

Leontief Exchange Markets Can Solve Multivariate Polynomial Equations, Yielding FIXP and ETR Hardness*

Jugal Garg[†] Ruta Mehta[‡] Vijay V. Vazirani[§] Sadra Yazdanbod[¶]

Abstract

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 [33]. As corollaries, we obtain FIXP-hardness for piecewise-linear concave (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 [17], 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.

The main technical part of our result is the following reduction: 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 a Leontief exchange market \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 is related to a classic result of Sonnenschein [32, 31].

*Supported by NSF Grants CCF-0914732 and CCF-1216019.

[†]Max-Planck-Institut für Informatik, Saarbrücken, Germany. jgarg@mpi-inf.mpg.de

[‡]College of Computing, Georgia Institute of Technology, Atlanta. rmehta@cc.gatech.edu

[§]College of Computing, Georgia Institute of Technology, Atlanta. vazirani@cc.gatech.edu

[¶]College of Computing, Georgia Institute of Technology, Atlanta. syazdanb@cc.gatech.edu

1 Introduction

In economics, it is customary to assume that utility functions are non-separable concave, not only because of their generality but also their nice properties, e.g., decreasing marginal utilities and convexity (of optimal bundles¹). Since computer science assumes a finite precision model of computation, we restrict attention to piecewise-linear concave (PLC) utility functions². Extensive study of special cases of PLC utility functions has led to a deep understanding of computability of market equilibria, ever since the commencement of this study twelve years ago; see details in Section 1.1. However, determining the exact complexity of computing equilibria for Arrow-Debreu markets under PLC utility functions has remained open. A subcase of the latter, which has been widely used in economic modeling [25], is Leontief utility functions, and its exact complexity has also remained open, e.g., see [33].

In this paper, we resolve both these problems, by showing them FIXP-hard. Very recently, Yannakakis [36] and Garg et. al. [17] gave proofs of membership in FIXP for Leontief and PLC utility functions, respectively. However, following the work of Etessami and Yannakakis [16], defining FIXP and proving FIXP-completeness of Arrow-Debreu markets whose excess demand functions are algebraic, there has been no progress on giving proofs of FIXP-hardness for other market equilibrium problems. Note that the latter result does not establish FIXP-completeness of Arrow-Debreu markets under any specific class of utility functions³

In this paper, we prove FIXP-hardness for Arrow-Debreu markets under Leontief utility functions. We also show FIXP-hardness for 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 [17]). 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.

Perhaps the most elementary way of stating the main technical part of our 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 [16] and that 3-Nash can be reduced to such a system \mathcal{S} , yield FIXP-hardness for the Leontief case.

Reduction \mathcal{R} is related to a classic result of Sonnenschein [32, 31] which states that a set of arbitrary multivariate polynomials can be generated as excess demand functions of an Arrow-

¹which is used crucially in fixed point theorems for proving existence of equilibria.

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

³In 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.

Debreu market with concave utilities (in this case, as in ours, simultaneous zeros of the polynomials correspond to equilibrium prices). This result led to the famous “Anything Goes Theorem” of Sonnenschein-Mantel-Debreu [24, 9] which states that very mild restrictions suffice to characterize excess demand functions of such markets. This theorem had wide ranging consequences to general equilibrium theory [29].

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 [11, 12, 22, 19, 21, 37, 27, 35, 13] and certain other cases [7, 23, 10, 34, 20], 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 [5, 33] for Papadimitriou’s class PPAD [28]. 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 [33]. The notion of SPLC production sets was defined in [18] 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 [15] and hence their equilibria can be approximated to any required degree in polynomial time [4, 2]. Arrow-Debreu markets under Leontief utilities were shown to be PPAD-hard [8]; however, since in this case equilibria are not rational numbers [14], its complexity is not characterize by PPAD (problems in PPAD have rational solutions). Leontief utilities are a limiting case of constant elasticity of substitution (CES) utilities [25]. Finding an approximate equilibrium under the latter was also shown to be PPAD-complete [6].

1.2 Technical contributions

We now describe the difficulties encountered in obtaining reduction \mathcal{R} and the ideas needed to overcome them; this should also help explain why FIXP-hardness of Leontief (and PLC) markets was a long-standing open problem. 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 [5]. At the outset, observe the latter is only dealing with linear functions of variables⁴ 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 need 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. Interestingly enough, in retrospect, we do not see how to create these units using only SPLC utility functions.

⁴Since 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.

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.

These closed submarkets enable us to ensure that variables denoting prices of goods satisfy specified arithmetic relations. The latter are equality, linear function and product; we show that these three arithmetic relations suffice to encode any polynomial equation. Under equality, we want two prices p_a and p_b to be equal, and 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.

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 [30] 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.

Next we briefly describe the classical Arrow-Debreu market model, the problem of 3-Nash and its relation with the complexity classes FIXP and ETR. Following are a few notations that we will use in the rest of the paper.

Notations. We mostly follow: capital letters denote matrices of constants, like W ; bold lower case letters denote vector of variables, like \mathbf{x}, \mathbf{y} ; and calligraphic capital letters denote sets like \mathcal{A}, \mathcal{G} . We use $[n]$ to denote the set $\{1, \dots, n\}$. Given an n -dimensional vector \mathbf{x} and a number $r \in \mathbb{R}$, by $\mathbf{x} \leq r$, we mean $\forall i \in [n], x_i \leq r$.

2 The Arrow-Debreu Market Model

The Arrow-Debreu (AD) market [1] consists of a set \mathcal{G} of divisible goods, a set \mathcal{A} of agents and a set \mathcal{F} of firms. Let g denote the number of goods in the market.

The production capabilities of a firm is defined by a convex set of production schedules and each firm wants to use a (optimal) production schedule that maximizes its profit — money earned from the production minus the money spent on the raw materials. Firms are owned by agents: Θ_{if} is the profit share of agent i in firm f such that $\forall f \in \mathcal{F}, \sum_{i \in \mathcal{A}} \Theta_{if} = 1$. Each agent i has a utility

function $U_i : \mathbb{R}_+^g \rightarrow \mathbb{R}_+$ over bundles, and comes with an initial endowment of goods; W_{ij} is amount of good j with agent i . Each agent wants to buy a (optimal) bundle of goods that maximizes her utility to the extent allowed by her earned money – from initial endowment and profit shares in the firms.

Given prices of goods, if there is an assignment of optimal production schedule to each firm and optimal affordable bundle to each agent so that there is neither deficiency nor surplus of any good, then such prices are called *market clearing* or *market equilibrium* prices; we note that a zero-priced good is allowed to be in surplus. 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 heavy machinery of Kakutani fixed point theorem. We note that an arbitrary market may not admit an equilibrium.

A well studied restriction of Arrow-Debreu model is **exchange economy**, i.e., markets without production firms. To work under finite precision it is customary to assume that utility functions are piecewise-linear concave (PLC) and production sets are polyhedral.

2.1 Piecewise-linear concave (PLC) utility function

As stated earlier, agent i 's utility function is $U_i : \mathbb{R}_+^g \rightarrow \mathbb{R}_+$ over bundle of goods. These functions are said to be piecewise-linear concave (PLC) if at bundle $\mathbf{x}_i = (x_{i1}, \dots, x_{ig})$ it is given by:

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

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 $U_i(\mathbf{0}) = 0$, and therefore at least one T_i^k is zero.

2.1.1 Leontief utility function

The Leontief utility function is a special subclass of PLC, where each good is required in a fixed proportion. Formally, it is given by:

$$U_i(\mathbf{x}_i) = \min_{j \in \mathcal{G}} \left\{ \frac{x_{ij}}{A_{ij}} \right\},$$

where A_{ij} 's are non-negative numbers. 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.

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

2.2 PLC production sets

A firm can produce a set of goods using another set of goods as raw materials, and these two sets are assumed to be disjoint. Let $\mathcal{P}_f \in \mathbb{R}^g$ be the set of production schedules for firm f , then if it can produce a bundle \mathbf{x}^s using a bundle \mathbf{x}^r then $\mathbf{x}^s - \mathbf{x}^r \in \mathcal{P}_f$. The set is assumed to be downward closed, contains the origin $\mathbf{0}$, no vector is strictly positive (*no production out of nothing*) and is polyhedral. We call these *PLC production sets*.

Let \mathcal{S}_f denote the set of goods that can be produced by firm f and \mathcal{R}_f be the set of goods it can use as raw material such that $\mathcal{S}_f \cap \mathcal{R}_f = \emptyset$. A PLC production set of firm f can be described as follows, where x_{fj}^s and x_{fj}^r denote the amount of good j produced and used respectively.

$$\mathcal{P}_f = \left\{ (\mathbf{x}^s - \mathbf{x}^r) \in \mathbb{R}^g \mid \sum_{j \in \mathcal{S}_f} D_{fj}^k x_{fj}^s \leq \sum_{j \in \mathcal{R}_f} C_{fj}^k x_{fj}^r + T_f^k, \forall k; \quad \begin{array}{l} \mathbf{x}^s \geq 0; \quad x_{fj}^s = 0, \forall j \notin \mathcal{S}_f; \\ \mathbf{x}^r \geq 0; \quad x_{fj}^r = 0, \forall j \notin \mathcal{R}_f \end{array} \right\}$$

where D_{fj}^k 's, C_{fj}^k 's and T_f^k 's are given non-negative rational numbers. Since there is no production if no raw material is consumed, it should be the case that for some k , $T_f^k = 0$.

