Market Equilibria with Hybrid Linear-Leontief Utilities

Xi Chen[1], Li-Sha Huang[1,⋆], and Shang-Hua Teng[2,⋆⋆]

[1] State Key Laboratory of Intelligent Technology and Systems,
Dept. of Computer Science and Technology, Tsinghua Univ., Beijing, 10084, China
[2] Computer Science Department, Boston University, Boston, Massachusetts, USA

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 and approximating market equilibria. Because this family of utility functions contains Leontief utility functions as special cases, finding approximate Arrow-Debreu equilibria with hybrid linear-Leontief utilities is **PPAD**-hard in general. In contrast, we show that, when the Leontief components are grouped, finite and well-conditioned, we can efficiently compute an approximate Arrow-Debreu equilibrium.

1 Introduction

In recent years, the problem of computing market equilibria has attracted many computer scientists. 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 the commodities to 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-Debreu [1] proved the existence of equilibrium prices under a mild condition. Since then, efficient algorithms have been developed for various settings. Naturally, the complexity for finding an equilibrium price is determined not just by the initial endowments, but also by traders' utility functions.

1.1 From Linear to Leontief Utilities

Two popular families of utility functions are the linear and Leontief utilities. Both utilities can be specified by an $m \times n$ demand matrix $\mathbf{D} = (d_{i,j})$, for m

[⋆] Supported by Natural Science Foundation of China (No.60135010,60321002) and the Chinese National Key Foundation Research and Development Plan (2004CB318108).

[⋆⋆] Also affiliated with Akamai Technologies Inc. Cambridge, Massachusetts, USA. 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.

P. Spirakis et al. (Eds.): WINE 2006, LNCS 4286, pp. 274–285, 2006.

goods and n traders. If trader $1 \leq j \leq n$ 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 utility functions are members a large family of utilities functions, referred to as CES utilities. The CES utility function with parameter $\rho \in (-\infty, 1] - \{0\}$ is:

$$u_j^\rho(\mathbf{x}_j) = \left(\sum_i d_{i,j} x_{i,j}^\rho\right)^{1/\rho} .$$

As $\rho \to -\infty$, CES utilities become the Leontief utilities. When $\rho = 1$, the utility functions are linear functions.

Although the Leontief utility functions and linear utility functions look similar, the complexity for finding their market equilibria might be very different. A market equilibrium with linear utilities can be approximated and computed in polynomial time, thanks to a collection of great algorithmic works by Nenakhov and Primak [16], Devanur *et al.* [10], Jain, Mahdian and Saberi [14], Garg and Kapoor [11], Jain [13], and Ye [17].

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 [5] from Nash equilibria to market equilibria, Huang and Teng [12] 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 [3], it is **PPAD**-hard to approximate a Leontief market equilibrium in fully polynomial time. In fact, the smoothed complexity of finding a market equilibrium in Leontief economies cannot be polynomial unless $\mathbf{PPAD} \subset \mathbf{RP}$.

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 of 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 terms in its Leontief components. If the number of terms in any of its Leontief components is at most k, we refer to it as a *k-wide linear-Leontief function.* We further focus on *grouped* hybridizations in which the commodities are divided into groups. Each trader's utility is the summation over the Leontief utilities of all groups. If each group has at most k commodities, we refer to the hybrid functions as *grouped k-wide linear-Leontief functions.*

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 hence a market equilibrium with 1-wide linear-Leontief functions can be found in polynomial time. On the other hand, market equilibria with general hybrid linear-Leontief utilities are **PPAD**-hard to find.

A market with grouped linear-Leontief utility functions can be viewed as a linear combination of several Leontief markets, one for each group of commodities. In an equilibrium, the supplies exactly satisfy the demands for each group of commodities. However, the 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.

- We show that a Fisher equilibrium of an exchange market with n traders, M commodities and hybrid linear-Leontief utility functions can be found in $O(\sqrt{Mn}(M+n)^3 L)$ time.
- We also show that, in the grouped hybridizations when the Leontief component is well-conditioned, we can compute an approximate Arrow-Debreu equilibrium in polynomial time either in M or n. (An interesting observation is that a recent result of Chen, Deng, and Teng [4] on sparse two-player games implies that it is **PPAD**-hard to approximate Arrow-Debreu equilibria in an exchange market with 10-wide linear-Leontief utilities in fully polynomial-time.)

