Quah [25], Proposition 2.4). *Assume that the utility function $u : \Re^n_{++} \to \Re$ is continuous, quasi-concave, increasing in all arguments, and has the property that for any $\overline{x} \in \Re^n_{++}$, the set $\{x \in \Re^n_{++} \mid u(x) \geq u(\overline{x})\}$ is closed. Let $\lambda \in \Re_{++}$ be a constant.*

1. *Then, $\widetilde{u}(\pi, \lambda)$ is convex in prices π* if and only if *the functions μ_x are convex for all x, where $\mu_x : \Re_{++} \to \Re$ is defined by $\mu_x(s) = u(x/s)$.*
2. *Suppose, in addition, that u is $\mathcal{C}^2$, a twice differentiable function. Then μ_x is convex* if and only if $\psi(x) = -\frac{x \cdot \partial^2 u(x) x}{\partial u(x) x} \leq 2$ *for all x.*

Remark 1. Contrast the condition $\psi(x) \leq 2$ above with the condition $\psi(x) < 4$ which is sufficient to guarantee that the induced demand function is *monotone* [7]. Recall that the demand function $x(\pi)$ is monotone if for any π, π', we have $(\pi - \pi') \cdot (x(\pi) - x(\pi')) \leq 0$. Thus if $\widetilde{u}$ is convex, the induced demand function is monotone.

Corollary 4 *1. A concave homogeneous utility function u of degree α where $0 \leq \alpha \leq 1$, i.e., $u(sx) = s^\alpha u(x)$, results in convex indirect utility function $\widetilde{u}$ if u satisfies the conditions in Proposition 1.*

2. If utility functions u_1 and u_2 satisfy the conditions in Proposition 1 and induce convex indirect utility functions, then so does $u_1 + u_2$.

Proof. For (1), note that $\mu_x(s) = s^{-\alpha} u(x)$ is a convex function of s. For (2), note that if $\mu_{1,x}$ and $\mu_{2,x}$ are convex functions then so is $\mu_{1,x} + \mu_{2,x}$.

Note, however, that some natural homogeneous utility functions of degree one (e.g., Leontief utilities and resource allocation utilities, defined later) do not satisfy the conditions in Proposition 1, in particular, the condition that the utility function is increasing in all arguments. However in the next theorem we show that the homogeneous utilities induce a convex indirect utility function even when they are not increasing in all arguments.

Theorem 5. *If the utility function $u : \Re^n_+ \to \Re$ is homogeneous (of degree one), i.e., $u(\alpha x) = \alpha u(x)$ for all $\alpha \in \Re_+$ and $x \in \Re^n_+$, then the indirect utility function $\widetilde{u}(\pi, \lambda)$ is convex in π for all $\lambda \in \Re_{++}$.*

Proof. Let price vectors $\pi, \pi_1, \pi_2 \in \Re^n_+$ satisfy $\pi = \alpha\pi_1 + (1-\alpha)\pi_2$ for some $0 \leq \alpha \leq 1$. Let $x \in \Re^n_+$ be such that $\pi \cdot x = \lambda$ and $u(x) = \widetilde{u}(\pi, \lambda)$. Define $x_1 = \frac{\lambda x}{\pi_1 \cdot x}$ and $x_2 = \frac{\lambda x}{\pi_2 \cdot x}$. Note that $\pi_1 \cdot x_1 = \pi_2 \cdot x_2 = \lambda$ and hence $\widetilde{u}(\pi_1, \lambda) \geq u(x_1)$ and $\widetilde{u}(\pi_2, \lambda) \geq u(x_2)$. Using the homogeneity of u, we also get that $u(x) = \frac{\pi_1 \cdot x}{\lambda} u(x_1) \leq \frac{\pi_1 \cdot x}{\lambda} \widetilde{u}(\pi_1, \lambda)$ and $u(x) = \frac{\pi_2 \cdot x}{\lambda} u(x_2) \leq \frac{\pi_2 \cdot x}{\lambda} \widetilde{u}(\pi_2, \lambda)$.