2.2.1 Leontief production

The Leontief production is a special subclass of PLC production sets, where a firm f produces a single good a using a subset of the rest of the goods as raw materials. To produce one unit of a , it requires D_{fj} units of good j , i.e.,

$$x_{fa}^s = \min_{j \neq a} \left\{ \frac{x_{fj}^r}{D_{fj}} \right\}.$$

3 3-player Nash Equilibrium (3-Nash)

In this section we describe 3-player finite games and characterize Nash equilibrium. 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 $\mathcal{S} = \mathcal{S}_1 \times \mathcal{S}_2 \times \mathcal{S}_3$. W.l.o.g. we assume that $|\mathcal{S}_1| = |\mathcal{S}_2| = |\mathcal{S}_3| = n_s$. Such a game can be represented by $n_s \times n_s \times n_s$ -dimensional tensors A_1, A_2 and A_3 representing payoffs of first, second and third player respectively. If players play $\mathbf{s} = (s_1, s_2, s_3) \in \mathcal{S}$, then the payoffs are $A_1(\mathbf{s})$, $A_2(\mathbf{s})$ and $A_3(\mathbf{s})$ respectively.

Since players may randomize among their strategies, 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 $\mathbf{z} = (z_1, z_2, z_3) \in \Delta$, let z_{ps} denote the probability with which player p plays strategy $s \in \mathcal{S}_p$, and let \mathbf{z}_{-p} be the strategy profile of all the players at \mathbf{z} except p . For player $p \in \{1, 2, 3\}$ the total payoff and payoff from strategy $s \in \mathcal{S}_p$ at \mathbf{z} are respectively,

$$\pi_p(\mathbf{z}) = \sum_{\mathbf{s} \in \mathcal{S}} A_p(\mathbf{s}) z_{1s_1} z_{2s_2} z_{3s_3} \quad \text{and} \quad \pi_p(s, \mathbf{z}_{-p}) = \sum_{\mathbf{t} \in \mathcal{S}_{-p}} A_p(s, \mathbf{t}) \prod_{q \neq p} z_{qt_q}$$

Definition 3.1 (Nash (1951) [26]) *A mixed-strategy profile $\mathbf{z} \in \Delta$ is a Nash equilibrium (NE) if no player gains by deviating unilaterally. Formally, $\forall p = 1, 2, 3 \quad \pi_p(\mathbf{z}) \geq \pi_p(\mathbf{z}', \mathbf{z}_{-p}), \forall \mathbf{z}' \in \Delta_p$.*

In 1951, John Nash [26] proved existence of an equilibrium in a finite game using Brouwer fixed-point theorem, which is highly non-constructive. Despite many efforts over the years, no efficient methods are obtained to compute a NE of finite games. Next we give a characterization of NE through multivariate polynomials and discuss its complexity.

NE Characterization. It is easy to see that in order to maximize the expected payoff, only *best moves* should be played with a non-zero probability; by *best moves* we mean the moves fetching maximum payoff. Formally,

$$\forall p \in \{1, 2, 3\}, \forall s \in \mathcal{S}_p, z_{ps} > 0 \Rightarrow \pi_p(s, \mathbf{z}_{-p}) = \delta_p, \quad \text{where} \quad \delta_p = \max_{s' \in \mathcal{S}_p} \pi_p(s', \mathbf{z}_{-p}) \quad (2)$$

Assumption. Since scaling all the co-ordinates of A_p 's with a positive number or adding a constant to them does not change the set of Nash equilibria of the game $\mathbf{A} = (A_1, A_2, A_3)$, w.l.o.g. we

assume that all the co-ordinates of each of A_p are in the interval $[0, 1]$. Using (2), we can define the following system of multivariate polynomials, where variables z_{ps} 's capture strategies, δ_p captures payoff of player p , and β_{ps} are slack variables:

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

Lemma 3.2 *Nash equilibria of \mathbf{A} are exactly the solutions of system $F_{NE}(\mathbf{A})$, projected onto \mathbf{z} .*

Proof : It is easy to check that NE of \mathbf{A} gives a solution of $F_{NE}(\mathbf{A})$ using (2); the upper bounds on variables of $F_{NE}(\mathbf{A})$ holds because all the entries in \mathbf{A} are in the interval $[0, 1]$. Similarly, given a solution $(\mathbf{z}, \boldsymbol{\beta}, \boldsymbol{\delta})$ of $F_{NE}(\mathbf{A})$, the first condition ensures that $\mathbf{z} \in \Delta$. The two parts of the second condition imply that \mathbf{z} satisfies (2) and therefore is a NE of game \mathbf{A} . \square

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

3.1 The class FIXP

The class FIXP to capture complexity of the exact fixed point problems with algebraic solutions [16]. An instance I of FIXP consists of an algebraic circuit C_I defining a function $F_I : [0, 1]^d \rightarrow [0, 1]^d$, 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.

The circuit C_I is a sequence of gates g_1, \dots, g_m , where for $i \in [d]$, $g_i := l_i$ is an input variable. For $d < i \leq d + r$, $g_i := c_i \in \mathbb{Q}$ is a rational constant, with numerator and denominator encoded in binary. For $i > d + r$ we have $g_i = g_j \circ g_k$, where $j, k < i$ and the binary operator $\circ \in \{\max, +, *\}$. The last d gates are the output gates. Note that the circuit forms a directed acyclic graph (DAG), when gates are considered as nodes, and there is an edge from g_j and g_k to g_i if $g_i = g_j \circ g_k$. Since circuit C_I represents function F_I it has to be the case that if we input $\boldsymbol{\lambda} \in [0, 1]^d$ to C_I then all the gates are well defined and the circuit outputs $C_I(\boldsymbol{\lambda}) = F_I(\boldsymbol{\lambda})$ in $[0, 1]^d$. We note that a circuit representing a problem in FIXP operates on real numbers.

Reduction requirements: A reduction from problem A to problem B consists of two polynomial-time computable functions: a function f that maps an instance I of A to an instance $f(I)$ of B , and another function g that maps a solution \mathbf{y} of $f(I)$ to a solution \mathbf{x} of I . If $x_i = g_i(\mathbf{y})$, then $g_i(\mathbf{y}) = a_i y_j + b_i$, for some j , where a_i and b_i are polynomial-size rational numbers; every coordinate of \mathbf{x} is a linear function of one coordinate of \mathbf{y} .

In order to remain faithful to Turing machine computation, Etessami and Yannakakis [16] defined following three discrete problems on FIXP.

Partial computation $FIXP_{pc}$: Given an instance I and a positive integer k in unary, compute the binary representation of some solution, up to the first k bits after the decimal point.

Decision $FIXP_d$: Given an instance I and a rational r return ‘Yes’ if $x_1 \geq r$ for all solutions, ‘No’ if $x_1 < r$ for all solutions, and otherwise either answer is fine.

(Strong) Approximation $FIXP_a$: Given an instance I and a rational $\epsilon > 0$ in binary, compute a vector \mathbf{x} that is within (additive) ϵ distance from some solution, i.e., $\exists \mathbf{x}^* \in Sol(I)$ such that $|\mathbf{x}^* - \mathbf{x}|_\infty \leq \epsilon$.

Whereas FIXP is a class of, in general, real-valued search problems, whose complexity can be studied in a real computation model, e.g., [3], 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. The following result was shown in [16].

Theorem 3.3 [16] *Given a 3-player game $\mathbf{A} = (A_1, A_2, A_3)$, 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 $FIXP_{pc}$.*

3.2 Existential Theory of Reals (ETR)

The class ETR was defined to capture the decision problems arising in *existential theory of reals* [30]. An instance I of class ETR consists of a sentence of the form,

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

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 whether the sentence is true. Following is an example of such an instance.

$$\exists(x_1, x_2), (x_1^4 + x_1^3x_2^2 - 3x_2^3 + 1 = 0 \wedge x_1x_2 \geq 3) \vee (x_2^4 - 3x_1 < 6)$$

[30] showed that disallowing the operation of $<$ does not change the class ETR. 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 [30] 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.

