Theoretical Computer Science 410 (2009) 1573–1580

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Market equilibria with hybrid linear-Leontief utilities

Xi Chen [a], Li-Sha Huang [a,*], Shang-Hua Teng [b,1]

[a] Institute for Theoretical Computer Science, Tsinghua University, Beijing, China

[b] Computer Science Department, Boston University, Boston, USA

ARTICLE INFO

Keywords:
Computational complexity
Market equilibrium
PPAD

ABSTRACT

We introduce a new family of utility functions for exchange markets. This family provides a natural and "continuous" hybridization of the traditional linear and Leontief utilities and might be useful in understanding the complexity of computing approximating market equilibria, although computing an equilibrium in a market with this family of utility functions, this is **PPAD**-hard in general. In this paper, we present an algorithm for finding an approximate Arrow–Debreu equilibrium when the Leontief components of the market are grouped, finite and well-conditioned.

1. Introduction

In recent years, the problem of computing and approximating market equilibria has attracted many researchers. In an exchange market, there is a set of traders and each trader comes with an initial endowment of commodities. They interact through some exchange process in order to maximize their own utility functions. In the state of an equilibrium, the traders can simply sell their initial endowments at a determined market price and buy commodities that maximize their utilities. Then, the market will clear — the price is so wisely set that the supplies exactly satisfy the demands. This price is called the *equilibrium price*.

Arrow and Debreu [1] proved the existence of equilibrium prices under some mild conditions. Since then, efficient algorithms have been developed for various families of utility functions.

1.1. From linear to Leontief utilities

Two popular families of utility functions are the *linear* and *Leontief* utilities. Both of them can be specified by an $m \times n$ demand matrix $\mathbf{D} = (d_{i,j})$, in an exchange market with m goods and n traders. If trader j, where $1 \le j \le n$, receives a bundle of goods $\mathbf{x}_j$, then its linear utility is $u_j(\mathbf{x}_j) = \sum_i x_{i,j}/d_{i,j}$, while its Leontief utility is $u_j(\mathbf{x}_j) = \min_i(x_{i,j}/d_{i,j})$. Both linear and Leontief utilities are members of a larger family of utility functions, referred to as *CES* utilities.

Although the two families of functions look similar, the complexities of computing market equilibria in these two settings might be very different. In the linear case, a market equilibrium can be approximated and computed in polynomial time, thanks to a collection of great algorithmic results by Nenakhov and Primak [12], Devanur, Papadimitriou, Saberi and Vazirani [6], Jain, Mahdian and Saberi [10], Garg and Kapoor [7], Jain [9], and Ye [13].

However, approximating market equilibria with Leontief utilities has proven to be hard, under some reasonable complexity assumptions. In particular, by analyzing a reduction of Codenotti, Saberi, Varadarajan and Ye [4] from Nash

* Corresponding author.
E-mail addresses: csxichen@gmail.com (X. Chen), huangstc@gmail.com (L.-S. Huang), steng@cs.bu.edu (S.-H. Teng).

1 Also affiliated with Akamai Technologies Inc. Cambridge, MA, USA.

0304-3975/$ – see front matter
doi:10.1016/j.tcs.2008.12.030

equilibria to market equilibria, Huang and Teng [8] showed that approximating Leontief market equilibria is as hard as approximating Nash equilibria of general two-player games. Thus by a recent result of Chen, Deng and Teng [2], it is **PPAD**-hard to approximate a Leontief market equilibrium in fully polynomial time. In fact, the smoothed complexity of finding a Leontief market equilibrium cannot be polynomial, unless **PPAD** $\subset$ **RP**.

1.2. Hybrid linear-Leontief utilities and our results

In this paper, we introduce a new family of utility functions, and study the computation and approximation of equilibria in exchange markets with these utilities. Our work is partially motivated by the complexity discrepancy between linear and Leontief utilities. In our market model, each trader's utility function is a linear combination of a collection of Leontief utility functions. We parameterize such a utility function by the maximum number of commodities in its Leontief components. If the number of commodities in any of its Leontief components is at most k, we refer to it as a *k-wide linear-Leontief function*.

Intuitively, the new utility function combines an "easy" linear function with several "hard" Leontief utility functions. Clearly, a 1-wide linear-Leontief function is a linear function and thus, a market with 1-wide linear-Leontief utilities can be solved in polynomial time. On the other hand, for markets with general linear-Leontief utilities, finding an equilibrium is **PPAD**-hard.

We further focus on *grouped hybridizations* in which the commodities of the exchange market are divided into groups. For any of these groups, each trader has a Leontief utility function over the commodities in the group. A trader's utility function is then the summation of all its Leontief utilities. If each group has at most k commodities, we refer to the utility functions (in the exchange market) as *grouped k-wide linear-Leontief functions*.

An exchange market with grouped linear-Leontief functions can be viewed as a linear combination of several smaller Leontief markets, one for each group of commodities. In an equilibrium, the supplies exactly satisfy the demands in each of these Leontief markets. However, a trader can invest the surplus it earned from one Leontief market to other Leontief markets.