In this paper, we only give formal definition for grouped linear-Leontief utility functions. It is easy to extend the definition and the first algorithmic result to hybrid ones.

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 $a_1, \ldots, a_n$. Matrices are denoted by bold upper-case Roman letters such as $\mathbf{A}$ and scalars are usually denoted by lower-case Roman letters.

We now enumerate some other notations that are used in this 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, $\left(\sum |x_i^p|\right)^{1/p}$ and $\|\mathbf{x}\|_\infty = \max_i |x_i|$.

2 Grouped Linear-Leontief Markets

Assume there are n traders in the market, denoted by $\mathbf{T} = \{1, 2, \ldots, n\}$. The market contains m groups of commodities, denoted by $\mathbf{G} = \{G_1, \ldots, G_m\}$. Each group G_j contains k_j kinds of commodities.

The trader i's initial endowment of goods is a collection of m vectors: $\{\mathbf{e}^i_j \in \mathbb{R}^{k_j}_+ \mid 1 \leq j \leq m\}$, where $\mathbf{e}^i_{j,k}$ is the amount of good k in group j held by trader i. For each group j, let the matrix $\mathbf{E}_j = (\mathbf{e}^1_j, \ldots, \mathbf{e}^n_j)$ denote the traders' initial endowments in the groups. We assume that the amount of each commodity is normalized to 1, i.e., $\langle \mathbf{E}_j | \mathbf{1} \rangle = \mathbf{1}$, or equivalently,

$$\sum_{i=1}^{n} \mathbf{e}_{j,k}^{i} = 1, \quad \forall 1 \le j \le m, 1 \le k \le k_j.$$

Similar to the initial endowments, the allocation to trader i is a collection of m vectors, denoted by $\mathbf{x}^i = \{\mathbf{x}_j^i \in \mathbb{R}_+^{k_j} \mid 1 \le j \le m\}$. The trader i's utility function is characterized by a tuple $\{\mathbf{a}^i \in \mathbb{R}_+^m, \{\mathbf{d}_j^i \in \mathbb{R}_+^{k_j} \mid 1 \le j \le m\}\}$. Given an allocation $\mathbf{x}^i = \{\mathbf{x}_j^i \in \mathbb{R}_+^{k_j} \mid 1 \le j \le m\}$, trader i's utility is defined as follows:

$$u_i(\mathbf{x}^i) = \sum_{j=1}^{m} a_j^i v_j^i, \text{ where } v_j^i = \min \left\{ \frac{x_{j,k}^i}{d_{j,k}^i} \,\middle|\, k = 1, 2, \ldots, k_j \right\}$$

In other words, trader i's utility function is a linear combination of m Leontief utility functions.

Locally, each group j is a Leontief economy. That is, every trader i demands the goods in group j in proportion to the vector $\mathbf{d}_j^i$. Therefore, we can introduce the matrix $\mathbf{D}_j = (\mathbf{d}_j^1, \ldots, \mathbf{d}_j^n)$ to characterize the traders' demands in group j. Let $\mathbf{v}_j = (v_j^1, v_j^2, \ldots, v_j^n)^\top$ be an n-dimensional column vector, which can be viewed as an allocation of goods in group j. Then a feasible allocation $\mathbf{v}_j$ of goods in group j should satisfy $\mathbf{D}_j \mathbf{v}_j \le \mathbf{1}$. The allocation of the whole market is denoted by $\mathbf{v} = \{\mathbf{v}_j \in \mathbb{R}_+^n \mid 1 \le j \le m\}$.

Let $\mathbf{D} = (\mathbf{D}_1, \ldots, \mathbf{D}_m)$, $\mathbf{E} = (\mathbf{E}_1, \ldots, \mathbf{E}_m)$ and $\mathbf{A} = (a_j^i)$, then the market can be denoted by 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.