Definition 3.4 (Decision 3-Nash) *Decision 3-Nash is the problem of checking if a given 3-player game \mathbf{A} admits a Nash equilibrium \mathbf{z} such that $\mathbf{z} \leq 0.5$, where \mathbf{z} is the mixed-strategy profile played.*

Theorem 3.5 [30] *Decision 3-Nash is ETR-complete.*

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

4 Multivariate Polynomials to Leontief Exchange Market

Consider the following system of m multivariate polynomials on n variables $\mathbf{z} = (z_1, \dots, z_n)$,

$$F : \{f_i(\mathbf{z}) = 0, \forall i \in [m]; \quad L_j \leq z_j \leq U_j, \forall j \in [n]\}, \quad \text{where } L_j, U_j \geq 0 \quad (4)$$

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 an exchange market with Leontief utility functions (see Section 2.1.1 for definition). The problems of 3-Nash and Decision 3-Nash (see Definition 3.4) can be characterized by a set similar to (4) (see

(3) in Section 3), in turn we obtain FIXP and ETR hardness results for Leontief markets, and in turn PLC markets, from the corresponding hardness of 3-Nash [16, 30].

Polynomial f_i is represented as sum of monomials, and a monomial $\alpha z_1^{d_1} \dots z_n^{d_n}$ is represented by tuple $(\alpha, d_1, \dots, d_n)$; here coefficient α is a rational number. Let \mathcal{M}_{f_i} denote the set of monomials of f_i , and $size[f_i] = \sum_{(\alpha, \mathbf{d}) \in \mathcal{M}_{f_i}} size(\alpha, \mathbf{d})$, 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 $deg(f_i) = \max_{(\alpha, \mathbf{d}) \in \mathcal{M}_{f_i}} \sum_j d_j$. The 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 a system F , next we construct an exchange market in time polynomial in $size[F]$, whose equilibria correspond to solutions of F .

4.1 Preprocessing

First we transform system F into a polynomial sized equivalent system that uses only the following three basic operations on *non-negative* variables.

$$\begin{aligned} (EQ.) \quad & z_a = z_b \\ (LIN.) \quad & z_a = Bz_b + Cz_c + D, \quad \text{where } B, C, D \geq 0 \\ (QD.) \quad & z_a = z_b * z_c \end{aligned} \tag{5}$$

Remark 4.1 *We note that even though (EQ.) is a special case of (LIN.), we consider it separately in order to convey the main ideas.*

Next we illustrate how to capture f_i 's using these basic operations through an example. Consider a polynomial

$$4z_1^2 z_2 + 3z_1 z_2 - z_1 - 2 = 0.$$

First, move all monomials with negative coefficients to the right hand side of the equality, so that all the coefficients become positive.

$$4z_1^2 z_2 + 3z_1 z_2 = z_1 + 2$$

Second, capture every monomials, with degree more than one, using basic operations:

$$\begin{aligned} z_{a_1} = z_1^2 z_2 & \equiv z_{a_2} = z_1 * z_1, \quad z_{a_1} = z_{a_2} * z_2 \\ z_{b_1} = z_1 z_2 & \equiv z_{b_1} = z_1 * z_2 \end{aligned}$$

Third, capture the equality $4z_{a_1} + 3z_{b_1} = z_1 + 2$ using (LIN.) and (EQ.):

$$4z_{a_1} + 3z_{b_1} = z_1 + 2 \quad \equiv \quad z_{e_1} = 4z_{a_1} + 3z_{b_1}, \quad z_{f_1} = z_1 + 2, \quad z_{e_1} = z_{f_1}$$

Finally, combine all of the above to represent f_i as follows:

$$\begin{aligned} 4z_1^2 z_2 + 3z_1 z_2 - z_1 - 2 = 0 & \equiv \begin{aligned} & z_{a_2} = z_1 * z_1, \quad z_{a_1} = z_{a_2} * z_2 \\ & z_{b_1} = z_1 * z_2 \\ & z_{e_1} = 4z_{a_1} + 3z_{b_1}, \quad z_{f_1} = z_1 + 2, \quad z_{e_1} = z_{f_1} \end{aligned} \end{aligned} \tag{6}$$

Since inequalities have to be captured through equalities with non-negative variables, the inequalities of (4) have to be transformed as follows:

$$\forall j \in [n], \quad z_j = s_j^l + L_j, \quad z_j + s_j^u = U_j, \quad z_j, s_j^l, s_j^u \geq 0 \tag{7}$$

Let $R(F)$ be a reformulation of F , using transformation similar to (6) for each f_i , and that of (7) for each inequality. All the 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 (as was done in (6)). Let the number of variables in $R(F)$ be N , and out of these let z_1, \dots, z_n be the original set of variables of F (4). 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} .

4.2 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 (5) (*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.

$$\begin{aligned} (EQ.) \quad & p_a = p_b \\ (LIN.) \quad & p_a = Bp_b + Cp_c + Dp_s, \quad \text{where } B, C, D \geq 0 \\ (QD.) \quad & p_a = \frac{p_b * p_c}{p_s} \end{aligned} \tag{8}$$

Let $R'(F)$ be a system of equalities after applying the transformation of (8) 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). Recall that \mathcal{M}_{f_i} denotes the set of monomials in polynomial f_i . To bound the values at a solution of $R'(F)$, define

$$H = M_{max} U_{max}^d + 1, \quad \text{where} \quad \begin{aligned} d &= \max_{f_i} deg(f_i), \quad M_{max} = \max_i |\mathcal{M}_{f_i}|, \\ U_{max} &= \max\{\max_j U_j, \max_{f_i, (\alpha, \mathbf{d}) \in \mathcal{M}_{f_i}} |\alpha|\}. \end{aligned}$$

Lemma 4.2 $size[R'(F)] = poly(size[F])$. Vector \mathbf{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, $p_j \leq H$, $\forall j \in [N]$.

Proof : For the first part, it is enough to bound $size[R(F)]$. Note that the number of auxiliary variables added to $R(F)$ to construct a monomial of f_i is at most its degree. Further, to construct the expression of f_i from these, the extra variables needed is at most the number of monomials. Further, the coefficients of (*LIN.*) type relations are coefficients of the monomials of f_i 's. Thus, we get that $size[R'(F)] = O(\sum_{i \in [m]} deg(f_i)^2 size[f_i] + \sum_{j \in [n]} size(U_j) + size(L_j)) = poly(size[F])$.

The second part follows by construction. For the third part note that (p_1, \dots, p_N) is a solution of $R(F)$. Since variables of the original system F is upper bounded by U_j s, it is easy to see that the maximum value of any variable in a non-negative solution of $R(F)$ is at most H . \square

Next we construct a market whose equilibria satisfy all the relations of $R'(F)$, and has $p_s > 0$.

4.3 Market construction

In this section we construct market \mathcal{M} consisting of goods G_1, \dots, G_N and G_s , such that the prices p_1, \dots, p_N and p_s , satisfy all the relations of $R'(F)$ at equilibrium.

First we want price of G_s to be always non-zero at equilibrium. To ensure this we add the following agent to market \mathcal{M} . Recall that W_{ij} is the amount of good G_j agent A_i brings to the

market, $U_i : \mathbb{R}_+^g \rightarrow \mathbb{R}_+$ is the utility function of A_i , and \mathbf{x}_i denotes the bundle of goods consumed by her.

$$A_s : W_{ss} = 1, W_{sj} = 0, \forall j \in [N]; \quad U_s(\mathbf{x}_s) = x_{ss} \quad (9)$$

Lemma 4.3 *At every equilibrium of market \mathcal{M} , we have $p_s > 0$, and $x_{ss} = W_{ss}$.*

Proof : At an equilibrium if $p_s = 0$, then agent A_s will demand infinite amount of good s , a contradiction. The second part follows using the fact that at any given prices, A_s wants to buy only good s , and has exactly $W_{ss}p_s$ amount of money to spend. \square

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 4.4 (submarket) *A submarket $\widetilde{\mathcal{M}}$ of a market \mathcal{M} consists of a subset of agents and goods such that the endowment and utility functions of agents in $\widetilde{\mathcal{M}}$ and the production functions of firms in $\widetilde{\mathcal{M}}$ are defined over goods only in $\widetilde{\mathcal{M}}$.*

Definition 4.5 (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 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 (9) is a closed submarket (due to Lemma 4.3) with only one agent and one good, namely A_s and G_s respectively. We will see that the submarket $\widetilde{\mathcal{M}}$ establishing a relation of type (*EQ.*), (*LIN.*) and (*QD.*) has a set of *exclusive* goods used only in $\widetilde{\mathcal{M}}$, in order to achieve the closed property. Before describing construction of closed submarkets for more involved relations, we first describe it for a simple and important *equality* relation. Furthermore, we will use equality to construct closed markets for (*QD.*).

Let there be K relations in $R'(F)$, numbered from 1 to K , and let \mathcal{M}_r denote the closed submarket establishing relation $r \in [K]$.

4.3.1 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.

\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 $U_1(\mathbf{x}) = \min\{x_a, x_r\}$ A_2 : $W_2 = (1, 0, 1)$ and $U_2(\mathbf{x}) = \min\{x_b, x_r\}$

Table 1: Closed submarket \mathcal{M}_r for r^{th} relation $p_a = p_b$

In \mathcal{M}_r , the endowment vector W_i 's 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; we use similar representation in the subsequent constructions.

Lemma 4.6 Consider the market \mathcal{M}_r of Table 1.