We present two algorithmic results on the computation and approximation of equilibria in markets with hybrid linear-Leontief utilities:

- In Fisher's model, we show that an equilibrium of an exchange market with n traders, M commodities and hybrid linear-Leontief utility functions can be computed in $O(\sqrt{Mn}(M+n)^3L)$ time, where L is the bit-length of the input data.
- We also present an algorithm for finding an approximate equilibrium in a *well-conditioned* Arrow–Debreu market with grouped linear-Leontief functions (see Section 4 for details). While the upper bound that we can prove is exponential, the time complexity of this algorithm is closely related to an interesting sampling problem (see Section 4.2 for details).

In this paper, we only prove the first result for grouped linear-Leontief utilities. It is easy to extend the proof to the general case. In the Arrow–Debreu model, we notice that, due to a recent result of Chen, Deng and Teng [3] on the complexity of sparse two-player games, approximating equilibria in fully polynomial-time is **PPAD**-hard even for Arrow–Debreu markets with 10-*wide* linear-Leontief utilities.

1.3. Notations

We will use bold lower-case Roman letters such as $\mathbf{x}$, $\mathbf{a}$, $\mathbf{b}_j$ to denote vectors. Whenever a vector, say $\mathbf{a} \in \mathbb{R}^n$ is present, its components will be denoted by lower-case Roman letters with subscripts, such as $a_1, a_2, \ldots, a_n$. Matrices are denoted by bold upper-case Roman letters such as $\mathbf{A}$ and scalars are usually denoted by lower-case Roman letters. We will also use the following notation in the paper:

- $\mathbb{R}^m_+$: the set of m-dimensional vectors with non-negative real entries;
- $\mathbb{P}^n$: the set of vectors $\mathbf{x} \in \mathbb{R}^n_+$ with $\sum_{i=1}^n x_i = 1$;
- $\langle \mathbf{a}|\mathbf{b}\rangle$: the dot-product of two vectors in the same dimension;
- $\|\mathbf{x}\|_p$: the p-norm of vector $\mathbf{x}$, that is, $(\sum_i |x_i|^p)^{1/p}$; and $\|\mathbf{x}\|_\infty = \max_i |x_i|$.

2. Grouped linear-Leontief markets

Assume there are n traders in the market, denoted by $\mathbf{T} = \{1, \ldots, n-1, n\}$. The market has m groups of commodities, denoted by $\mathbf{G} = \{G_1, \ldots, G_m\}$, and each group G_j contains k_j kinds of commodities.

Trader i's initial endowment of goods is a collection of m vectors: $\{\mathbf{e}^i_j \in \mathbb{R}^{k_j}_+, 1 \le j \le m\}$, where $e^i_{j,k}$ is the amount of good k in group j held by trader i. For each group j, we use matrix $\mathbf{E}_j = (\mathbf{e}^1_j, \ldots, \mathbf{e}^n_j)$ to denote the traders' initial endowments in this group. We also assume that the amount of each commodity is normalized to 1, i.e., $\sum_{i=1}^n e^i_{j,k} = 1$ for all $j : 1 \le j \le m$ and $k : 1 \le k \le k_j$. Similarly, the allocation to trader i is also a collection of m vectors, denoted by $\mathbf{x}^i = \{\mathbf{x}^i_j \in \mathbb{R}^{k_j}_+, 1 \le j \le m\}$.

Trader i's utility function u_i is described by vectors $\{\mathbf{d}^i_j \in \mathbb{R}^{k_j}_+, 1 \le j \le m\}$ and $\mathbf{a}^i \in \mathbb{R}^m_+$: given an allocation $\mathbf{x}^i = \{\mathbf{x}^i_j, 1 \le j \le m\}$, we have

$$u_i(\mathbf{x}^i) = \sum_{j=1}^{m} a_j^i v_j^i, \quad \text{where } v_j^i = \min_{1\le k\le k_j} \{x_{j,k}^i / d_{j,k}^i\}. \tag{1}$$

In other words, u_i is a linear combination of m Leontief functions. Locally, we have a Leontief market for each group j, in which trader i demands the goods in proportion to vector $\mathbf{d}_j^i$. For each group j, we use matrix $\mathbf{D}_j = (\mathbf{d}_j^1, \ldots, \mathbf{d}_j^n)$ to denote the traders' demands in this group.

Let $\mathbf{D} = (\mathbf{D}_1, \ldots, \mathbf{D}_m)$, $\mathbf{E} = (\mathbf{E}_1, \ldots, \mathbf{E}_m)$, and $\mathbf{A} = (\mathbf{a}^1, \ldots, \mathbf{a}^n)$, then the market is a tuple $\mathbf{M} = (\mathbf{T}, \mathbf{G}, \mathbf{D}, \mathbf{E}, \mathbf{A})$. Now we define the exchange equilibrium and approximate equilibrium in this market model.

Let $\mathbf{x}^1, \ldots, \mathbf{x}^n$ be an allocation of the market, then we use $\mathbf{v}_j = (v_j^1, v_j^2, \ldots, v_j^n)^\top$ to denote the n-dimensional column vector whose ith component v_j^i is defined in Eq. (1). We also use $\mathbf{v}$ to denote $\{\mathbf{v}_j \in \mathbb{R}_+^n, 1 \le j \le m\}$.

