A Complementary Pivot Algorithm for Market Equilibrium under Separable, Piecewise-Linear Concave Utilities

Jugal Garg[1*] Ruta Mehta[1†] Milind Sohoni[2] Vijay V. Vazirani[1‡]

[1]Georgia Institute of Technology
jgarg,rmehta,vazirani@cc.gatech.edu
[2]Indian Institute of Technology, Bombay
sohoni@cse.iitb.ac.in

Abstract

Using Lemke's scheme, we give a complementary pivot algorithm for computing an equilibrium for Arrow-Debreu markets under separable, piecewise-linear concave (SPLC) utilities. Despite the PPAD-completeness of this case, experiments indicate that our algorithm is practical – on randomly generated instances, the number of iterations it needs is linear in the total number of segments (i.e., pieces) in all the utility functions specified in the input.

Our paper settles a number of open problems:

- Eaves [16] gave an LCP formulation and a Lemke-type algorithm for linear Arrow-Debreu model. We generalize both to the SPLC case, hence settling the relevant part of his open problem.
- Our path following algorithm for SPLC markets, together with a result of Todd [54], gives a direct proof of membership of such markets in PPAD and settles a question of Vazirani and Yannakakis [55].
- We settle a question of Devanur and Kannan [10] of obtaining a "systematic way of finding equilibrium instead of the brute-force way" for the separable case and we obtain a strongly polynomial algorithm if the number of goods or agents is constant.
- We give a combinatorial way of interpreting Eaves' algorithm for the linear case, hence answering Eaves' question [16], "That the algorithm can be interpreted as a 'global market adjustment mechanism' might be interesting to explore."

1 Introduction

The study of computability of market equilibria started twelve years ago in theoretical computer science, and once polynomial time algorithms were found for markets under linear utility

*Research supported by Algorithms and Randomness Center, Georgia Tech.
†Research supported by NSF Grants CCF-0914732 and CCF-1216019.
‡Research supported by NSF Grants CCF-0914732 and CCF-1216019, and a Guggenheim Fellowship.

functions [11, 12, 31, 24, 30, 57, 42, 56, 14], interest shifted to more general utility functions. In economics, it is customary to assume that utility functions are concave, since they capture the important condition of decreasing marginal utilities[1]. Since we are studying computability in a finite precision model of computation, we need to restrict attention to piecewise-linear concave (PLC) utility functions; clearly, by making the pieces fine enough, we can obtain a good approximation to the original utility functions.

Within the class of PLC utilities, it is important to distinguish between the separable and non-separable cases. As detailed in Section 1.1, whereas the former always admit rational equilibria, if all parameters of the market are rational numbers, the latter may have only irrational equilibria and are difficult to deal with. For these reasons, understanding the complexity of markets under separable PLC (SPLC) utility functions became the next major challenge. This long-standing open question was settled in [5], showing that the problem is PPAD-complete for both Fisher and Arrow-Debreu market models.

As a result, under the assumption P $\neq$ PPAD, a polynomial time algorithm for this case is not possible. On the other hand, efficiently computing market equilibria is of practical importance, e.g., see [18, 51], and its computability had been the subject of intense work in economics as well [4, 17, 20, 22, 37, 49]. Our main result is a complementary pivot algorithm for computing an equilibrium in an Arrow-Debreu market under SPLC utility functions. Experimental results on randomly generated instances suggest that our algorithm will be fast in practice.

Starting with the (pivoting-based) simplex algorithm for linear programming [8], by now several prominent algorithms exhibiting the following phenomena are known: They perform well in practice even though their worst case behavior is exponential; the latter is exhibited via intricately doctored up instances that are designed to make the algorithm perform poorly, e.g., the Klee-Minty example for simplex [33]. Another algorithm exhibiting this phenomena is the classical Lemke-Howson algorithm for computing a Nash equilibrium of a 2-person bimatrix game, which will henceforth be called 2-Nash [36, 44]; this is also a complementary pivot algorithm. We expect our algorithm to also be exponential in the worst case, and we leave the open problem of finding such a family of instances.

In addition to being practical, the Lemke-Howson algorithm has yielded deep structural properties of 2-Nash equilibria, such as oddness of the number of equilibria as well as index, degree and stability [25, 48, 50]. It also motivated the definition of the complexity class PPAD [43] as described in the following quote from [55]:

> *The definition of the class PPAD was designed to capture problems that allow for path following algorithms, in the style of the algorithms of Lemke-Howson [36] and Scarf [47]. Our result, showing membership in PPAD for both market models under*

[1] Furthermore, such utilities introduce convexity in the problem, which is a natural condition without which even fixed point theorems are not applicable. Additionally, convexity is crucial for designing algorithms for the problem.

separable, piecewise-linear, concave utility functions, establishes the existence of such path following algorithms for finding equilibria for these market models; however, it does so indirectly, by appealing to the characterization of PPAD given in [21]. It will be interesting to obtain natural, direct algorithms for this task (hence leading to a more direct proof of membership in PPAD), which may be useful for computing equilibria in practice.

The starting point of our work was the open problem described in this quote. Our algorithm is a path following algorithm and yields, together with Todd's result [54], a direct proof of membership of the problem in PPAD. We build on the work of Eaves [15], who obtained a path following algorithm for Arrow-Debreu markets under linear utilities, using Lemke's algorithm.

Both Lemke-Howson and Lemke's algorithms are complementary pivot algorithms; however, their mode of operation has some basic differences. The former requires a starting dummy solution, while the latter constructs a starting solution by introducing an extra dimension, see Section 2. Also, in Lemke's algorithm, all the complementarity conditions are satisfied on the followed path, whereas in the Lemke-Howson algorithm, all but one condition are satisfied on the path. Lemke's algorithm is useful for solving a broader class of problems, including 2-Nash [35, 53, 34].

We note that whereas several complementary pivot algorithms have been given for Nash equilibria after the work of Lemke and Howson [35, 19, 28, 34, 53], no such algorithms were obtained for market equilibria following Eaves' work. In the next section, we will attempt to give a reason for this.

1.1 The importance of rationality

A complementary pivot algorithm is possible for a problem only if it exhibits *rationality*, i.e., if all parameters are set to rational numbers, the solution must be rational.

Eaves ends his 1975 report [15] with the following sentence:

Also under study are extensions of the overall method to include piecewise linear concave utilities, production, etc., if successful, this avenue could prove important in real economic modeling.

However, in his 1976 journal paper [16], he drops this sentence and instead adds:

... now let us suppose that each trader has a piecewise linear concave utility function in lieu of a linear one. We asked the same question, given rational data, does there exist a rational equilibrium? Andreu Mas-Colell (personal communication, 1975) generated the following 3-trader and 2-good example to demonstrate the negative.

He goes on to stating an example of a market with Leontief utilities that has only irrational equilibria and concludes:

Consequently, one can conclude that Lemke's algorithm cannot be used to solve this class of exchange problems.

One can surmise that Eaves did not consider the case of SPLC markets. Moreover, rationality of equilibria for this case was established only in 2009 – independently in [10] and [55]. Indeed, rationality had played a crucial role in his own work on the linear case, since rationality is essential for obtaining an LCP formulation for a problem.

Stymied in an effort to compute an equilibrium of the linear pure exchange model using Lemke's algorithm, the author approached David Gale with the following question. If W and U are rational, does there exist a rational equilibrium? The success of the present paper rests upon the argument given in Gale (private communication 1974) which supplied an affirmative answer ...

1.2 Our results

As stated above, our main result is a Lemke-type algorithm for SPLC Arrow Debreu markets, hence solving the relevant subcase of the question posed by Eaves [15]. At a technical level, this involves two main ideas: The first is to derive an LCP formulation for the given problem, and the second is to prove that the polyhedron associated with the augmented LCP has no secondary rays. For the second part, we need to assume that the SPLC market satisfies *strong connectivity* – this is among the weakest known conditions for existence of a market equilibrium (see Section 3.2). Without imposing any condition, an SPLC market may not admit an equilibrium, and determining if it has one is NP-complete [55].

We note that a deficiency of Lemke's algorithm is that in general it does not guarantee a solution: this happens if the path starting with the primary ray ends in a secondary ray, see Sections 2 and 5. For several classes of LCPs it is known that their polyhedra do not have secondary rays, and hence Lemke's algorithm is guaranteed to terminate in a solution [7]. However, our LCP does not lie in any of these classes [1], hence necessitating a separate proof of this fact.

Our algorithm yields several additional results analogous to those yielded by the Lemke-Howson algorithm. First, together with a result of Todd [54], it yields a path following algorithm for SPLC markets and therefore a direct proof of membership of this case in PPAD, hence settling the open problem of [55]. Second, it yields the first elementary proof of existence of equilibrium for SPLC markets, i.e., without using fixed point theorems. The best known example of an elementary proof of existence is Lemke and Howson's proof for existence of an equilibrium for 2-Nash, which follows from their algorithm [36]. Scarf has used their algorithm to derive other elementary proofs, e.g., for showing that balanced games have a non-empty core [45, 46].

Third, it enables us to prove that SPLC markets have an odd number of equilibria (up to scaling), assuming non-degeneracy, and we believe it should yield other insights as well, see Section 10. In the past, economists have considered the issue of oddness of equilibria for regular markets, i.e., markets whose demand functions are continuously differentiable. Debreu [9] showed that such markets have a finite number of equilibria and then using index theorems, Dierker [13] showed that the number of equilibria is odd. We note that in general, in an SPLC market an agent can have multiple optimal bundles, hence these markets don't even have a well defined demand function and are not regular. It is worth mentioning that economists have also obtained algorithms for computing equilibria in regular markets, typically via homotopy-based methods. Once again these methods are not applicable to our markets for reasons stated above.

At this point, it is natural to ask whether the pivoting steps of our algorithm have an interpretation in the market itself. Indeed, this question was asked by Eaves [16] as well, "That the algorithm can be interpreted as a 'global market adjustment mechanism' might be interesting to explore." We give a combinatorial way of interpreting Eaves' algorithm for the linear case. We note that it is quite different from the combinatorial interpretation of Garg et. al's algorithm, which is based on the Lemke-Howson approach, for the linear case [23].

For SPLC markets, Devanur and Kannan [10] gave a polynomial time algorithm to compute an equilibrium when the number of goods or agents is a constant. Their algorithm resorts to an exhaustive search of all possible configurations of allocations, and they leave the question of obtaining a "systematic way of finding equilibrium instead of the brute-force way." We settle this question and we improve their running time to strongly polynomial.

This is achieved by showing that if the number of goods or agents is a constant, say c, then the number of vertices (of the polyhedron) on the path starting with the primary ray is polynomially bound. Of course, c occurs in the exponent of this polynomial, i.e., it is $n^{O(c)}$. However unlike their algorithm, which explores all of the polynomially bounded (with c in the exponent) configurations on each input, our algorithm does not do exhaustive search and will terminate very quickly on typical inputs.

