Settling the Complexity of Leontief and PLC Exchange Markets under Exact and Approximate Equilibria

Jugal Garg
University of Illinois at Urbana-Champaign
Urbana, IL, USA
jugal@illinois.edu

Ruta Mehta
University of Illinois at Urbana-Champaign
Urbana, IL, USA
rutameht@illinois.edu

Vijay V. Vazirani*
Georgia Institute of Technology
Atlanta, GA, USA
vazirani@cc.gatech.edu

Sadra Yazdanbod
Georgia Institute of Technology
Atlanta, GA, USA
syazdanb@cc.gatech.edu

ABSTRACT

Our first result shows membership in PPAD for the problem of computing approximate equilibria for an Arrow-Debreu exchange market for piecewise-linear concave (PLC) utility functions. As a corollary we also obtain membership in PPAD for Leontief utility functions. This settles an open question of Vazirani and Yannakakis (2011).

Next we show FIXP-hardness of computing equilibria in Arrow-Debreu exchange markets under Leontief utility functions, and Arrow-Debreu markets under linear utility functions and Leontief production sets, thereby settling these open questions of Vazirani and Yannakakis (2011). As corollaries, we obtain FIXP-hardness for PLC utilities and for Arrow-Debreu markets under linear utility functions and polyhedral production sets. In all cases, as required under FIXP, the set of instances mapped onto will admit equilibria, i.e., will be "yes" instances. If all instances are under consideration, then in all cases we prove that the problem of deciding if a given instance admits an equilibrium is ETR-complete, where ETR is the class Existential Theory of Reals.

As a consequence of the results stated above, and the fact that membership in FIXP has been established for PLC utilities, the entire computational difficulty of Arrow-Debreu markets under PLC utility functions lies in the Leontief utility subcase. This is perhaps the most unexpected aspect of our result, since Leontief utilities are meant for the case that goods are perfect complements, whereas PLC utilities are very general, capturing not only the cases when goods are complements and substitutes, but also arbitrary combinations of these and much more.

Finally, we give a polynomial time algorithm for finding an equilibrium in Arrow-Debreu exchange markets under Leontief

utility functions provided the number of agents is a constant. This settles part of an open problem of Devanur and Kannan (2008).

CCS CONCEPTS

• **Theory of computation → Exact and approximate computation of equilibria**; **Market equilibria**;

KEYWORDS

Market Equilibria, Leontief, PLC, PPAD, FIXP, ETR

1 INTRODUCTION

A decade and a half of work in TCS has led to a deep understanding of computability of market equilibria for classic market models under fundamental utility functions; see details in Section 1.1. At this point, perhaps the most basic utility functions whose complexity remains unresolved are Leontief[1] and piecewise-linear concave (PLC). For both exact and approximate computation of equilibria, only partial results are know as detailed below. In this paper, we resolve the remaining open questions, thereby pinning down the classes which characterize their complexity.

In economics, concave utilities occupy a special place because of their generality and because they capture the natural condition of decreasing marginal utilities. Since computer science assumes a finite precision model of computation, one is forced to restrict attention to PLC[2] utility functions. Price equilibria are clearly quintessential to economics, and therefore it is important to obtain a precise understanding of the complexity of computing Arrow-Debreu equilibria under PLC utility functions[3]. Leontief utilities

---

[1]Leontief utility function for a bundle x of goods is given by $U(x) = \min_j x_j/A_j$, where A_j's are non-negative constants. It captures the situation when goods are complements and required in a fixed proportion.

[2]Clearly, by making the pieces fine enough, we can obtain a good approximation to the original utility functions.

[3]In the past, for central problems, simply establishing evidence of intractability has not been enough – researchers have gone the extra step to exactly identify the classes

form a subcase of PLC utilities and are very widely used in economic modeling [33].

For both these utility functions, computation of approximate equilibria has been known to be PPAD-hard for more than a decade [11, 13, 28]; however, membership in PPAD has not been established yet. In fact, [39] goes further to say that these problems may not even be in PPAD[4]. Furthermore, certain consequences stated in the literature are not true without establishing this result (see Section 1.2). Our first result shows membership of these problems in PPAD. The only fixed point formulation known for these problems was obtained in the context of proving membership in FIXP [23]. This formulation is our starting point; however, working with it is not straightforward. The main technical challenge lies in showing that an approximate fixed point captures an approximate market equilibrium. This turns out to be quite involved and technical, and requires new ideas as elaborated in Section 1.2. On the other hand, for both these utility functions, exact computation of equilibria is known to be in FIXP [23, 42]; however, FIXP-hardness was not established before and was stated as an open problem in [39]. This is our second result.

Proofs of membership in FIXP for Leontief and PLC utility functions were given by Yannakakis [42] and Garg et. al. [23], respectively. In this paper, we prove FIXP-hardness for Arrow-Debreu exchange markets under Leontief utility functions, and Arrow-Debreu markets under linear utility functions and Leontief production sets. As corollaries, we obtain FIXP-hardness for PLC utilities and for Arrow-Debreu markets under linear utility functions and polyhedral production sets (membership in FIXP for production was also shown in [23]). In all cases, as required under FIXP, the set of instances mapped onto will admit equilibria, i.e., will be "yes" instances. If all instances are under consideration, then we prove that the problem of deciding if a given instance admits an equilibrium is ETR-complete, where ETR is the class Existential Theory of Reals.

As a consequence of the results stated above, the entire computational difficulty of Arrow-Debreu markets under PLC utility functions lies in the Leontief utility subcase. This is perhaps the most unexpected aspect of our result, since Leontief utilities are meant only for the case that goods are perfect complements, whereas PLC utilities are very general, capturing not only the cases when goods are complements and substitutes, but also arbitrary combinations of these and much more.[5]

The class PPAD was defined by Papadimitriou [36]; he also proved PPAD-completeness of computing an approximate equilibrium of Arrow-Debreu exchange markets given by aggregate excess demand functions. The class FIXP was defined by Etessami

and Yannakakis [21] and they proved FIXP-completeness of Arrow-Debreu exchange markets whose aggregate excess demand functions are algebraic. However, these results do not establish PPAD or FIXP-completeness of Arrow-Debreu markets under any specific class of utility functions[6]. We note that there has been no progress on giving proofs of FIXP-hardness for market equilibria under any specific utility functions.

Perhaps the most elementary way of stating the main technical part of our second result is the following reduction, which we will denote by $\mathcal{R}$: Given a set $\mathcal{S}$ of simultaneous multivariate polynomial equations in which the variables are constrained to be in a closed bounded region in the positive orthant, we construct an Arrow-Debreu market with Leontief utilities, say $\mathcal{M}$, which has one good corresponding to each variable in $\mathcal{S}$. We prove that the equilibria of $\mathcal{M}$, when projected onto prices of these latter goods, are in one-to-one correspondence with the set of solutions of the polynomials. This reduction, together with the fact that the 3-player Nash equilibrium problem (3-Nash) is FIXP-complete [21] and that 3-Nash can be reduced to such a system $\mathcal{S}$, yield FIXP-hardness for the Leontief case.

On positive results for PLC utilities, Devanur and Kannan had given a polynomial time algorithm for finding an equilibrium in Arrow-Debreu markets under these utility functions provided the number of goods is a constant, using algebraic cell decomposition [15]. They had stated the open problem of handling the case of constant number of agents. Our third result settles a part of this open problem by obtaining a polynomial time algorithm for the subcase of Leontief utilities.

1.1 Previous Results on Computability of Market Equilibria

The first utility functions to be studied were linear. Once polynomial time algorithms were found for markets under such functions [14, 16–18, 25, 29, 30, 35, 41, 43] and certain other cases [10, 15, 26, 31, 40], the next question was settling the complexity of Arrow-Debreu markets under separable, piecewise-linear concave (SPLC) utility functions. This problem was shown to be complete [7, 39] for PPAD. Also, when all instances are under consideration, the problem of deciding if a given SPLC market admits an equilibrium was shown to be NP-complete [39]. The notion of SPLC production sets was defined in [24] and Arrow-Debreu markets under such production sets and linear utility functions were shown to be PPAD-complete.

Previous computability results for Leontief utility functions were the following: In contrast to our result, Fisher markets under Leontief utilities admit a convex program [20] and hence their equilibria can be approximated to any required degree in polynomial time [4, 6]. Arrow-Debreu markets under Leontief utilities were shown to be PPAD-hard [11]. They reduce 2-Nash to a special case called "pairing economy" in which each agent brings one unit of a distinct good. For this case, equilibria are rational; however, in general they are irrational for Leontief markets [19], and hence their complexity is not characterized by PPAD. We note that the two complexity

---

that characterize their complexity. Two prominent examples are the simplex method for LP and the Lemke-Howson algorithm for 2-Nash. Both were shown to require exponential time [32, 37] and later the complexity of finding the solutions produced by both were shown to be PSPACE-complete [22, 27].

[4][39] says: *"For the Arrow-Debreu model, checking existence of an equilibrium is NP-hard, and for instances satisfying the standard Arrow-Debreu sufficient conditions, the computation of approximate equilibria is PPAD-hard [Codenotti et al. 2006; Huang and Teng 2007; Deng and Du 2008]. Note that these are hardness, rather than completeness, results because these problems for Leontief markets do not lie necessarily in NP and PPAD."*

[5]We had expected the precise complexity of computing an equilibrium in Arrow-Debreu exchange markets to be easier in case of Leontief utilities than in case of PLC utilities.

[6]In the economics literature, there are two parallel streams of results on market equilibria, one assumes being given an excess demand function and the other a specific class of utility functions.

classes PPAD and FIXP appear to be quite disparate – whereas solutions to problems in the former are rational numbers, those to the latter are algebraic numbers. And whereas the former is contained in function classes NP ∩ co-NP, the latter lies somewhere between P and PSPACE, and is likely to be closer to the harder end of PSPACE [42].