Definition 1 (Exchange Equilibrium). *An equilibrium is a pair* $(\mathbf{p}, \mathbf{v})$*, where* $\mathbf{p} = \{\mathbf{p}_j \in \mathbb{R}_+^{k_j} \mid 1 \le j \le m\}$ *is a collection of* m *price vectors and* $\mathbf{v} = \{\mathbf{v}_j \in \mathbb{R}_+^n \mid 1 \le j \le m\}$ *is the allocation of the whole market, satisfying that:*

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

Definition 2 (ε-approximate Equilibrium). *An* ε*-equilibrium is a pair* $(\mathbf{p}, \mathbf{v})$*, satisfying that:*

$$\begin{cases} u_i = \sum\limits_{j=1}^{m} a_j^i v_j^i, \forall i = 1, \ldots, n \\ u_i \ge (1 - \varepsilon) \max\{ \sum\limits_{j=1}^{m} a_j^i z_j^i \mid \sum\limits_{j=1}^{m} \langle \mathbf{p}_j | \mathbf{d}_j^i \rangle z_j^i \le \sum\limits_{j=1}^{m} \langle \mathbf{p}_j | \mathbf{e}_j^i \rangle \}, \forall i = 1, \ldots, n \\ \mathbf{D}_j \mathbf{v}_j \le (1 + \varepsilon) \mathbf{1}, \forall j = 1, \ldots, m \end{cases}$$

3 An Equivalent Equilibrium Condition

Next, we prove a necessary and sufficient condition of an equilibrium. This condition will be useful in our equilibrium computation algorithms.

Theorem 1. *A pair* $(\mathbf{p}, \mathbf{v})$ *is an equilibrium if and only if it satisfies that*

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

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

$$\sum_{j=1}^{m} v_j^i \left\langle \mathbf{d}_j^i | \mathbf{p}_j \right\rangle \leq \sum_{j=1}^{m} \left\langle \mathbf{e}_j^i | \mathbf{p}_j \right\rangle \tag{2}$$

$$v_j^i > 0 \Rightarrow a_j^i / \left\langle \mathbf{d}_j^i | \mathbf{p}_j \right\rangle \geq a_k^i / \left\langle \mathbf{d}_k^i | \mathbf{p}_k \right\rangle (\forall k) \tag{3}$$

The first equation is the trader's budget constraint, and the second equation implies that the trader buys only those groups that maximizes his utility gained per unit money spent on the groups.

Note that the equation (2) and (3) can be replaced equivalently by

$$\begin{cases} u_i = \sum_{j=1}^{m} a_j^i v_j^i, \quad w_i = \sum_{j=1}^{m} \left\langle \mathbf{p}_j | \mathbf{e}_j^i \right\rangle \\ \dfrac{a_j^i}{\left\langle \mathbf{p}_j | \mathbf{d}_j^i \right\rangle} \leq \dfrac{u_i}{w_i}, \ \forall j \\ \dfrac{a_j^i}{\left\langle \mathbf{p}_j | \mathbf{d}_j^i \right\rangle} v_j^i \leq \dfrac{u_i}{w_i} v_j^i, \ \forall j \end{cases}$$

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

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

Now, it suffices to prove that the last equation can be derived from the other four equations. By $w_i a_j^i \leq u_i \left\langle \mathbf{p}_j | \mathbf{d}_j^i \right\rangle$, we have