2 The Linear Complementarity Problem and Lemke's Algorithm

Given an $n \times n$ matrix $\boldsymbol{M}$, and a vector $\boldsymbol{q}$, the linear complementarity problem[2] asks for a vector $\boldsymbol{y}$ satisfying the following conditions:[3]

$$\boldsymbol{My} \leq \boldsymbol{q}, \quad \boldsymbol{y} \geq 0 \quad \text{and} \quad \boldsymbol{y} \cdot (\boldsymbol{q} - \boldsymbol{My}) = 0. \tag{1}$$

The problem is interesting only when $\boldsymbol{q} \not\geq 0$, since otherwise $\boldsymbol{y} = 0$ is a trivial solution. Let us introduce slack variables $\boldsymbol{v}$ to obtain the equivalent formulation.

$$\boldsymbol{My} + \boldsymbol{v} = \boldsymbol{q}, \quad \boldsymbol{y} \geq 0, \quad \boldsymbol{v} \geq 0 \quad \text{and} \quad \boldsymbol{y} \cdot \boldsymbol{v} = 0. \tag{2}$$

[2]We refer the reader to [7] for a comprehensive treatment of notions presented in this section.

[3]The standard way of formulating an LCP is '$\boldsymbol{My} \geq -\boldsymbol{q},\ \boldsymbol{y} \geq 0$ and $\boldsymbol{y} \cdot (\boldsymbol{My} + \boldsymbol{q}) = 0$'. However, the LCP we construct in Section 4 is more amenable to the form given in (1).

The reason for imposing non-negativity on the slack variables is that the first condition in (1) implies $\boldsymbol{q} - \boldsymbol{M}\boldsymbol{y} \geq 0$. Let $\mathcal{P}$ be the polyhedron in $2n$ dimensional space defined by the first three conditions; we will assume that $\mathcal{P}$ is non-degenerate. Under this condition, any solution to (2) will be a vertex of $\mathcal{P}$, since it must satisfy $2n$ equalities. Note that the set of solutions may be disconnected.

An ingenious idea of Lemke was to introduce a new variable and consider the system, which is called the *augmented LCP*:

$$\boldsymbol{M}\boldsymbol{y} + \boldsymbol{v} - z\mathbf{1} = \boldsymbol{q}, \quad \boldsymbol{y} \geq 0, \quad \boldsymbol{v} \geq 0, \quad z \geq 0 \quad \text{and} \quad \boldsymbol{y} \cdot \boldsymbol{v} = 0. \tag{3}$$

Let $\mathcal{P}'$ be the polyhedron in $2n+1$ dimensional space defined by the first four conditions of the augmented LCP; again we will assume that $\mathcal{P}'$ is non-degenerate. Since any solution to (3) must still satisfy $2n$ equalities, the set of solutions, say S, will be a subset of the one-skeleton of $\mathcal{P}'$, i.e., it will consist of edges and vertices of $\mathcal{P}'$. Any solution to the original system must satisfy the additional condition $z = 0$ and hence will be a vertex of $\mathcal{P}'$.

Now S turns out to have some nice properties. Any point of S is *fully labeled* in the sense that for each i, $y_i = 0$ or $v_i = 0$. We will say that a point of S *has double label* i if $y_i = 0$ and $v_i = 0$ are both satisfied at this point. Clearly, such a point will be a vertex of $\mathcal{P}'$ and it will have only one double label. Since there are exactly two ways of relaxing this double label, this vertex must have exactly two edges of S incident at it. Clearly, a solution to the original system (i.e., satisfying $z = 0$) will be a vertex of $\mathcal{P}'$ that does not have a double label. On relaxing $z = 0$, we get the unique edge of S incident at this vertex.

As a result of these observations, we can conclude that S consists of paths and cycles. Of these paths, Lemke's algorithm explores a special one. An unbounded edge of S such that the vertex of $\mathcal{P}'$ it is incident on has $z > 0$ is called a *ray*. Among the rays, one is special – the one on which $\boldsymbol{y} = 0$. This is called the *primary ray* and the rest are called *secondary rays*. Now Lemke's algorithm explores, via pivoting, the path starting with the primary ray. This path must end either in a vertex satisfying $z = 0$, i.e., a solution to the original system, or a secondary ray. In the latter case, the algorithm is unsuccessful in finding a solution to the original system; in particular, the original system may not have a solution.

Remark: Observe that $z\mathbf{1}$ can be replaced by $z\boldsymbol{a}$, where vector $\boldsymbol{a}$ has a 1 in each row in which $\boldsymbol{q}$ is negative and has either a 0 or a 1 in the remaining rows, without changing its role; in our algorithm, we will set a row of $\boldsymbol{a}$ to 1 if and only if the corresponding row of $\boldsymbol{q}$ is negative. As mentioned above, if $\boldsymbol{q}$ has no negative components, (1) has the trivial solution $\boldsymbol{y} = 0$. Additionally, in this case Lemke's algorithm cannot be used for finding a non-trivial solution, since it is simply not applicable. However, Lemke-Howson scheme is applicable for such a case; it follows a complementary path in the original polyhedron (2) starting at $\boldsymbol{y} = 0$. It guarantees termination at a non-trivial solution if the polyhedron is bounded.

3 Arrow-Debreu Markets with SPLC Utility Functions

The Arrow-Debreu market model [2] consists of a set $\mathcal{G}$ of divisible goods and a set $\mathcal{A}$ of agents; let $|\mathcal{G}| = n$ and $|\mathcal{A}| = m$. Assume that the goods are numbered from 1 to n and the agents are numbered from 1 to m. Each agent $i \in \mathcal{A}$ has an initial endowment of goods, say $(w^i_1, \ldots, w^i_n)$ where $w^i_j \geq 0,\ \forall j \in \mathcal{G}$. These and other parameters defined below will be assumed to be rational numbers. Without loss of generality (w.l.o.g.) we assume that each agent i has a positive amount of at least one good and the total quantity of every good is unit.

In this paper, we deal with the case of SPLC utility functions. For each agent i and good j we are specified a function $f^i_j : \mathbf{R}^+ \rightarrow \mathbf{R}^+$ which is (non-negative) non-decreasing, piecewise-linear and concave, and gives the utility that i derives as a function of the amount of good j that she receives. Her overall utility, $u_i(\boldsymbol{x})$, for a bundle $\boldsymbol{x} = (x_1, \ldots, x_n)$ of goods is additively separable over the goods, i.e., $u_i(\boldsymbol{x}) = \sum_{j \in G} f^i_j(x_j)$.

Given prices $\boldsymbol{p} = (p_1, \ldots, p_n)$ for all the goods, define the *earning of agent* i to be $\sum_{j\in\mathcal{G}} w^i_j p_j$, i.e., the amount of money she earns by selling her initial endowment. Agent i uses this money to buy an *optimal bundle of goods*, i.e., a bundle that maximizes her utility. We say that $\boldsymbol{p}$ is a *market clearing price vector* if there are choices of optimal bundles for the agents such that after each agent is given such a bundle, there is no deficiency of any good and no surplus of any good having a positive price (note that surplus quantities of any good having zero price can be freely disposed off without decreasing agents' utilities), i.e., the market clears. The problem is to find such market clearing or *equilibrium prices*.

We call each piece of f^i_j a *segment*. The number of segments in function f^i_j is denoted by s_{ij} and the k^{th} segment of f^i_j is denoted by the triple (i, j, k). The slope of a segment specifies the rate at which the agent derives utility per unit of additional good received. Suppose segment (i, j, k) has domain $[a, b] \subseteq \mathbf{R}^+$, and slope c. Then, we define $u^i_{jk} = c$ and $l^i_{jk} = b - a$. Note that for each function f^i_j the amount of the last segment is infinity. However, since the total amount of good j available in the market is unit, we assume w.l.o.g. that the amount of the last segment is a small constant greater than one. Clearly, $\forall k < s_{ij},\ u^i_{jk} > u^i_{j(k+1)} \geq 0$. We will denote this market by $\mathcal{M}$.

3.1 Characterizing optimal baskets

Next, we characterize optimal baskets of an agent i relative to prices $\boldsymbol{p}$. Define the *bang-per-buck of agent* i *from segment* (j, k) *relative to prices* $\boldsymbol{p}$ to be

$$\mathrm{bpb}^i_{jk} = \frac{u^i_{jk}}{p_j}.$$

We take $0/0$ as 0. The value bpb^i_{jk} represents the utility derived by agent i per unit of money while obtaining good j corresponding to segment (j, k).

Clearly, i's optimal bundle will consist of goods obtained on segments yielding highest possible bang per buck and can be computed as follows. Sort i's segments by decreasing bang per buck and partition the segments by equality, i.e., each equivalence class will consist of all segments having equal bang-per-buck. Let the classes be: $Q_1, Q_2, \ldots$. At prices $\boldsymbol{p}$, the segments in Q_l make i equally happy, and strictly happier than those in $Q_{l+1}, Q_{l+2}, \ldots$. Hence, she would start buying partitions in order, until all her money ($\sum_j w^i_j p_j$) is exhausted. Suppose she exhausts all her money at k_i^{th} partition. The segments in partitions 1 to k_{i-1} will be called *forced*, those in partition k_i will be called *flexible* and those in partitions k_{i+1} and higher will be called *undesirable*. Indeed, every optimal bundle is obtained in this manner: it must fully allocate all segments in the forced partitions; the money left over after this allocation is spent on segments in the flexible partition in any manner, since all these segments have equal bang per buck; and no allocation is made corresponding to segments in undesirable partitions.

3.2 Strong connectivity

In general there may not exist market equilibrium prices; in fact, for SPLC utilities, it is NP-hard to determine if they exist [55]. However, an equilibrium is guaranteed to exist under certain sufficient conditions. Let us say that *agent i is non-satiated by good j* if the last segment of f^i_j has positive slope, i.e., $u^i_{js_{ij}} > 0$.

- **Strong connectivity:** Construct a directed graph whose nodes correspond to agents of market $\mathcal{M}$ and there is an edge from i' to i only if there is a good possessed by agent i' for which agent i is non-satiated. Market $\mathcal{M}$ satisfies strong connectivity if this graph is strongly connected.

Strong connectivity is among the weakest known sufficient conditions for existence of market equilibrium, see Maxfield [39][4]. Henceforth we will assume that market $\mathcal{M}$ satisfies this condition.

4 LCP Formulation

Building on Eaves' formulation for the linear utilities case, we derive an LCP formulation for Arrow-Debreu markets with SPLC utility functions. Since the formulation turns out to be quite complex, we will do it in stages. As stated in Section 3.2, we have assumed that the given market $\mathcal{M}$ satisfies strong connectivity, hence equilibrium does exist.