Leontief utilities are a limiting case of constant elasticity of substitution (CES) utilities [33]. Finding an approximate equilibrium under the latter was also shown to be PPAD-complete [9].

1.2 Technical Contributions

Let $\mathcal{M}$ denote an Arrow-Debreu exchange market under piecewise-linear concave (PLC) utilities. Daskalakis, Goldberg and Papadimitriou [12] proved that computation of approximate fixed point of a Lipschitz continuous function, whose Lipschitz constant is polynomial sized, is in PPAD. In showing membership of the problem of computing an equilibrium of $\mathcal{M}$ in FIXP, [23] had given a particular fixed point formulation F such that the fixed points of F give equilibria of $\mathcal{M}$ and vice-versa. We start with F and to show that the problem of computing an ϵ-approximate market equilibrium for $\mathcal{M}$ is in PPAD, we need to show two things: (*i*) F is Lipschitz continuous with constant K, where $size(K) = poly(size(\mathcal{M}))$, and (*ii*) a δ-approximate fixed point of F gives an ϵ-approximate market equilibrium, where δ and ϵ are polynomially related, i.e., $size(\delta) = poly(size(\epsilon, \mathcal{M}))$.

The first step is easy to show; however, the second step is quite involved and technical because F is rather intricate. We note that even in the case of Nash equilibrium, whose fixed point formulation is relatively simple, it was non-trivial to show membership in PPAD [12]. Informally, at a market equilibrium (prices, allocation of goods to agents), each agent obtains an optimal bundle of goods and demand of each good meets its supply (market clearing). At given prices, optimal bundles of an agent can be captured through a linear program (LP). At a fixed point of F, the primal and dual constraints, and complementary slackness conditions of this LP are satisfied. This ensures that each agent receives an optimal bundle. We show that at an approximate fixed point of F, both feasibility and complementary slackness constraints are approximately satisfied. This proves that each agent gets an approximately optimal bundle. Further, we show that market clearing is also approximately satisfied at an approximate fixed point of F. Together, they imply approximate market equilibrium and hence membership in PPAD.

As stated in the Introduction, some claims made in the literature do not hold without formally proving membership in PPAD of the above-stated problem. As an example, [28] state that the Leontief exchange market problem does not have a fully polynomial-time approximation scheme, unless PPAD ⊆ P, and that the smoothed complexity of any algorithm for computing a market equilibrium in a Leontief economy, is not polynomial, unless PPAD ⊆ RP.

We next move to our FIXP-hardness result and describe the difficulties encountered in obtaining reduction $\mathcal{R}$ and the ideas needed to overcome them. For this purpose, it will be instructive to draw a comparison between reduction $\mathcal{R}$ and the reduction from 2-Nash to SPLC markets given in [7]. At the outset, observe the

latter is only dealing with linear functions of variables[7] and hence is much easier than the former.

Both reductions create one market with numerous agents and goods, and the amount of each good desired by an agent gets determined only after the prices are set. Yet, at the desired prices, corresponding to solutions to the problem reduced from, the supply of each good needs to be exactly equal to its demand. In the latter reduction, the relatively constrained utility functions give a lot more "control" on the optimal bundles of agents. Indeed, it is possible to create one large market with many agents and many goods and still argue how much of each good is consumed by each agent at equilibrium.

We do not see a way of carrying out similar arguments when all agents have Leontief utility functions. The key idea that led to our reduction was to create several modular units within the large market and ensure that each unit would have a very simple and precise interaction with the rest of the market. Leontief utilities, which seemed hard to manage, in fact enabled this in a very natural manner as described below.

Closed submarket: A closed submarket is a set S of agents satisfying the following: At every equilibrium of the complete market, the union of initial endowments of all agents in S exactly equals the union of optimal bundles of all these agents.

Observe that the agents in S will not be sequestered in any way — they are free to choose their optimal bundles from all the goods available. Yet, we will show that at equilibrium prices, they will only be exchanging goods among themselves. We note that the proof of PPAD-hardness for Nash equilibrium computation also uses small game gadgets to accomplish arithmetic operations of addition, multiplication and comparison [8, 12]; however, since there variables are captured through strategies of different players, these gadgets did not interfere. In our case, the primary challenge is to prevent flow of goods across gadgets and to ensure desired price dependencies even when same goods are used across gadgets. We achieve this through the notion of closed submarkets.

These closed submarkets enable us to ensure that variables denoting prices of goods satisfy specified arithmetic relations. The latter are linear function and product; we show that these two arithmetic relations suffice to encode any polynomial equation. Under linear functions, we want that $p_a = Bp_b + Cp_c + D$, where B, C and D are constants.

Under product, we want that $p_a = p_b \cdot p_c$. Designing this closed submarket, say $\mathcal{M}$, requires several ideas, which we now describe. $\mathcal{M}$ has an agent i whose initial endowment is one unit of good a and she desires only good c. We will ensure that the *amount* of good c leftover, after all other agents in the submarket consume what they want, is exactly p_b, i.e., the *price* of good b. At equilibrium, i must consume all the leftover good c, whose total cost is $p_b \cdot p_c$. Therefore the price of her initial endowment, i.e., one unit of good a, must be $p_b \cdot p_c$, hence establishing the required product relation. The tricky part is ensuring that exactly p_b amount of good c is leftover, without knowing what p_b will be at equilibrium. This is non-trivial, and this submarket needs to have several goods and agents in addition to the ones mentioned above.

---

[7]Since the payoff of the row player from a given strategy is a linear function of the variables denoting the probabilities played by the column player.

Once reduction $\mathcal{R}$ is established, FIXP-hardness follows from the straightforward observation that a 3-Nash instance can be encoded via polynomials, where each variable, which represents the probability of playing a certain strategy, is constrained in the interval $[0, 1]$. To get ETR-hardness, we appeal to the result of Schaefer and Štefankovič [38] that checking if a 3-Nash instance has a solution in a ball of radius half in l_∞-norm is ETR-hard; this entails constraining the variables to be in the interval $[0, 1/2]$. By Nash's theorem, in the former case, the market will admit an equilibrium and in the latter case, it will admit an equilibrium iff the 3-Nash instance has a solution in the ball of radius half in l_∞-norm. Membership in ETR follows by essentially showing a reduction in the reverse direction: given a Leontief market, we obtain a set of simultaneous multivariate polynomial equations whose roots capture its equilibria.

Our final result gives a polynomial time algorithm for computing an equilibrium for Arrow-Debreu exchange markets under Leontief utility functions provided the number of agents is a constant, say d. Using the property that equilibrium allocation of an agent can be written in terms of her equilibrium utility, we show that if equilibrium exists, then there is one where the number of goods with positive prices is at most d. Next we iterate over all subsets of size d of goods, and for each set we reduce the problem of checking existence of equilibrium to checking feasibility of a set of polynomial inequalities in $2d$ dimension. Since this can be done in polynomial time [2, 3], we get a polynomial time algorithm.

1.3 Organization of the Paper

Due to lack of space, we only give an overview of our results here and refer the reader to the full version of the paper for complete details and proofs. In Section 2.1 we define the Arrow-Debreu exchange market model, and the relevant utility functions. The definition of 3-player Nash equilibrium problem and its relation with the complexity classes FIXP and ETR are given in Section 2.2. Section 3 contains an overview of our second result where we show FIXP-hardness of computing an equilibrium in Leontief exchange markets. An overview of our first result where we show membership in PPAD for computing approximate equilibrium in exchange markets under PLC utilities is given in Section 4. Finally, Section 5 contains an overview of our third result where we give a polynomial time algorithm for computing an equilibrium for Arrow-Debreu exchange markets under Leontief utilities provided the number of agents is a constant.

2 PRELIMINARIES

2.1 The Arrow-Debreu Market Model

An Arrow-Debreu (AD) *exchange* market[8] [1] consists of a set $\mathcal{G}$ of divisible goods and a set $\mathcal{A}$ of agents. Let g denote the number of goods in the market. Each agent i comes with an initial endowment of goods; W_{ij} is amount of good j with agent i. The preference of an agent i over bundles of goods is captured by a non-negative, non-decreasing and concave utility function $U_i : \mathbb{R}_+^g \to \mathbb{R}_+$. Non-decreasingness is due to free disposal property, and concavity captures the law of diminishing marginal returns. Each agent wants to

---
[8]Refer to the full version for the definition of Arrow-Debreu markets with production firms.

buy a (optimal) bundle of goods that maximizes her utility to the extent allowed by her earned money from the initial endowment.

Let $\boldsymbol{p} \in \mathbb{R}_+^g$ denote the prices of goods, where p_j is the price of good j. Given $\boldsymbol{p}$, let $OPT_i(\boldsymbol{p}) = \arg\max_{\boldsymbol{y}}\{U_i(\boldsymbol{y}) \mid \sum_j y_j p_j = \sum_j W_{ij} p_j\}$ denote the set of optimal bundles of agent i. Let $\boldsymbol{x}_i \in \mathbb{R}^g$ denote the assignment of goods to agent i, where x_{ij} is the amount of good j.

If there is an assignment $\boldsymbol{x}_i \in OPT_i(\boldsymbol{p})$ to each agent i so that there is neither deficiency nor surplus of any good, then such prices are called *market clearing* or *market equilibrium* prices. Formally,

Definition 2.1 (Market Equilibrium). $(\boldsymbol{x}, \boldsymbol{p})$ is an equilibrium of an Arrow-Debreu exchange market $\mathcal{M}$ if

$$\forall i \in \mathcal{A} \quad : \quad \boldsymbol{x}_i = \arg\max_{\boldsymbol{y}}\{U_i(\boldsymbol{y}) \mid \sum_j y_j p_j \leq \sum_j W_{ij} p_j\}$$

$$\forall j \in \mathcal{G} \quad : \quad \sum_{i \in \mathcal{A}} x_{ij} = \sum_{i \in \mathcal{A}} W_{ij}.$$

The market equilibrium problem is to find such prices when they exist. In a celebrated result, Arrow and Debreu [1] proved that market equilibrium always exists under some mild conditions, however the proof is non-constructive and uses the machinery of Kakutani fixed point theorem. We note that an arbitrary market may not admit an equilibrium.