Note that $\alpha(\pi_1 \cdot x) + (1-\alpha)(\pi_2 \cdot x) = \lambda$. Now

$$\begin{aligned}
\left(\frac{\alpha\lambda}{\pi_1 \cdot x} + \frac{\lambda(1-\alpha)}{\pi_2 \cdot x}\right) &= \left(\frac{\alpha\lambda}{\pi_1 \cdot x} + \frac{\lambda(1-\alpha)}{\pi_2 \cdot x}\right)\left(\frac{\alpha(\pi_1 \cdot x)}{\lambda} + \frac{(1-\alpha)(\pi_2 \cdot x)}{\lambda}\right) \\
&= \alpha^2 + \alpha(1-\alpha)\left(\frac{\pi_1 \cdot x}{\pi_2 \cdot x} + \frac{\pi_2 \cdot x}{\pi_1 \cdot x}\right) + (1-\alpha)^2 \\
&\geq \alpha^2 + 2\alpha(1-\alpha) + (1-\alpha)^2 \\
&= 1.
\end{aligned}$$

To complete the proof we now observe

$$\begin{aligned}
\widetilde{u}(\pi, \lambda) = u(x) &\leq u(x)\left(\frac{\alpha\lambda}{\pi_1 \cdot x} + \frac{\lambda(1-\alpha)}{\pi_2 \cdot x}\right) \\
&\leq \left(\frac{\pi_1 \cdot x}{\lambda}\widetilde{u}(\pi_1, \lambda)\right)\frac{\alpha\lambda}{\pi_1 \cdot x} + \left(\frac{\pi_2 \cdot x}{\lambda}\widetilde{u}(\pi_2, \lambda)\right)\frac{\lambda(1-\alpha)}{\pi_2 \cdot x} \\
&= \alpha\widetilde{u}(\pi_1, \lambda) + (1-\alpha)\widetilde{u}(\pi_2, \lambda).
\end{aligned}$$

The set of **homogeneous utility functions** of degree one includes the following well-studied utility functions. Here let $a \in \Re^n_+$. Linear utilities $u(x) = a \cdot x$, Leontief utilities $u(x) = \min_{j \in S} a_j x_j$ where $S \subseteq \{1, \ldots, n\}$, Cobb-Douglas utilities $u(x) = \prod_j x_j^{a_j}$ assuming $\sum_j a_j = 1$, CES utilities $u(x) = (\sum_j a_j x_j^\rho)^{1/\rho}$ for $-\infty < \rho < 1$ and $\rho \neq 0$, and nested CES utilities [5] [6].

It also includes the **resource allocation utilities** defined as follows. Let k be a positive integer and let $A \in \Re^{n \times k}_+$ be a matrix and $c \in \Re^k_+$ be a vector. The resource allocation utility $u : \Re^n_+ \to \Re$ is defined as

$$u(x) = \max\{c \cdot y \mid y \in \Re^k_+, Ay \leq x\}. \tag{3}$$

The columns of matrix A can be thought of as "objects" that the trader wants to "build". A unit of an object l needs A_{jl} units of resource (or good) j and accrues c_l units of utility. The trader then builds y_l units of object l such that the total need for resources is at most x and the total utility $c \cdot y$ is maximized. This framework includes interesting markets like

1. Multi-commodity flow markets (in directed or undirected capacitated networks). Here trader i wants to send maximum amount of flow from source s_i to sink t_i such that the total cost of routing the flow under the prices π is at most her budget. The objects here are s_i-t_i paths and the resources are the edges.
2. Steiner-tree markets in undirected (resp. directed) capacitated networks. Here trader i is associated with a subset S_i of nodes and wants to build maximum fractional packing of Steiner trees connecting S_i (resp. fractional arborescences rooted at some $r_i \in S_i$ connecting S_i to r_i) such that the total cost of building under the prices π is at most her budget. The objects here are Steiner trees (resp. arborescences). Note that computing a profit maximizing demand in undirected Steiner-tree market is NP-hard. Therefore the running times of the algorithms are only oracle-polynomial.

From Corollary 4, the **additive separable concave** utilities also induce a convex indirect utility functions: (1) $u(x_1, \ldots, x_n) = \sum_j a_j x_j^{\rho_j}$ where $a_j \in \Re_{++}$ and $0 \leq \rho_j \leq 1$; (2) $u(x_1, \ldots, x_n) = \sum_j \log(1 + a_j x_j)$ where $a_j \in \Re_{++}$ [3] — follows from the fact that $\log(1 + \frac{a_j x_j}{s})$ is a convex function of s.