[4]Earlier, Arrow and Debreu [2] had proved that the following are the sufficient conditions:

- **Non-zero initial endowments:** Every agent posses a non-zero amount of every good.
- **Non-satiation:** Every agent is non-satiated by some good.

The LCP needs to accomplish two main tasks: ensuring market clearing (i.e., that every good is fully sold and the money of each agent, obtained by selling her initial endowment, is fully spent) and ensuring that each agent obtains an optimal bundle of goods.

The first task is easy and in fact it does not even need complementarity – just non-negativity suffices. Let p_j be a variable that denotes good j's price and let q^i_{jk} be a variable that denotes the amount of money spent by agent i for buying good j corresponding to segment (j, k). All variables introduced will have a non-negativity constraint; for the sake of brevity, we will not write them explicitly. Also, for each agent i, let us introduce a variable λ_i. For now, our intention is that $1/\lambda_i$ will be the bang-per-buck of the flexible partition of i's allocation[5]. Because of the assumptions made on $\mathcal{M}$ in Section 3, λ_i will turn out to be positive.

The first task is accomplished by the following constraints; we have included the corresponding complementarity conditions in order to obtain an LCP in the standard form. (We will refer to these as follows: the equation number will refer to the constraint and the equation number with a prime will refer to the complementarity condition, e.g., (4) refers to the first constraint below and (4') refers to the corresponding complementarity condition.)

$$\forall j \in \mathcal{G}: \quad \sum_{i,k} q^i_{jk} \le p_j \quad \text{and} \quad p_j \left(\sum_{i,k} q^i_{jk} - p_j \right) = 0 \tag{4}$$

$$\forall i \in \mathcal{A}: \quad \sum_{j} w^i_j p_j \le \sum_{j,k} q^i_{jk} \quad \text{and} \quad \lambda_i \left(\sum_{j} w^i_j p_j - \sum_{j,k} q^i_{jk} \right) = 0 \tag{5}$$

Lemma 1 *The sets of constraints given in (4) and (5) hold if and only if the market clears.*

Proof : Adding the constraints in (4) over all goods and those in (5) over all agents we get

$$\sum_{i,j,k} q^i_{jk} \le \sum_j p_j \quad \text{and} \quad \sum_{i,j} w^i_j p_j \le \sum_{i,j,k} q^i_{jk},$$

respectively. Since $\sum_{i,j} w^i_j p_j = \sum_j p_j$, both these inequalities are equalities. Finally, by non-negativity, all the constraints in (4) and (5) must hold with equality, hence proving the lemma. □

Ensuring optimal bundles is somewhat more involved and requires the full power of complementarity. Consider a segment (i, j, k) in i's flexible partition. By the remarks made above, we want

$$\frac{1}{\lambda_i} = \text{bpb}^i_{jk} = \frac{u^i_{jk}}{p_j}.$$

[5] Eventually, in Lemma 3 we will show that $1/\lambda_i$ is a lower bound on the bang-per-buck of the flexible partition of i's allocation.

Let (i, j', k') be a segment in one of i's forced partitions. Clearly, $\text{bpb}^i_{j'k'} > \text{bpb}^i_{jk}$. To compensate for this, for each segment (i, j, k) in i's utility functions let us introduce variable γ^i_{jk} which can be viewed as a supplementary price associated with this segment.

Consider the following constraints and complementarity conditions:

$$\forall (i, j, k): \quad u^i_{jk}\lambda_i \leq p_j + \gamma^i_{jk} \quad \text{and} \quad q^i_{jk}(u^i_{jk}\lambda_i - p_j - \gamma^i_{jk}) = 0 \tag{6}$$

$$\forall (i, j, k): \quad q^i_{jk} \leq l^i_{jk} p_j \quad \text{and} \quad \gamma^i_{jk}(q^i_{jk} - l^i_{jk} p_j) = 0 \tag{7}$$

Let us denote the LCP defined by the sets of constraints and complementarity conditions given in (4), (5), (6) and (7), together with non-negativity on all variables, as **LCP (1)**.

Lemma 2 *Any equilibrium of market $\mathcal{M}$ yields a solution to LCP (1).*

Proof : Consider an equilibrium of $\mathcal{M}$. Substitute for the variables p_j, q^i_{jk}, λ_i in the manner described above. Since the market clears, by Lemma 1, (4) and (5) are satisfied. Substitute for the variables γ^i_{jk} as follows: if segment (i, j, k) is flexible or undesirable, set it to zero, and if it is forced, set it so that the following equality is satisfied

$$\frac{1}{\lambda_i} = \frac{u^i_{jk}}{p_j + \gamma^i_{jk}}.$$

Clearly, all the γ^i_{jk}s satisfy non-negativity. Now, it is easy to verify that in each of the three cases – that the segment (i, j, k) is forced, flexible or undesirable – the constraints (6) and (7), and complementarity conditions (6') and (7') are all satisfied. □

LCP (1) suffers from two shortcomings. First, since the rhs vector of the constraints, denoted by $\boldsymbol{q}$ in Section 2, is zero, the polyhedron is highly degenerate – in fact, it is a cone with its vertex at the origin. Another eventuality that results from $\boldsymbol{q}$ being zero is that Lemke's algorithm is simply not applicable, as mentioned in the Remark in Section 2 – for that $\boldsymbol{q}$ needs to have some negative entries. Second, LCP (1) admits solutions that don't correspond to equilibria, e.g., pick an arbitrary subset $\mathcal{G}' \subset \mathcal{G}$ and find an equilibrium for the market consisting of agents $\mathcal{A}$ and goods $\mathcal{G}'$. Set the corresponding variables in accordance with this equilibrium. For each good $j \in (\mathcal{G} \setminus \mathcal{G}')$, set $p_j = 0$ and for each segment (i, j, k) of this good, set $q^i_{jk} = 0$ and $\gamma^i_{jk} = u^i_{jk}\lambda_i$. One can verify that this is a solution to LCP (1).

Both these shortcomings can be circumvented as follows. For a good j, define desire(j) to be the total amount represented by its non-zero utility segments, *i.e.*, $\text{desire}(j) = \sum_{(i,k): u^i_{jk} > 0} l^i_{jk}$. Assuming strong connectivity, we observe the following:

- If for good j desire$(j) \leq 1$, then there is an equilibrium of market $\mathcal{M}$ in which $p_j = 0$.

- If desire(j) > 1 then $p_j > 0$ in every equilibrium of market $\mathcal{M}$.

If for some good j, desire(j) ≤ 1, then we may safely set $p_j = 0$ and solve the rest of the market. Therefore we further assume that desire of every good is more than one and hence market $\mathcal{M}$ admits an equilibrium with each good having a positive price. Furthermore, since in the Arrow-Debreu model, a non-zero scaling of an equilibrium price vector yields an equilibrium price vector, we can impose the condition $\forall j \in \mathcal{G} :\ p_j \geq 1$, or equivalently $\forall j \in \mathcal{G} :\ p_j = p'_j + 1$ and $p'_j \geq 0$, where p'_j is a new variable.

The new LCP, **LCP (2)** is given below; non-negativity is imposed on each of the variables occurring in it. Observe that on substituting $p'_j = p_j - 1$, the only change from the previous LCP is in (8'). As in LCP (1), this complementarity condition is not needed for establishing market clearing. However, it does play important role in proving Corollary 7; see also the Remark in Section 4.2.

$$\forall j \in \mathcal{G} : \quad \sum_{i,k} q^i_{jk} - p'_j \leq 1 \quad \text{and} \quad p'_j \left(\sum_{i,k} q^i_{jk} - p'_j - 1 \right) = 0 \tag{8}$$

$$\forall i \in \mathcal{A} : \quad \sum_j w^i_j p'_j - \sum_{j,k} q^i_{jk} \leq - \sum_j w^i_j \quad \text{and} \quad \lambda_i \left(\sum_j w^i_j (p'_j + 1) - \sum_{j,k} q^i_{jk} \right) = 0 \tag{9}$$

$$\forall (i,j,k) : \quad u^i_{jk}\lambda_i - p'_j - \gamma^i_{jk} \leq 1 \quad \text{and} \quad q^i_{jk}(u^i_{jk}\lambda_i - p'_j - 1 - \gamma^i_{jk}) = 0 \tag{10}$$

$$\forall (i,j,k) : \quad q^i_{jk} - l^i_{jk}p'_j \leq l^i_{jk} \quad \text{and} \quad \gamma^i_{jk}(q^i_{jk} - l^i_{jk}p'_j - l^i_{jk}) = 0 \tag{11}$$

It is easy to check that Lemmas 1 and 2 still hold. Additionally, we prove:

Lemma 3 *In any solution to LCP (2), each agent receives an optimal bundle of goods w.r.t. the prices of goods given by this solution.*

Proof : Consider an agent i. First observe that $\lambda_i > 0$, for otherwise (10) will be satisfied as a strict inequality hence forcing, via (10'), $q^i_{jk} = 0$ for each segment of i and hence contradicting market clearing.

Among all segments of i on which a positive allocation has been made, consider one having the lowest bang-per-buck, say it is (i, j, k). Let Q be the partition it belongs to and let its bang-per-buck be

$$\frac{u^i_{jk}}{p_j} = \frac{1}{\sigma_i}.$$

Now, by the constraint (10) for this segment, and the non-negativity of γ^i_{jk}, we get that $\lambda_i \geq \sigma_i$.

Define Q to be the flexible partition, all partitions having bang-per-buck strictly higher than $1/\sigma_i$ to be forced partitions, and all partitions having bang-per-buck strictly lower than $1/\sigma_i$

to be undesirable partitions. Next, we will prove that the names given are in accordance with those in Section 3.1.

Consider an arbitrary segment of i, say (i, j, k). If it is in a forced partition, it must have $\gamma^i_{jk} > 0$ in order to satisfy (10). As a result, in order to satisfy (11'), the inequality (11) must be satisfied with equality, i.e., this segment is fully allocated. And if (i, j, k) is in an undesirable partition, it must satisfy (10) as a strict inequality. Hence, $q^i_{jk} = 0$ by (10'), i.e., it is totally unallocated.

Finally, if $(i, j, k) \in Q$, there are two cases. If $\lambda_i > \sigma_i$, then in order to satisfy (10), $\gamma^i_{jk} > 0$. Again, in order to satisfy (11'), the inequality (11) must be satisfied with equality, i.e., all segments in partition Q must be fully allocated. And if $\lambda_i = \sigma_i$, $\gamma^i_{jk} = 0$ in order to satisfy (10). As a result, the only constraints on q^i_{jk} are that $0 \leq q^i_{jk} \leq l^i_{jk} p_j$, i.e., the allocation on this segment is flexible. In order to satisfy market clearing, in both cases, the total money spent on segments in Q must exhaust all the money of i that is remaining after all forced partitions are allocated.