- \mathcal{M}_r is a closed submarket.
- At equilibrium, \mathcal{M}_r enforces $p_a = p_b$.
- Every non-negative solution of $p_a = p_b$ gives an equilibrium of \mathcal{M}_r .

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:

$$p_b + p_r = \alpha(p_a + p_r)$$

$$p_a + p_r = \beta(p_b + p_r)$$

Clearly the above 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,

$$\alpha + \beta \leq 2$$

Replacing $\beta = \frac{1}{\alpha}$ in the above condition gives $(\alpha - 1)^2 \leq 0 \Rightarrow \alpha = \beta = 1$. 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 $p_a = p_b \geq 0$, then choosing $p_r = 1$, and $x_{1a} = x_{1r} = x_{2b} = x_{2r} = 1$ gives a market equilibrium of \mathcal{M}_r . \square

4.3.2 Submarket for relation (LIN.) $p_a = Bp_b + Cp_c + Dp_s$

The gadget for (LIN.) is an extension of (EQ.) having two agents with Leontief utility functions, as given in Table 2, where $B, C, D \geq 0$.

Remark 4.7 For simplicity, we denote agents of each submarket by A_1, A_2, \dots , and sometimes exclusive goods by G_1, G_2, \dots , however they are different across submarkets.

\mathcal{M}_r : 2 Agents (A_1, A_2) and 5 Goods (G_a, G_b, G_c, G_s, G_r) // G_r : an exclusive good A_1 : $W_1 = (1, 0, 0, 0, 1)$ and $U_1(\mathbf{x}) = \min\{\frac{x_b}{B}, \frac{x_c}{C}, \frac{x_s}{D}, x_r\}$ A_2 : $W_2 = (0, B, C, D, 1)$ and $U_2(\mathbf{x}) = \min\{x_a, x_r\}$
--

Table 2: \mathcal{M}_r : Closed market for r^{th} relation $p_a = Bp_b + Cp_c + Dp_s$, $B, C, D \geq 0$

Lemma 4.8 Consider the market \mathcal{M}_r of Table 2 with $B, C, D \geq 0$.

- \mathcal{M}_r is a closed submarket.
- At equilibrium, \mathcal{M}_r enforces $p_a = Bp_b + Cp_c + Dp_s$.
- Every non-negative solution of $p_a = Bp_b + Cp_c + Dp_s$ gives an equilibrium of \mathcal{M}_r .

Proof : The proof is similar to that of Lemma 4.6. Let α and β denote the utility obtained by A_1 and A_2 at equilibrium respectively. Then using (1) together with market clearing conditions of the two agents, we get:

$$\begin{aligned} p_a + p_r &= \alpha(Bp_b + Cp_c + Dp_s + p_r) \\ Bp_b + Cp_c + Dp_s + p_r &= \beta(p_a + p_r) \end{aligned}$$

Clearly the above conditions imply $\alpha\beta = 1 \Rightarrow \beta = 1/\alpha$. Since G_r is exclusive to this market, using similar argument as the proof of Lemma 4.6 we get that $\alpha + \beta \leq 2$. This together with $\beta = 1/\alpha$ gives $\alpha = \beta = 1$. Thus every equilibrium of \mathcal{M}_r enforces $p_a + p_r = Bp_b + Cp_c + Dp_s + p_r \Rightarrow p_a = Bp_b + Cp_c + Dp_s$. Hence, \mathcal{M}_r is a closed submarket.

For the last part, if $p_a = Bp_b + Cp_c + Dp_s \geq 0$, then setting $p_r = 1$, $x_{1r} = x_{2r} = x_{2a} = 1$, and $x_{1b} = B, x_{1c} = C, x_{1s} = D$ gives a market equilibrium of \mathcal{M}_r . \square

Using Lemma 4.8, we easily get the following.

Corollary 4.9 *There is a simple closed submarket to establish any linear relation of form $p_a = E_1p_{b_1} + \dots + E_np_{b_n} + E_0p_s$ for any $n \geq 1$, where E_0, E_1, \dots, E_n are non-negative rational constants.*

4.3.3 Submarket for relation (QD.) $p_a = \frac{p_b p_c}{p_s}$

In this section, we derive a closed submarket for establishing the (QD.) relation. In order to simplify the market construction, which is quite involved, we first make the following two assumptions, which are removed later. First, that $p_s = 1$ and second, that $p_b \neq 0$. The first assumption violates the scale invariance of prices, see Section 4.2, but simplifies the relation needed to $p_a = p_b p_c$. The second assumption ensures that no agent can demand an infinite amount of a good of price p_b (Note that in the reduction, since the price of a good corresponds to the probability of playing a certain strategy, eventually we do need to allow for $p_b = 0$).

The main idea for enforcing the simpler relation, $p_a = p_b p_c$, is to ensure that there is an agent A whose initial endowment is one unit of Good 1 priced at p_a , and she desires to consume only Good 2 priced at p_b . The left over amount of Good 2 after everyone, except agent A , consume is exactly p_c . Since $p_b > 0$, agent A has to buy all of this left over amount which requires her to spend $p_b p_c$. On the other hand her earning from the sell of Good 1 is p_a , implying $p_a = p_b p_c$. Figure 1 illustrates the idea. The difficulty in implementing this idea lies in the fact that p_b and p_c are variables; if they were both constants, the construction of the submarket would have been easy.

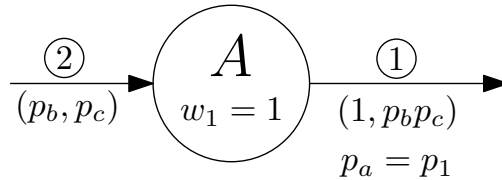


Figure 1: The main idea for enforcing the relation $p_a = p_b p_c$. Wires are numbered in circle, and wire i carries good G_i . The tuple on each wire represents (amount, price).

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.

Submarkets for the devices

In this section, we show implementation of three devices to be used in the submarket for **(QD.)** relation.

Converter (Conv(q)): The net consumption of this device is one 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 3 and Figure 2 illustrate the implementation. In the figure tuple on edges represent *(amount, price)* of the corresponding good shown in circle. Table 3 has two parts: Part 1 describes the market and Part 2 enforces linear relations among prices using the submarkets described in Sections 4.3.1 and 4.3.2.

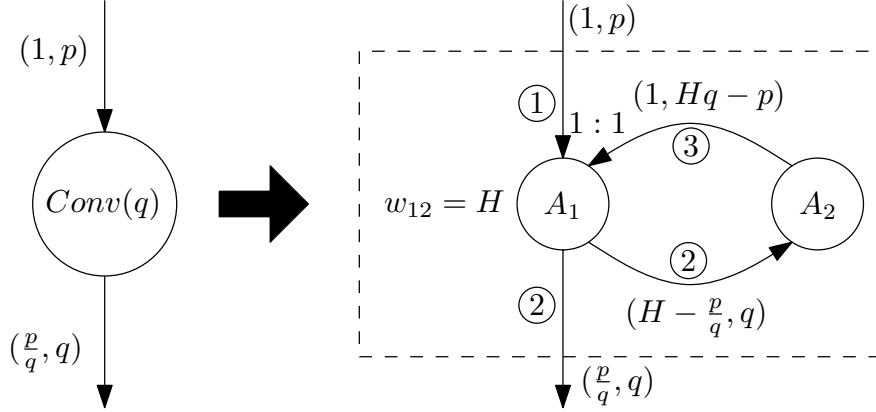


Figure 2: Flow of goods in Part 1 of Table 3 for $\text{Conv}(q)$. Wires are numbered in circle, and wire i carries good G_i . The tuple on each wire represents *(amount, price)*.

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

Table 3: A closed submarket for $\text{Conv}(q)$

There are two agents A_1 and A_2 , and three goods G_1, G_2 and G_3 . The endowment of A_1 is H units of G_2 , whose price is set to q . Recall that H is a constant defined in Section 4.2. A_1 likes to consume G_1 and G_3 in the ratio of 1:1. The net consumption of this device, i.e., one unit of G_1 at price p , is consumed by A_1 . Agent A_2 's endowment is one unit of G_3 , whose price is set to $Hq - p$. A_2 wants to consume G_2 , whose price is q . 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.

Combiner ($\text{Comb}(l, p_a, p_b)$): The net consumption of this device 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 4 and Figure 3 illustrate the implementation.