4 The Algorithm

In this section, we present our algorithm to compute a *weak approximate market equilibrium* defined as follows. To simplify the definition, we assume that $\mathcal{K}_i \subseteq \Re^n_+$, i.e., we let x_{ij} take only non-negative values. For some technical reason, we assume that the set Π satisfies the following property: for any vector $p \in \Re^n_+$, there exists $\alpha \in \Re_{++}$ such that $\alpha p \in \Pi$. Note that this requirement is satisfied by the sets Π for the utilities in Fisher markets.

Definition 6 (Weak $(1+\epsilon)$-approximate market equilibrium) *A price vector $\pi \in \Pi$ and bundles $x_i \in \mathcal{K}_i$ for each trader i are called a weak $(1+\epsilon)$-approximate market equilibrium in the exchange economy if*

1. *The utility of x_i to trader i is at least that of the utility-maximizing bundle under prices π: $u_i(x_i) \geq \widetilde{u}_i(\pi, \pi \cdot w_i)$ for each i,*
2. *The total demand is at most $(1+\epsilon)$ times the supply: $\sum_i x_i \leq (1+\epsilon) \sum_i w_i$, and*
3. *The market clears: $\pi \cdot \sum_i w_i \leq \pi \cdot \sum_i x_i$.*

Note that item 3 above follows directly from item 1 and Assumption 1. If $\mathcal{K}_i \not\subseteq \Re^n_+$, we use a standard technique of "shifting" the space so that x_{ij} are non-negative. This, however, needs that $\mathcal{K}_i$ is bounded below and we know these bounds. It also weakens the notion of approximate market equilibrium and we omit the details from this extended abstract. Shifting has also been used to address similar problems arising while solving linear programs with negative entries [24].

Without loss of generality, we scale the endowments w_i so that $\sum_i w_i = \mathbf{1}$, the vector of all ones. This also implies that we scale the vectors in $\mathcal{K}_i$. We emphasize that the algorithm also works without scaling; however the scaling simplifies the presentation. The algorithm is given in Figure 1. Here $\delta > 0$ is a constant to be fixed later. The algorithm goes in N iterations. In each iteration, we first scale the current price vector p so that it lies in Π. We then "announce" this price vector and receive in response the utility-maximizing bundles $x_i \in \mathcal{K}_i$. We then update the price vector p according to the aggregate demands X_j of goods j as given in Step 2d.

Note that this update is essentially same as (within a $(1+\delta)$ factor) to the following natural update in terms of *excess demand*. Let $Z_j = X_j - \sum_i w_{ij} = X_j - 1$ be the excess demand of good j. We can update p as:

$$p_j \leftarrow p_j(1 + \delta\sigma Z_j).$$

1. Initialize $p_j = 1$ for $1 \le j \le n$.
2. Repeat for $N = \frac{n}{\delta} \log_{1+\delta} n$ iterations:
   (a) Find $\alpha > 0$ such that $\alpha p \in \Pi$. Announce prices $\pi = \alpha p$.
   (b) Each trader i computes $x_i \in \text{argmax}\{u_i(x) \mid x \in \mathcal{K}_i, \pi \cdot x \le \pi \cdot w_i\}$.
   (c) Compute the aggregate demand $X = \sum_i x_i$ and let $\sigma = \frac{1}{\max_j X_j}$ where X_j denotes the aggregate demand of good j.
   (d) Update for each good j: $p_j \leftarrow p_j\,(1 + \delta\sigma X_j)$.
3. Output for each i: $\overline{x}_i = \frac{\sum_{r=1}^N \sigma(r) x_i(r)}{\sum_{r=1}^N \sigma(r)}$ where $x_i(r)$ and $\sigma(r)$ are the values of x_i and σ in the rth iteration.
4. Output $\overline{\pi} = \frac{\sum_{r=1}^N \sigma(r)\pi(r)}{\sum_{r=1}^N \sigma(r)}$ where $\pi(r)$ and $\sigma(r)$ are the values of $\hat{\pi}$ and σ in the rth iteration.

