SIAM J. Comput.
Vol. 44, No. 6, pp. 1820–1847

A COMPLEMENTARY PIVOT ALGORITHM FOR MARKET EQUILIBRIUM UNDER SEPARABLE, PIECEWISE-LINEAR CONCAVE UTILITIES*

JUGAL GARG†, RUTA MEHTA†, MILIND SOHONI‡, AND VIJAY V. VAZIRANI†

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 polynomial parity argument on directed graphs (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: (1) Eaves (1976) gave an LCP formulation and a Lemke-type algorithm for the linear Arrow–Debreu model. We generalize both to the SPLC case, hence settling the relevant part of his open problem. (2) Our path following algorithm for SPLC markets, together with a result of Todd (1976), gives a direct proof of membership of such markets in PPAD and settles a question of Vazirani and Yannakakis (2011). (3) We settle a question of Devanur and Kannan (2008) 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. (4) We give a combinatorial way of interpreting Eaves' algorithm for the linear case, hence answering Eaves' question (1976), "That the algorithm can be interpreted as a 'global market adjustment mechanism' might be interesting to explore."

Key words. market equilibrium, exchange markets, linear complementarity problem, Lemke's scheme

AMS subject classifications. 68Q25, 91B50

DOI. 10.1137/140971002

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 functions [15, 16, 34, 27, 33, 63, 46, 60, 18], 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 nonseparable cases. As detailed in section 1.1, whereas the former always admit rational equilibria, if all parameters of the market are rational numbers, the

*Received by the editors May 30, 2014; accepted for publication (in revised form) September 17, 2015; published electronically December 17, 2015. A preliminary version of this paper appeared in the Proceedings of the 44th ACM Symposium on Theory of Computing, 2012.
http://www.siam.org/journals/sicomp/44-6/97100.html

†College of Computing, Georgia Institute of Technology, Atlanta, GA 30332 (jugal.garg@gmail.com, rmehta@cc.gatech.edu, vazirani@cc.gatech.edu). The first author was supported by Algorithms and Randomness Center, Georgia Tech. The second author was supported by NSF grants CCF-0914732 and CCF-1216019. The fourth author was supported by NSF grants CCF-0914732 and CCF-1216019, and a Guggenheim Fellowship.

‡Indian Institute of Technology, Bombay, India (sohoni@cse.iitb.ac.in).

[1] Furthermore, such utilities introduce convexity into 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.

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, 7, 59], showing that the problem is polynomial parity argument on directed graphs (PPAD) complete for both Fisher and Arrow–Debreu market models, where PPAD is the class defined by Papadimitriou [47].

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 [22, 55]), and its computability had been the subject of intense work in economics as well [4, 21, 23, 25, 41, 53]. 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 [12], 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 [36]. Another algorithm exhibiting this phenomenon is the classical Lemke–Howson algorithm for computing a Nash equilibrium of a 2-person bimatrix game, which will henceforth be called 2-Nash [39, 49, 9]; 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 [28, 61, 54]. It also motivated the definition of the complexity class PPAD [47] as described in the following quote from [59]:

> *"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* [39] *and Scarf* [51]. *Our result, showing membership in PPAD for both market models under 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* [24]. *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 [57], a direct proof of membership of the problem in PPAD. We build on the work of Eaves [19], 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 [38, 10, 11, 40].

We note that whereas several complementary pivot algorithms have been given for Nash equilibria after the work of Lemke and Howson [38, 58, 30, 37, 62], 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 [19] 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 [20], 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 state 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 [14] and [59]. 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 [19]. 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 [59].

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 [10]. 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 [57], 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 [59]. 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 [39]. Scarf has used their algorithm to derive other elementary proofs, e.g., for showing that balanced games have a nonempty core [50, 52].

Third, it enables us to prove that SPLC markets have an odd number of equilibria (up to scaling), assuming nondegeneracy, 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 [13] showed that such markets have a finite number of equilibria and then using index theorems, Dierker [17] 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.

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 [20] 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 [26].