$$
\begin{array}{llll}
 & w_i a_j^i v_j^i & \leq u_i \left\langle \mathbf{d}_j^i | \mathbf{p}_j \right\rangle v_j^i, & \forall i, j \\
\Rightarrow & w_i \sum\limits_{j=1}^{m} a_j^i v_j^i & \leq u_i \sum\limits_{j=1}^{m} \left\langle \mathbf{d}_j^i | \mathbf{p}_j \right\rangle v_j^i, & \forall i \\
\Rightarrow & w_i = \sum\limits_{j=1}^{m} \left\langle \mathbf{e}_j^i | \mathbf{p}_j \right\rangle & \leq \sum\limits_{j=1}^{m} \left\langle \mathbf{d}_j^i | \mathbf{p}_j \right\rangle v_j^i, & \forall i \\
\Rightarrow & \sum\limits_{i=1}^{n} \sum\limits_{j=1}^{m} \left\langle \mathbf{e}_j^i | \mathbf{p}_j \right\rangle & \leq \sum\limits_{i=1}^{n} \sum\limits_{j=1}^{m} \left\langle \mathbf{d}_j^i v_j^i | \mathbf{p}_j \right\rangle & \\
\Rightarrow & \sum\limits_{j=1}^{m} \langle \mathbf{1} | \mathbf{p}_j \rangle & \leq \sum\limits_{j=1}^{m} \langle \mathbf{D}_j \mathbf{v}_j | \mathbf{p}_j \rangle &
\end{array}
$$

Since $\mathbf{D}_j \mathbf{v}_j \leq \mathbf{1}$ for all j, we have

$$
\begin{cases}
\langle \mathbf{D}_j \mathbf{v}_j | \mathbf{p}_j \rangle = \langle \mathbf{1} | \mathbf{p}_j \rangle \\
w_i = \sum\limits_{j=1}^{m} \left\langle \mathbf{e}_j^i | \mathbf{p}_j \right\rangle = \sum\limits_{j=1}^{m} \left\langle \mathbf{d}_j^i | \mathbf{p}_j \right\rangle v_j^i, \forall i
\end{cases}
$$

Again, by $w_i a_j^i \leq u_i \left\langle \mathbf{p}_j | \mathbf{d}_j^i \right\rangle$, we have

$$
\begin{array}{lll}
w_i a_j^i v_j^i & \leq u_i \left\langle \mathbf{d}_j^i | \mathbf{p}_j \right\rangle v_j^i, & \forall i, j \\
\Rightarrow w_i u_i = w_i \sum\limits_{j=1}^{m} a_j^i v_j^i & \leq u_i \sum\limits_{j=1}^{m} \left\langle \mathbf{d}_j^i | \mathbf{p}_j \right\rangle v_j^i = u_i w_i, & \forall i
\end{array}
$$

This forced that $w_i a_j^i v_j^i = u_i \left\langle \mathbf{d}_j^i | \mathbf{p}_j \right\rangle v_j^i$ for all i, j. ■

3.1 Solving the Fisher's Model

The Fisher's model is a special case of the Arrow-Debreu's exchange market model. In the Fisher's model, the commodities are held by a seller initially. The traders come to the market with the initial endowments of money, instead of the endowments of commodities in the general setting. The traders buy goods from the seller to maximize each's utility, under the budget constraints. The market is in an equilibrium if the supplies satisfy the demands. Usually, the computation of equilibria in the Fisher's setting is much easier than that in the general case.

Assume in the Fisher's model, trader i has w_i dollars initially. As shown in [15], the equilibrium can be approximated by solving the following convex programming problem:

$$
\begin{array}{l}
\max \sum\limits_{i=1}^{n} w_i \log(u_i) \\
\text{s.t.} \begin{cases}
u_i = \sum\limits_{j=1}^{m} a_j^i v_j^i & , \forall i = 1, \ldots, n \\
\mathbf{D}_j \mathbf{v}_j \leq \mathbf{1} & , \forall j = 1, \ldots, m \\
\mathbf{v}_j \geq 0 & , \forall j = 1, \ldots, m
\end{cases}
\end{array}
\tag{4}
$$

With the same argument as in Ye [17], we can prove that

Theorem 2 (Fisher's Equilibrium). *The Fisher's model can be solved by the interior-point algorithm in time $O(\sqrt{Mn}(M+n)^3 L)$, where $M = \sum\limits_{j=1}^{m} k_j$ is the total number of commodities, n is the number of traders and L is the bit-length of the input data.* ■

4 An Approximation Algorithm

Since Leontief economy is a special case of the hybrid linear-Leontief economy, the hardness results [5,8,12] in Leontief economy can be imported to our case. For example, it is NP-hard to determine the existence of equilibria [5], and there is no algorithm to compute the equilibrium in smoothed polynomial time, unless **PPAD** $\subset$ **RP** [12]. In this section, we propose an approximation algorithm for the grouped linear-Leontief economy, running in

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

where $M = \sum_{j=1}^{m} k_j$ is the total number of commodities and $\tau = \min_{i,j,k}\{\mathbf{e}^i_{j,k}, \mathbf{d}^i_{j,k}\}$.