To work under finite precision it is customary to assume that utility functions are piecewise-linear concave.

2.1.1 Piecewise-Linear Concave (PLC) Utility Function.
The utility function U_i of agent i is said to be piecewise-linear concave (PLC) if at bundle $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{ig})$ it is given by

$$U_i(\boldsymbol{x}_i) = \min_k\{\sum_j U_{ij}^k x_{ij} + T_i^k\},$$

where U_{ij}^k's and T_i^k's are given non-negative rational numbers. Since the agent gets zero utility when she gets nothing, we have $U_i(\boldsymbol{0}) = 0$, and therefore at least one T_i^k is zero.

2.1.2 Leontief Utility Function.
It is a special subclass of PLC, where each good is required in a fixed proportion. Formally, it is given by:

$$U_i(\boldsymbol{x}_i) = \min_{j \in \mathcal{G}: A_{ij} > 0}\{\frac{x_{ij}}{A_{ij}}\}, \quad A_{ij} \geq 0.$$

In other words the agent wants good j in A_{ij} proportion. Clearly, the agent has to spend $\sum_j A_{ij} p_j$ amount of money to get one unit of utility. Thus, optimal bundle satisfies the following condition.

$$\forall j \in \mathcal{G}, \ x_{ij} = \beta_i A_{ij}, \quad \text{where} \ \beta_i = \frac{\sum_{j \in \mathcal{G}} W_{ij} p_j}{\sum_{j \in \mathcal{G}} A_{ij} p_j} \quad (1)$$

Since all market equilibria may be irrational even in the special case of Leontief utilities [19], it may not be possible to compute them exactly. Next we define a notion of approximate market equilibrium.

Definition 2.2 (ϵ-approximate market equilibrium). $(\boldsymbol{x}, \boldsymbol{p})$ is an ϵ-approximate equilibrium of an Arrow-Debreu exchange market $\mathcal{M}$ if each agent receives an approximately optimal bundle and

the aggregate demand of each good is approximately its aggregate supply, i.e.,

$$
\forall i \in \mathcal{A} \quad : \quad U_i(\boldsymbol{x}_i) \geq (1 - \epsilon)U_i(\tilde{\boldsymbol{x}}_i), \text{ where}
$$

$$
\tilde{\boldsymbol{x}}_i = \arg\max_{\boldsymbol{y}}\{U_i(\boldsymbol{y}) \mid \sum_j y_j p_j \leq \sum_j W_{ij}p_j\}
$$

$$
\forall i \in \mathcal{A} \quad : \quad \sum_{j \in \mathcal{G}} x_{ij}p_j \leq (1 + \epsilon)\sum_{j \in \mathcal{G}} W_{ij}p_j
$$

$$
\forall j \in \mathcal{G} \quad : \quad \sum_{i \in \mathcal{A}} x_{ij} \leq (1 + \epsilon)\sum_{i \in \mathcal{A}} W_{ij} .
$$

2.1.3 Sufficiency conditions. Market equilibrium may not exist, and it is NP-complete to decide whether there exists an equilibrium even in the exchange markets with SPLC utility functions [39]. Arrow-Debreu [1] gave the following sufficiency conditions for the existence of equilibrium: $W > 0$ and each agent is non-satiated. In case of PLC markets, non-satiation implies that for every k, there exists a j such that $U_{ij}^k > 0$.

2.2 3-Player Nash Equilibrium (3-Nash)

Given a 3-player finite game, let the set of strategies of player $p \in \{1, 2, 3\}$ be denoted by $\mathcal{S}_p$. Let $\mathcal{S} = \mathcal{S}_1 \times \mathcal{S}_2 \times \mathcal{S}_3$. Such a game can be represented by 3-dimensional tensors A_1, A_2 and A_3, representing payoffs of first, second and third players respectively. If players play $\boldsymbol{s} = (s_1, s_2, s_3) \in \mathcal{S}$, then the payoffs are $A_1(\boldsymbol{s})$, $A_2(\boldsymbol{s})$ and $A_3(\boldsymbol{s})$ respectively. Without loss of generality, we assume that $0 \leq A_p(\boldsymbol{s}) \leq 1, \forall p$.

Let Δ_p denote the probability distribution over set $\mathcal{S}_p$, $\forall p \in \{1, 2, 3\}$ (the set of mixed-strategies for player p), and let $\Delta = \Delta_1 \times \Delta_2 \times \Delta_3$. Given a mixed-strategy profile $\boldsymbol{z} = (z_1, z_2, z_3) \in \Delta$, let z_{ps} denote the probability with which player p plays strategy $s \in \mathcal{S}_p$, and let $\boldsymbol{z}_{-p}$ be the strategy profile of all the players at $\boldsymbol{z}$ except p. For player $p \in \{1, 2, 3\}$ the total payoff and payoff from strategy $s \in \mathcal{S}_p$ at $\boldsymbol{z}$ are respectively,

$$
\pi_p(\boldsymbol{z}) = \sum_{\boldsymbol{s} \in \mathcal{S}} A_p(\boldsymbol{s})z_{1s_1}z_{2s_2}z_{3s_3} \quad \text{and}
$$

$$
\pi_p(s, \boldsymbol{z}_{-p}) = \sum_{\boldsymbol{t} \in \mathcal{S}_{-p}} A_p(s, \boldsymbol{t})\prod_{q \neq p} z_{qt_q} .
$$

Definition 2.3 (Nash (1951) [34]). A mixed-strategy profile $\boldsymbol{z} \in \Delta$ is a Nash equilibrium (NE) if no player gains by deviating unilaterally. Formally, $\forall p = 1, 2, 3 \ \pi_p(\boldsymbol{z}) \geq \pi_p(z', \boldsymbol{z}_{-p}), \forall z' \in \Delta_p$.

Consider the following system of multivariate polynomials, where $A = (A_1, A_2, A_3)$:

$$
F_{NE}(A) : \begin{array}{ll}
\forall p \in \{1, 2, 3\}, & \sum_{s \in S_p} z_{ps} = 1 \\
\forall p \in \{1, 2, 3\}, \ s \in S_p, & \pi_p(s, \boldsymbol{z}_{-p}) + \beta_{ps} = \delta_p \\
\forall p \in \{1, 2, 3\}, \ s \in S_p, & z_{ps}\beta_{ps} = 0 \\
\forall p \in \{1, 2, 3\}, \ s \in S_p, & 0 \leq z_{ps} \leq 1 \\
\forall p \in \{1, 2, 3\}, \ s \in S_p, & 0 \leq \beta_{ps} \leq 1 \\
\forall p \in \{1, 2, 3\}, \ s \in S_p, & 0 \leq \delta_p \leq 1 .
\end{array} \quad (2)
$$

LEMMA 2.4. *Nash equilibria of A are exactly the solutions of system $F_{NE}(A)$, projected onto $\boldsymbol{z}$.*

Let 3-Nash denote the problem of computing a Nash equilibrium of a 3-player game. Next we describe its relation with the complexity classes FIXP and ETR.

2.3 The Class FIXP

The class FIXP was defined to capture complexity of the exact fixed point problems with algebraic solutions [21]. An instance I of FIXP consists of an algebraic circuit C_I defining a function $F_I : [0, 1]^d \to [0, 1]^d$, and the problem is to compute a fixed-point of F_I. The circuit is a finite representation of function F_I (like a formula), consisting of $\{\max, +, *\}$ operations, rational constants, and d inputs and outputs. We note that a circuit representing a problem in FIXP operates on real numbers.

In order to remain faithful to Turing machine computation, [21] also defined three discrete problems on FIXP, namely $FIXP_{pc}$ (partial computation), $FIXP_d$ (decision) and $FIXP_a$ ((strong) approximation). We refer to the full version for their definitions.

Whereas FIXP is a class of, in general, real-valued search problems, whose complexity can be studied in a real computation model, e.g., [5], note that $FIXP_{pc}$, $FIXP_d$ and $FIXP_a$ are classes of discrete search problems, hence their complexity can be studied in the standard Turing machine model. This is precisely the reason to define these three classes. [21] showed the following result.

THEOREM 2.5 (ETESSAMI-YANNAKAKIS (2010) [21]). *Given a 3-player game $A = (A_1, A_2, A_3)$, computing its NE is FIXP-complete. In particular, the corresponding Decision, (Strong) Approximation, and Partial Computation problems are complete respectively for $FIXP_d$, $FIXP_a$ and $FIXP_{pc}$.*

2.4 Existential Theory of Reals (ETR)

The class ETR was defined to capture the decision problems arising in *existential theory of reals* [38]. An instance I of class ETR consists of a sentence of the form

$$
(\exists x_1, \ldots, x_n)\phi(x_1, \ldots, x_n),
$$

where ϕ is a quantifier-free $(\wedge, \vee, \neg)$-Boolean formula over the predicates (sentences) defined by signature $\{0, 1, -1, +, *, <, \leq, =\}$ over variables that take real values. The question is if the sentence is true. The size of the problem is $n + size(\phi)$, where n is the number of variables and $size(\phi)$ is the minimum number of signatures needed to represent ϕ (we refer the readers to [38] for detailed description of ETR, and its relation with other classes like PSPACE). Schaefer and Štefankovič showed the following result; the first result on the complexity of a *decision* version of 3-Nash.