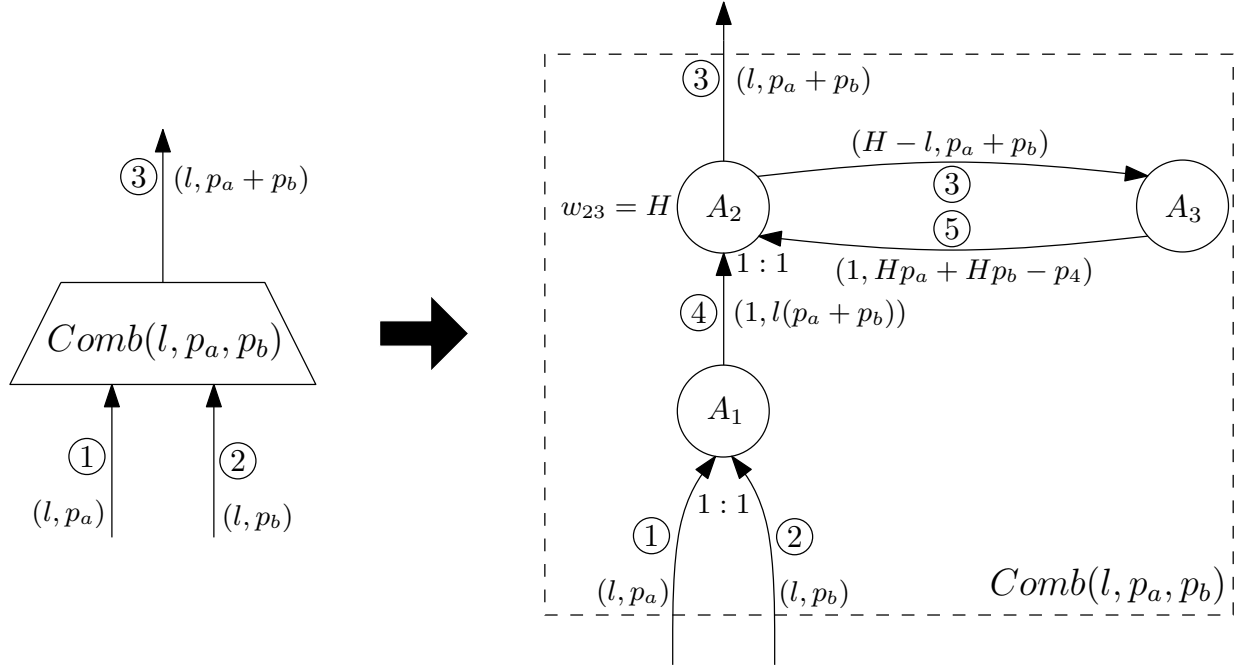


Figure 3: Flow of goods in Part 1 of Table 4 for $\text{Comb}(l, p_a, p_b)$. Wires are numbered, and wire i carries good G_i . The tuple on each wire represents (amount, price).

Part 1:	<p>Input: l units of G_1 and G_2 at price p_a and p_b respectively</p> <p>Output: l units of G_3 at price $p_a + p_b$</p> <p>3 Agents (A_1, A_2, A_3), 5 goods (G_1, G_2, G_3, G_4, G_5)</p> <p>A_1: $W_{14} = 1$ and $U_1(\mathbf{x}) = \min\{x_1, x_2\}$</p> <p>$A_2$: $W_{23} = H$ and $U_2(\mathbf{x}) = \min\{x_4, x_5\}$</p> <p>$A_3$: $W_{35} = 1$ and $U_3(\mathbf{x}) = x_3$</p>
Part 2:	<p>Closed submarkets for the following linear relations</p> <p>$p_3 = p_a + p_b$</p> <p>$p_5 = Hp_a + Hp_b - p_4$</p>

Table 4: A closed submarket for $\text{Comb}(l, p_a, p_b)$

Agent A_1 wants G_1 and G_2 in the ratio 1:1, and no other agent wants these goods. Therefore, A_1 will consume all of the available G_1 and G_2 and hence the price of her endowment, i.e., one unit of G_4 , will be $l(p_a + p_b)$ (observe that there is a multiplication involved in this price; however, it is not performed explicitly).

Agent A_2 wants G_4 and G_5 in the ratio 1:1. The price of A_3 's endowment, i.e., one unit of G_5 is set to $H(p_a + p_b) - p_4$. Hence the endowment of A_2 , i.e., H units of G_3 , has a price of $(p_a + p_b)$. Of this, A_3 must consume $(H - l)$, leaving l amount of G_3 as the net endowment of this device.

Splitter ($\text{Spl}(l, p_a, p_b)$): The net endowment of this device 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 5 and Figure 5 illustrate the implementation.

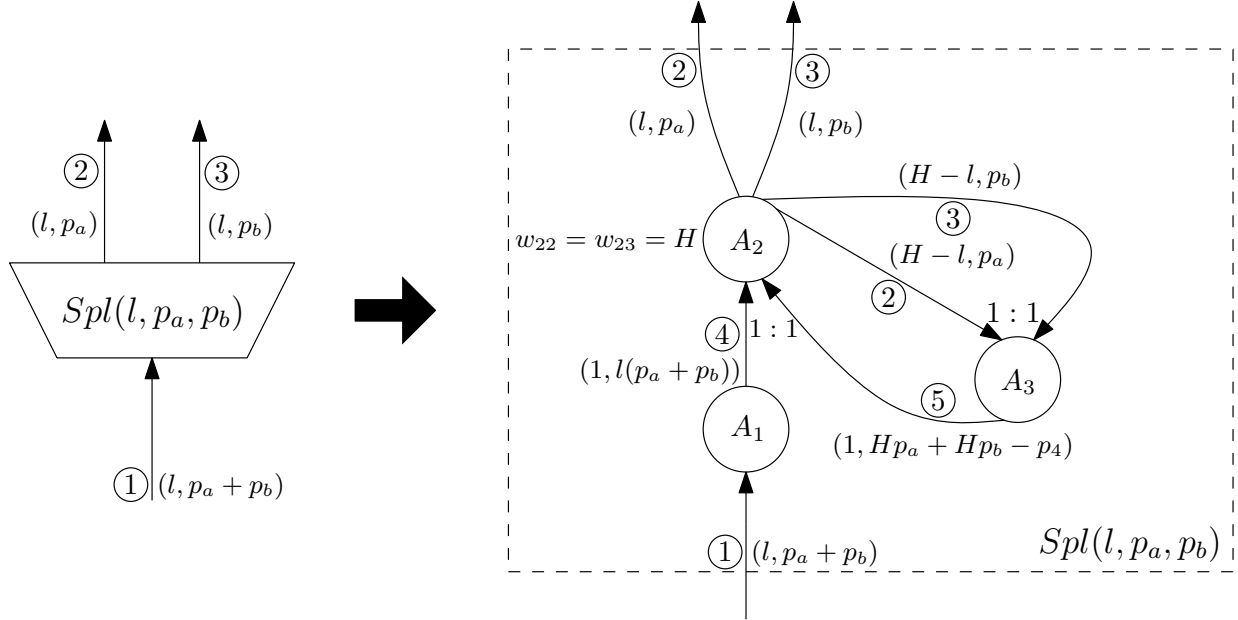


Figure 4: Flow of goods in Part 1 of Table 5 for $\text{Spl}(l, p_a, p_b)$. Wires are numbered, and wire i carries good G_i . The tuple on each wire represents (amount, price).

Good G_1 is desired only by Agent A_1 . Hence, the price of her initial endowment, i.e., one unit of G_4 , is forced to be $l(p_a + p_b)$ (observe that the multiplication involved is not done explicitly). Agent A_2 wants goods G_4 and G_5 in the ratio 1:1. The price of G_5 is set explicitly to $H(p_a + p_b) - p_4$. The endowment of A_2 is H units each of G_2 and G_3 , whose prices have been set to p_a and p_b , respectively. Agent A_3 wants these two goods in the ratio 1:1, and because of the setting of the price of her initial endowment, she must consume $(H - l)$ units of each of these two goods. The remaining amounts, i.e., l each, form the net endowment of the device, as required.

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

Table 5: A closed submarket for $\text{Spl}(l, p_a, p_b)$

Submarket construction for $p_a = p_b p_c$

Now we are ready to describe a closed submarket that enforces $p_a = p_b p_c$. Consider the submarket given in Table 6. In this market, the 7 goods, G_1, \dots, G_7 are exclusive to the submarket; the price of good G_j is p_j . The prices of $G_1, G_2, G_4, G_5, G_6, G_7$ are set to appropriate using (*EQ.*) and (*LIN.*) relations and prices p_a, p_b and p_c , as specified in the second part of the table. The submarket uses two Converters, one Combiner and one Splitter. Each of these devices is specified by giving its (net endowment, net consumption). Besides the agents needed to implement these devices, the submarket requires two additional agents, A_1 and A_2 .

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, \dots, G_7) $A_1: W_{12} = 1$ and $U_1(\mathbf{x}) = x_1$ $A_2: W_{26} = 1$ and $U_2(\mathbf{x}) = x_5$ $Conv_1 = Conv(1): (G_1, G_2)$ $Conv_2 = Conv(p_b): (G_6, G_7)$ $Comb(p_c, p_b, 1): ((G_2, G_7), G_3)$ $Spl(p_c, p_b, 1): (G_3, (G_4, G_5))$
Part 2:	Closed submarkets for the following linear relations $p_1 = p_c$ $p_2 = 1$ $p_4 = 1$ $p_5 = p_b$ $p_6 = p_a$ $p_7 = p_b$

Table 6: A closed submarket \mathcal{M}'_r that enforces $p_a = p_b p_c$

Lemma 4.10 *The submarket given in Table 6 (and illustrated in Figure 5) enforces $p_a = p_b p_c$ and is closed at equilibrium under the assumption $p_b \neq 0$.*

Proof : Let \mathbf{p} be an equilibrium price vector of the entire market, where $p_b > 0$. It is easy to see that prices p_2, p_4, p_5 and p_7 are strictly greater than zero, so these have to be consumed completely. Further, since the (*EQ.*) and (*LIN.*) submarkets implementing the second part of Table 6 are closed (Lemmas 4.6 and 4.8), the net endowment of goods G_1, \dots, G_7 for agents in this submarket, including those in devices, is exactly what they bring.