Fig. 1. Algorithm for the convex program (2)

This is so because $(1 + \delta\sigma Z_j) \approx (1 + \delta\sigma X_j)(1 - \delta\sigma)$, which is in turn true since $Z_j = X_j - 1$ and $\delta\sigma$ is small. The extra factor $(1 - \delta\sigma)$ is common to all goods j and is factored away in the price scaling step. The algorithm in the end outputs, $\overline{\pi}$ and $\overline{x}_i$ for all i, the weighted average of the prices and allocations computed in N iterations.

Lemma 1. *The outputs $\overline{x}_i$ and $\overline{\pi}$ satisfy $u_i(\overline{x}_i) \ge \widetilde{u}_i(\overline{\pi}, \overline{\pi} \cdot w_i)$ for each i.*

Proof. Since $\widetilde{u}_i(\pi, \pi \cdot w_i)$ is convex when $\pi \in \Pi$ and $u_i(x_i)$ is concave when $x_i \in \mathcal{K}_i$, we have $\widetilde{u}_i(\overline{\pi}, \overline{\pi} \cdot w_i) \le \frac{\sum_r \sigma(r)\widetilde{u}_i(\pi(r), \pi(r)\cdot w_i)}{\sum_r \sigma(r)} = \frac{\sum_r \sigma(r) u_i(x_i(r))}{\sum_r \sigma(r)} \le u_i(\overline{x}_i)$.

The following main lemma about the output is proved below. The proof is based on the standard application of "experts theorem" or "multiplicative update" technique used previously in solving packing and covering linear programs [24, 12, 11].

Lemma 2. *The outputs $\overline{x}_i$ satisfy $\sum_i \overline{x}_i \le \frac{1}{1-2\delta} \sum_i w_i$.*

We set $\delta = \frac{\epsilon}{2(1+\epsilon)}$ so that $\frac{1}{1-2\delta} = 1 + \epsilon$. The proof of Theorem 7 now follows from Lemmas 1, 2, and Assumption 1 on the utility functions.

The main result of this section is summarized in the following theorem.

Theorem 7. *Our algorithm computes a weak $(1+\epsilon)$-approximate market equilibrium in an economy for which a set Π containing an equilibrium price is known such that for each i, the indirect utility function $\widetilde{u}_i(\pi, \pi \cdot w_i)$ is a convex function of π when restricted to $\pi \in \Pi$.*

In the algorithm, each trader i makes $O(\epsilon^{-2} n \log n)$ calls to her "demand" oracle: given prices $\pi \in \Pi$, compute $x_i \in$ argmax$\{u_i(x) \mid x \in \mathcal{K}_i, \pi \cdot x \le \pi \cdot w_i\}$.

Proof of Lemma 2. Let $\overline{x} = \sum_i \overline{x}_i$ and let $(\overline{x})_j$ denote the jth coordinate of $\overline{x}$. To this end, let us define a potential $\Phi(r) = \sum_j p_j(r)$ where $p_i(r)$ denote the value of p_i in the beginning of rth iteration. From the step 2d in the algorithm, we have

$$\Phi(r+1) = \Phi(r) + \delta\sigma(r)\sum_j p_j(r)X_j(r)$$

where $X_j(r)$ denotes the value of X_j in the rth iteration. Thus

$$\frac{\Phi(r+1)}{\Phi(r)} = 1 + \delta\sigma(r)\sum_j \frac{p_j(r)}{\Phi(r)}X_j(r) \;=\; 1 + \delta\sigma(r) \;\le\; \exp(\delta\sigma(r)).$$

The second equality follows from the fact that $\sum_j p_j(r)X_j(r) = \frac{1}{\alpha(r)}\sum_j \pi_j(r)X_j(r)$ which is, by Assumption 1, equal to $\frac{1}{\alpha(r)}\sum_j \pi_j(r)\sum_i w_{ij} = \frac{1}{\alpha(r)}\sum_j \pi_j(r) = \sum_j p_j(r) = \Phi(r)$. Here $\alpha(r)$ is the value of α in rth iteration.