For SPLC markets, Devanur and Kannan [14] 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]

$$\boldsymbol{M}\boldsymbol{y} \le \boldsymbol{q}, \quad \boldsymbol{y} \ge 0, \quad \text{and} \quad \boldsymbol{y} \cdot (\boldsymbol{q} - \boldsymbol{M}\boldsymbol{y}) = 0. \tag{2.1}$$

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:

$$\boldsymbol{M}\boldsymbol{y} + \boldsymbol{v} = \boldsymbol{q}, \quad \boldsymbol{y} \ge 0, \quad \boldsymbol{v} \ge 0, \quad \text{and} \quad \boldsymbol{y} \cdot \boldsymbol{v} = 0. \tag{2.2}$$

The reason for imposing nonnegativity on the slack variables is that the first condition in (2.1) implies $\boldsymbol{q} - \boldsymbol{M}\boldsymbol{y} \ge 0$. Let $\mathcal{P}$ be the polyhedron in $2n$-dimensional

[2] We refer the reader to [10] for a comprehensive treatment of notions presented in this section.

[3] The standard way of formulating an LCP is $M\boldsymbol{y} \ge -\boldsymbol{q}$, $\boldsymbol{y} \ge 0$, and $\boldsymbol{y} \cdot (M\boldsymbol{y} + \boldsymbol{q}) = 0$. However, the LCP we construct in section 4 is more amenable to the form given in (2.1).

space defined by the first three conditions; we will assume that $\mathcal{P}$ is nondegenerate.[4] Under this condition, any solution to (2.2) 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*:

$$\boldsymbol{My} + \boldsymbol{v} - z\mathbf{1} = \boldsymbol{q}, \quad \boldsymbol{y} \geq 0, \quad \boldsymbol{v} \geq 0, \quad z \geq 0, \quad \text{and} \quad \boldsymbol{y} \cdot \boldsymbol{v} = 0. \tag{2.3}$$

Let $\mathcal{P}'$ be the polyhedron defined by the first four conditions of the augmented LCP. Even though $\mathcal{P}'$ is defined in a $(2n+1)$-dimensional space, because of the first n equalities, the dimension of the polyhedron is $(n+1)$. Again we will assume that $\mathcal{P}'$ is nondegenerate, i.e., for any $0 \leq d \leq (n+1)$-dimensional face of $\mathcal{P}'$ exactly $n+1-d$ of conditions $\{y_i \geq 0, v_i \geq 0, \ \forall i \in \{1, \ldots, n\}; \ z \geq 0\}$ hold with equality. Since any solution to (2.3) must still satisfy $2n$ equalities, the set of solutions, say S, will be a subset of the 1-skeleton of $\mathcal{P}'$, i.e., it will consist of edges (1-dimensional face) and vertices (0-dimensional face) 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 $v_i = 0$.[5] 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, it follows 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 $(\boldsymbol{y_*}, \boldsymbol{v_*}, z_*)$ of $\mathcal{P}'$ it is incident on has $z_* > 0$, is called a *ray*, which can be formally represented by

$$\left\{ \begin{bmatrix} \boldsymbol{y_*} \\ \boldsymbol{v_*} \\ z_* \end{bmatrix} + \delta \begin{bmatrix} \boldsymbol{y_\circ} \\ \boldsymbol{v_\circ} \\ z_\circ \end{bmatrix} \quad \text{such that} \quad \delta \geq 0 \right\},$$

where $(\boldsymbol{y_\circ}, \boldsymbol{v_\circ}, z_\circ) \neq \mathbf{0}$ is a solution of the above LCP with $\boldsymbol{q} = \mathbf{0}$. Among the rays, one is special—the one on which $\boldsymbol{y} = \mathbf{0}$. This is called the *primary ray* and the rest are called *secondary rays.* The vertex incident at the primary can be obtained by setting $\boldsymbol{y} = \mathbf{0}$, $z = |\min_i q_i|$, and $v_i = q_i + z \ \forall i \in \{1, \ldots, n\}$. 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

[4] A representation of a polyhedron given by inequality and equality constraints in n dimensions is said to be *nondegenerate* if on its d-dimensional face exactly $n - d$ of its constraints hold with equality. For example on vertices (0-dimensional face) exactly n constraints hold with equality. There are many other equivalent ways to describe this notion.