In both cases we get that $1/\lambda_i$ is a lower bound on the bang-per-buck of the flexible partition, i.e., $1/\sigma_i$, as was promised. Also, by the characterization given in Section 3.1, i receives an optimal bundle of goods. □

4.1 Our non-degeneracy assumption

Recall that in Section 2 while outlining Lemke's algorithm, we had assumed that the polyhedron corresponding to the LCP was non-degenerate. Now, it turns out that the polyhedron corresponding to LCP (2) has an inherent degeneracy, so we need to clarify the non-degeneracy assumption we are making. The degeneracy comes about because of the following fact established in the proof of Lemma 1: adding the constraints in (4) over all goods and those in (5) over all agents yields two identical equations. Henceforth, we will say that the polyhedron corresponding to LCP (2) is non-degenerate if it has no other degeneracies.

Lemmas 1, 2 and 3 give us:

Theorem 4 *The set of solutions of LCP (2) capture exactly the set of equilibria of market $\mathcal{M}$, up to scaling. Furthermore, if polyhedron $\mathcal{P}$ corresponding to LCP (2) is non-degenerate, then the solutions of LCP (2) will be in one-to-one correspondence with the equilibria of $\mathcal{M}$, up to scaling.*

Theorem 4 settles the appropriate subcase of the open problem posed by Eaves (1975) [15], of formulating an LCP to capture equilibria of markets with piecewise-linear, concave utility functions.

Remark: Observe that in the case that all segments of the flexible partition, Q, are fully allocated, the setting of λ_i for agent i is not unique – it can be set to any value in the range $[\sigma_i, \delta_i)$, where $1/\delta_i$ is the bang-per-buck of the first undesirable partition. It is for this reason that for an arbitrary market $\mathcal{M}$, the statement of Theorem 4 cannot be strengthened to claim a one-to-one correspondence between the solutions of LCP (2) and the equilibria of market $\mathcal{M}$, up to scaling. However, in the non-degenerate case, the segments of partition Q will not be fully allocated and the one-to-one correspondence will hold.

4.2 The augmented LCP

LCP (2) has the same form as the formulation given in (1) in Section 2. Next, we obtain its augmented LCP, i.e., in the form of the formulation given in (3); however, for simplicity we will not add slack variables at this stage.

Observe that the rhs vector, i.e., $\boldsymbol{q}$, of LCP (2) does have negative entries and therefore, as stated in the Remark in Section 2, Lemke's algorithm is applicable. Further, as stated in that remark, the z variable needs to be added only in the constraints and complementarity conditions that have a negative rhs; in our case these are precisely (9) (with slack variables added). Hence, we make two changes to LCP (2) to obtain the augmented LCP, which we call **LCP (3)**. First, we change (9) as follows:

$$\forall i \in \mathcal{A}: \quad \sum_j w^i_j p'_j - \sum_{j,k} q^i_{jk} - z \leq -\sum_j w^i_j \quad \text{and} \quad \lambda_i \left(\sum_j w^i_j (1 + p'_j) - \sum_{j,k} q^i_{jk} - z \right) = 0 \quad (12)$$

Second, we impose non-negativity on z. Our algorithm will traverse a path on the 1-skeleton of the polyhedron defined by the constraints of LCP (3), as described in Section 2.

Remark: Clearly, the set of solutions of LCP (3) in which $z = 0$ enjoy all properties established in Theorem 4 and its lemmas. However, if $z > 0$, even market clearing, which was shown in Lemma 1, does not hold. Despite the non-applicability of Lemma 1, observe that by (8'), if $p'_j > 0$, then good j must be fully sold. This fact will be used at multiple places, e.g., for proving Corollary 7.

5 Proving Non-existence of Secondary Rays

Recall from Section 2 that the set of solutions of LCP (3), called S, consists of paths and cycles. A crucial fact needed for the correctness of our algorithm is that the polyhedron $\mathcal{P}'$, defined by the constraints of LCP (3), has no secondary rays and hence the path starting with the primary ray must lead to a solution with $z = 0$, i.e., an equilibrium for market $\mathcal{M}$. Our proof will critically use the fact that $\mathcal{M}$ satisfies strong connectivity. Recall that a ray is an unbounded edge of S such that the vertex of $\mathcal{P}'$ it is incident on satisfies $z > 0$.

Consider an arbitrary ray that is incident on the vertex $(\boldsymbol{y_*}, z_*)$, with $z_* > 0$, and has the direction vector $(\boldsymbol{y_\circ}, z_\circ)$. The set of points on the ray is:

$$R = \{(\boldsymbol{y_*}, z_*) + \alpha(\boldsymbol{y_\circ}, z_\circ) \mid \forall \alpha \geq 0\}.$$

Clearly, every one of these points is a solution of LCP (3). Now, this fact imposes such heavy constraints on $\boldsymbol{y}_*, \boldsymbol{y}_\circ, z_*$ and $z_\circ$ that only one possibility results, namely, $\boldsymbol{y}_* = \boldsymbol{y}_\circ = 0$, $z_* > 0$ and $z_\circ = 0$, i.e., this ray is the primary ray. This is precisely the reason that $\mathcal{P}'$ has no secondary rays.

We prove this fact below. The proof is long since we need to show that each of the other possibilities leads to a contradiction. All but one of the contradictions[6] uses the following simple fact: $(\boldsymbol{y}_*, z_*) + \alpha(\boldsymbol{y}_\circ, z_\circ)$ needs to be a solution of LCP (3) for unbounded values of α. Let us start by showing that $\boldsymbol{y}_\circ \geq 0$ and $z_\circ \geq 0$. If not, for sufficiently large α we will get a point that has a negative coordinate, contradicting a non-negativity constraint in LCP (3).

The vector $\boldsymbol{y}$ consists of four types of variables, i.e., $\boldsymbol{y} = (\boldsymbol{\lambda}, \boldsymbol{p}', \boldsymbol{q}, \boldsymbol{\gamma})$. Let $\boldsymbol{p}'_\circ$ denote the price variables in the direction vector $\boldsymbol{y}_\circ$. At the top level, we will consider the three cases: (*i*) $\boldsymbol{p}'_\circ > 0$, (*ii*) $\boldsymbol{p}'_\circ \not> 0$, $\boldsymbol{p}'_\circ \neq 0$, and (*iii*) $\boldsymbol{p}'_\circ = 0$. The first two cases lead to contradictions, the second through strong connectivity. Finally, in the third case we show that only one alternative can hold: that R is the primary ray.

W.r.t. a solution T to LCP (3), define the *surplus of agent* i to be the difference of her earnings and the amount of money she spends, i.e., $\sum_j w^i_j p_j - \sum_{j,k} q^i_{jk}$.

Claim 5 *W.r.t. a solution T to LCP (3):*

- *if $\lambda_i = 0$ then the surplus of i equals her earnings.*
- *if $\lambda_i > 0$ then the surplus of i equals z.*

Proof : If $\lambda_i = 0$ then for each segment (i, j, k) of i, (10) is satisfied with strict inequality. Hence, by (10'), $q^i_{jk} = 0$. Hence i does not spend any money and her surplus equals her earnings.

If $\lambda_i > 0$ then by (12'), $z = \sum_j w^i_j (1 + p'_j) - \sum_{j,k} q^i_{jk}$. Hence, by (12), z represents her unspent money. Since $z \geq 0$, i's surplus is non-negative. □

Lemma 6 *If in a solution, T, to LCP (3), each good is fully sold then $z = 0$.*

Proof : Since each good j is fully sold, $p_j = \sum_{i,k} q^i_{jk}$. Adding over all goods we get $\sum_{i,j,k} q^i_{jk} = \sum_j p_j$. The l.h.s. of this equation is the total money spent by all agents. Since there is one unit of each good, $\sum_{i,j} w^i_j p_j = \sum_j p_j$. The l.h.s. of this equation is the total money earned by all agents. The two equations imply that the total surplus of all agents is zero. Since $z \geq 0$, by Claim 5, each agent has non-negative surplus. Therefore, each agent must have zero surplus. Hence by Claim 5, $z = 0$. □

Corollary 7 $\boldsymbol{p}'_\circ \not> 0$.

[6] The exception is the contradiction in Corollary 7.

Proof : Suppose $\boldsymbol{p}'_\circ > 0$. Then, at every point of R with $\alpha > 0$, $\boldsymbol{p}' > 0$ and therefore by (8') every good is fully sold. Hence, by Lemma 6, $z = 0$, Now, we have already established that $\boldsymbol{p}'_* \geq 0$ and $z_\circ \geq 0$, and by definition of a ray, $z_* > 0$. Therefore, at every point of R with $\alpha > 0$, $z > 0$ leading to a contradiction. □

Lemma 8 *It cannot be the case that $\boldsymbol{p}'_\circ \not> 0$ and $\boldsymbol{p}'_\circ \neq 0$.*

Proof : Assume that $\boldsymbol{p}'_\circ \not> 0$ and $\boldsymbol{p}'_\circ \neq 0$. Let $S \subset G$ be the set of goods for which the vector $\boldsymbol{p}'_\circ$ is zero and $\overline{S}$ be the remaining goods; by assumption, both these sets are non-empty. Let $A_1 \subseteq A$ be the set of agents who are non-satiated by at least one good in S. Clearly, the prices of goods in S remain constant throughout R and those of goods in $\overline{S}$ go to infinity. Hence eventually, the bang-per-buck of all segments corresponding to goods from S will dominate that of goods from $\overline{S}$.

By (8), each good in $\overline{S}$ is fully sold. Now, since only goods in S can remain unsold, the total surplus of all agents is bounded. Since $z \geq 0$, by Claim 5 each agent has a non-negative surplus and hence the surplus of each agent is bounded. Now, consider an agent i who has a good from $\overline{S}$ in her initial endowment. Since her earnings go to infinity and her surplus is bounded, she must eventually buy up all segments corresponding to goods in S for which she has positive utility. We will use this observation to derive a contradiction by considering the following three cases.

Case 1, $A_1 = A$: By the observation made above, any agent having a non-zero amount of a good from $\overline{S}$ must eventually demand more than one unit of some good in S, contradicting (8).

Case 2, $\emptyset \subset A_1 \subset A$: By strong connectivity there must be $i_1 \in A_1$ and $i_2 \in (A \setminus A_1)$ such that i_1 has a good for which i_2 is non-satiated. Since $i_2 \notin A_1$, this good must be from $\overline{S}$. Since i_1 has a good from $\overline{S}$, by the observation made above, i_1 must eventually demand more than one unit of some good in S, contradicting (8).