Nash equilibrium always exists [34], however there are many non-trivial *decision* questions.

Definition 2.6 (Decision 3-Nash). Decision 3-Nash is the problem of checking if a given 3-player game A admits a Nash equilibrium $\boldsymbol{z}$ such that $\boldsymbol{z} \leq 0.5$.

THEOREM 2.7 (SCHAEFER-ŠTEFANKOVIČ (2015) [38]). *Decision 3-Nash is ETR-complete.*

Note that changing the upper bound on all z_{ps}'s from 1 to 0.5 in $F_{NE}(A)$ (2), exactly captures the NE with $\boldsymbol{z} \leq 0.5$. Thus *Decision 3-Nash* can be reduced to checking if such a system of polynomials admits a solution. Next we show a construction of Leontief

exchange markets to exactly capture the solutions of a system of polynomials, similar to that of $F_{NE}(A)$, at its equilibria.

3 MULTIVARIATE POLYNOMIALS TO LEONTIEF EXCHANGE MARKET

Consider the following system of m multivariate polynomials on n variables $z = (z_1, \ldots, z_n)$:

$$F : \{f_i(z) = 0, \forall i \in [m]; \quad 0 \le L_j \le z_j \le U_j, \forall j \in [n]\} . \quad (3)$$

The coefficients of f_i's, and the upper and lower bounds U_j's and L_j's are assumed to be rational numbers. In this section we show that solutions of F can be captured as equilibrium prices of a Leontief exchange market. The problems of 3-Nash and Decision 3-Nash can be characterized by a set similar to (3) (Lemma 2.4), in turn we obtain FIXP and ETR hardness results for Leontief exchange markets, from the corresponding hardness of 3-Nash (Theorems 2.5 and 2.7).

Polynomial f_i is represented as sum of monomials, and a monomial $\alpha z_1^{d_1} \ldots z_n^{d_n}$ is represented by tuple $(\alpha, d_1, \ldots, d_n)$; here coefficient α is a rational number.[9] Let $\mathcal{M}_{f_i}$ denote the set of monomials of f_i, and $size[f_i] = \sum_{(\alpha,d) \in \mathcal{M}_{f_i}} size(\alpha, d)$, where $size(r)$ for a rational number r is the minimum number of bits needed to represent its numerator and denominator. Degree of f_i is $deg(f_i) = \max_{(\alpha,d) \in \mathcal{M}_{f_i}} \sum_j d_j$. Size of F, denoted by $size[F]$, is $m + n + \sum_j (size(U_j) + size(L_j)) + \sum_i (deg(f_i) + size[f_i])$. Given F, next we construct an exchange market in time polynomial in $size[F]$, whose equilibria correspond to solutions of F.

Preprocessing. First we transform system F into a polynomial sized equivalent system that uses only the following basic operations on *non-negative* variables (refer to the full version for more details).

$$\begin{array}{ll} \text{(LIN.)} & z_a = Bz_b + Cz_c + D, \quad \text{where} \ B, C, D \ge 0 \\ \text{(QD.)} & z_a = z_b * z_c \end{array} \quad (4)$$

Let $R(F)$ be a reformulation of F using these basic operations. All variables in $R(F)$ are constrained to be non-negative. In order to construct $R(F)$ from F, we need to introduce many auxiliary variables. Let the number of variables in $R(F)$ be N, and out of these let $z_1, \ldots, z_n$ be the original set of variables of F (3). Given a system $R(F)$ of equalities, we will construct an exchange market $\mathcal{M}$, such that value of each variable z_j, $j \in [N]$ is captured as price p_j of good G_j in $\mathcal{M}$. Further, we make sure that these prices satisfy all the relations in $R(F)$ at every equilibrium of $\mathcal{M}$.

Ensuring scale invariance. Since equilibrium prices of an exchange market are scale invariant, the relations that these prices satisfy have to be scale invariant too. However note that in (4) (LIN.) and (QD.) are not scale invariant. To handle this we introduce a special good G_s, such that when its price p_s is set to 1 we get back the original system.

$$\begin{array}{ll} \text{(LIN.)} & p_a = Bp_b + Cp_c + Dp_s, \quad \text{where} \ B, C, D \ge 0 \\ \text{(QD.)} & p_a = \dfrac{p_b * p_c}{p_s} \end{array} \quad (5)$$

[9]In fact, our reduction is also applicable to a succinct representation of polynomials, e.g., $f_i = (x + 1)^d$, however the monomial representation is enough to obtain the hardness result due to (2).

Let $R'(F)$ be a system of equalities after applying the transformation of (5) to $R(F)$. Note that, $R'(F)$ has exactly one extra variable than $R(F)$, namely p_s, and solutions of $R'(F)$ with $p_s = 1$ are exactly the solutions of $R(F)$. Let the size of $R'(F)$ be (# variables + # relations in $R'(F)$ + $size(B, C, D)$ in each of (LIN.)-type relations). To bound the values at a solution of $R'(F)$, define

$$H = M_{max} U_{max}^d + 1, \quad \text{where} \ M_{max} = \max_i |\mathcal{M}_{f_i}|,$$

$$d = \max_{f_i} deg(f_i), \ \text{and} \ U_{max} = \max\{\max_j U_j, \max_{f_i, (\alpha, d) \in \mathcal{M}_{f_i}} |\alpha|\}.$$

LEMMA 3.1. $size[R'(F)] = poly(size[F])$. *Vector* p *is a non-negative solution of* $R'(F)$ *with* $p_s = 1$ *iff* $z_j = p_j$, $\forall j \in [n]$ *is a solution of* F. *Further,* $p_j \le H$, $\forall j \in [N]$.

3.1 Market Construction

In this section we construct market $\mathcal{M}$ consisting of goods $G_1, \ldots, G_N$ and G_s, such that the prices $p_1, \ldots, p_N$ and p_s, satisfy all the relations of $R'(F)$ at equilibrium. To ensure $p_s > 0$ at equilibrium, we add the following agent to $\mathcal{M}$. Recall that W_{ij} is the amount of good G_j agent A_i brings to the market, x_i is the bundle of goods consumed by her, and $U_i : \mathbb{R}_+^g \to \mathbb{R}_+$ is her utility function.

$$A_s : \ W_{ss} = 1, \ W_{sj} = 0, \ \forall j \in [N]; \quad U_s(x_s) = x_{ss} \quad (6)$$

LEMMA 3.2. *At every equilibrium of market* $\mathcal{M}$, *we have* $p_s > 0$, *and* $x_{ss} = W_{ss}$.

Since a price p_j may be used in multiple relations of $R'(F)$, the corresponding good has to be used in many different gadgets. When we combine all these gadgets to form market $\mathcal{M}$, the biggest challenge is to analyze the flow of goods among these gadgets at equilibrium. We overcome this all together by forming *closed submarket* for each gadget.

Definition 3.3 (Submarket). A submarket $\widetilde{\mathcal{M}}$ of a market $\mathcal{M}$ consists of a subset of agents and goods such that endowment and utility functions of agents in $\widetilde{\mathcal{M}}$ are defined over goods only in $\widetilde{\mathcal{M}}$.

Definition 3.4 (Closed Submarket). A submarket $\widetilde{\mathcal{M}}$ of a market $\mathcal{M}$ is said to be closed if at every equilibrium of the entire market $\mathcal{M}$, the submarket $\widetilde{\mathcal{M}}$ is locally at equilibrium, i.e., its total demand equals its total supply. The total demand of $\widetilde{\mathcal{M}}$ is the sum of demands of agents in $\widetilde{\mathcal{M}}$ and its total supply is the sum of initial endowments of agents in $\widetilde{\mathcal{M}}$.

In other words, $\widetilde{\mathcal{M}}$ does not interfere with the rest of the market in terms of supply and demand, even if some goods in $\widetilde{\mathcal{M}}$ are used outside as well. Note that the market of (6) is a closed submarket (Lemma 3.2) with only one agent A_s and one good G_s. We will see that each submarket $\widetilde{\mathcal{M}}$ establishing a relation of (LIN.) and (QD.) has a set of *exclusive* goods used only in $\widetilde{\mathcal{M}}$, to achieve the *closed* property. Before describing construction of closed submarkets for more involved relations, first we describe it for a simple *equality* relation.

3.1.1 Submarket for relation (EQ.) $p_a = p_b$. The gadget for (EQ.) consists of two agents with Leontief utility functions, as given in Table 1, where good G_r is exclusive to this submarket. The endowment vector W_i of agent A_i should be interpreted as (amount

of G_a, amount of G_b, amount of G_r), i.e., in the same order of goods as listed on the first line of the table.

Table 1: Closed submarket (EQ.) $p_a = p_b$

$\mathcal{M}_{EQ}$:	2 Agents (A_1, A_2) and 3 Goods (G_a, G_b, G_r)
	(G_r: an exclusive good)
	A_1: $W_1 = (0, 1, 1)$ and $U_1(\boldsymbol{x}) = \min\{x_a, x_r\}$
	A_2: $W_2 = (1, 0, 1)$ and $U_2(\boldsymbol{x}) = \min\{x_b, x_r\}$

LEMMA 3.5. *The market $\mathcal{M}_{EQ}$ of Table 1: (i) is a closed submarket, (ii) at equilibrium, it enforces $p_a = p_b$, and (iii) every non-negative solution of $p_a = p_b$ gives an equilibrium.*

PROOF. Let α and β denote the utility obtained by A_1 and A_2 at equilibrium respectively. Then using (1) which characterizes optimal bundles for Leontief functions, the market clearing conditions of the two agents give: $p_b + p_r = \alpha(p_a + p_r)$ and $p_a + p_r = \beta(p_b + p_r)$.