4.1 Intuition

Since the market can be viewed as a linear combination of several Leontief markets, we may expect that it can be reduced to a linear market when the equilibrium information of the sub-markets are given. We first discuss this intuition in this subsection.

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 more convenient to replace $\mathbf{p}_j$ by $q_j\mathbf{p}_j$, where $q_j \in \mathbb{R}_+$ and $\|\mathbf{p}_j\|_1 = 1$ is a normalized vector in $\mathbb{R}^{k_j}_+$. Thus the equilibrium $(\mathbf{p}, \mathbf{v})$ is replaced by $(\mathbf{q}, \mathbf{p}, \mathbf{v})$, where $\mathbf{q} \in \mathbb{R}^m_+$.

We define a market $\hat{\mathbf{M}}$ with linear utilities as follows. The set of traders are same as $\mathbf{M}$. For each group $G_j \in \mathbf{G}$, we introduce a commodity j to $\hat{\mathbf{M}}$. The trader i's initial endowment of commodity j is defined by $\hat{e}^i_j = \left\langle \mathbf{e}^i_j | \mathbf{p}_j \right\rangle$ and his preference to commodity j is $\hat{a}^i_j = a^i_j / \left\langle \mathbf{d}^i_j | \mathbf{p}_j \right\rangle$. The following lemma is obvious.

Lemma 1 (Market Reduction). *Let* $\hat{x}^i_j = v^i_j \left\langle \mathbf{d}^i_j | \mathbf{p}_j \right\rangle$ *for all* $1 \le i \le n$ *and* $1 \le j \le m$*, then* $(\mathbf{q}, \hat{\mathbf{x}})$ *is an equilibrium for the linear market* $\hat{\mathbf{M}}$*.* ■

The above lemma shows that if we are so lucky that we know the *internal* price $\mathbf{p}_j$ of every group G_j, the hybrid market can be transformed to a linear market, where every group G_j in the market $\mathbf{M}$ is replaced by a special commodity j in $\hat{\mathbf{M}}$, which plays the role of currency of this group. The traders' endowments and preferences to this group are changed to the endowments and preferences to this currency. $\hat{\mathbf{M}}$ can be viewed as the foreign currency exchange market. The equilibrium price $\mathbf{q}$ of $\hat{\mathbf{M}}$ is the exchange rate between groups, which can be computed in polynomial time, since the market $\hat{\mathbf{M}}$ is linear.

This fact leads to the following approximation heuristic. We exhaustively *enumerate* the internal prices of every group j in the simplex $\mathbb{P}^{k_j}$. With the collection of sampled internal prices $\mathbf{p} = \{\mathbf{p}_j \mid 1 \leq j \leq m\}$, we transform the market $\mathbf{M}$ to the linear market $\hat{\mathbf{M}}$. Then we compute the equilibrium price $\mathbf{q}$ and allocation $\hat{\mathbf{x}}$ in the market $\hat{\mathbf{M}}$. Let $v_j^i = \hat{x}_j^i / \langle \mathbf{d}_j | \mathbf{p}_j \rangle$ for all $1 \leq i \leq n$ and $1 \leq j \leq m$. For every group j, let $\mathbf{v}_j = (v_j^1, \ldots, v_j^n)^\top$. Finally, we check that if $\mathbf{D}_j \mathbf{v}_j \leq \mathbf{1}$ are approximately satisfied. If true, we have found an approximate equilibrium $(\mathbf{q}, \mathbf{p}, \mathbf{v})$ for the original market $\mathbf{M}$.

Before we explicitly present the algorithm, we prepare two important tools in the following two subsections.