Definition 1 (*Exchange Equilibrium*). An exchange equilibrium of market $\mathbf{M} = (\mathbf{T}, \mathbf{G}, \mathbf{D}, \mathbf{E}, \mathbf{A})$ is a pair $(\mathbf{p}, \mathbf{v})$, where $\mathbf{p} = \{\mathbf{p}_j \in \mathbb{R}_+^{k_j}, 1 \le j \le m\}$ is a collection of m price vectors and $\mathbf{v} = \{\mathbf{v}_j \in \mathbb{R}_+^n, 1 \le j \le m\}$, such that

$$\begin{cases} u_i = \sum_{j=1}^{m} a_j^i v_j^i, & \forall i = 1, \ldots, n \\ u_i = \max \left\{ \sum_{j=1}^{m} a_j^i z_j^i \mid \sum_{j=1}^{m} \langle \mathbf{p}_j | \mathbf{d}_j^i \rangle z_j^i \le \sum_{j=1}^{m} \langle \mathbf{p}_j | \mathbf{e}_j^i \rangle \right\}, & \forall i = 1, \ldots, n \\ \mathbf{D}_j \mathbf{v}_j \le \vec{\mathbf{1}}, & \forall j = 1, \ldots, m. \end{cases}$$

It is easy to see that, given any exchange equilibrium $(\mathbf{p}, \mathbf{v})$, one can find an allocation $\mathbf{x}^1, \ldots, \mathbf{x}^n$ of the market efficiently, such that, every trader's utility is maximized and the market clears.

Definition 2 (*ε-Approximate Equilibrium*). An ε-approximate equilibrium of market $\mathbf{M} = (\mathbf{T}, \mathbf{G}, \mathbf{D}, \mathbf{E}, \mathbf{A})$ is a pair $(\mathbf{p}, \mathbf{v})$ such that

$$\begin{cases} u_i = \sum_{j=1}^{m} a_j^i v_j^i, & \forall i \\ u_i \ge (1-\varepsilon) \max \left\{ \sum_{j=1}^{m} a_j^i z_j^i \mid \sum_{j=1}^{m} \langle \mathbf{p}_j | \mathbf{d}_j^i \rangle z_j^i \le \sum_{j=1}^{m} \langle \mathbf{p}_j | \mathbf{e}_j^i \rangle \right\}, & \forall i \\ \mathbf{D}_j \mathbf{v}_j \le (1+\varepsilon) \vec{\mathbf{1}}, & \forall j. \end{cases}$$

3. An equivalent equilibrium characterization

In this section, we first prove a theorem that gives a necessary and sufficient condition for $(\mathbf{p}, \mathbf{v})$ being an exchange equilibrium. Then we present the algorithm for Fisher's model, whose definition will be presented later.

Theorem 3. *A pair $(\mathbf{p}, \mathbf{v})$ is an equilibrium if and only if it satisfies*

$$\begin{cases} \mathbf{D}_j \mathbf{v}_j \le \vec{\mathbf{1}}, & \forall j \\ u_i = \sum_{j=1}^{m} a_j^i v_j^i, & \forall i \\ w_i = \sum_{j=1}^{m} \langle \mathbf{p}_j | \mathbf{e}_j^i \rangle, & \forall i \\ w_i a_j^i \le u_i \langle \mathbf{p}_j | \mathbf{d}_j^i \rangle, & \forall i, j. \end{cases} \tag{2}$$

Proof. For each trader i, $(\mathbf{p}, \mathbf{v})$ maximizes its utility if and only if

$$\begin{cases} u_i = \sum_{j=1}^{m} a_j^i v_j^i, \qquad w_i = \sum_{j=1}^{m} \langle \mathbf{p}_j | \mathbf{e}_j^i \rangle, \\ \dfrac{a_j^i}{\langle \mathbf{p}_j | \mathbf{d}_j^i \rangle} \le \dfrac{u_i}{w_i}, & \forall j \\ \dfrac{a_j^i}{\langle \mathbf{p}_j | \mathbf{d}_j^i \rangle} v_j^i = \dfrac{u_i}{w_i} v_j^i, & \forall j. \end{cases}$$

Therefore, $(\mathbf{p}, \mathbf{v})$ is an equilibrium of $\mathbf{M}$ if and only if

$$\begin{cases} \mathbf{D}_j \mathbf{v}_j \le \vec{\mathbf{1}}, & \forall j \\ u_i = \sum_{j=1}^{m} a_j^i v_j^i, & \forall i \\ w_i = \sum_{j=1}^{m} \langle \mathbf{p}_j | \mathbf{e}_j^i \rangle, & \forall i \\ w_i a_j^i \le u_i \langle \mathbf{p}_j | \mathbf{d}_j^i \rangle, & \forall i, j \\ w_i a_j^i v_j^i = u_i \langle \mathbf{p}_j | \mathbf{d}_j^i \rangle v_j^i, & \forall i, j. \end{cases}$$