Clearly these conditions imply that $\alpha\beta = 1 \Rightarrow \beta = 1/\alpha$. Note that A_1 and A_2 consume α and β amounts of good G_r respectively. And since this good is exclusive to $\mathcal{M}_r$, no other agent will consume it. Further, there are exactly two units of G_r available in the entire market $\mathcal{M}$. Hence we get $\alpha + \beta \le 2$. Replacing $\beta = \frac{1}{\alpha}$ gives $(\alpha - 1)^2 \le 0 \Rightarrow \alpha = \beta = 1$. Therefore, we get that every equilibrium of $\mathcal{M}_r$ enforces $p_a + p_r = p_b + p_r \Rightarrow p_a = p_b$. Further, $\mathcal{M}_r$ is a closed submarket because at equilibrium, demand of every good *in* $\mathcal{M}_r$ is equal to its supply *in* $\mathcal{M}_r$ even though every good except G_r might participate in the rest of the market as well. For the last part, if $p_a = p_b \ge 0$, then choosing $p_r = 1$, and $x_{1a} = x_{1r} = x_{2b} = x_{2r} = 1$ gives a market equilibrium of $\mathcal{M}_r$. □

We refer the reader to full version for the (LIN.) submarket, which is an extension of (EQ.).

3.1.2 Submarket for Relation (QD.) $p_a = \frac{p_b p_c}{p_s}$. The market construction is quite involved, so we simplify it using the two assumptions (refer to the full version for complete details). First, that $p_s = 1$ and second, that $p_b \ne 0$.

As mentioned earlier, establishing this relation in the market turns out to be complex, even with the above two assumptions. For this, we need to make sure that at every equilibrium price p_a of a good a is same as the product of prices of two other goods b and c. One way to establish this is by creating an agent who brings 1 unit of good a and desires only good c. For this to work, the challenge is to ensure that the amount of good c leftover, after all other agents in the submarket consume what they want, is exactly p_b, without knowing what p_b will be at equilibrium. We show that this is indeed possible, but for this we need to create several gadgets and combine them in a particular way.

In order to present the submarket in a modular manner, we will first define some devices. Each of these devices will be implemented via a set of agents with Leontief utility functions. Each device ensures a certain relationship between the net endowment left over by these agents and the net consumption of these agents; for convenience, we will call these the *net endowment and net consumption of the device*. Clearly, at equilibrium prices, for each device, the total

worth of its net endowment and net consumption must be equal. The first device converts price of a good to amount of another good whose price is one.

Converter (Conv(q)): The net consumption of this device is 1 unit of good G_1, whose price is p, and the net endowment is p/q units of good G_2, whose price is q. Table 2 and Figure 1 illustrate the implementation. In the figure tuple on edges represent *(amount, price)* of the goods whose number is shown in circle. Table 2 has two parts: Part 1 describes the market and Part 2 enforces linear relations using (LIN.) submarkets.
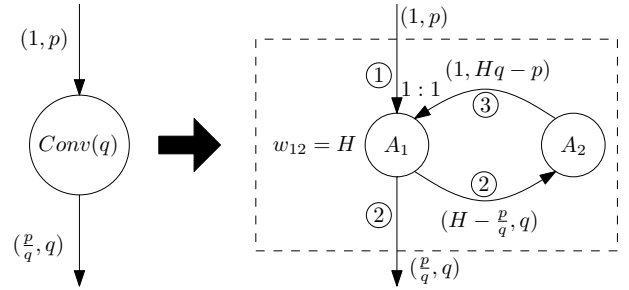

Figure 1: Flow of goods in Part 1 of Table 2 for Conv(q). Wires are numbered in circle, and wire i carries good G_i. The tuple on each wire represents (amount, price).

Table 2: A closed submarket for Conv(q)

Part 1:	**Input:** 1 unit of G_1 at price p
	Output: p/q units of G_2 at price q
	2 Agents (A_1, A_2), 3 goods (G_1, G_2, G_3)
	A_1: $W_{12} = H$ and $U_1(\boldsymbol{x}) = \min\{x_1, x_3\}$
	A_2: $W_{23} = 1$ and $U_2(\boldsymbol{x}) = x_2$
Part 2:	Closed submarkets for these linear relations
	$p_2 = q$
	$p_3 = Hq - p$

There are two agents A_1 and A_2, and three goods G_1, G_2 and G_3. A_1 brings H units of G_2, whose price is set to q (H is a constant defined in Section 3). A_1 wants to consume G_1 and G_3 in the ratio of 1:1. The net consumption of this device, i.e., 1 unit of G_1 at price p, is consumed by A_1. A_2 brings 1 unit of G_3, whose price is enforced to $Hq - p$. A_2 wants to consume only G_2, hence it consumes $H - p/q$ units of G_2 (observe that there is no need to perform the division involved in p/q explicitly). The remaining p/q units of G_2 form the net endowment of the device, as required.

We will use *Conv* to convert price p_c to endowment of a good with price 1. In order to convert this endowment to an endowment of a good with price p_b and to make the entire submarket closed, we need the following two more devices. Their construction is a bit more involved, and we refer the reader to full version for their complete details.

Combiner (Comb(l, p_a, p_b)): The net consumption is l units each of goods G_1 and G_2, whose prices are p_a and p_b, respectively. The net endowment is l units of a good G_3, whose price is $p_a + p_b$.

Table 3: A closed submarket for Comb(l, p_a, p_b)

Part 1:	**Input:** l units of G_1 and G_2 at price p_a and p_b **Output:** l units of G_3 at price $p_a + p_b$ 3 Agents (A_1, A_2, A_3), 5 goods $(G_1, G_2, G_3, G_4, G_5)$ A_1: $W_{14} = 1$ and $U_1(\boldsymbol{x}) = \min\{x_1, x_2\}$ A_2: $W_{23} = H$ and $U_2(\boldsymbol{x}) = \min\{x_4, x_5\}$ A_3: $W_{35} = 1$ and $U_2(\boldsymbol{x}) = x_3$
Part 2:	Closed submarkets for these linear relations $p_3 = p_a + p_b$ $p_5 = Hp_a + Hp_b - p_4$

Splitter (Spl(l, p_a, p_b)): The net endowment is l units each of two goods G_2 and G_3, whose prices are p_a and p_b, respectively. The net consumption is l units of Good 1, whose price is $p_a + p_b$.

Table 4: A closed submarket for Spl(l, p_a, p_b)

Part 1:	**Input:** l units of G_1 at price $p_a + p_b$ **Output:** l units of G_2 and G_3 at price p_a and p_b 3 Agents (A_1, A_2, A_3), 5 goods $(G_1, G_2, G_3, G_4, G_5)$ A_1: $W_{14} = 1$ and $U_1(\boldsymbol{x}) = x_1$ A_2: $W_{22} = W_{23} = H$ and $U_2(\boldsymbol{x}) = \min\{x_4, x_5\}$ A_3: $W_{35} = 1$ and $U_2(\boldsymbol{x}) = \min\{x_2, x_3\}$
Part 2:	Closed submarkets for these linear relation $p_2 = p_a$ $p_3 = p_b$ $p_5 = Hp_a + Hp_b - p_4$

Submarket construction for $p_a = p_b p_c$: Consider the submarket given in Table 5 and Figure 2. In this market, the 7 goods, $G_1, \ldots G_7$ are exclusive to the submarket. The prices of some goods are set using $(LIN.)$ relations as specified in Part 2. The submarket uses 2 Converters, 1 Combiner and 1 Splitter. Each device is specified by its (net endowment, net consumption). In addition to the agents needed for implementing these devices, the submarket requires 2 additional agents, A_1 and A_2.

LEMMA 3.6. *The submarket of Table 5 enforces $p_a = p_b p_c$ and is closed under assumption $p_b \neq 0$.*

For each relation r of $R'(F)$, depending on its type, construct a closed submarket $\mathcal{M}_r$ as described in Section 3.1. Combine all the $\mathcal{M}_r$'s and add the agent of equation (6) to form one market $\mathcal{M}$.

Next we prove the main theorem of this section which will give all the desired hardness results.

THEOREM 3.7. *Equilibrium prices of market $\mathcal{M}$, projected onto $(p_1, \ldots, p_n)$, are in one-to-one correspondence with the solutions of F. Furthermore, $size[\mathcal{M}] = poly(size[F])$.*
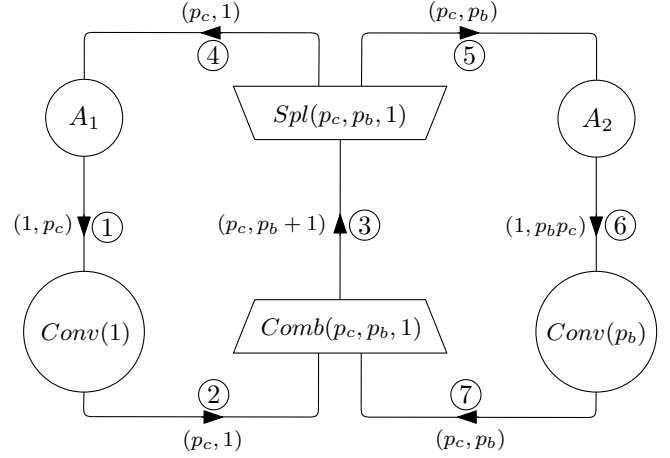

Figure 2: Flow of goods in Part 1 of Table 5. Wires are numbered, and wire i carries good G_i. The tuple on each wire represents (amount, price).