4.2 Efficient Sampling Problem

Problem: Given $\varepsilon > 0$ and n vectors $\{\mathbf{x}_i \in \mathbb{P}^k \mid i = 1, \ldots, n\}$, called anchor points, find a sampling set $S \subseteq \mathbb{P}^k$ such that for any $\mathbf{p} \in \mathbb{P}^k$, there exists a sample point $\hat{\mathbf{p}} \in S$ satisfying $1 - \varepsilon \leq \langle \mathbf{p} | \mathbf{x}_i \rangle / \langle \hat{\mathbf{p}} | \mathbf{x}_i \rangle \leq 1 + \varepsilon$ for any anchor point $\mathbf{x}_i$, $1 \leq i \leq n$. The set S is called the *efficient sampling set* of $\{\mathbf{x}_i\}$ and ε. Our goal is to minimize the size of S.

We give two constructions for set S.

Lemma 2. *If* $\tau = \min_{i,j}\{\mathbf{x}_{i,j}\} > 0$, *then we can construct an efficient sampling set* S *of size* $O(\dfrac{\log(1/\tau)}{\varepsilon})^n$.

Proof. For $\mathbf{x}_1$ and any $\mathbf{p} \in \mathbb{P}^k$, we have $\tau \leq \langle \mathbf{p} | \mathbf{x}_1 \rangle \leq 1$.

Define $\log(1/\tau)/\log(1+\varepsilon) \approx \log(1/\tau)/\varepsilon$ planes:

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

These planes cut $\mathbb{P}^k$ into $O(\log(1/\tau)/\varepsilon)$ polytopes, denoted by $P_0, P_1, \ldots$.

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

Repeat this process for n rounds, we divide simplex $\mathbb{P}^k$ to $O(\log(1/\tau)/\varepsilon)^n$ polytopes. The sampling set S is constructed by picking an inner point from each polytope. ■

Lemma 3. *If* $\tau = \min_{i,j}\{\mathbf{x}_{i,j}\} > 0$, *then we can construct 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 $\hat{\mathbf{p}} \in S$ satisfying $\|\mathbf{p} - \hat{\mathbf{p}}\|_\infty \leq \varepsilon\tau$. Obviously, S is an efficient sampling set and is of size $O((\tau\varepsilon)^{1-k})$. ■

In our algorithm, we are going to construct the efficient sampling set S_j for $\{\mathbf{e}_j^i \mid i = 1, \ldots, n\} \cup \{\mathbf{d}_j^i \mid i = 1, \ldots, n\}$ and ε. Let S be $S_1 \times \cdots \times S_m$. The time complexity of the algorithm is dominated by the size of S.

4.3 Convex Optimization Problem

Consider the equilibrium condition in Theorem 1. As in the above discussion, an equilibrium $(\mathbf{p}, \mathbf{v})$ is replaced by a 3-tuple $(\mathbf{q}, \mathbf{p}, \mathbf{v})$. We now introduce the following optimization problem:

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

The quantity θ can be viewed as the surplus of the demands. We can prove that θ is always nonnegative for any feasible solution of problem (5). The proof is omitted here since it is similar to the one in Ye [17].

Lemma 4. *For any feasible solution* $(\mathbf{q}, \mathbf{p}, \mathbf{v})$ *of (5),* $\theta \geq 0$. *Moreover,* $(\mathbf{q}, \mathbf{p}, \mathbf{v})$ *is an equilibrium if and only if* $\theta = 0$.

Assume we have *guessed* a set of internal prices $\hat{\mathbf{p}} = \{\hat{\mathbf{p}}_j \mid 1 \leq j \leq m\}$, then problem (5) is reduced to the following convex optimization problem, denoted by Opt$(\hat{\mathbf{p}})$:

$$\begin{array}{rll} \mathit{min} & \theta & \\ \mathit{s.t.}\ \mathbf{D}_j\mathbf{v}_j & \leq (1+\theta)\mathbf{1}, & \forall j \\ u_i & = \sum_{j=1}^{m} a_j^i v_j^i, & \forall i \\ w_i & = \sum_{j=1}^{m} q_j \left\langle \hat{\mathbf{p}}_j | \mathbf{e}_j^i \right\rangle, & \forall i \\ w_i a_j^i & \leq u_i q_j \left\langle \hat{\mathbf{p}}_j | \mathbf{d}_j^i \right\rangle, & \forall i, j \\ \mathbf{q} & > 0, & \forall j \end{array} \tag{6}$$

Opt$(\hat{\mathbf{p}})$ can be solved in polynomial time [17]. Note that since $\hat{\mathbf{p}}$ may not be an equilibrium internal prices, the optimum of Opt$(\hat{\mathbf{p}})$ may not be zero.