To prove the theorem, it suffices to show that the last condition can be derived from the other four. By $w_i a_j^i \le u_i \langle \mathbf{p}_j | \mathbf{d}_j^i \rangle$, we have $w_i a_j^i v_j^i \le u_i \langle \mathbf{d}_j^i | \mathbf{p}_j \rangle v_j^i$ and

$$\begin{aligned} & w_i \sum_{j=1}^{m} a_j^i v_j^i \le u_i \sum_{j=1}^{m} \langle \mathbf{d}_j^i | \mathbf{p}_j \rangle v_j^i, \quad \forall i \\ \Rightarrow \quad & \sum_{j=1}^{m} \langle \mathbf{e}_j^i | \mathbf{p}_j \rangle = w_i \le \sum_{j=1}^{m} \langle \mathbf{d}_j^i | \mathbf{p}_j \rangle v_j^i, \quad \forall i \\ \Rightarrow \quad & \sum_{i=1}^{n} \sum_{j=1}^{m} \langle \mathbf{e}_j^i | \mathbf{p}_j \rangle \le \sum_{i=1}^{n} \sum_{j=1}^{m} \langle \mathbf{d}_j^i v_j^i | \mathbf{p}_j \rangle \\ \Rightarrow \quad & \sum_{j=1}^{m} \langle \vec{\mathbf{1}} | \mathbf{p}_j \rangle \le \sum_{j=1}^{m} \left\langle \mathbf{D}_j \mathbf{v}_j | \mathbf{p}_j \right\rangle. \end{aligned}$$

On the other hand, since $\mathbf{D}_j \mathbf{v}_j \le \vec{\mathbf{1}}$ for all j, we have $\langle \mathbf{D}_j \mathbf{v}_j | \mathbf{p}_j \rangle = \langle \vec{\mathbf{1}} | \mathbf{p}_j \rangle$ for all j. This forces $w_i a_j^i v_j^i = u_i \langle \mathbf{d}_j^i | \mathbf{p}_j \rangle v_j^i$ for all i, j. ■

3.1. Solving Fisher's model

Fisher's model is a special case of the Arrow–Debreu model. In Fisher's model, the commodities are initially held by a single seller, and all other traders come to the market with money. Each trader buys goods from the seller to maximize its utility, under the budget constraint. The market is in an equilibrium if the supplies satisfy the demands. Usually, the computation of equilibria in Fisher's setting is much easier than that in the general case. In fact, as shown in [11], one can find an approximate equilibrium of the market by solving the following convex program:

$$\begin{aligned} & \max \sum_{i=1}^{n} w_i \log(u_i) \\ \text{s.t.} \quad & \begin{cases} u_i = \sum_{j=1}^{m} a_j^i v_j^i, & \forall i = 1, \ldots, n \\ \mathbf{D}_j \mathbf{v}_j \le \vec{\mathbf{1}}, & \forall j = 1, \ldots, m \\ \mathbf{v}_j \ge 0, & \forall j = 1, \ldots, m \end{cases} \end{aligned} \tag{3}$$

where we assume trader i comes to the market with w_i dollars. With the same argument as in [13], we can prove the following theorem:

Theorem 4 (*Fisher's Equilibrium*). *Fisher's model can be solved using the interior-point algorithm in $O(\sqrt{Mn}(M+n)^3 L)$ time, where $M = \sum_j k_j$ is the number of commodities, n is the number of traders, and L is the bit-length of the input data.*

4. An approximation algorithm for Arrow–Debreu markets

Since Leontief economy is a special case of the hybrid linear-Leontief economy, the hardness results [4,5,8] for Leontief economy also apply to our setting. For example, it is NP-hard to determine the existence of equilibria [4], and there is no algorithm to compute an equilibrium in smoothed polynomial time, unless **PPAD** $\subset$ **RP** [8]. In this section, we present an approximation algorithm for *well-conditioned* grouped linear-Leontief markets. We say a market $\mathbf{M} = (\mathbf{T}, \mathbf{G}, \mathbf{D}, \mathbf{E}, \mathbf{A})$ is well-conditioned, if $\tau = \min_{i,j,k} \{e_{j,k}^i, d_{j,k}^i\} > 0$. The time complexity of our algorithm is

$$\min \left\{ O\left(\left(\frac{1}{\tau \varepsilon} \right)^{M-m} \text{poly}(M, n) \right), O\left(\left(\frac{\log(1/\tau)}{\varepsilon} \right)^{2mn} \text{poly}(M, n) \right) \right\},$$

where $M = \sum_{j=1}^{m} k_j$ is the number of commodities.

4.1. Intuition

Assume the market is $\mathbf{M} = (\mathbf{T}, \mathbf{G}, \mathbf{D}, \mathbf{E}, \mathbf{A})$ and $(\mathbf{p}, \mathbf{v})$ is one of its equilibria. In the following discussion, it is convenient to replace each $\mathbf{p}_j$ with $q_j\mathbf{p}_j$, where $q_j \in \mathbb{R}_+$ and $\|\mathbf{p}_j\|_1 = 1$ is a normalized vector, and rewrite equilibrium $(\mathbf{p}, \mathbf{v})$ as $(\mathbf{q}, \mathbf{p}, \mathbf{v})$, where $\mathbf{q} \in \mathbb{R}_+^m$. We call $\mathbf{p}$ an equilibrium internal price of $\mathbf{M}$.