Thus after taking telescoping product, we get

$$\Phi(N+1) \le \Phi(1)\cdot\exp\left(\delta\sum_r \sigma(r)\right) = n\cdot\exp\left(\delta\sum_r \sigma(r)\right). \tag{4}$$

On the other hand, observe that

$$\begin{aligned}\Phi(N+1) = \sum_j p_j(N+1) &= \sum_j \prod_{r=1}^{N}(1+\delta\sigma(r)X_j(r)) \\ &\ge \sum_j \exp\left(\delta(1-\delta)\sum_r \sigma(r)X_j(r)\right) \\ &\ge \max_j \exp\left(\delta(1-\delta)\sum_r \sigma(r)X_j(r)\right) \\ &= \exp\left(\delta(1-\delta)\max_j\sum_r \sigma(r)X_j(r)\right).\end{aligned}$$

The first inequality follows from the elementary fact that $1+\mu \ge \exp(\mu(1-\delta))$ for all $0<\mu<\delta<\frac{1}{2}$. Combining the above observation with (4), we get

$$\delta(1-\delta)\max_j\sum_r \sigma(r)X_j(r) \le \log\Phi(N+1) \le \log n + \delta\sum_r \sigma(r).$$

Therefore,

$$\max_j(\overline{x})_j \;=\; \max_j \frac{\sum_r \sigma(r)X_j(r)}{\sum_r \sigma(r)} \le \frac{1}{1-\delta} + \left(\frac{\log n}{\delta(1-\delta)\sum_r \sigma(r)}\right). \tag{5}$$

Now we "charge" the second term on the right-hand-side in (5) to $\max_j(\overline{x})_j$ as follows. Note that at least one p_j increases by a factor $(1+\delta)$ in any iteration. Thus after $N = \frac{n}{\delta}\log_{1+\delta} n$ iterations, $\max_j p_j(N+1) \ge n^{1/\delta}$. Also

$$(\overline{x})_j = \frac{\sum_r \sigma(r)X_j(r)}{\sum_r \sigma(r)} = \frac{\log\prod_r \exp(\delta\sigma(r)X_j(r))}{\delta\sum_r \sigma(r)} \ge \frac{\log p_j(N+1)}{\delta\sum_r \sigma(r)}.$$

Thus $\max_j(\overline{x})_j \geq \frac{\log n}{\delta^2 \sum_r \sigma(r)}$. Putting all pieces together, we get

$$\max_j(\overline{x})_j \leq \frac{1}{1-\delta} + \left(\frac{\delta \max_j(\overline{x})_j}{1-\delta}\right).$$

Thus $\max_j(\overline{x})_j \leq \frac{1}{1-2\delta}$.

5 Future Work

Our definitions of approximate market equilibrium is weak because the budget constraints of traders are satisfied only in the aggregate sense. Some of the traders may be spending significantly more than their budget. Moreover, some positively priced items may not be fully allocated. A notion of strongly approximate market equilibrium may be defined on the lines of [13], where budget constraints of no trader may exceed by a factor more than $(1+\epsilon)$ and no item with positive price is under-demanded. Under this definition it might be possible to prove the "closeness" of the discovered prices to the equilibrium prices (see e.g., [14]). If we set $\delta = O(\frac{\epsilon \min_i \lambda_i}{\sum_i \lambda_i})$, where λ_i is the income of trader i in a Fisher economy, our tâtonnement algorithm obtains a strong $(1+\epsilon)$ approximate market equilibrium in the above sense in $O((\frac{\epsilon \min_i \lambda_i}{\sum_i \lambda_i})^{-2} n \log n)$ iterations. It will be very interesting to develop a tâtonnement algorithm that converges to a strong approximate market equilibrium in near linear number of iterations. Finally, it is interesting to note that the continuous time version of our process can be described as $\frac{d\pi_j}{dt} = \pi_j Z_j$ where $Z_j = \sum_j x_{ij} - \sum_i w_{ij}$ is the excess demand of good j. Under what conditions is this process or its "time-average" $\hat{\pi}_j = \frac{1}{t}\int_{\tau=0}^{t} \pi_j d\tau$ stable and does converge to the equilibrium?