4.4 The Algorithm

Finally, our algorithm is described in Figure 1. Its correctness is guaranteed by the following lemma. The lemma shows that there exists an internal price $\hat{\mathbf{p}} \in S$ such that the solution of Opt$(\hat{\mathbf{p}})$ is an ε-approximate equilibrium, according to Definition 2.

Lemma 5. *Assume* $(\mathbf{q}^*, \mathbf{p}^*, \mathbf{v}^*)$ *is an equilibrium. If* $\hat{\mathbf{p}}$ *satisfies that*

$$1 - \varepsilon \leq \frac{\left\langle \mathbf{p}_j^* | \mathbf{d}_j^i \right\rangle}{\left\langle \hat{\mathbf{p}}_j | \mathbf{d}_j^i \right\rangle} \leq 1 + \varepsilon \quad \mathit{and} \quad 1 - \varepsilon \leq \frac{\left\langle \mathbf{p}_j^* | \mathbf{e}_j^i \right\rangle}{\left\langle \hat{\mathbf{p}}_j | \mathbf{e}_j^i \right\rangle} \leq 1 + \varepsilon$$

for all i *and* j, *then the optimum of the problem Opt*$(\hat{\mathbf{p}})$ *satisfies* $\hat{\theta} \leq 3\varepsilon$.

```
for each group $G_j$ do
    Construct the efficient sampling set $S_j$ for
    $\{\mathbf{e}_j^i \mid i = 1, \ldots, n\} \cup \{\mathbf{d}_j^i \mid i = 1, \ldots, n\}$ and $\varepsilon/3$.
end
Let $S = S_1 \times \cdots \times S_m$.
for each $\hat{\mathbf{p}} \in S$ do
    Solve the convex optimization problem $\mathrm{Opt}(\hat{\mathbf{p}})$;
    If the optimum $\hat{\theta} < \varepsilon$, break the loop and output.
end
```

Fig. 1. An Approximation Algorithm

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

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

We explicitly construct a feasible solution $(\mathbf{q}, \mathbf{v})$ of the problem (6) as follows:

$$\begin{cases} q_j = q_j^*, \quad \forall j \\ \lambda_i = \min\{q_j \left\langle \hat{\mathbf{p}}_j | \mathbf{d}_j^i \right\rangle / a_j^i \mid 1 \le j \le m\}, \quad \forall i \\ w_i = \sum_{j=1}^{m} q_j \left\langle \hat{\mathbf{p}}_j | \mathbf{e}_j^i \right\rangle, \quad \forall i \\ u_i = w_i / \lambda_i, \quad \forall i \\ v_j^i = v_j^{i*} \dfrac{u_i}{u_i^*}, \quad \forall i, j \end{cases}$$

Since $\lambda_i / \lambda_i^* \ge 1 - \varepsilon$ and $w_i / w_i^* \ge 1 + \varepsilon$, we have

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

and thus, $\mathbf{D}_j \mathbf{v}_j \le \mathbf{D}_j \mathbf{v}_j^* (1 + 3\varepsilon) \le 1 + 3\varepsilon$. Therefore, the optimum of (6) must be less or equal to 3ε. ■

The time complexity of our algorithm is $|S|\mathrm{poly}(M, n)$, where $\mathrm{poly}(M, n)$ is spent on solving each optimization problem $\mathrm{Opt}(\hat{\mathbf{p}})$ and $M = \sum_{j=1}^{m} k_j$ is the total number of commodities. According to Lemma 2 and Lemma 3, the size of S is

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

where $\tau = \min_{i,j,k}\{\mathbf{e}_{j,k}^i, \mathbf{d}_{j,k}^i\}$.