As we mentioned earlier, market $\mathbf{M}$ can be viewed as a linear combination of several Leontief markets. Therefore, it is not surprising that given an equilibrium $(\mathbf{q}, \mathbf{p}, \mathbf{v})$ of $\mathbf{M}$, we can construct a new linear market $\overline{\mathbf{M}}$ such that $\mathbf{q}$ is the price vector of an equilibrium of $\overline{\mathbf{M}}$.

Given $(\mathbf{q}, \mathbf{p}, \mathbf{v})$, we build $\overline{\mathbf{M}}$ with linear utilities as follows. The set of traders is the same as $\mathbf{M}$. We introduce a commodity g_j for each group $G_j \in \mathbf{G}$ and thus, $\overline{\mathbf{M}}$ has m types of commodities $g_1, \ldots, g_m$. Trader i's initial endowment of commodity g_j is $\bar{e}_j^i = \langle \mathbf{e}_j^i | \mathbf{p}_j \rangle$, and its preference to commodity g_j is $\bar{a}_j^i = a_j^i / \langle \mathbf{d}_j^i | \mathbf{p}_j \rangle$. The lemma below follows directly from the definition of equilibria.

Lemma 5 (*Market Reduction*). *Let* $\bar{x}_j^i = v_j^i \langle \mathbf{d}_j^i | \mathbf{p}_j \rangle$ *for all* $1 \le i \le n$ *and* $1 \le j \le m$, *then* $(\mathbf{q}, \bar{\mathbf{x}})$ *is an equilibrium of the linear market* $\overline{\mathbf{M}}$.

Lemma 5 shows that if we are so lucky that we know the *internal* price $\mathbf{p}_j$ of each group G_j, then the hybrid market $\mathbf{M}$ can be somehow reduced to a linear market $\overline{\mathbf{M}}$: By computing an equilibrium $(\mathbf{q}, \bar{\mathbf{x}})$ of $\overline{\mathbf{M}}$ (which can be done in polynomial time, since $\overline{\mathbf{M}}$ is a linear market), we get an equilibrium $(\mathbf{q}, \mathbf{p}, \mathbf{v})$ of the hybrid market $\mathbf{M}$, where $v_j^i = \bar{x}_j^i / \langle \mathbf{d}_j^i | \mathbf{p}_j \rangle$ for all $1 \le i \le n$, $1 \le j \le m$.

This observation leads us to the following approximation algorithm. First, we introduce a discrete set $S_j \subset \mathbb{P}^{k_j}$, for each group G_j, to approximately *cover* the simplex $\mathbb{P}^{k_j}$. More exactly, S_j is a dense subset of $\mathbb{P}^{k_j}$ in the sense that, for any $\mathbf{p}_j \in \mathbb{P}^{k_j}$, there exists a $\bar{\mathbf{p}}_j \in S_j$ such that $\mathbf{p}_j$ and $\bar{\mathbf{p}}_j$ are close enough (for the desired accuracy). Then, we exhaustively *enumerate* all the internal prices $\bar{\mathbf{p}} = \{\bar{\mathbf{p}}_j, 1 \le j \le m\}$ in $S_1 \times \cdots \times S_m$. For each $\bar{\mathbf{p}}$, we reduce $\mathbf{M}$ to a linear market $\overline{\mathbf{M}}$ using $\bar{\mathbf{p}}$ (as described above). Finally, we check whether there is an equilibrium $(\mathbf{q}, \bar{\mathbf{x}})$ of $\overline{\mathbf{M}}$, which can be combined with $\bar{\mathbf{p}}$ to produce an approximate equilibrium of the original market $\mathbf{M}$.

The time complexity of this algorithm clearly depends on the size of the *sampling* set $S_1 \times \cdots \times S_m$, so the construction of sets S_j is very important. We address this problem in the following subsection.

4.2. The efficient sampling problem

The Efficient Sampling Problem: Given $\varepsilon > 0$ and n points $\{\mathbf{x}_i \in \mathbb{P}^k, i = 1, \ldots, n\}$, called *anchor points*, find a subset $S \subseteq \mathbb{P}^k$ such that for any $\mathbf{p} \in \mathbb{P}^k$, there exists a point $\bar{\mathbf{p}} \in S$ satisfying $1 - \varepsilon \le \langle \mathbf{p} | \mathbf{x}_i \rangle / \langle \bar{\mathbf{p}} | \mathbf{x}_i \rangle \le 1 + \varepsilon$ for all anchor points $\mathbf{x}_i$, $1 \le i \le n$. Such a set S is called an *efficient sampling set* of $\{\mathbf{x}_i, i = 1, \ldots, n\}$ with accuracy ε. The goal is to minimize the size of S.

We now give two constructions for S.

Lemma 6. *If* $\tau = \min_{i,j}\{x_{i,j}\} > 0$, *then we can build an efficient sampling set* S *of size* $O((\log(1/\tau)/\varepsilon)^n)$.