[5] These are also known as *almost complementary solutions* in the literature.

components, (2.1) has the trivial solution $\boldsymbol{y} = 0$. Additionally, in this case Lemke's algorithm cannot be used for finding a nontrivial solution, since it is simply not applicable. However, the Lemke–Howson scheme is applicable for such a case; it follows a complementary path in the original polyhedron (2.2) starting at $\boldsymbol{y} = 0$, and guarantees termination at a nontrivial 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 $(w^i_1, \dots, w^i_n)$, 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 (nonnegative) nondecreasing, 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 $\boldsymbol{x} = (x_1, \dots, x_n)$ of goods is additively separable over the goods, i.e., $u_i(\boldsymbol{x}) = \sum_{j \in \mathcal{G}} f^i_j(x_j)$.

Given prices $\boldsymbol{p} = (p_1, \dots, p_n)$ 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 the *market clears* at given demand bundles if 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). At *equilibrium prices* there are choices of optimal bundles for the agents such that after each agent is given such a bundle, the market clears.

We call each piece of f^i_j a *segment*. The number of segments in function f^i_j is denoted by $|f^i_j|$ and the kth 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 length 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 length of the last segment is a small constant greater than one. Clearly, $\forall k < |f^i_j|,\ u^i_{jk} > u^i_{j(k+1)} \geq 0$. We will denote this market by $\mathcal{M}$.

3.1. Characterizing optimal bundles. Next, we characterize optimal bundles 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 the k_ith 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. Note that, even though the agent buys segments of a good separately, the resulting allocation is a valid allocation. This is because $u^i_{jk} < u^i_{j(k-1)} \Rightarrow \mathrm{bpb}^i_{jk} < \mathrm{bpb}^i_{j(k-1)}$.

The above characterization may also be obtained using KKT conditions of the following linear program calculating an optimal bundle of agent i:

$$\begin{array}{ll} \max: & \sum_{j,k} u^i_{jk} x^i_{jk} \\ s.t. & \sum_{j,k} x^i_{jk} p_j \le \sum_j w^i_j p_j, \\ & 0 \le x^i_{jk} \le l^i_{jk},\ \forall (j,k). \end{array}$$

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 [59]. However, an equilibrium is guaranteed to exist under certain sufficient conditions. Let us say that *agent i is nonsatiated by good j* if the last segment of f^i_j has positive slope, i.e., $u^i_{j|f^i_j|} > 0$.

Construct a directed graph whose nodes correspond to agents of market $\mathcal{M}$ and there is an edge from i' to i if and only if there is a good possessed by agent i' in its initial endowment for which agent i is nonsatiated. 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 [43].[6] Henceforth we will assume that market $\mathcal{M}$ satisfies this condition.

4. LCP formulation.

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. To start with, we obtain an LCP that captures all the market equilibria, but also admits nonequilibrium solutions. Later we modify it so that the only solutions that remain correspond to market equilibria. As stated in section 3.2, we have assumed that the given market $\mathcal{M}$ satisfies strong connectivity, hence equilibrium does exist.

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 nonnegativity 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 nonnegativity

[6] Earlier, Arrow and Debreu [2] had proved that the following are sufficient conditions: (1) *Nonzero initial endowments:* Every agent possesses a non-zero amount of every good. (2) *Nonsatiation:* Every agent is nonsatiated by some good.