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 equilibria in the Fisher's model can be found in polynomial time. We also develop an approximation algorithm for approximating equilibria in the Arrow-Debreu's exchange market model. The time complexity of this approximation algorithm depends on the answer to the efficient sampling problem, which is described in Section 4.2. At this moment, it is exponential to either the number of commodities or the number of traders. Any improvement to the sampling problem will improve the performance of our approximation algorithm.

As a grouped hybrid market is a linear combination of Leontief economies, given the fact that linear markets are easy to solve [16,13,17], we conjecture that there exists an approximation algorithm that runs in polynomial time to the number of groups and the number of traders, with an 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 plus 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 the family of hierarchical utility functions to characterize more complicated market behaviors. With the same technique used in Section 3.1, an equilibrium in the Fisher's setting can be computed efficiently. We hope that, the study to these utilities will lead us to a better understanding of the complexity of computing market equilibria.

References

1. Arrow, J., Debreu, G.: Existence of an equilibrium for a competitive economy. *Econometrica*, 22(3):265–290, 1954.
2. Chen, X., Deng, X.: Settling the Complexity of 2-Player Nash-Equilibrium. To appear in FOCS 2006.
3. Chen, X., Deng, X., Teng, S.-H.: Computing Nash Equilibria: Approximation and smoothed complexity. To appear in FOCS 2006.
4. Chen, X., Deng, X., Teng, S.-H.: Sparse games are hard. To appear in WINE 2006.
5. Codenotti, B., Saberi, A., Varadarajan, K., Ye, Y.: Leontief economies encode nonzero sum two-player games. In: *Proceedings of SODA'06*, pages 659–667, 2006.
6. Codenotti, B., Varadarajan, K.: Efficient computation of equilibrium prices for markets with Leontief utilities. In: *Proceedings of ICALP'04*, pages 371–382, 2004.
7. Deng, X., Papadimitriou, C., Safra, S.: On the complexity of price equilibria. *Journal of Computer and System Sciences*, 67(2):311–324, 2003.
8. Deng, X., Huang, L.-S.: On Complexity of Market Equilibria with Maximum Social Welfare. *Information Processing Letters*, 97 : 4–11, 2006.

9. Deng, X., Huang, L.-S.: Approximate Economic Equilibrium Algorithms. In T. Gonzalez, editor, *Approximation Algorithms and Metaheuristics*. 2005.
10. Devanur, N. R., Papadimitriou, C., Saberi, A., Vazirani, V. V.: Market equilibria via a primal-dual-type algorithm. In *Proceedings of FOCS'02*, pages 389–395, 2002.
11. Garg, R., Kapoor, S.: Auction Algorithms for Market Equilibrium. In: *Proceeding of STOC'04*, pages 511–518, 2004.
12. Huang, L.-S., Teng, S.-H.: On the Approximation and Smoothed Complexity of Leontief Market Equilibria. ECCC TR 06-031, 2006. Also available online http://arXiv.org/abs/cs/0602090, 2006
13. Jain, K.: A polynomial time algorithm for computing the Arrow-Debreu market equilibrium for linear utilities. In: *Proceeding of FOCS'04*, pages 286–294, 2004.
14. Jain, K., Mahdian, M., Saberi, A.: Approximating market equilibria. In: *Proceedings of APPROX'03*, pages 98-108, 2003.
15. Jain, K., Vazirani, V. V., Ye, Y.: Market equilibria for homothetic, quasi-concave utilities and economies of scale in production. In: *Proceedings of SODA'05*, pages 63–71, 2005.
16. Nenakhov, E., Primak, M.: About one algorithm for finding the solution of the Arrow-Debreu model. *Kibernetica*, (3):127–128, 1983.
17. Ye, Y.: A path to the Arrow-Debreu competitive market equilibrium. *Math. Programming*, 2004.
18. Ye, Y.: On exchange market equilibria with Leontief's utility: Freedom of pricing leads to rationality. In: *Proceedings of WINE'05*, pages 14–23, 2005.