Table 5: A closed submarket for $p_a = p_b p_c$

Part 1:	2 Agents (A_1, A_2), 2 Converters $(Conv_1, Conv_2)$, 1 Combiner $(Comb)$, 1 Splitter (Spl), and 7 Goods $(G_1, \ldots, G_7)$ A_1: $W_{11} = 1$ and $U_1(\boldsymbol{x}) = x_4$ A_2: $W_{26} = 1$ and $U_2(\boldsymbol{x}) = x_5$ $Conv_1 = Conv(1)$: (G_1, G_2) $Conv_2 = Conv(p_b)$: (G_6, G_7) $Comb(p_c, p_b, 1)$: $((G_2, G_7), G_3)$ $Spl(p_c, p_b, 1)$: $(G_3, (G_4, G_5))$
Part 2:	Closed submarkets for these linear relations $p_1 = p_c$ $p_2 = 1$ $p_4 = 1$ $p_5 = p_b$ $p_6 = p_a$ $p_7 = p_b$

Theorem 3.7 shows that finding solutions of F can be reduced to finding equilibria of a Leontief exchange market. As discussed in Section 2.2, the problem of computing a NE of a 3-player game A can be formulated as finding a solution of system $F_{NE}(A)$ (2) of polynomials in which variables take values in $[0, 1]$ (Lemma 2.4). Note that $size[F_{NE}(A)] = O(size(A))$. The next theorem follows using the formulation of (2), together with Lemma 2.4, and Theorems 2.5, 2.7 and 3.7.

THEOREM 3.8. *Computing an equilibrium of an exchange market under Leontief utility functions is FIXP-hard. In particular, the*

corresponding Decision, (Strong) Approximation, and Partial Computation problems are hard for $FIXP_d$, $FIXP_a$ and $FIXP_{pc}$, respectively. Furthermore, checking existence of an equilibrium in an arbitrary Leontief exchange market (and in market with PLC utilities) is ETR-complete[10].

4 MEMBERSHIP IN PPAD

In this section, we show that computing an approximate equilibrium in an exchange market with PLC utilities is in PPAD. This resolves an open question of [39].

The problem of computing an approximate market equilibrium under sufficiency conditions has been known to be PPAD-hard [7, 11, 13, 28] for more than a decade; however, membership in PPAD has not been established yet. The only fixed point formulation known for this problem was obtained in the context of proving membership in FIXP [23]. We use this formulation to show the result. Let $\mathcal{M}$ denote an exchange market with PLC utilities, and $size(\mathcal{M})$ denote the bit length of input parameters of $\mathcal{M}$. Let F be the fixed point formulation for $\mathcal{M}$ given in [23]. For this, we need to show the following:

(1) F is Lipschitz continuous with constant K, where $size(K) = poly(size(\mathcal{M}))$.
(2) δ-approximate fixed point of F gives an ϵ-approximate equilibrium, where $size(\delta) = poly(size(\epsilon, \mathcal{M}))$.

The first task is easy to show and it implies that finding an ϵ-approximate fixed point of F is in PPAD [36]. Showing the second step is quite involved and technical because F is rather intricate. For the second task, we need to show approximate market clearing of every good and approximately optimal bundle to each agent at an approximate fixed point of F.

There is a simple linear program (LP) that captures optimal bundles of each agent at a given price vector. Using this LP and its dual, F captures optimal bundles of each agent as feasibility of primal and dual constraints, and complementary slackness conditions. We show that at a δ-approximate fixed point of F, both feasibility and complementary slackness constraints are approximately satisfied. This essentially proves that each agent gets an approximately optimal bundle.

Next we briefly describe the description of fixed point formulation F, given in [23]. Recall the market parameters from Section 2.1. Given prices $\boldsymbol{p}$, the optimal utility of agent i is a solution of the following LP, where variables $\boldsymbol{x} = \{x_{ij} \mid i \in \mathcal{A}, j \in \mathcal{G}\}$ capture the assignment of goods to agents, and λ_i's and γ_i^k's are dual variables.

$$
\begin{aligned}
&\max\ u_i & &\min\ \lambda_i \sum_j W_{ij} p_j + \sum_k T_i^k \gamma_i^k \\
&\forall k:\ u_i \le \sum_j U_{ij}^k x_{ij} + T_i^k & &\forall j:\ \sum_k U_{ij}^k \gamma_i^k \le \lambda_i p_j \\
&\sum_j x_{ij} p_j \le \sum_j W_{ij} p_j & \xleftrightarrow{dual} & & (7) \\
&\forall j:\ x_{ij} \ge 0 & &\sum_k \gamma_i^k = 1 \\
& & &\lambda_i \ge 0;\ \forall k:\ \gamma_i^k \ge 0
\end{aligned}
$$

---

[10]Membership in ETR is obtained by characterizing the set of equilibria as simultaneous solutions of a set of multivariate polynomial equations (refer to the full version for details).

Without loss of generality, we may assume that the total initial endowment of every good is 1, i.e, $\sum_{i \in \mathcal{A}} W_{ij} = 1, \forall j \in \mathcal{G}$. Let $m \stackrel{\text{def}}{=} |\mathcal{A}|$, and $n \stackrel{\text{def}}{=} |\mathcal{G}|$. Let H denote the maximum number of hyperplanes in an agent's PLC utility function, and wlog we may assume that it is same for each agent. Let $\boldsymbol{\lambda} = \{\lambda_i \mid i \in \mathcal{A}\}$ and $\boldsymbol{\gamma} = \{\gamma_i^k \mid i \in \mathcal{A}, k \in [H]\}$. Let $x_{max} \stackrel{\text{def}}{=} 1.1$, $W_{min} \stackrel{\text{def}}{=} \min_{(i,j)} W_{ij}$, $U_{max} \stackrel{\text{def}}{=} \max_{(i,j,k)} U_{ij}^k$, $U_{min} \stackrel{\text{def}}{=} \min_{(i,j,k):U_{ij}^k>0} U_{ij}^k$, $T_{max} \stackrel{\text{def}}{=} \max_{(i,k)} T_i^k$, and $\lambda_{max} \stackrel{\text{def}}{=} 2n(U_{max}+T_{max})/W_{min}$. Note that $W_{min} > 0$ under sufficiency conditions.

Let $D \stackrel{\text{def}}{=} \{(\boldsymbol{p}, \boldsymbol{x}, \boldsymbol{\gamma}, \boldsymbol{\lambda}) \in \mathbb{R}_+^N \mid \sum_j p_j = 1;\ x_{ij} \le x_{max};\ \sum_k \gamma_i^k = 1;\ \lambda_i \le \lambda_{max}\}$, where N is the total number of variables, and $F : D \to D$ is a continuous function such that $(\overline{\boldsymbol{p}}, \overline{\boldsymbol{x}}, \overline{\boldsymbol{\gamma}}, \overline{\boldsymbol{\lambda}}) \stackrel{\text{def}}{=} F(\boldsymbol{p}, \boldsymbol{x}, \boldsymbol{\gamma}, \boldsymbol{\lambda})$ as given in Table 6.

Table 6: FIXP Circuit for Exchange Markets

$$
\begin{aligned}
\overline{p}_j &= \frac{p_j + \max\{\sum_i x_{ij} - 1, 0\}}{\sum_j (p_j + \max\{\sum_i x_{ij} - 1, 0\})} \\[2mm]
\overline{\gamma}_i^k &= \frac{\gamma_i^k + \max\{u_i - \sum_j U_{ij}^k x_{ij} - T_i^k, 0\}}{\sum_k (\gamma_i^k + \max\{u_i - \sum_j U_{ij}^k x_{ij} - T_i^k, 0\})} \\[2mm]
\overline{x}_{ij} &= \min\left\{\max\left\{x_{ij} + \sum_k U_{ij}^k \gamma_i^k - \lambda_i p_j, 0\right\}, x_{max}\right\} \\[2mm]
\overline{\lambda}_i &= \min\left\{\max\left\{\lambda_i + \sum_j x_{ij} p_j - \sum_j W_{ij} p_j, 0\right\}, \lambda_{max}\right\}
\end{aligned}
$$

THEOREM 4.1 (GARG-MEHTA-VAZIRANI (2014) [23]). *Assuming sufficient conditions of the existence of equilibrium, every fixed point of F gives an equilibrium of $\mathcal{M}$ and vice versa.*

LEMMA 4.2. *F is Lipschitz continuous with constant K, where $size(K) = poly(size(\mathcal{M}))$.*

4.1 Approximate Market Clearing

In this section, we show that market clears approximately at an approximate fixed point of F. Let $\boldsymbol{y} = (\boldsymbol{p}, \boldsymbol{x}, \gamma, \boldsymbol{\lambda})$. First we define ϵ-approximate fixed point of F.

Definition 4.3 (ϵ-Approximate Fixed Point). $\boldsymbol{y}$ is an ϵ-approximate fixed point of F if $\|F(\boldsymbol{y}) - \boldsymbol{y}\|_\infty \le \epsilon$.

LEMMA 4.4. *At an ϵ-approximate fixed point $(\boldsymbol{x}, \boldsymbol{p}, \boldsymbol{\lambda}, \boldsymbol{\gamma})$ of F,*
- $|\sum_j x_{ij} p_j - \sum_j W_{ij} p_j| \le \epsilon,\ \forall i \in \mathcal{A}$
- $\sum_i x_{ij} \le 1 + 5mn\sqrt{\epsilon n},\ \forall j \in \mathcal{G}$.

4.2 Approximately Optimal Bundle

In this section, we show that each agent gets an approximately optimal bundle at an approximate fixed point of F. We achieve this by showing that feasibility and complementary slackness of (7) are approximately satisfied.