Case 3, $A_1 = \emptyset$: Consider an arbitrary agent i. For strong connectivity to hold, there must be some agent i_1 such that i has a good for which i_1 is non-satiated. Since $A_1 = \emptyset$, this good is from $\overline{S}$. Hence each agent has a good from $\overline{S}$ in her initial endowment. Let $j \in S$. Now, by the observation made above, all agents will eventually buy all segments of j for which they have positive utility, contradicting (8), since desire$(j) > 1$. □

Lemma 9 *If $\boldsymbol{p}'_\circ = 0$ then R is the primary ray, i.e., $\boldsymbol{y}_\circ = 0$ and $\boldsymbol{y}_* = 0$.*

Proof : If $\boldsymbol{p}'_\circ = 0$ then the price of each good remains constant on ray R. Since by (8) no good can be oversold, $\boldsymbol{q}_\circ = 0$. Furthermore, the money earned by each agent i remains unchanged throughout R. Therefore, the forced, flexible and undesirable partitions of i remain unchanged and hence, corresponding to each of her undesirable and partially allocated segments, $\gamma^i_{jk} = 0$ throughout R.

A consequence of strong connectivity is that each agent i must be non-satiated for some good, say j. Hence there must be a segment (i, j, k), with $u^i_{jk} > 0$, that is undesirable or partially allocated. Now, in order to satisfy the constraint (10), λ_i cannot increase, forcing $\boldsymbol{\lambda}_\circ = 0$. As a result, for a forced segment (i, j, k), γ^i_{jk} cannot increase – otherwise (10') will force $q^i_{jk} = 0$. Putting this together with the assertion about undesirable and partially allocated segments made above, we get that $\boldsymbol{\gamma}_\circ = 0$. Hence, $\boldsymbol{y}_\circ = 0$. Therefore $z_\circ > 0$, or else the direction vector will be the all zero vector.

Next we show that $\boldsymbol{y}_* = 0$. Throughout R, for each agent i the money spent and money earned remain unchanged; however, z increases. Therefore, $\sum_j w^i_j p'_j - \sum_{jk}(q^i_{jk}) - z < \sum_j w^i_j$ at each point of R except possibly at the vertex of polyhedron $\mathcal{P}'$. Hence λ_i has to be zero on the rest of the ray, forcing $\boldsymbol{\lambda}_* = 0$. Therefore, for each segment, (10) is satisfied as a strict inequality, which forces $\boldsymbol{q}_* = 0$ by (10'). Now, by (8'), this forces $\boldsymbol{p'_*} = 0$, and in turn (11') forces $\boldsymbol{\gamma}_* = 0$. Altogether we get $\boldsymbol{y}_* = 0$. □

Corollary 7, Lemma 8 and Lemma 9 give:

Theorem 10 *The polyhedron $\mathcal{P}'$, defined by the constraints of the augmented LCP for SPLC market $\mathcal{M}$, LCP (3), has no secondary rays.*

6 Algorithm and Results

Before presenting the algorithm, let us add slack variables to the constraints of LCP (3) – assume that the slack variable that is added to the i^{th} constraint is v_i. This gives us an LCP in the form of the formulation given in (3); call it **LCP (3')**. The algorithm appears in Table 1. Here T_0 is the solution to LCP (3') corresponding to the vertex of polyhedron $\mathcal{P}'$ at which the primary ray is incident.

Table 1: Complementary Pivot Algorithm for SPLC Utilities

```
Initialization: Let T ← T_0
While  z > 0  in the current solution T to LCP (3'),  do
  Let i be the double label in solution T, i.e., v_i = y_i = 0 at T.
  If v_i just became 0 at the current vertex, then pivot by relaxing y_i = 0.
  Else, pivot by relaxing v_i = 0.

  Let T' be the solution to LCP (3') at the newly reached vertex. T ← T'.
Endwhile
Output solution T.
```

Theorem 10 directly yields:

Theorem 11 *If an SPLC market $\mathcal{M}$ satisfies strong connectivity and the desire of each good exceeds one, then $\mathcal{M}$ admits an equilibrium and Algorithm 1 terminates with one.*

Theorem 11 settles the appropriate case of the open problem, posed by Eaves (1975) [15], as described in the Introduction. The algorithm also gives a constructive proof of the existence of equilibrium for SPLC markets that satisfy strong connectivity.

6.1 The class PPAD

As stated in the Introduction, the class PPAD [43] attempts to capture the complexity of problems that are total, i.e., always have solutions, and moreover admit "path-following" algorithms for finding one. The latter notion, and the class are defined by the following canonical problem.

END OF A LINE: Let G be a directed graph defined by two polynomial time Turing machines P (predecessor) and S (successor). The indegree and outdegree of every node in G is at most one, and there are no isolated nodes. Clearly, G consists of a set of paths and cycles. Turing machines P and S take a node as input and output its predecessor and successor, respectively. Given a node with no predecessor and the machines P and S, the problem is to find another node with either no predecessor or no successor.

Every problem in PPAD is polynomial time reducible to END OF A LINE, e.g., the Lemke-Howson algorithm and Todd's result show membership in PPAD for 2-Nash. The graph G in this case is formed by the set of nodes and edges where at most one fixed "label" is missing.

6.2 Further Results

Theorem 12 *Assuming strong connectivity, the problem of computing an equilibrium of a market with SPLC utilities is in PPAD.*

Proof : By Theorem 10, Algorithm 1 must converge to an equilibrium. Now, by Todd's result [54] on the orientability of the path followed by a complementary pivot algorithm, we get a proof of membership of the problem in PPAD. □

This settles the question posed by Vazirani and Yannakakis in [55] of obtaining a direct proof of the membership of the problem in PPAD.

Observe that the polyhedron corresponding to LCP (3) has the same inherent degeneracy as that of LCP (2), and explained in Section 4.1. The reason is that at any solution to LCP (3) at which $z = 0$, the market clearing conditions are satisfied and the dependence in the constraints established in Lemma 1 holds. Once again, we will say that the polyhedron corresponding to LCP (3) is non-degenerate if it has no other degeneracies. Let v be a vertex solution to LCP (3) with $z = 0$. Then it is easy to show that there is exactly one $j \in \mathcal{G}$ with $p'_j = 0$ at v. Relaxing $p'_j = 0$ gives an unbounded edge, starting at v, at which z remains zero. Therefore, every point of this edge corresponds to a market equilibrium in which the prices at v are appropriately scaled.

Theorem 13 *If the polyhedron $\mathcal{P}'$ corresponding to LCP (3) of an SPLC market is non-degenerate, then $\mathcal{M}$ has an odd number of equilibria, up to scaling.*

Proof : As observed in Section 2, the set of solutions, S, to LCP (3) consists of paths and cycles. The solutions of LCP (3) satisfying $z = 0$ are precisely the solutions to LCP (2). By Theorem 4, the latter are in one-to-one correspondence with the equilibria of market $\mathcal{M}$, up to scaling. Now, solutions of LCP (3) satisfying $z = 0$ occur at endpoints of such paths (under non-degeneracy). One of the paths starts with the primary ray and ends with an equilibrium. Since by Theorem 10 $\mathcal{P}'$ has no secondary rays, the rest of the equilibria must be paired up. Hence there are an odd number of equilibria. $\square$

7 Strongly Polynomial Bound

Devanur and Kannan [10] gave a polynomial time algorithm for SPLC markets when either the number of goods or the number of agents is a constant, using the "cell decomposition" technique and the fact that the number of nonempty regions (cells) formed by n hyperplanes in $\mathbf{R}^d$ is at most $O(n^d)$. We use similar techniques to show a strongly polynomial bound on the number of fully labeled vertices in the polytope $\mathcal{P}'$ of LCP (3), when number of goods or agents is constant. This in turn gives a strongly polynomial bound for our algorithm for this case.

Suppose the number of goods, i.e., n, is a constant. The idea is to decompose the $(p_1, \ldots, p_n, z)$-space (i.e., $\mathbf{R}_+^{n+1}$) into cells by a set of polynomially many hyperplanes such that every cell corresponds to a unique setting of forced, flexible and undesirable partitions. Then we show that every fully labeled vertex of $\mathcal{P}'$ can be naturally mapped (by projection) to a cell and this mapping maps at most two vertices to any given cell. We describe below how to get the cell decomposition.

7.1 Constant number of goods

Consider $\mathbf{R}_+^{n+1}$ with coordinates $p_1, \cdots, p_n, z$. For each 5-tuple (i, j, j', k, k'), where $i \in \mathcal{A}$, $j \neq j' \in \mathcal{G}$, $k \leq s_{ij}$ and $k' \leq s_{ij'}$, introduce hyperplane $u^i_{jk} p_{j'} - u^i_{j'k'} p_j = 0$. These hyperplanes divide the space into cells and each cell has one of the signs $<, =, >$ for each hyperplane. For each agent, these signs give partial order on the bang-per-buck of her segments. Using this information for a given cell, we can sort all segments (j, k) of agent i by decreasing bang-per-buck, and partition them by equality into classes: $Q^i_1, Q^i_2, \ldots$. Let $Q^i_{<l}$ denote $Q^i_1 \cup Q^i_2 \cup \ldots \cup Q^i_{l-1}$. Similarly, we define $Q^i_{\leq l}$ and $Q^i_{>l}$.

Next we want to capture the flexible partition. To do this, we further subdivide a cell by adding hyperplane $\sum_{(j,k) \in Q^i_{<l}} l^i_{jk} p_j = \sum_{j \in \mathcal{G}} w^i_j p_j - z$, for each agent i and each of her partitions Q^i_l. For any given subcell, let $Q^i_{l_i}$ be the right most partition such that $\sum_{(j,k) \in Q^i_{<l_i}} l^i_{jk} p_j <$

$\sum_{j\in\mathcal{G}} w^i_j p_j - z$, then $Q^i_{l_i}$ is the flexible partition for agent i. In addition, we add hyperplanes $p_j = 1,\ \forall j \in \mathcal{G}$ and $z = 0$, and consider only those cells where $p_j \geq 1$ and $z \geq 0$.

Given a vertex $(\boldsymbol{y}, z)$ on the path traced by our algorithm, there is a natural cell associated with it, namely due to projection of it on $(\boldsymbol{p}, z)$-space.