Proof. For any $\mathbf{p} \in \mathbb{P}^k$, it is easy to see that $\tau \le \langle \mathbf{p} | \mathbf{x}_i \rangle \le 1$. To construct S, we first introduce the following $\log(1/\tau)/\log(1+\varepsilon) \approx \log(1/\tau)/\varepsilon$ planes:

$$\begin{cases} a_0 = \tau, & \text{plane}_0 = \{\mathbf{y}, \langle \mathbf{y} | \mathbf{x}_1 \rangle = a_0\}; \\ a_i = (1+\varepsilon)a_{i-1}, & \text{plane}_i = \{\mathbf{y}, \langle \mathbf{y} | \mathbf{x}_1 \rangle = a_i\}. \end{cases}$$

These planes cut $\mathbb{P}^k$ into $O(\log(1/\tau)/\varepsilon)$ polytopes $(P_1, P_2, \ldots)$.

Then, we define $O(\log(1/\tau)/\varepsilon)$ planes for $\mathbf{x}_2$ similarly. They further cut each P_i into at most $O(\log(1/\tau)/\varepsilon)$ smaller polytopes (denoted by $P_{i,0}, P_{i,1}, \ldots$).

We repeat this process for vectors $\mathbf{x}_3, \mathbf{x}_4, \ldots, \mathbf{x}_n$ and finally, $\mathbb{P}^k$ is divided into $O((\log(1/\tau)/\varepsilon)^n)$ polytopes. The sampling set S is then constructed by picking an inner point from each of these polytopes. ■

Lemma 7. *If* $\tau = \min_{i,j}\{x_{i,j}\} > 0$, *then we can build an efficient sampling set* S *of size* $O((\tau\varepsilon)^{1-k})$.

Proof. The sampling set S is constructed by meshing the simplex $\mathbb{P}^k$, such that for any $\mathbf{p} \in \mathbb{P}^k$, there exists a $\bar{\mathbf{p}} \in S$ satisfying $\|\mathbf{p} - \bar{\mathbf{p}}\|_\infty \le \varepsilon\tau$. It is easy to show that S is an efficient sampling set of size $O((\tau\varepsilon)^{1-k})$. ■

In the algorithm, as we will see, we need to build an efficient sampling set S_j for $\{\mathbf{e}_j^i, i = 1, \ldots, n\} \bigcup \{\mathbf{d}_j^i, i = 1, \ldots, n\}$ with accuracy ε. Let S be $S_1 \times \cdots \times S_m$, then the time complexity of the algorithm is dominated by $|S|$.

4.3. A convex optimization problem

Now suppose $\mathbf{p}^*$ is an equilibrium internal price of market $\mathbf{M}$ and $\bar{\mathbf{p}}$ is a vector in $S = S_1 \times \cdots \times S_m$ such that

$$1 - \varepsilon \le \frac{\langle \mathbf{p}_j^* | \mathbf{d}_j^i \rangle}{\langle \bar{\mathbf{p}}_j | \mathbf{d}_j^i \rangle} \le 1 + \varepsilon \quad \text{and} \quad 1 - \varepsilon \le \frac{\langle \mathbf{p}_j^* | \mathbf{e}_j^i \rangle}{\langle \bar{\mathbf{p}}_j | \mathbf{e}_j^i \rangle} \le 1 + \varepsilon \tag{4}$$

for all $1 \le i \le n$ and $1 \le j \le m$. In this subsection, we show how to use $\bar{\mathbf{p}}$ to compute an approximate equilibrium of $\mathbf{M}$.

Recall the equivalent equilibrium condition stated in Theorem 3. After replacing $(\mathbf{p}, \mathbf{v})$ with $(\mathbf{q}, \mathbf{p}, \mathbf{v})$, we get the following convex optimization problem:

$$\begin{aligned} \min \quad & \theta \\ \text{s.t.} \quad & \mathbf{D}_j \mathbf{v}_j \le (1+\theta)\vec{\mathbf{1}}, && \forall j \\ & u_i = \sum_{j=1}^{m} a_j^i v_j^i, && \forall i \\ & w_i = \sum_{j=1}^{m} q_j \langle \mathbf{p}_j | \mathbf{e}_j^i \rangle, && \forall i \\ & w_i a_j^i \le u_i q_j \langle \mathbf{p}_j | \mathbf{d}_j^i \rangle, && \forall i, j \\ & \langle \mathbf{p}_j | \vec{\mathbf{1}} \rangle = 1,\ \mathbf{q} > 0, && \forall j. \end{aligned} \tag{5}$$

The variable θ can be viewed as the surplus of the demands in the market $\mathbf{M}$. We can prove that θ is always nonnegative in any feasible solution to problem (5). The proof is similar to the one in Ye [13], and is omitted here.