LEMMA 4.5. *At an ϵ-approximate fixed point $(\boldsymbol{x}, \boldsymbol{p}, \boldsymbol{\lambda}, \boldsymbol{\gamma})$ of F,*
- $\forall(i, j):$ If $x_{ij} > \epsilon$ then $\lambda_i p_j \le \sum_k U_{ij}^k \gamma_i^k + \epsilon$

- $\forall (i,j) : \lambda_i p_j \geq \sum_k U_{ij}^k \gamma_i^k - \epsilon$
- $\forall (i,k) : u_i - \sum_j U_{ij}^k x_{ij} - T_i^k \leq 12nH^2 U_{max}^2 x_{max}\sqrt{\epsilon}$
- $\forall (i,k) : $ If $\gamma_i^k > \sqrt[4]{\epsilon}$ then
  $u_i \geq \sum_j U_{ij}^k x_{ij} + T_i^k - 24nH^2 U_{max}^2 x_{max}(nx_{max}+\lambda_{max})\sqrt[4]{\epsilon}.$

Lemmas 4.4 and 4.5 give

THEOREM 4.6. *At an ϵ^8-approximate fixed point of F, where $\epsilon < $*
$1/24mnH^2 U_{max}^2 x_{max}(nx_{max}+\lambda_{max})$,

(1) $\forall i : |\sum_j x_{ij}p_j - \sum_j W_{ij}p_j| \leq \epsilon$
(2) $\forall j : \sum_i x_{ij} \leq 1 + \epsilon, \forall j$
(3) $\forall i : u_i \leq \sum_j U_{ij}^k x_{ij} + T_i^k + \epsilon, \forall i$
(4) $\forall (i,j) : \lambda_i p_j \geq \sum_k U_{ij}^k \gamma_i^k - \epsilon, \forall (i,j)$
(5) $\forall (i,j) : $ If $x_{ij} > \epsilon$ then $\lambda_i p_j \leq \sum_k U_{ij}^k \gamma_i^k + \epsilon$
(6) $\forall (i,k) : $ If $\gamma_i^k > \epsilon$ then $u_i \geq \sum_j U_{ij}^k x_{ij} + T_i^k - \epsilon.$

Let u_i^1 be the optimal utility of (7). For a given ϵ, consider the following modified LP, where feasibility constraints of (7) are perturbed:

$$\max u_i - \epsilon \sum_j x_{ij}$$
$$\forall k : u_i \leq \sum_j U_{ij}^k x_{ij} + T_i^k + \epsilon \tag{8}$$
$$\sum_j x_{ij}p_j \leq \sum_j W_{ij}p_j + \epsilon$$
$$\forall j : x_{ij} \geq 0 .$$

The dual is

$$\min \lambda_i \sum_j W_{ij}p_j + \sum_k \gamma_i^k (T_i^k + \epsilon)$$
$$\forall j : \sum_k U_{ij}^k \gamma_i^k \leq \lambda_i p_j + \epsilon$$
$$\sum_k \gamma_i^k = 1$$
$$\lambda_i \geq 0; \forall k : \gamma_i^k \geq 0$$

Let u_i^2 be the optimal utility of (8).

LEMMA 4.7. $u_i^2 \geq u_i^1 - \epsilon n x_{max}, \forall i.$

PROOF. Let $\dot{x}$ and $\ddot{x}$ be optimal solutions of (7) and (8) respectively. Since $\dot{x}$ is a feasible point in (8), we have $u_i^2 - \epsilon \sum_j \ddot{x}_{ij} \geq u_i^1 - \epsilon \sum_j \dot{x}_{ij}$, that implies $u_i^2 - u_i^1 \geq \epsilon(\sum_j \dot{x}_{ij} - \sum_j \ddot{x}_{ij}) \geq \epsilon n x_{max}.$ □

Suppose we have a candidate point $(\tilde{x}, \tilde{p}, \tilde{\lambda}, \tilde{\gamma})$, which satisfies all feasibility constraints of (8) but approximately satisfies complementary slackness constraints as follows:

$$\forall (i,j) : \text{ If } \tilde{x}_{ij} > \epsilon \text{ then } \tilde{\lambda}_i \tilde{p}_j \leq \sum_k U_{ij}^k \tilde{\gamma}_i^k + \epsilon$$
$$\forall (i,k) : \text{ If } \tilde{\gamma}_i^k > \epsilon \text{ then } u_i^3 \geq \sum_j U_{ij}^k \tilde{x}_{ij} + T_i^k - \epsilon$$
$$\forall i : \text{ If } \tilde{\lambda}_i > \epsilon \text{ then } \sum_j \tilde{x}_{ij}\tilde{p}_j \geq \sum_j W_{ij}\tilde{p}_j - \epsilon,$$

where u_i^3 is the utility obtained at this point. At prices $p = \tilde{p}$, let u_i^1 and u_i^2 be the optimal value of (7) and (8) respectively. Next we show that $\tilde{x}$ gives an approximately optimal utility at $\tilde{p}$.

LEMMA 4.8. $\forall i : u_i^3 \geq u_i^1 - \epsilon(2 + 2\lambda_{max} + 4nx_{max} + HT_{max} + nHx_{max}U_{max}).$

Finally, using Theorem 4.6 and Lemma 4.8, we get

THEOREM 4.9. *At an ϵ^{16}-approximate fixed point (x, p, γ, λ) of F, where $\epsilon < 1/(24mn^2 H^2 U_{max}^2 x_{max}^2 \lambda_{max} T_{max})$, we have*

(i) $\forall i : \sum_j x_{ij}p_j \leq \sum_j W_{ij}p_j + \epsilon$
(ii) $\forall j : \sum_i x_{ij} \leq 1 + \epsilon$
(iii) $\forall i : u_i \geq u_i^{opt} - \epsilon,$

where u_i^{opt} is the optimal utility of agent i at prices p.

LEMMA 4.10. *For an ϵ, if we have a solution (x, p) such that*

(i) $\forall i : \sum_j x_{ij}p_j \leq \sum_j W_{ij}p_j + \epsilon$
(ii) $\forall j : \sum_i x_{ij} \leq 1 + \epsilon$
(iii) $\forall i : u_i \geq u_i^{opt} - \epsilon,$

then it gives an ϵ'-approximate equilibrium, where $\epsilon' = U_{min}W_{min}\epsilon.$

From Theorem 4.9, Theorem 4.1 and Lemma 4.10, we get

THEOREM 4.11. *Assuming sufficiency conditions for the existence of equilibrium, for any $0 < \epsilon < 1$, an ϵ-approximate equilibrium of exchange markets with piecewise-linear concave utilities can be obtained from a δ-approximate fixed point of F, where $\delta = (\frac{\epsilon}{\Lambda})^{16}$ and $\Lambda = \frac{24mn^2 H^2 U_{max}^2 x_{max}^2 \lambda_{max} T_{max}}{W_{min}U_{min}}.$*

Note that finding an ϵ-approximate fixed point of a Lipschitz-continuous function from a convex compact domain to itself is in PPAD [36]. Using Lemma 4.2 and Theorem 4.11, together with [36], we get

THEOREM 4.12. *Assuming sufficiency conditions for the existence of equilibrium, finding an approximate equilibrium in exchange markets with PLC utilities is in PPAD.*

Since finding an n^{-13}-approximate equilibrium of SPLC markets is PPAD-hard [7], we get

THEOREM 4.13. *Finding an n^{-13}-approximate equilibrium of exchange markets with PLC utilities is PPAD-complete, where number of agents and goods are $O(n)$.*

5 LEONTIEF UTILITIES UNDER CONSTANT NUMBER OF AGENTS

In this section, we show that there is a polynomial time algorithm for finding an equilibrium in Arrow-Debreu exchange markets under Leontief utility functions provided the number of agents is a constant. This settles part of an open problem of Devanur and Kannan [15]. Consider a Leontief exchange market with n goods and d agents, where d is a constant. The Leontief utility function of agent i is given by

$$U_i(x_i) = \min_{j=1}^n \left\{ \frac{x_{ij}}{A_{ij}} \right\},$$

where $A_{ij} \geq 0$ is the fraction of good j that agent i wants. Let W_{ij} be the amount of good j agent i owns. We assume wlog that

$\sum_i W_{ij} = 1$, $\forall j$. Let us capture the equilibrium utility of agent i in variable β_i, then the optimal bundle condition gives,

$$x_{ij} = A_{ij}\beta_i \tag{9}$$

at an equilibrium. Further, if $(p_1, \ldots, p_n)$ are corresponding equilibrium prices, then the market clearing conditions can be written as,

$$\forall j \in \mathcal{G}: \qquad \sum_i x_{ij} = \sum_i A_{ij}\beta_i \leq 1 \tag{10}$$
$$\text{if } p_j > 0 \text{ then } \sum_i A_{ij}\beta_i = 1 .$$

Further, since Leontief utility function is non-satiated (given any bundle, there exists another bundle where utility increases), the agents will spend all of their earned money. This gives the following relation in $\boldsymbol{\beta}$ and $\boldsymbol{p}$:

$$\forall i \in \mathcal{A}: \sum_j p_j x_{ij} = \sum_j W_{ij}p_j \Rightarrow \beta_i = \frac{\sum_j W_{ij}p_j}{\sum_j A_{ij}p_j} .$$

First, we show that if there is an equilibrium, then there is one where at most d prices are non-zero.