Lemma 14 *Each cell is mapped onto from at most two fully labeled vertices of the polyhedron $\mathcal{P}'$ corresponding to LCP (3). Furthermore, if a cell is mapped onto from two vertices, then they must be adjacent.*

Proof : Every fully labeled vertex and every cell has its own settings of forced, flexible and undesirable partitions, for each agent. Hence, if a fully labeled vertex maps onto a cell, then these two settings, one coming from the cell and the other from the vertex, must match. A fully labeled vertex $v = (\boldsymbol{\lambda}, \boldsymbol{p'}, \boldsymbol{q}, \boldsymbol{\gamma}, z)$, which maps onto a given cell must satisfy the following equalities. In the cell,

- If $p_j > 1$ then $\sum_{ik} q^i_{j,k} - p'_j - 1 = 0$ else $p'_j = 0$ at v.
- If $\sum_j w^i_j p_j - z \geq 0$ (second set of hyperplanes for the tuple $(i, 1)$) then $\sum_j w^i_j (p'_j + 1) - \sum_{j,k} q^i_{jk} - z = 0$ else $\lambda_i = 0$ at v.
- If $u^i_{jk} p_{j'} - u^i_{j'k'} p_j \geq 0$ for a $(j', k') \in Q^i_{l_i}$ then $u^i_{jk}\lambda_i - p'_j - 1 - \gamma^i_{jk} = 0$ else $q^i_{jk} = 0$ at v.
- If $u^i_{jk} p_{j'} - u^i_{j'k'} p_j > 0$ for a $(j', k') \in Q^i_{l_i}$ then $q^i_{jk} - l^i_{jk} p'_j - l^i_{jk} = 0$ else $\gamma^i_{jk} = 0$ at v.

Since from each of complementary conditions given above one equality is enforced, their intersection forms a line. If this line does not intersect $\mathcal{P}'$, no fully labeled vertex gets mapped to the cell under the consideration. If it does then intersection can be either a fully labeled vertex, say v, or a fully labeled edge – we will say that an edge of the polyhedron $\mathcal{P}'$ is *fully labeled* if the solution represented by each point of this edge is fully labeled. In the former case only vertex v gets mapped to the cell and in the latter case only the endpoints of the fully labeled edge map to the cell – clearly these are two adjacent vertices of $\mathcal{P}'$. $\square$

Note that the total number of hyperplanes we introduced is strongly polynomial, thereby creating strongly polynomially many cells.

7.2 Constant number of agents

So far, we had considered a partitioning of the segments corresponding to each agent. For this case, we will consider a partitioning of the segments corresponding to each good, as detailed below. Besides this change, the analysis for this case is similar to that of the previous case.

Consider $\mathbf{R}^m_+$ with coordinates $\lambda_1, \cdots, \lambda_m$. Every fully labeled vertex $(\boldsymbol{\lambda}, \boldsymbol{p'}, \boldsymbol{q}, \boldsymbol{\gamma}, z)$ naturally gets mapped to this space by taking projection on $\boldsymbol{\lambda}$. Before getting into cell decomposition we discuss some properties of fully labeled vertices.

Given a fully labeled vertex, for every good j sort all its segments (i, j, k) in decreasing order of $u^i_{jk}\lambda_i$, and partition them by equality into classes: $Q^j_1, Q^j_2, \ldots$. It is easy to verify that at this vertex, good j gets allocated in the order of partitions, starting from the first. If a segment $(i, j, k) \in Q^j_l$ is allocated (i.e., $q^i_{jk} > 0$), then all the segments in partitions before Q^j_l must be completely allocated. We call the last allocated partition as flexible partition, all the partitions before it as forced partitions and all partitions after it as undesirable partitions for good j. Further, let (i, j, k) be a segment in flexible partition of good j. Then, we have $u^i_{jk}\lambda_i = 1 + p'_j$, otherwise all the segments in this partition are either undesirable or all of them are forced for the corresponding agents. Therefore, the flexible partition of any good defines its price.

Next we decompose the $\mathbf{R}^m_+$ space into cells (by introducing hyperplanes) such that every cell captures the segment configurations for each good. Introduce hyperplanes of type $u^i_{jk}\lambda_i - u^{i'}_{jk'}\lambda_{i'} = 0$ for each 5-tuple (i, i', j, k, k'), where $i \neq i' \in \mathcal{A},\ j \in \mathcal{G},\ k \leq s_{ij}$ and $k' \leq s_{i'j}$. Given a cell, the signs of these hyperplanes in the cell give partial order of segments (i, k) for every good j based on $u^i_{jk}\lambda_i$. For each good j sort its segments in decreasing value of $u^i_{jk}\lambda_i$ using this partial order, and partition them by equality into classes: $Q^j_1, Q^j_2, \ldots$.

Next we capture the flexible partition for every good. For a fully sold good, it may be computed easily by just summing up the segment lengths starting from the first partition until it becomes one. However, a fully labeled vertex may have undersold goods. Since the price of such a good is fixed to one (p' is zero), segments in its flexible partition have $u^i_{jk}\lambda_i = 1$. To capture this we introduce $u^i_{jk}\lambda_i - 1 = 0$ for each (i, j, k). In general the flexible partition for good j is the earlier one of the two: partition when good is fully sold and the partition with $u^i_{jk}\lambda_i = 1$. This can be easily deduced for a given cell from the signs of the hyperplanes. Further, we put $\lambda_i = 0$ for each $i \in \mathcal{A}$.

From the above discussion it is clear that given a cell the equalities of the fully labeled vertices mapping to it may be worked out as done in Lemma 14. Further, we get one equality for every complementary condition, since every cell captures complete segment configuration, status of goods, and agents in the market. Thus, we get the following lemma.

Lemma 15 *Each cell is mapped onto from at most two fully labeled vertices of the polyhedron $\mathcal{P}'$ corresponding to LCP (3). Furthermore, if a cell is mapped onto from two vertices, then they must be adjacent.*

It is clear that our algorithm follows a systematic path instead of brute force enumeration of all the cells. The next theorem follows directly from the above discussion, since the number of hyperplane introduced is strongly polynomial in both the cases.

Theorem 16 *For an SPLC market with a constant number of agents or goods, our algorithm computes an equilibrium in strongly polynomial time.*

8 Combinatorial Interpretation for Linear Case

In this section, we give a complete combinatorial interpretation for Eaves' algorithm for linear case. We then provide an example to illustrate that our algorithm for the SPLC case has a much more complex mechanism and we leave the open problem of obtaining its combinatorial interpretation.

Dropping index k, since each utility function has only one segment, we specialize LCP (3) to this case below; let us call it **LCP (4)** and denote its polyhedron by $\mathcal{P}^l$. Two additional simplifications are made: First, $1/\lambda_i$ will be the maximum *bang-per-buck of agent* i, which is defined to be

$$\max_{j \in \mathcal{G}} \frac{u_{ij}}{p_j}.$$

Second, we don't need the variables γ^i_{jk} and the constraints and complementarity conditions given in (11). This was precisely the LCP derived by Eaves [16].

$$\forall j \in \mathcal{G}: \quad \sum_i q^i_j - p'_j \leq 1 \quad \text{and} \quad p'_j \left(\sum_i q^i_j - p'_j - 1 \right) = 0 \tag{13}$$

$$\forall i \in \mathcal{A}: \ \sum_j w^i_j p'_j - \sum_j q^i_j - z \leq - \sum_j w^i_j \quad \text{and} \quad \lambda_i \left(\sum_j w^i_j (p'_j + 1) - \sum_j q^i_j - z \right) = 0 \tag{14}$$

$$\forall (i,j): \quad u^i_j \lambda_i - p'_j \leq 1 \quad \text{and} \quad q^i_j (u^i_j \lambda_i - p'_j - 1) = 0 \tag{15}$$

Construct a bipartite graph whose vertices are $\mathcal{A} \cup \mathcal{G}$ and (i,j) is an edge iff inequality (15) is satisfied as an equality. Call this the *tight graph* and its edges the *tight edges*. By (15'), goods can only be sold along tight edges. Hence, if $p'_j > 0$, then the total sales on edges incident at j must be one unit. Also, agents having $\lambda_i = 0$ do not have any edges incident at them and goods having $p'_j = 0$ may or may not have edges incident at them.

Next, let us analyze the changes that take place while moving along an edge e of polyhedron $\mathcal{P}^l$ during the algorithm. Since an equilibrium is not reached yet, $z > 0$ on e. The set of inequalities that are tight remain unchanged at all points of e except for the two end vertices where an extra inequality is tight. Consider the connected components of the tight graph; singleton agents are not considered as a components.

By (13'), if $p'_j > 0$, j must be fully sold. If all goods are fully sold, all of the agents' money must be spent, making $z = 0$. However, since $z > 0$, there is at least one undersold good, say j and $p'_j = 0$ at all points of edge e. Hence, in any component containing such a good, the prices of goods and the λ_is of agents must remain unchanged while moving along e.

Next consider the remaining components. Since each good has edges to all agents who buy this good, the goods of such a component are sold precisely to agents in this component. Now, if these agents do not own any goods from outside this component, then they cannot have positive surplus, contradicting $z > 0$. Hence they must own goods from outside this component. Furthermore, in such a component $1+p'_j$ and λ_is change by the same factor in order to maintain equalities of type (15) corresponding to tight edges. Since the agents in all these components have the same surplus, namely z, the prices of all goods in these components either monotonically increase or decrease while moving along e. Furthermore, they increase iff the surplus, and hence z, monotonically decreases.

Next, let us analyze the changes on the entire path π followed by the algorithm. Let e_1 and e_2 be two adjacent edges on π with a common vertex $v = (\boldsymbol{y}, z)$. While moving on π suppose the algorithm enters v through e_1 and leaves it through e_2.

Lemma 17 *If z is decreasing on e_1 while moving towards v, then it keeps decreasing on e_2 while moving away from v.*

Proof : The new tight inequality at v corresponds to its double label, and determines what to relax to move on e_2. There are six possibilities for the new tight inequality. For each of them we argue that if z is decreasing while moving along e_1 towards v, then it will monotonically decrease while moving away from v on e_2. Let the new tight inequality is at v be

1. $\sum_i q^i_j - p'_j \leq 1$: Then we relax $p'_j = 0$ to obtain e_2. Hence, prices monotonically increase and z monotonically decreases on e_2.

2. $-\sum_j q^i_j + \sum_j w^i_j p'_j - z \leq -\sum_j w^i_j$: Then we relax λ_i to move on e_2. In this case agent i is not part of any component yet, so all the prices and z remain constant.

3. $p'_j \geq 0$: This case never arises since then the prices should be decreasing on e_1. A contradiction to z decreasing on e_1.

4. $\lambda_i \geq 0$: This case never arises since then λ_is should be decreasing on e_1. This in turn implies prices are decreasing and z is increasing on e_1, a contradiction.

5. $u^k_l y_k - p'_l \leq 1$: Then we relax $q^k_l = 0$. This case is little involved.

   On e_1 let C_0 be the set of components containing undersold goods, i.e., constant prices. Let $C_1, \ldots, C_h$ be the rest of the components. Clearly, $k \notin C_0$. Suppose, $k \in C_1$ (w.l.o.g.). There are two cases. The case when $l \in C_0$ is easy. Since prices of goods in C_1 are fixed on e_2, if rest of the prices decrease then the income and inturn surplus of agents in C_1 decreases, a contradiction.