Lemma 8. *For any feasible solution $(\mathbf{q}, \mathbf{p}, \mathbf{v})$ of (5), we have $\theta \ge 0$. Moreover, $(\mathbf{q}, \mathbf{p}, \mathbf{v})$ is an equilibrium if and only if $\theta = 0$.*

Now suppose we are given an internal price $\overline{\mathbf{p}} = \{\overline{\mathbf{p}}_j, 1 \le j \le m\} \in S$. We can reduce (5) to the following convex optimization problem, denoted by $\text{Opt}(\overline{\mathbf{p}})$:

$$\begin{aligned} \min \quad & \theta \\ \text{s.t.} \quad & \mathbf{D}_j \mathbf{v}_j \le (1+\theta)\vec{\mathbf{1}}, && \forall j \\ & u_i = \sum_{j=1}^{m} a_j^i v_j^i, && \forall i \\ & w_i = \sum_{j=1}^{m} q_j \langle \overline{\mathbf{p}}_j | \mathbf{e}_j^i \rangle, && \forall i \\ & w_i a_j^i \le u_i q_j \langle \overline{\mathbf{p}}_j | \mathbf{d}_j^i \rangle, && \forall i, j \\ & \mathbf{q} > 0, && \forall j. \end{aligned} \tag{6}$$

$\text{Opt}(\overline{\mathbf{p}})$ can be solved in polynomial time [13]. Note that since $\overline{\mathbf{p}}$ may not be an equilibrium internal price of $\mathbf{M}$, the optimum of $\text{Opt}(\overline{\mathbf{p}})$ may not be zero. However, we prove in the following lemma that, if $\overline{\mathbf{p}}$ is *close* to an equilibrium internal price $\mathbf{p}^*$, then the optimum of $\text{Opt}(\overline{\mathbf{p}})$ is small.

Lemma 9. *Suppose $(\mathbf{q}^*, \mathbf{p}^*, \mathbf{v}^*)$ is an equilibrium of market $\mathbf{M}$, and $\varepsilon \le 1/3$. If $\overline{\mathbf{p}}$ satisfies (4) for all i, j, then the optimum of $\text{Opt}(\overline{\mathbf{p}})$ is at most 3ε.*

Proof. Since $(\mathbf{q}^*, \mathbf{p}^*, \mathbf{v}^*)$ is an equilibrium, it satisfies

$$\begin{cases} \mathbf{D}_j \mathbf{v}_j^* \le \vec{\mathbf{1}}, & \forall j \\ u_i^* = \sum_{j=1}^{m} a_j^i v_j^{i*}, & \forall i \\ w_i^* = \sum_{j=1}^{m} q_j^* \langle \mathbf{p}_j^* | \mathbf{e}_j^i \rangle, & \forall i \\ \lambda_i^* = w_i^* / u_i^*, & \forall i \\ \lambda_i^* = \min_{1 \le j \le m} \left\{ q_j^* \langle \mathbf{p}_j^* | \mathbf{d}_j^i \rangle \big/ a_j^i \right\}, & \forall i. \end{cases}$$

We now use $(\mathbf{q}^*, \mathbf{p}^*, \mathbf{v}^*)$ to construct a feasible solution to (6) as follows:

$$\begin{cases} q_j = q_j^*, & \forall j \\ \lambda_i = \min_{1 \le j \le m} \left\{ q_j \langle \overline{\mathbf{p}}_j | \mathbf{d}_j^i \rangle \big/ a_j^i \right\}, & \forall i \\ w_i = \sum_{j=1}^{m} q_j \langle \overline{\mathbf{p}}_j | \mathbf{e}_j^i \rangle, & \forall i \\ u_i = w_i / \lambda_i, & \forall i \\ v_j^i = v_j^{i*} \frac{u_i}{u_i^*}, & \forall i, j. \end{cases}$$

Using (4), we have $\lambda_i / \lambda_i^* \ge 1 - \varepsilon$ and $w_i / w_i^* \le 1 + \varepsilon$. As a result, we have

$$\frac{u_i}{u_i^*} = \frac{w_i}{w_i^*} \frac{\lambda_i^*}{\lambda_i} \le \frac{1+\varepsilon}{1-\varepsilon} \le 1 + 3\varepsilon,$$

since we assumed $\varepsilon \le 1/3$. Therefore, $\mathbf{D}_j \mathbf{v}_j \le \mathbf{D}_j \mathbf{v}_j^* (1 + 3\varepsilon) \le (1 + 3\varepsilon)\vec{\mathbf{1}}$, and the optimum of problem (6) is at most 3ε. ■

for *each group* G_j **do**
  Construct an efficient sampling set S_j for
  $\{\mathbf{e}_j^i, i = 1, ..., n\} \cup \{\mathbf{d}_j^i, i = 1, ..., n\}$ with accuracy $\varepsilon/3$
end
Let $S = S_1 \times ... \times S_m$
for *each internal price* $\overline{\mathbf{p}} \in S$ **do**
  Solve the convex optimization problem $\mathrm{Opt}(\overline{\mathbf{p}})$
  If the optimum $\overline{\theta} \le \varepsilon$, then break the loop and output $(\mathbf{q}, \overline{\mathbf{p}}, \mathbf{v})$
end

Fig. 1. An approximation algorithm for well-conditioned Arrow–Debreu markets.