LEMMA 5.1. *If an exchange market with Leontief utilities has an equilibrium, then there is one where at most d goods have non-zero prices.*

Due to Lemma 5.1, to find an equilibrium, it suffices to check for every set S of d goods if there is an equilibrium by setting the prices of goods outside S to zero. And this can be achieved by checking the feasibility of the following system, where β_is and p_js are variables.

$$\forall j \notin S, \ p_j = 0; \quad \forall j \in S, \ p_j \geq 0.$$
$$\forall j \notin S, \ \sum_i A_{ij}\beta_i \leq 1; \quad \forall j \in S, \ \sum_i A_{ij}\beta_i = 1 \tag{11}$$
$$\forall i, \ \beta_i = \frac{\sum_j W_{ij}p_j}{\sum_j A_{ij}p_j}$$

LEMMA 5.2. *If $\exists \boldsymbol{\beta}^*$, $\boldsymbol{p}^*$ satisfying (11) then they constitute an equilibrium.*

PROOF. Let $x_{ij}^* = \beta_i^* A_{ij}$, then for every agent i we get

$$\sum_j x_{ij}^* p_j^* = \beta_i^* \sum_j A_{ij}p_j^* = \sum_j W_{ij}p_j^* ,$$

using the third condition of (11). This together with the fact that $x_{ij}^* = \beta_i^* A_{ij}$ it follows that $\boldsymbol{x}_i^*$ is an optimal bundle of agent i at prices $\boldsymbol{p}^*$. Market clearing for goods follows from the first two conditions of (11). □

Note that system (11) remains unchanged if we remove price variables that are set to zero. Then it will have $2d$ variables, The first two conditions are linear in these variables, while the third condition is of degree two. Since d is a constant, checking non-emptiness of (11) can be done in polynomial time [2, 3, 15].

If (11) turns out to be non-empty then by Lemma 5.2 we get an equilibrium. Lemma 5.1 implies that we need to check feasibility of this system for every subset of goods of size d. There are at most $\binom{n}{d} \leq n^d$ such systems need to be checked, which is a polynomial in number because d is a constant. Therefore, overall we can find an equilibrium in polynomial time, and the next theorem follows:

THEOREM 5.3. *Consider an Arrow-Debreu exchange market under Leontief utility functions in which the number of agents is a constant. Then, in polynomial time we can determine if an equilibrium exists, and if so, we can find one.*

6 DISCUSSION

Is computing an equilibrium for a Fisher market under PLC utilities FIXP-hard? Clearly the problem is in FIXP since Fisher markets are a subcase of Arrow-Debreu markets. We believe that existing techniques, for example of [39] establishing hardness for Fisher markets under SPLC utilities via reduction from Arrow-Debreu markets, will not work and new ideas are needed. As stated in Section 1.1, finding an approximate equilibrium under CES utilities was also shown to be PPAD-complete [9]. Is computing an exact equilibrium FIXP-complete?

In economics, uniqueness of equilibria plays an important role. In this vein, we ask what is the complexity of deciding if a PLC or Leontief market has more than one equilibria. We note that the reduction given in this paper blows up the number of equilibria and hence it will not answer this question in a straightforward manner.

REFERENCES

[1] K. Arrow and G. Debreu. 1954. Existence of an equilibrium for a competitive economy. *Econometrica* 22 (1954), 265–290.
[2] S. Basu, R. Pollack, and M.-F. Roy. 1996. On the combinatorial and algebraic complexity of quantifier elimination. *JACM* 43(6) (1996), 1002–1045.
[3] S. Basu, R. Pollack, and M.-F. Roy. 2004. *Quantifier Elimination and Cylindrical Algebraic Decomposition.* Springer.
[4] M. S. Bazarra, H. D. Sherali, and C. M. Shetty. 2006. *Nonlinear Programming: Theory and Algorithms.* John Wiley & Sons.
[5] Lenore Blum, F. Cucker, Mike Shub, and Steve Smale. 1998. *Complexity and Real Computation.* Springer-Verlag.
[6] S. Boyd and L. Vandenberghe. 2009. *Convex Optimization.* Cambridge University Press.
[7] X. Chen, D. Dai, Y. Du, and S.-H. Teng. 2009. Settling the complexity of Arrow-Debreu equilibria in markets with additively separable utilities. In *IEEE Annual Symposium on Foundations of Computer Science.*
[8] X. Chen, X. Deng, and S.-H. Teng. 2009. Settling the complexity of computing two-player Nash equilibria. *Journal of the ACM* 56(3) (2009).
[9] Xi Chen, Dimitris Paparas, and Mihalis Yannakakis. 2013. The complexity of non-monotone markets. In *ACM Symposium on the Theory of Computing.* 181–190.
[10] Bruno Codenotti, Sriram Pemmaraju, and Kasturi Varadarajan. 2005. On the polynomial time computation of equilibria for certain exchange economies. In *ACM-SIAM Annual Symposium on Discrete Algorithms.* 72–81.
[11] B. Codenotti, A. Saberi, K. Varadarajan, and Y. Ye. 2006. Leontief economies encode two-player zero-sum games. In *SODA.*
[12] C. Daskalakis, P. W. Goldberg, and C. H. Papadimitriou. 2009. The Complexity of Computing a Nash Equilibrium. *SIAM J. Comput.* 39(1) (2009), 195–259.
[13] X. Deng and Y. Du. 2008. The computation of approximate competitive equilibrium is PPAD-hard. *Inform. Process Letters* 108 (2008), 369–373.
[14] Xiaotie Deng, Christos Papadimitriou, and Shmuel Safra. 2002. On the complexity of equilibria. In *Proceedings of ACM Symposium on Theory of Computing.*
[15] N. Devanur and R. Kannan. 2008. Market Equilibria in Polynomial Time for Fixed Number of Goods or Agents. In *FOCS.* 45–53.
[16] N. Devanur, C. H. Papadimitriou, A. Saberi, and V. V. Vazirani. 2008. Market Equilibrium via a Primal-Dual Algorithm for a Convex Program. *Journal of the ACM* 55, 5 (2008).
[17] N. Devanur and V. Vazirani. 2003. An improved approximation scheme for computing Arrow-Debreu prices for the linear case. In *Proceedings of 23rd FSTTCS.*
[18] Ran Duan and Kurt Mehlhorn. 2013. A combinatorial polynomial algorithm for the linear Arrow-Debreu market. In *ICALP.*
[19] B. C. Eaves. 1976. A Finite Algorithm for the Linear Exchange Model. *Journal of Mathematical Economics* 3 (1976), 197–203.

[20] E. Eisenberg. 1961. Aggregation of utility functions. *Management Sciences* 7 (1961), 337–350.

[21] K. Etessami and M. Yannakakis. 2010. On the complexity of Nash equilibria and other fixed points. *SIAM J. Comput.* 39(6) (2010), 2531–2597.

[22] John Fearnley and Rahul Savani. 2015. The complexity of the simplex method. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC.* 201–208.

[23] Jugal Garg, Ruta Mehta, and Vijay V. Vazirani. 2014. Dichotomies in Equilibrium Computation, and Complementary Pivot Algorithms for a New Class of Non-Separable Utility Functions. In *ACM Symposium on the Theory of Computing.*

[24] Jugal Garg and Vijay V. Vazirani. 2014. On Equilibrium Computation in Markets with Production. In *SODA.*

[25] R. Garg and S. Kapoor. 2004. Auction algorithms for market equilibrium. In *ACM Symposium on the Theory of Computing.*

[26] R. Garg, S. Kapoor, and V. V. Vazirani. 2004. An Auction-Based Market Equilbrium Algorithm for the Separable Gross Substitutibility Case. In *Proceedings, APPROX.*

[27] Paul W. Goldberg, Christos H. Papadimitriou, and Rahul Savani. 2011. The complexity of the homotopy method, equilibrium selection, and Lemke-Howson solutions. In *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS.* 67–76.

[28] Li-Sha Huang and Shang-Hua Teng. 2007. On the approximation and smoothed complexity of Leontief market equilibria. In *Frontiers in Algorithmics, First Annual International Workshop, FAW.* 96–107.

[29] Kamal Jain. 2007. A polynomial time algorithm for computing the Arrow-Debreu market equilibrium for linear utilities. *SIAM J. Comput.* 37, 1 (2007), 306–318.

[30] K. Jain, M. Mahdian, and A. Saberi. 2003. Approximating market equilibrium. In *Proceedings of 6th APPROX.*

[31] K. Jain and K. Varadarajan. 2006. Equilibria for economies with production: Constant-returns technologies and production planning constraints. In *ACM-SIAM Annual Symposium on Discrete Algorithms.* 688–697.

[32] V. Klee and G. J. Minty. 1972. How good is the simplex algorithm? In *(O. Shisha, ed.) Inequalities III* (1972), 159–175.

[33] A. Mas-Colell, M. D. Whinston, and J. R. Green. 1995. *Microeconomic Theory.* Oxford University Press.

[34] John Nash. 1951. Non-cooperative Games. *Annals of Mathematics* 54, 2 (September 1951), 289–295.

[35] J. B. Orlin. 2010. Improved algorithms for computing Fisher's market clearing prices. In *ACM Symposium on the Theory of Computing.* 291–300.

[36] C. H. Papadimitriou. 1994. On the complexity of the parity argument and other inefficient proofs of existence. *JCSS* 48(3) (1994), 498–532.

[37] R. Savani and B. von Stengel. 2004. Exponentially Many Steps for Finding a Nash Equilibrium in a Bimatrix Game. In *FOCS.* 258–267.

[38] Marcus Schaefer and Daniel Štefankovič. 2015. Fixed points, Nash equilibria, and the existential theory of the reals. (2015). Theory of Computing Systems.

[39] Vijay V. Vazirani and Mihalis Yannakakis. 2011. Market equilibrium under separable, piecewise-linear, concave utilities. *Journal of ACM* 58, 3 (2011), 10:1–10:25.

[40] Laszlo A. Vegh. 2012. Concave Generalized Flows with Applications to Market Equilibria. In *IEEE Annual Symposium on Foundations of Computer Science.*

[41] Laszlo A. Vegh. 2012. Strongly polynomial algorithm for a class of minimum-cost flow problems with separable convex objectives. In *ACM Symposium on the Theory of Computing.*

[42] M. Yannakakis. 2013. (2013). Personal communication.

[43] Yinyu Ye. 2008. A path to the Arrow-Debreu competitive market equilibrium. *Math. Program.* 111(1-2) (2008), 315–348.