   For the other case, let $l \in C_2$ (w.l.o.g.). In component C_x suppose the $1 + p'_j$s and λ_is change by a multiplicative factor α_x. Clearly, $\alpha_1 > \alpha_2 \geq 1$. Consider all α_x relative to α_1, or in other words all the price changes relative to price changes in C_1. Now, on e_2 since C_1 and C_2 merge into one component say C_{12}, the prices of goods in C_2 changes by a faster rate. Therefore, prices of all the goods outside C_{12} also changes by a faster rate. If all of them decrease on e_2 then earning of agents in C_1 decreases by a faster rate than the spending, implying that either their surplus decreases or they need to pull money from C_2 through edge (k, l). Both leads to contradictions; negative flow on (k, l) in the latter case.

6. $q^i_j = 0$: Then we relax $u^i_j y_i - p_j = 0$. Similar analysis as the previous case works to show that z monotonically decreases in this case too.

□

We know that on the primary ray z monotonically decreases. Hence by Lemma 17, it decreases on the entire path starting with the primary ray, and at the end of the path, when equilibrium is achieved, it becomes zero. Now, if there are more equilibria, they must be at the endpoints of other paths, as shown in Theorem 13. Therefore, $z = 0$ at both the endpoints and must increase monotonically while moving away from each endpoint, leading to a contradiction. Hence we get:

Lemma 18 *If the polyhedron $\mathcal{P}^l$ of LCP (4) for a linear market is non-degenerate, then the market has a unique equilibrium up to scaling.*

Finally, we give an example to illustrate that our algorithm for the SPLC case has a much more complex mechanism. In this example, neither does z decrease monotonically nor do prices increase monotonically on the path starting with the primary ray.

Example 19 Consider a simple market with 2 agents, 3 goods and 2 segments for every pair of agent and good, where

$$W = \begin{bmatrix} 0.4 & 0.3 & 0.1 \\ 0.6 & 0.7 & 0.9 \end{bmatrix}, U^1_{\mathcal{G}} = \begin{bmatrix} 0.2 & 0.1 \\ 0.9 & 0.7 \\ 0.8 & 0.6 \end{bmatrix}, U^2_{\mathcal{G}} = \begin{bmatrix} 0.9 & 0.7 \\ 0.8 & 0.1 \\ 0.4 & 0.2 \end{bmatrix},$$

$$L^1_{\mathcal{G}} = \begin{bmatrix} 0.3 & 1 \\ 0.6 & 1 \\ 0.9 & 1 \end{bmatrix}, L^2_{\mathcal{G}} = \begin{bmatrix} 0.3 & 1 \\ 0.9 & 1 \\ 1 & 1 \end{bmatrix}.$$

The changes in values of z, p'_1, p'_2 and p'_3, during a run of the algorithm for this market, are recorded in Table 2. Neither z nor p'_js are monotonic as shown by the gray cells in the table.

□

8.1 A combinatorial algorithm

The various observations made above lead to a combinatorial algorithm that is equivalent to Eaves' complementary pivot algorithm. We assume that the given market is non-degenerate; this can be achieved by perturbation of u^i_js.

Table 2: The values of z, p_1, p_2 and p_3 as the algorithm runs on Example 19

iter.	1	2	3	4	5	6	7	8	9	10	11	12
$\boldsymbol{z}$	2.2	1.9	1.9	1	1	0.8	0.8	0.7	0.725	0.275	0.05	0
$\boldsymbol{1+p'_1}$	1	1	1	1	1	1	1	1	1	1.5625	1.375	1.5
$\boldsymbol{1+p'_2}$	1	1	1	1	1	1	1	1.125	1.125	1.125	1.5	1.5
$\boldsymbol{1+p'_3}$	1	1	1	1	1	1	1	1	1	1	1	1

Let $\mathcal{A}_0$ be the set of agents with surplus less than z, and let $\mathcal{A}_1 = \mathcal{A} \setminus \mathcal{A}_0$. Consider the bipartite graph between agents of $\mathcal{A}_1$ and all the goods consisting of tight edges. Say that edge (i, j) is *non-zero* if agent i is spending money on good j[7].

Let E be the set of all the non-zero edges. Let C_0 be the set of components of E containing undersold goods; prices of goods in C_0 remain unchanged. Let components of $E \setminus C_0$ be $C_1, \ldots, C_l$. Let α_i be the factor with which prices of goods in C_i change (p_j changes to $\alpha_i p_j$). Note that $\alpha_0 = 1$. Using market clearing conditions, every α_i can be written as a linear function of z, namely $c_i z + d_i$. Hence, as z changes α_i changes accordingly.

Note that as the prices change the money owned by the agents change. Therefore maintaining the feasible money allocation (*i.e.*, q^i_j's) is a challenging task. Define network $N(E, \boldsymbol{p}, z)$ as follows: if $(i, j) \in E$, then put a directed edge from i to j with infinity capacity, $cap(i, j) = \infty$. Add a source node s and a sink node t to the network. From s, put a directed edge to every $i \in \mathcal{A}_1$ with capacity $cap(s, i) = \sum_{k=0}^{l} \sum_{j \in C_k} w^i_j \alpha_k p_j - z$, which is equivalent to $\sum_{k=0}^{l} \sum_{j \in C_k} w^i_j p_j (c_k z + d_k) - z$, a linear equation in z. From every good $j \in \mathcal{G}$, put a directed edge to t with capacity $cap(j, t) = \alpha_k p_j$ if $g_j \in C_k$. As per the flow the following invariant is maintained.

I. The cut $(s, \mathcal{A}_1 \cup \mathcal{G} \cup t)$ is always a min cut in $N(E, \boldsymbol{p}, z)$.

The complete algorithm is given in Table 3. Next, we describe how the threshold z for the four events can be efficiently computed. For Events 1 and 2 this is straightforward. To check Event 3 in any component except C_0, do the following for every edge $(i, j) \in E$: Let S be the set of agents of component containing j after removal of edge (i, j). Calculate z using $\sum_{i \in S} cap(s, i) = \sum_{j \in \Gamma(S)} cap(j, t)$, where $\Gamma(S)$ is the set of neighbors of S in the graph formed by E. The maximum z gives the threshold for Event 3 in components other than C_0. In case of C_0 we observe that each of its component cannot have more than one undersold good. Further, if flow on (i, j) becomes zero then good j has to be fully sold. In that case after removal of (i, j) from E the component containing j has only fully sold goods, and hence the same procedure applies. Since every component of C_0 has exactly one undersold good, the z triggering Event 4 may be easily calculated: for each of its components calculate z that clears the market from the goods side, and pick the maximum z among them.

[7] If the given market is non-degenerate, then at any point during a run of the algorithm there can be at most one tight edge without any money flow.

Table 3: Combinatorial Algorithm for Linear Utilities

Initialize:
$p_j \leftarrow 1, \forall j \in \mathcal{G}$; $e_i \leftarrow \sum_j w_j^i p_j, \forall i \in \mathcal{A}$; $z \leftarrow \max_{i \in \mathcal{A}} e_i$;
$\mathcal{A}_0 \leftarrow \{i \in \mathcal{A} \mid e_i < z\}$; $\mathcal{A}_1 = \mathcal{A} \setminus \mathcal{A}_0$; $C_0 \leftarrow \mathcal{G}$;
$E \leftarrow \{(i,j) \in (\mathcal{A}_1 \times \mathcal{G}) \mid bpb_j^i = \max_{k \in \mathcal{G}} bpb_k^i\}$;
while $z \neq 0$ **do**
Decrease z, while maintaining I, until one of the following events occurs:
Event 1: For an $i \in \mathcal{A}_0$ e_i becomes z;
$\mathcal{A}_1 \leftarrow \mathcal{A}_1 \cup \{i\}$; $\mathcal{A}_0 \leftarrow \mathcal{A}_0 \setminus \{i\}$; $E \leftarrow E \cup$ maximum bpb edges of i;
Event 2: A new edge $(i,j) \in (\mathcal{A}_1 \times \mathcal{G})$ becomes tight;
$E \leftarrow E \cup \{(i,j)\}$;
Event 3: Flow on a non-zero edge (i,j) becomes zero;
$E \leftarrow E \setminus \{(i,j)\}$;
Event 4: An undersold good $j \in C_0$ becomes fully sold;
$C_0 \leftarrow C_0 \setminus \{j\}$; Recompute C_0;
endwhile

9 Experimental Results

Table 4 summarizes the results of experiments, done over randomly generated instances, with a Matlab implementation of our algorithm. For each choice of number of agents and goods, the total number of segments in all the utility functions was kept the same and is denoted by #Seg in the table. The values of u_{jk}^is, l_{jk}^is and w_j^is were drawn uniformly at random from the intervals $[0,1]$, $[0, \frac{1}{\#\text{Seg}}]$ and $[0,1]$, respectively. The w_j^i values were scaled so that the total amount of each good is unit. Finally, for each agent i and good j, corresponding u_{jk}^is were sorted in decreasing order to get the SPLC utility function f_j^i.

Table 4: Experimental Results over Random Instances

$\lvert\mathcal{A}\rvert$ x $\lvert\mathcal{G}\rvert$ x #Seg	#Instances	Min Iters	Avg Iters	Max Iters
5 x 5 x 5	1000	107	142.7	199
10 x 5 x 5	1000	130	154.3	197
10 x 10 x 5	1000	254	321.9	401
10 x 10 x 10	50	473	515.8	569
15 x 15 x 5	100	413	509.7	582
15 x 15 x 10	50	775	991	1090
15 x 15 x 15	10	1197	1261.3	1382
20 x 20 x 5	10	719	764	853
20 x 20 x 10	10	1093	1208.8	1473

Note that, even in the worst case the number of iterations is of the order of the total number of segments of the utility functions, *i.e.*, $\sum_{i,j} s_{ij} = |\mathcal{A}| \times |\mathcal{G}| \times \#\text{Seg}$. Figure 1 plots $\log_2(\sum_{i,j} s_{ij})$

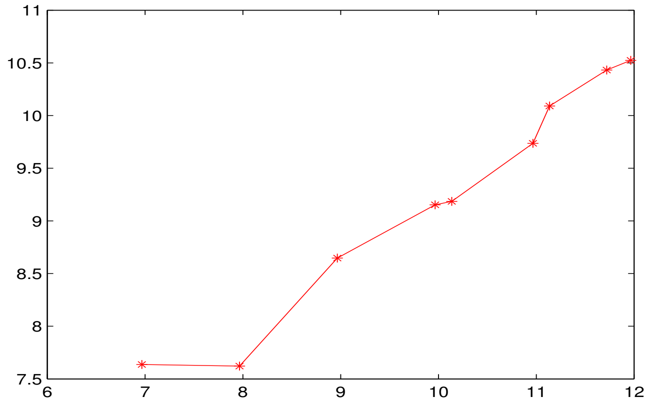

Figure 1: Plot of $\log_2(\sum_{i,j} s_{ij})$ vs. $\log_2$(Max Iters)

vs. $\log_2$(Max Iters) for a comparative analysis.