References

1. K. Arrow and G. Debreu. Existence of an Equilibrium for a Competitive Economy. *Econometrica*, 22:265–290, 1954.
2. K. J. Arrow, H. D. Block, and L. Hurwicz. On the stability of the competitive equilibrium, II. *Econometrica*, 27(1):82–109, 1959.
3. N. Chen, X. Deng, X. Sun, and A. Yao. Fisher Equilibrium Price with a class of Concave Utility Functions. In *ESA*, 2004.
4. B. Codenotti and K. Varadarajan. Computation of market equilibria by convex programming. In N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, editors, *Algorithmic Game Theory*. Cambridge University Press, 2007.
5. B. Codenotti, B. McCune, S.V. Pemmaraju, R. Raman, and K. Varadarajan. An experimental study of different approaches to solve the market equilibrium problem. In *ALENEX/ANALCO*, pages 167–179, 2005.
6. B. Codenotti, B. McCune, R. Raman, and K. Varadarajan. Computing equilibrium prices: Does theory meet practice? In *ESA*, 2005.
7. B. Codenotti, B. McCune, and K. Varadarajan. Market equilibrium via the excess demand function. In *STOC*, 2005.

8. R. Cole and L. Fleischer. Fast-converging tatonnement algorithms for the market problem. Technical Report TR2007-602, Dept. of Computer Science, Dartmouth College, 2007. Available at http://www.cs.dartmouth.edu/reports/.
9. X. Deng, C. Papadimitriou, and S. Safra. On the Complexity of Equilibria. In *STOC*, 2002.
10. N. Devanur, C. Papadimitriou, A. Saberi, and V. Vazirani. Market Equilibrium via a Primal-Dual-Type Algorithm. In *FOCS*, pages 389–395, 2002. Journal version to appear in the Journal of the ACM.
11. L. Fleischer. Approximating fractional multicommodity flow independent of the number of commodities. *SIAM J. Discrete Math.*, 13:505–520, 2000.
12. N. Garg and J. Könemann. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. In *FOCS*, pages 300–309, 1998.
13. R. Garg and S. Kapoor. Auction algorithms for market equilibrium. *Math. Oper. Res.*, 31(4):714–729, 2006.
14. R. Garg and S. Kapoor. Price roll-backs and path auctions: An approximation scheme for computing the market equilibrium. In *WINE*, pages 225–238, 2006.
15. R. Garg, S. Kapoor, and V. Vazirani. An Auction-Based Market Equilibrium Algorithm for the Separable Gross Substitutability Case. In *APPROX*, 2004.
16. M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag, Berlin, 1988.
17. K. Jain, M. Mahdian, and A. Saberi. Approximating Market Equilibrium. In *APPROX*, 2003.
18. K. Jain and V. Vazirani. Eisenberg-gale markets: Algorithms and structural properties. In *STOC*, 2007.
19. F.P. Kelly and V. Vazirani. Rate control as a market equilibrium. Manuscript, 2002.
20. A. Mas-Colell. *The Theory of General Economic Equilibrium: A Differential Approach*. Cambridge University Press, Cambridge, 1985.
21. T. Negishi. A note on the stability of an economy where all goods are gross substitutes. *Econometrica*, 26(3):445–447, 1958.
22. T. Negishi. The stability of a competitive economy: A survey article. *Econometrica*, 30(4):635–669, 1962.
23. H. Nikaido and H. Uzawa. Stability and non-negativity in a Walrasian process. *International Econ. Review*, 1:50–59, 1960.
24. S. Plotkin, D. Shmoys, and E. Tardos. Fast approximation algorithms for fractional packing and covering problems. *Math. Oper. Res.*, 20:257–301, 1995.
25. J. K.-H. Quah. The monotonicity of individual and market demand. *Econometrica*, 68(4):911–930, 2000.
26. P.A. Samuelson. Foundations of Economic Analysis. Harward University Press, Cambridge, Mass., 1947.
27. H. Varian. *Microeconomic Analysis*. W. W. Norton, New York, 1992.
28. V. Vazirani. Combinatorial algorithms for market equilibria. In N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, editors, *Algorithmic Game Theory*. Cambridge University Press, 2007.
29. L. Walras. Elements of Pure Economics, or the Theory of Social Wealth (in French). Lausanne, Paris, 1874.
30. Y. Ye. A path to the Arrow-Debreu competitive market equilibrium. *Mathematical Programming*, 2006.