We have set $p_1 = p_c$ and the endowment of A_1 is one unit of G_2 . Good G_1 is desired only by the first agent in $Conv_1$; hence its net consumption costs p_c . Furthermore, since the price of G_2 is set to 1 and the parameter of $Conv_1$ is 1, the net endowment of $Conv_1$ will be p_c units of G_2 .

The first agent in $Comb$ wants G_2 and G_7 in the ratio 1:1 both of whose prices are positive. Moreover, net endowment p_c of Good G_2 is desired only by this agent, and therefore using (1) at equilibrium the agent has to consume p_c units of both the goods. Since prices of G_2 and G_7 are 1 and p_b respectively, the price of the net endowment of $Comb$ will be $p_b + 1$, and the amount will be p_c .

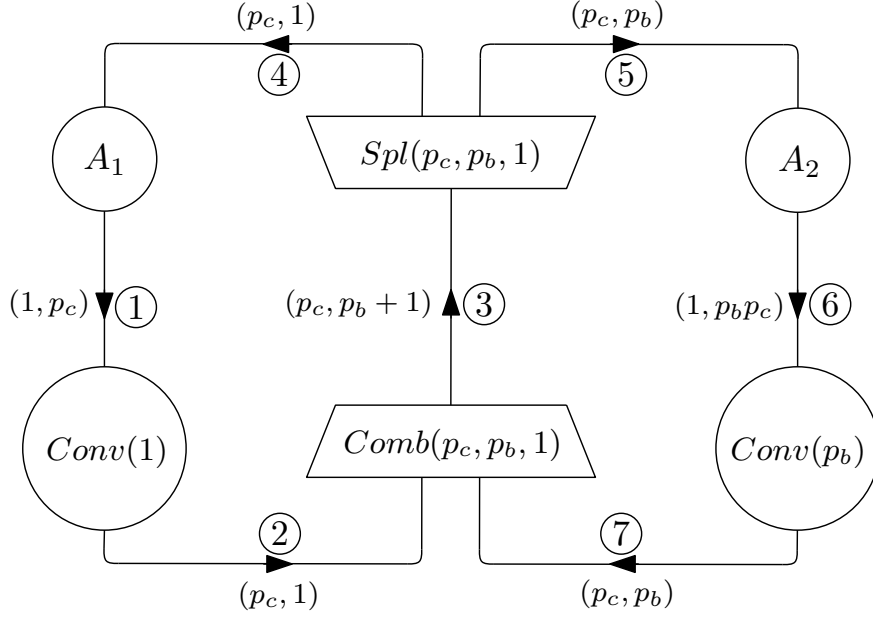


Figure 5: Flow of goods in Part 1 of Table 6. Wires are numbered, and wire i carries good r_i . The tuple on each wire represents (amount, price).

Thus, output of $Comb$ is p_c amount of good G_3 priced at $p_b + 1$, which has to be consumed by Spl . The prices p_4 and p_5 are set to 1 and p_b , respectively, thereby ensuring that the total worth of the net endowment and net consumption of Spl are equal. Finally, agent A_2 gets p_c amount of G_5 whose price is p_b and has one unit of G_6 as her endowment. Hence the price of G_6 must be $p_6 = p_a = p_b p_c$, as required. Good G_6 is only desired by the first agent in $Conv_2$. The net endowment of this device is p_c units of G_7 whose price is set to p_b , and this good is fully consumed by the first agent of $Comb$.

Note that it may be possible that p_c is zero which may force prices of some of G_1, \dots, G_7 goods to be zero, like G_1 . However, whoever consumes this good also want to consume another good with non-zero price, in the same proportion. And therefore demand of no good will exceed the supply. Since the total supply and demand of each of the seven goods are equal, the submarket is closed at these (equilibrium) prices. \square

Now we will modify the construction of Table 6 in order to remove the assumption $p_b > 0$ and $p_s = 1$. Consider the implementation of Table 7.

Lemma 4.11 *Consider the submarket \mathcal{M}_r of Table 7,*

- \mathcal{M}_r is a closed submarket.
- At equilibrium, \mathcal{M}_r enforces $p_a = \frac{p_b p_c}{p_s}$, and $p_c/p_s \leq H$.
- Every non-negative solution of $p_a = p_b p_c$, where $p_s > 0$ and $p_c \leq H$, gives an equilibrium of \mathcal{M}_r with $p_s = 1$.

Proof : The only difference between the market of Tables 7 and 6 are that p_b is replaced with $p_b + p_s$, and p_c with p_c/p_s . As $p_s > 0$ (Lemma 4.3) we have that $p_b + p_s > 0$, and $\frac{p_c}{p_s}$ is well-defined.

	<p>2 Agents (A_1, A_2), 2 Converters ($Conv_1, Conv_2$), 1 Combiner ($Comb$), 1 Splitter (Spl), and 7 Goods (G_1, \dots, G_7)</p>
Part 1:	<p>A_1: $W_{12} = 1$ and $U_1(\mathbf{x}) = x_1$ A_2: $W_{26} = 1$ and $U_2(\mathbf{x}) = x_5$ $Conv_1 = Conv(p_s)$: (G_1, G_2) $Conv_2 = Conv(p_b + p_s)$: (G_6, G_7) $Comb(\frac{p_c}{p_s}, p_b + p_s, p_s)$: ($(G_2, G_7), G_3$) $Spl(\frac{p_c}{p_s}, p_b + p_s, p_s)$: ($G_3, (G_4, G_5)$)</p>
Part 2:	<p>Closed submarkets for the following linear relations $p_1 = p_c$ $p_2 = p_s$ $p_4 = p_s$ $p_5 = p_b + p_s$ $p_7 = p_b + p_s$ $p_a + p_c = p_6$</p>

Table 7: A closed submarket \mathcal{M}_r that enforces $p_a = \frac{p_b p_c}{p_s}$

Further, the prices to be set in Part 2 of devices are still linear, even when $l = p_c/p_s$ in $Comb$ and Spl . Thus using Lemma 4.10 it follows that the market is closed and $p_6 = (p_b + p_s) \frac{p_c}{p_s}$. Then using the last linear relation enforced in Part 2 of Table 7 we get $p_a = \frac{p_b p_c}{p_s}$.

For the last part, consider a non-negative p_a, p_b and p_c such that $p_a = p_b p_c$. Set $p_s = 1$, the prices of goods G_1, G_2, G_4, G_5, G_7 as per Part 2 of Table 7, and $p_3 = p_b + 2p_s$ and $p_6 = (p_b + p_s) \frac{p_c}{p_s}$. For goods within devices, set their prices as per Part 2 of Tables 3, 4 and 5 respectively. For good G_4 in $Comb(\frac{p_c}{p_s})$ and good G_4 in $Spl(\frac{p_c}{p_s})$, set their prices to to $(p_b + 2p_s) * \frac{p_c}{p_s}$. It is easy to verify that this gives an equilibrium for market \mathcal{M}_r of Table 7. \square

4.4 Results

In this section we prove the main results using the claims established in Section 4.3. Given a system F of multivariate polynomials as in (4), construct an equivalent set of relations $R'(F)$ consisting of only three types of basic relations given in (8). Let K be the number of relations in $R'(F)$. For each relation $r \in [K]$, depending on its type, we construct a market \mathcal{M}_r as described in Tables 1, 2 and 7. Further, for each r of type $(QD.)$ replace the corresponding devices with agents of Tables 3, 4 and 5 respectively. Combine all the \mathcal{M}_r 's to form one market \mathcal{M} . Also add the agent of (9) in \mathcal{M} .

Since equilibrium prices of an Arrow-Debreu market are scale invariant, *i.e.*, if $\mathbf{p} = (p_1, \dots, p_g)$ is an equilibrium price vector then so is $\alpha \mathbf{p}$, $\forall \alpha > 0$, it is without loss of generality (wlog) to assume some kind of normalization. For example, $\sum_j p_j = 1$, or choose a good to be *numéraire*, *i.e.*, fix its price to 1.⁵ For the latter case, we require price of the numéraire good to be non-zero

⁵We note that, the reduction in [16] from fixed-point to Arrow-Debreu market with algebraic excess demand function, it is assumed that $\sum_j p_j = 1$ at equilibrium.

at equilibrium, and any good, for which an agent is non-satiated⁶, qualifies.