4.4. The algorithm

Finally, our algorithm is described in Fig. 1. The construction of $S = S_1 \times \cdots \times S_m$ guarantees that, if $\mathbf{M}$ has an equilibrium $(\mathbf{q}^*, \mathbf{p}^*, \mathbf{v}^*)$, then there must exist a $\overline{\mathbf{p}} \in S$ that satisfies (4). By Lemma 9, the optimum of problem $\mathrm{Opt}(\overline{\mathbf{p}})$ is at most ε and thus, the algorithm outputs an ε-approximate equilibrium.

The time complexity of our algorithm is $|S| \cdot \mathrm{poly}(M, n)$, where $\mathrm{poly}(M, n)$ is spent on solving each optimization problem $\mathrm{Opt}(\overline{\mathbf{p}})$, and $M = \sum_{j=1}^m k_j$ is the total number of commodities. By Lemmas 6 and 7, we can bound the size of S by

$$\min \left\{ O\left((\tau\varepsilon)^{-M+m}\right), O\left((\log(1/\tau)/\varepsilon)^{2mn}\right) \right\},$$

where $\tau = \min_{i,j,k}\{e_{j,k}^i, d_{j,k}^i\} > 0$.

5. Discussion

In this paper, we introduce a new family of utility functions — hybrid linear-Leontief functions. We study the computation and approximation of exchange equilibria in markets with grouped linear-Leontief utilities, which are special cases of the hybrid ones. We show that, in Fisher's model, an equilibrium can be found in polynomial time. We also develop an algorithm for approximating market equilibria in the Arrow–Debreu model. The time complexity of our algorithm is closely related to the efficient sampling problem discussed in Section 4.2. At this moment, it is exponential in either the number of commodities or the number of traders. But any improvement to the construction of efficient sampling sets will improve the performance of our algorithm.

As a grouped hybrid market can be viewed as a linear combination of Leontief economies, given the fact that linear markets are easy to solve [12,9,13], we conjecture that there exists an approximation algorithm that runs in time polynomial in the number of groups and the number of traders, with access to an oracle that can compute equilibria in Leontief economies.

More generally, we can extend the concept of hybrid linear-Leontief utility functions to *hierarchical* linear-Leontief utility functions. Such a function can be specified by a tree whose internal vertices are either $+$ or *max* operators. Each of its leaves is associated with one commodity. Given an allocation vector, one can evaluate the utility function from bottom up.

Clearly, we can use this family of hierarchical utility functions to characterize more complicated market behaviors. With the same technique used in Section 3.1, a market equilibrium in Fisher's setting can be computed efficiently. We hope the study of these utilities could lead us to a better understanding of the complexity of computing market equilibria.

Acknowledgements

The first and second authors were supported by the Chinese National Key Foundation Plan (2003CB317807 and 2004CB318108), the National Natural Science Foundation of China Grant 60553001 and the National Basic Research Program of China Grant (2007CB807900 and 2007CB807901). The third author was partially supported by NSF grants CCR-0311430 and ITR CCR-0325630. Part of this work done while visiting Tsinghua University and Microsoft Research Asia Lab.

References

[1] J. Arrow, G. Debreu, Existence of an equilibrium for a competitive economy, Econometrica 22 (3) (1954) 265–290.
[2] X. Chen, X. Deng, S.-H. Teng, Computing Nash equilibria: Approximation and smoothed complexity, in: Proceedings of the 47th Annual Symposium on Foundations of Computer Science, 2006.
[3] X. Chen, X. Deng, S.-H. Teng, Sparse games are hard, in: Proceedings of the 2nd International Workshop on Internet and Network Economics, 2006.
[4] B. Codenotti, A. Saberi, K. Varadarajan, Y. Ye, Leontief economies encode nonzero sum two-player games, in: Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms, 2006.
[5] X. Deng, L.-S. Huang, On complexity of market equilibria with maximum social welfare, Inform. Process. Lett. 97 (2006) 4–11.
[6] N. Devanur, C. Papadimitriou, A. Saberi, V. Vazirani, Market equilibria via a primal-dual-type algorithm, in: Proceedings of the 43rd Annual Symposium on Foundations of Computer Science, 2002.
[7] R. Garg, S. Kapoor, Auction algorithms for market equilibrium, in: Proceedings of the 36th Annual ACM Symposium on Theory of Computing, 2004.
[8] L.-S. Huang, S.-H. Teng, On the approximation and smoothed complexity of Leontief market equilibria, in: Proceedings of the 1st Annual International Frontiters of Algorithmics WorkShop, 2007.

[9] K. Jain, A polynomial time algorithm for computing the Arrow–Debreu market equilibrium for linear utilities, in: Proceedings of the 45th Annual Symposium on Foundations of Computer Science, 2004.
[10] K. Jain, M. Mahdian, A. Saberi, Approximating market equilibria, in: Proceedings of the 6th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, 2003.
[11] K. Jain, V. Vazirani, Y. Ye, Market equilibria for homothetic, quasi-concave utilities and economies of scale in production, in: Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms, 2005.
[12] E. Nenakhov, M. Primak, About one algorithm for finding the solution of the Arrow–Debreu model, Kibernetica 3 (1983) 127–128.
[13] Y. Ye, A path to the Arrow–Debreu competitive market equilibrium, Math. Program. 111 (1–2) (2008) 315–348.