10 Discussion

Besides parity [50], the Lemke-Howson algorithm has yielded numerous insights into structural properties of 2-Nash equilibria, such as index, degree and stability [25, 48, 50]. It has been the subject of much other work, e.g., [6] determine its smoothed complexity [52] and [44] give an example on which it takes exponential time. All these issues are worth exploring for SPLC equilibria and our algorithm as well. We note that the notion of index has already been studied for regular markets [32, 38]; however, it uses the Hessian of the demand functions and hence is not applicable to our case.

The question of whether any of the tracing procedures for Nash equilibrium computation [19, 26, 27, 28, 35, 36, 40] can locate all the Nash equilibria of a game has been studied extensively [3, 29, 41]. It would be interesting to determine if our algorithm can locate all the equilibria of a given SPLC market by trying out all possible coefficients vectors for the z variable.

A natural question that arises from our result is whether one can obtain a practical algorithm for the case of non-separable, piecewise-linear concave utilities via the following scheme: Obtain a rational approximation of an equilibrium that is amenable to an LCP formulation and a complementary pivot algorithm. Another open question, first raised in [55], is to prove that this case is FIXP-complete; it is known to have algebraic equilibria.

An obvious approach to answering the question of [55] was to build on the work of [11], i.e., obtain a flow-based algorithm that iteratively adjusts prices, responding to certain min-cuts in a network analogous to the one used in [11]. To show termination for such an algorithm, one would need a potential function that changes monotonically, achieving its optimal value when the algorithm finds an equilibrium. However, it turns out that as prices of goods change, the value of forced allocations changes in such a way that it seems impossible to construct a suitable potential function. Another approach was to generalize the work of [23], who gave a Lemke-Howson-type algorithm for the linear case, to SPLC utilities. However, the same obstacle, i.e., changing value of forced allocations, thwarts this attempt as well.

At present, we do not understand how our algorithm finesses the obstacle mentioned above. However, we believe that obtaining a combinatorial understanding of our algorithm will clarify this. Additionally, it is likely that even in the linear case, in practice, Eaves' and our algorithms are competitive over provably polynomial time algorithms – experiments are needed to confirm or refute this.

The decade-long endeavor, within TCS, of understanding the computability of market equilibria has successfully addressed almost all broad, general classes of markets – the main exception being markets with production. Following up on Eaves [15], we restate the question of studying such markets.

References

[1] Adler, I.: (2014), Personal Communication.

[2] Arrow, K., Debreu, G.: Existence of an equilibrium for a competitive economy. Econometrica 22, 265–290 (1954)

[3] Balthasar, A.: Equilibrium tracing in strategic-form games. Economic Theory 42(1), 39–54 (2010)

[4] Brown, D.J., Demarzo, P.M., Eaves, B.C.: Computing equilibria when asset markets are incomplete. Econometrica 64(1), 1–27 (1996)

[5] Chen, X., Dai, D., Du, Y., Teng, S.H.: Settling the complexity of Arrow-Debreu equilibria in markets with additively separable utilities. In: FOCS (2009)

[6] Chen, X., Deng, X., Teng, S.H.: Settling the complexity of computing two-player Nash equilibria. Journal of the ACM 56(3) (2009)

[7] Cottle, R., Pang, J., Stone, R.: The Linear Complementarity Problem. Academic Press, Boston (1992)

[8] Dantzig, G.B.: Linear Programming and Extensions. Princeton University Press (1963)

[9] Debreu, G.: Economies with a finite set of equilibria. Econometrica 38, 387–393 (1980)

[10] Devanur, N., Kannan, R.: Market equilibria in polynomial time for fixed number of goods or agents. In: FOCS. pp. 45–53 (2008)

[11] Devanur, N., Papadimitriou, C.H., Saberi, A., Vazirani, V.V.: Market equilibrium via a primal-dual algorithm for a convex program. JACM 55(5) (2008)

[12] Devanur, N., Vazirani, V.: An improved approximation scheme for computing Arrow-Debreu prices for the linear case. In: Proceedings of 23rd FSTTCS (2003)

[13] Dierker, E.: Two remarks on the number of equilibria of an economy. Econometrica 40(5), 951–953 (1972)

[14] Duan, R., Mehlhorn, K.: A combinatorial polynomial algorithm for the linear Arrow-Debreu market. In: ICALP (2013)

[15] Eaves, B.C.: A finite algorithm for the linear exchange model. Cowles Foundation Discussion Paper No. 389 (1975)

[16] Eaves, B.C.: A finite algorithm for the linear exchange model. Journal of Mathematical Economics 3, 197–203 (1976)

[17] Eaves, B.C., Schmedders, K.: General equilibrium models and homotopy methods. Journal of Economic Dynamics & Control 23, 1249–1279 (1999)

[18] Ellickson, B.: Competitive Equilibrium: Theory and Applications. Cambridge University Press (1994)

[19] van den Elzen, A.H., Talman, A.J.J.: A procedure for finding Nash equilibria in bimatrix games. ZOR-Methods and Models of Operations Research 35, 27–43 (1991)

[20] Esteban-Bravo, M.: An interior-point algorithm for computing equilibria in economies with incomplete asset markets. Journal of Economic Dynamics & Control 32, 677–694 (2008)

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

[22] Garcia, C., Zangwill, W.: Pathways to Solutions, Fixed Points, and Equilibria. Prentice-Hall, Englewood Cliffs, NJ (1981)

[23] Garg, J., Mehta, R., Sohoni, M., Vishnoi, N.K.: Towards polynomial simplex-like algorithms for market equilibria. In: SODA (2013)

[24] Garg, R., Kapoor, S.: Auction algorithms for market equilibrium. In: ACM Symposium on the Theory of Computing (2004)

[25] Govindan, S., Wilson, R.: Equivalence and invariance of the index and degree of Nash equilibria. Games and Economic Behavior 21(1), 56–61 (1997)

[26] Govindan, S., Wilson, R.: A global Newton method to compute Nash equilibria. Journal of Economic Theory 110(1), 65–86 (2003)

[27] Herings, P.J.J., van den Elzen, A.H.: Computation of the Nash equilibrium selected by the tracing procedure in n-person games. Games Econ Behav 38, 89–117 (2002)

[28] Herings, P.J.J., Peeters, R.: A differentiable homotopy to compute Nash equilibria of n-person games. Econ Theory 18, 159–185 (2001)

[29] Hofbauer, J.: Some thoughts on sustainable/learnable equilibria. (2003), Available at: http://www.econ.uniurb.it/imgta/PlenaryLecture/Hofbauer.pdf

[30] Jain, K.: A polynomial time algorithm for computing the Arrow-Debreu market equilibrium for linear utilities. SIAM Journal on Computing 37(1), 306–318 (2007)

[31] Jain, K., Mahdian, M., Saberi, A.: Approximating market equilibrium. In: Proceedings of 6th APPROX (2003)

[32] Kehoe, T.J.: An index theorem for general equilibrium models with production. Econometrica 48(5), 1211–1232 (1980)

[33] Klee, V., Minty, G.J.: How good is the simplex algorithm? In: Inequalities, III ed. by O. Shisha. pp. 159–175. New York: Academic Press (1972)

[34] Koller, D., Megiddo, N., Stengel, B.V.: Efficient computation of equilibria for extensive two-person games. Games and Economic Behavior 14(2), 247–259 (1996)

[35] Lemke, C.E.: Bimatrix equilibrium points and mathematical programming. Management Science 11(7), 681–689 (1965)

[36] Lemke, C.E., J. T. Howson, J.: Equilibrium points of bimatrix games. SIAM J. on Applied Mathematics 12(2), 413–423 (1964)

[37] Mansur, A., Whalley, J.: A decomposition algorithm for general equilibrium computation with application to international trade models. Econometrica 50(6), 1547–1557 (1982)

[38] Mas-Colell, A.: The Theory of General Economic Equilibrium: A Differentiable Approach. Econometric Society Monograph, Cambridge University Press (1985)

[39] Maxfield, R.R.: General equilibrium and the theory of directed graphs. Journal of Mathematical Economics 27(1), 23–51 (1997)

[40] McKelvey, R.D., Palfrey, T.R.: Quantal response equilibria for normal form games. Games Econ Behav 10, 6–38 (1995)

[41] Myerson, R.B.: Sustainable equilibria in culturally familiar games. In: Understanding Strategic Interaction: Essays in Honor of Reinhard Selten, eds. W. Albers et al., Springer, Heidelberg. pp. 111–121 (1997)

[42] Orlin, J.B.: Improved algorithms for computing Fisher's market clearing prices. In: ACM Symposium on the Theory of Computing. pp. 291–300 (2010)

[43] Papadimitriou, C.H.: On the complexity of the parity argument and other inefficient proofs of existence. JCSS 48(3), 498–532 (1992)

[44] Savani, R., von Stengel, B.: Hard-to-solve bimatrix games. Econometrica 74(2), 397–429 (2006)

[45] Scarf, H.: An elementary proof of a theorem on the core of an n person game (1965), available at: http://cowles.econ.yale.edu/˜hes/pub/pp650426.pdf

[46] Scarf, H.: The Computation of Economic Equilibria. Yale University Press (1973)

[47] Scarf, H.E.: The approximation of fixed points of a continuous mapping. SIAM Journal on Applied Mathematics 15, 1328–1343 (1967)

[48] von Schemde, A.: Index and Stability in Bimatrix Games: A Geometric-Combinatorial Approach. Springer (2005)

[49] Schmedders, K.: A homotopy algorithm and an index theorem for the general equilibrium model with incomplete asset markets. Journal of Mathematical Economics 32, 225–241 (1999)

[50] Shapley, L.S.: A note on the Lemke-Howson algorithm. In: Mathematical Programming Studies. pp. 175–189 (1974)

[51] Shoven, J., Whalley, J.: Applying General Equilibrium. Cambridge University Press (1992)

[52] Spielman, D.A., Teng, S.H.: Smoothed analysis: Why the simplex algorithm usually takes polynomial time. Journal of the ACM 51(3), 385–463 (2004)

[53] von Stengel, B., van den Elzen, A.H., Talman, A.J.J.: Computing normal form perfect equilibria for extensive two-person games. Econometrica 70(2), 693–715 (2002)

[54] Todd, M.J.: Orientation in complementary pivot algorithms. Mathematics of Operation Research 1(1), 54–66 (1976)

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

[56] Vegh, L.A.: Strongly polynomial algorithm for a class of minimum-cost flow problems with separable convex objectives. In: ACM Symposium on the Theory of Computing (2012)

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