Given an equilibrium price vector \mathbf{p} of \mathcal{M} , we know that $p_s > 0$ due to Lemma 4.3. Henceforth by equilibrium prices \mathbf{p} of \mathcal{M} w.l.o.g. we mean equilibrium prices with $p_s = 1$. In the next two lemmas we establish that the equilibria of market \mathcal{M} exactly capture the solutions of system F .

Lemma 4.12 *If \mathbf{p} is an equilibrium price vector of \mathcal{M} , then $z_j = p_j, \forall j \in [n]$ is a solution of F .*

Proof : Due to Lemma 4.2, it is enough to show that \mathbf{p} is a solution of $R'(F)$. The submarket \mathcal{M}_r , constructed for relation r of $R'(F)$, is closed and enforces r at \mathbf{p} (first two statements of Lemmas 4.6, 4.8, and 4.11). Since \mathcal{M} is a union of \mathcal{M}_r 's, \mathbf{p} has to satisfy each of the relation of $R'(F)$. \square

Next we map solutions of F to equilibria of market \mathcal{M} .

Lemma 4.13 *If \mathbf{z} is a solution of F , then there exists equilibrium prices \mathbf{p} of market \mathcal{M} , where $p_s = 1$ and $p_j = z_j, \forall j \in [n]$.*

Proof : Using Lemma 4.2, we can construct a non-negative solution \mathbf{p}' of $R'(F)$ using \mathbf{z} such that $p'_s = 1, p'_j = z_j, \forall j \in [n]$, and $p'_j \leq H, \forall j \in [N]$.

Construct prices \mathbf{p} of market \mathcal{M} , where set $p_j = p'_j, \forall j \in [N]$ and $p_s = 1$. Set $x_{ss} = 1$ for agent A_s of (9). Last statement of Lemmas 4.6, 4.8, and 4.11, imply that in each \mathcal{M}_r , \mathbf{p} can be extended to yield an equilibrium. Since equilibrium in \mathcal{M} consists of equilibrium in each \mathcal{M}_r with same prices for common goods, combining these gives an equilibrium of \mathcal{M} . \square

Thus establishing the strong relation between solutions of F and equilibria of market \mathcal{M} , next we prove the main theorem of the paper which will give all the desired hardness results as corollaries.

Theorem 4.14 *Equilibrium prices of market \mathcal{M} , projected onto (p_1, \dots, p_n) , are in one-to-one correspondence with the solutions of F . Further \mathcal{M} can be expressed using polynomially many bits in the size $[F]$, i.e., $size[\mathcal{M}] = poly(size[F])$.*

Proof : First part follows using Lemmas 4.12 and 4.13. For the second part, it is enough to show that $size[\mathcal{M}] = poly(size[R'(F)])$, due to Lemma 4.2. Let L be the size of $R'(F)$.

For each of the relation $r \in [K]$ of system $R'(F)$, we add $O(1)$ agents and goods in \mathcal{M}_r , as described in Tables 1, 2, and 7 together with Tables 3, 4 and 5. Let n_g and n_a be the #goods and #agents in \mathcal{M} , then we have $n_g = O(K)$ and $n_a = O(K)$.

Clearly, each agent of \mathcal{M}_r brings $O(1)$ goods and has utility for exactly those many goods, and these markets are closed submarkets (see Definition 4.5). The utility functions of the agents are Leontief which can be written as (good id, coefficient). The endowments can be captured similarly.

Thus the encoding of endowments and utility functions of agents in \mathcal{M}_r requires: (i) $O(\log K)$ if r is of type (EQ.), (ii) $O(\log K + size(B, C, D))$ if r is of type (LIN.), and (iii) $O(\log K + L)$ if r is of type (QD.) as $size[H] = O(L)$, where H is a constant defined in Section 4.2.

Since \mathcal{M} is a union of $\mathcal{M}_r, \forall r \in [K]$, and the agent of (9), the size of \mathcal{M} is at most $O(KL)$. \square

Theorem 4.14 shows that finding solutions of F can be reduced to finding equilibria of an exchange market with Leontief utility functions.

⁶An agent is said to be non-satiated for good j if at any given bundle she can obtain more utility by consuming additional amount of good j .

As discussed in Section 3, the problem of computing a Nash equilibrium of a 3-player game \mathbf{A} can be formulated as finding a solution of system $F_{NE}(\mathbf{A})$ (3) of multivariate polynomials in which variables are bounded between $[0, 1]$ (Lemma 3.2). Note that $size[F_{NE}(\mathbf{A})] = O(size(\mathbf{A}))$. Further, since taking projection on a set of coordinates is a linear function, the next theorem follows using the formulation of (3), together with Lemma 3.2, and Theorems 3.3 and 4.14 (see Section 3.1 for the reduction requirements for class FIXP).

Theorem 4.15 *Computing an equilibrium of an exchange market with 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.*

Further since the class of Leontief utility functions is a special subclass of piecewise-linear concave (PLC) utility functions, under which goods need not be only complementary (like Leontief) and substitute (like separable PLC), but can be arbitrary combination of these and much more, the next theorem follows.

Theorem 4.16 *Computing equilibrium of an exchange market with piecewise-linear concave 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.*

Note that in Theorems 4.15 and 4.16 the resulting market is guaranteed to have an equilibrium, since it was constructed from an instance of 3-Nash which always has a NE (Nash's theorem [26]). However in general an AD market may not have an equilibrium. Checking if an arbitrary exchange market with SPLC utility function has an equilibrium is known to be NP-complete [33]. We study the analogous question for Leontief (and in turn PLC) markets. It turns out that the complexity of these questions is captured by the class ETR.

Theorem 3.5 shows that checking if a 3-player game \mathbf{A} has NE within 0.5-ball at origin in l_∞ norm is ETR-complete. Clearly, this problem can be reduced to finding a solution of $F_{NE}(\mathbf{A})$ of (3) with upper-bound on z_{ps} 's changed from 1 to 0.5 (Lemma 3.2). If this system is reduced to a market \mathcal{M} , then \mathcal{M} will have an equilibrium if and only if game \mathbf{A} has a NE within 0.5-ball at origin (Theorem 4.14). Thus we get the following result.

Theorem 4.17 *Checking existence of an equilibrium in an exchange market with Leontief utility functions, and in market with PLC utility functions is ETR-hard.*

[18] gave a reduction from an exchange market \mathcal{M} with arbitrary concave utility functions to an equivalent Arrow-Debreu market \mathcal{M}' with firms, where utility functions of all the agents are linear. It turns out that \mathcal{M}' has all the goods of \mathcal{M} , in addition to others, and equilibrium prices of \mathcal{M} are in one-to-one correspondence with the equilibrium prices of \mathcal{M}' projected onto the prices of common goods. Further the production functions of \mathcal{M}' are precisely the utility functions in \mathcal{M} , hence representation of \mathcal{M}' is in the order of the representation of \mathcal{M} . Therefore, this reduction together with Theorems 4.15 and 4.17, gives the next two results.

Corollary 4.18 *Computing equilibrium of an Arrow-Debreu market with linear utility functions and Leontief production, and in turn PLC (polyhedral) production sets, is FIXP-hard. In particular, the corresponding Decision, (Strong) Approximation, and Partial Computation problems are hard respectively for $FIXP_d$, $FIXP_a$ and $FIXP_{pc}$.*

Corollary 4.19 *Checking existence of an equilibrium in an Arrow-Debreu market with linear utility functions and Leontief production, and in turn PLC (polyhedral) production sets, is ETR-hard.*

Next we show that checking existence of an equilibrium in markets with PLC utility functions and PLC production sets is in ETR.

5 Existence of Equilibrium in ETR

Using the nonlinear complementarity problem (NCP) formulation of [17] to capture equilibria of PLC markets, in this section we show that checking for existence of equilibrium in PLC markets is in ETR, and therefore ETR-complete using Corollary 4.19. For the sake of completeness next we present the NCP formulation derived in [17].

Recall the PLC utility functions and PLC production sets defined in Sections 2.1 and 2.2 respectively. Using the optimal bundle and optimal production plan conditions at equilibrium for such a market, [17] derived the nonlinear complementarity problem (NCP) formulation AD-NCP for market equilibrium as shown in Table 8, and showed the Lemma 5.1. All the variables in the NCP of Table 8 are non-negative, and we omit this condition for the sake of brevity.

Table 8: AD-NCP

$$\begin{aligned}
\forall(f, k) : \sum_j D_{fj}^k x_{fj}^s &\leq \sum_j C_{fj}^k x_{fj}^r + T_f^k & \text{and} & & \delta_f^k (\sum_j D_{fj}^k x_{fj}^s - \sum_j C_{fj}^k x_{fj}^r - T_f^k) &= 0 \\
\forall(f, j) : p_j &\leq \sum_k D_{fj}^k \delta_f^k & \text{and} & & x_{fj}^s (p_j - \sum_k D_{fj}^k \delta_f^k) &= 0 \\
\forall(f, j) : \sum_k C_{fj}^k \delta_f^k &\leq p_j & \text{and} & & x_{fj}^r (\sum_k C_{fj}^k \delta_f^k - p_j) &= 0 \\
\forall(i, j) : \sum_k U_{ij}^k \gamma_i^k &\leq \lambda_i p_j & \text{and} & & x_{ij} (\sum_l U_{ij}^k \gamma_i^k - \lambda_i p_j) &= 0 \\
\forall(i, k) : u_i &\leq \sum_j U_{ij}^k x_{ij} + T_i^k & \text{and} & & \gamma_i^k (u_i - \sum_j U_{ij}^k x_{ij} - T_i^k) &= 0 \\
\forall i : \sum_j x_{ij} p_j &\leq \sum_j W_{ij} p_j + \sum_f \Theta_{if} \phi_f & \text{and} & & \lambda_i (\sum_j x_{ij} p_j - \sum_j W_{ij} p_j - \sum_f \Theta_{if} \phi_f) &= 0 \\
\forall j : \sum_i x_{ij} + \sum_f x_{fj}^r &\leq 1 + \sum_f x_{fj}^s & \text{and} & & p_j (\sum_i x_{ij} + \sum_f x_{fj}^r - 1 - \sum_f x_{fj}^s) &= 0 \\
\forall i : \sum_k \gamma_i^k &= 1 & \text{and} & & u_i = \lambda_i (\sum_j W_{ij} p_j + \sum_f \Theta_{if} \phi_f) + \sum_k \gamma_i^k T_i^k \\
\forall f : \phi_f &= \sum_k \delta_f^k T_f^k & \text{and} & & \sum_j p_j &= 1
\end{aligned}$$

Lemma 5.1 [17] *If $(\mathbf{p}, \mathbf{x}, \mathbf{x}^s, \mathbf{x}^r, \boldsymbol{\lambda}, \boldsymbol{\gamma}, \boldsymbol{\delta})$ is a solution of AD-NCP, then $(\mathbf{p}, \mathbf{x}, \mathbf{x}^s, \mathbf{x}^r)$ is a market equilibrium. Further, if $(\mathbf{p}, \mathbf{x}, \mathbf{x}^s, \mathbf{x}^r)$ is a market equilibrium, then $\exists(\boldsymbol{\lambda}, \boldsymbol{\gamma}, \boldsymbol{\delta})$ such that $(\mathbf{p}, \mathbf{x}, \mathbf{x}^s, \mathbf{x}^r, \boldsymbol{\lambda}, \boldsymbol{\gamma}, \boldsymbol{\delta})$ is a solution of AD-NCP.*

Due to Lemma 5.1, checking if the market has an equilibrium is equivalent to checking if AD-NCP admits a solution. Since all the inequalities and equalities in AD-NCP are polynomial, and all the coefficients in these polynomials are rational numbers, AD-NCP can be represented using signature $\{0, 1, -1, +, *, <, \leq, =\}$. The denominators of the coefficients can be removed by taking least common multiple (LCM) while keeping the size of coefficients polynomial in the original size. Therefore, checking if AD-NCP has a solution can be formulated in ETR (see Section 3.2 for definition), and we get the following result using Theorem 4.17.

Theorem 5.2 *Checking existence of an equilibrium in an exchange market with piecewise linear concave utility functions is ETR-complete.*

The next result follows using Corollary 4.19.

Theorem 5.3 *Checking existence of an equilibrium in an Arrow-Debreu market with PLC utility functions and PLC (polyhedral) production sets is ETR-complete.*

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 [33] 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 [6]. 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.

Acknowledgement: We wish to thank Mihalis Yannakakis for valuable discussions.

References

- [1] K. Arrow and G. Debreu. Existence of an equilibrium for a competitive economy. *Econometrica*, 22:265–290, 1954.
- [2] M. S. Bazarra, H. D. Sherali, and C. M. Shetty. *Nonlinear Programming: Theory and Algorithms*. John Wiley & Sons, 2006.
- [3] L. Blum, F. Cucker, M. Shub, and S. Smale. *Complexity and Real Computation*. Springer-Verlag, 1998.
- [4] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2009.
- [5] X. Chen, D. Dai, Y. Du, and S.-H. Teng. Settling the complexity of Arrow-Debreu equilibria in markets with additively separable utilities. In *IEEE Annual Symposium on Foundations of Computer Science*, 2009.
- [6] X. Chen, D. Paparas, and M. Yannakakis. The complexity of non-monotone markets. In *ACM Symposium on the Theory of Computing*, pages 181–190, 2013.
- [7] B. Codenotti, S. Pemmaraju, and K. Varadarajan. On the polynomial time computation of equilibria for certain exchange economies. In *ACM-SIAM Annual Symposium on Discrete Algorithms*, pages 72–81, 2005.
- [8] B. Codenotti, A. Saberi, K. Varadarajan, and Y. Ye. The complexity of equilibria: Hardness results for economies via a correspondence with games. *Theoretical Computer Science*, 408(2-3):188–198, 2008.

- [9] G. Debreu. Excess demand functions. *Journal of Mathematical Economics*, 1:15–22, 1974.
- [10] N. Devanur and R. Kannan. Market equilibria in polynomial time for fixed number of goods or agents. In *FOCS*, pages 45–53, 2008.
- [11] N. Devanur, C. H. Papadimitriou, A. Saberi, and V. V. Vazirani. Market equilibrium via a primal-dual algorithm for a convex program. *Journal of the ACM*, 55(5), 2008.
- [12] N. Devanur and V. Vazirani. An improved approximation scheme for computing Arrow-Debreu prices for the linear case. In *Proceedings of 23rd FSTTCS*, 2003.
- [13] R. Duan and K. Mehlhorn. A combinatorial polynomial algorithm for the linear Arrow-Debreu market. In *ICALP*, 2013.
- [14] B. C. Eaves. A finite algorithm for the linear exchange model. *Journal of Mathematical Economics*, 3:197–203, 1976.
- [15] E. Eisenberg. Aggregation of utility functions. *Management Sciences*, 7:337–350, 1961.
- [16] K. Etessami and M. Yannakakis. On the complexity of Nash equilibria and other fixed points. *SIAM Journal on Computing*, 39(6):2531–2597, 2010.
- [17] J. Garg, R. Mehta, and V. V. Vazirani. 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*, 2014.
- [18] J. Garg and V. V. Vazirani. On equilibrium computation in markets with production. In *SODA*, 2014.
- [19] R. Garg and S. Kapoor. Auction algorithms for market equilibrium. In *ACM Symposium on the Theory of Computing*, 2004.
- [20] R. Garg, S. Kapoor, and V. V. Vazirani. An auction-based market equilibrium algorithm for the separable gross substitutability case. In *Proceedings, APPROX*, 2004.
- [21] K. Jain. A polynomial time algorithm for computing the Arrow-Debreu market equilibrium for linear utilities. *SIAM Journal on Computing*, 37(1):306–318, 2007.
- [22] K. Jain, M. Mahdian, and A. Saberi. Approximating market equilibrium. In *Proceedings of 6th APPROX*, 2003.
- [23] K. Jain and K. Varadarajan. Equilibria for economies with production: Constant-returns technologies and production planning constraints. In *ACM-SIAM Annual Symposium on Discrete Algorithms*, pages 688–697, 2006.
- [24] R. Mantel. On the characterization of excess demand. *Journal of Economic Theory*, 7:348–353, 1974.
- [25] A. Mas-Colell, M. D. Whinston, and J. R. Green. *Microeconomic Theory*. Oxford University Press, 1995.
- [26] J. Nash. Non-cooperative games. *Annals of Mathematics*, 54(2):289–295, September 1951.
- [27] J. B. Orlin. Improved algorithms for computing Fisher’s market clearing prices. In *ACM Symposium on the Theory of Computing*, pages 291–300, 2010.

- [28] C. H. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *JCSS*, 48(3):498–532, 1992.
- [29] S. A. T. Rizvi. The Sonnenschein-Mantel-Debreu results after thirty years. *History of Political Economy*, 38, 2006.
- [30] M. Schaefer and D. Štefankovič. Fixed points, Nash equilibria, and the existential theory of the reals. Manuscript, 2011.
- [31] H. Sonnenschein. Market excess-demand functions. *Econometrica*, 40(3):549–563, 1972.
- [32] H. Sonnenschein. Do Walras’ identity and continuity characterize the class of community excess demand functions? *Journal of Economic Theory*, 6:345–354, 1973.
- [33] V. V. Vazirani and M. Yannakakis. Market equilibrium under separable, piecewise-linear, concave utilities. *Journal of ACM*, 58(3):10:1–10:25, 2011.
- [34] L. A. Vegh. Concave generalized flows with applications to market equilibria. In *IEEE Annual Symposium on Foundations of Computer Science*, 2012.
- [35] L. A. Vegh. Strongly polynomial algorithm for a class of minimum-cost flow problems with separable convex objectives. In *ACM Symposium on the Theory of Computing*, 2012.
- [36] M. Yannakakis. Personal communication, 2013.
- [37] Y. Ye. A path to the Arrow-Debreu competitive market equilibrium. *Math. Program.*, 111(1-2):315–348, 2008.