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.[7]

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., (4.1) refers to the first constraint below and (4.1′) refers to the corresponding complementarity condition.)

$$\text{(4.1)} \qquad \forall j \in \mathcal{G}: \quad \sum_{i,k} q^i_{jk} \le p_j \quad \text{and} \quad p_j \left(\sum_{i,k} q^i_{jk} - p_j \right) = 0.$$

$$\text{(4.2)} \qquad \forall i \in \mathcal{A}: \quad \sum_{j} w^i_j p_j \le \sum_{j,k} q^i_{jk} \quad \text{and} \quad \lambda_i \left(\sum_{j} w^i_j p_j - \sum_{j,k} q^i_{jk} \right) = 0.$$

LEMMA 4.1. *If $\boldsymbol{p}$ is an equilibrium price vector, then $\exists \boldsymbol{q}$ such that* (4.1) *and* (4.2) *are satisfied. Further, if $(\boldsymbol{p}, \boldsymbol{q})$ satisfy* (4.1) *and* (4.2), *and $\boldsymbol{p} > 0$, then the market clears.*

Proof. For the first part, let $\boldsymbol{x}$ be the corresponding equilibrium allocation. Distribute x^i_j to obtain an allocation on individual segments x^i_{jk} as follows: start filling up from the first segment until all of x^i_j is used up, i.e.,

$$x^i_{jk} = \min \left\{ \max \left\{ x^i_j - \sum_{k' < k} l^i_{jk'}, 0 \right\}, l^i_{jk} \right\}.$$

Then the market clearing condition ensures that $q^i_{jk} = x^i_{jk} p_j$ together with $\boldsymbol{p}$ satisfies (4.1) and (4.2).

For the second part, adding the constraints in (4.1) over all goods and those in (4.2) 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 nonnegativity, all the constraints in (4.1) and (4.2) must hold with equality. Thus setting $x^i_{jk} = q^i_{jk}/p_j$ ensures market clearing. $\square$

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

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

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.

[7] Eventually, in Lemma 4.3 we will show that $1/\lambda_i$ is a lower bound on the bang-per-buck of the flexible partition of i's allocation.

Consider the following constraints and complementarity conditions:

$$(4.3)\qquad \forall(i,j,k): \quad u^i_{jk}\lambda_i \le p_j + \gamma^i_{jk} \quad \text{and} \quad q^i_{jk}(u^i_{jk}\lambda_i - p_j - \gamma^i_{jk}) = 0.$$

$$(4.4)\qquad \forall(i,j,k): \quad q^i_{jk} \le l^i_{jk}p_j \quad \text{and} \quad \gamma^i_{jk}(q^i_{jk} - l^i_{jk}p_j) = 0.$$

Let us denote the LCP defined by the sets of constraints and complementarity conditions given in (4.1), (4.2), (4.3) and (4.4), together with nonnegativity on all variables, as *LCP* (1).

Lemma 4.2. *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. Because of the strong connectivity assumption, the flexible partition of agent i contains a segment with nonzero u^i_{jk} and therefore $\lambda_i > 0$. Since the market clears, by Lemma 4.1, (4.1) and (4.2) are satisfied. Further, (4.1′) and (4.2′) are also satisfied because each agent is nonsatiated for some good and the market clears.

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 nonnegativity. 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 (4.3) and (4.4), and complementarity conditions (4.3′) and (4.4′), are all satisfied. □

LCP (1) suffers from two shortcomings. First, since the right-hand-side (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., for a subset $\mathcal{G}' \subset \mathcal{G}$ if the market consisting of agents $\mathcal{A}$ and goods $\mathcal{G}'$ satisfies strong connectivity, then this submarket has an equilibrium. Find such an equilibrium, and 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 nonzero utility segments, i.e., desire$(j) = \sum_{(i,k):u^i_{jk}>0} l^i_{jk}$. Assuming strong connectivity, we observe the following:

- If for good j, desire$(j) \le 1$, 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 $\mathcal{M}$, otherwise its demand will be more than its supply at $p_j = 0$.

If for some good j, desire$(j) \le 1$, 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 nonzero scaling of an equilibrium price vector yields an equilibrium price vector, we can impose the condition $\forall j \in \mathcal{G}: \ p_j \ge 1$, or equivalently $\forall j \in \mathcal{G}: \ p_j = p'_j + 1$ and $p'_j \ge 0$, where p'_j is a new variable.

The new LCP, *LCP* (2), is given below; nonnegativity 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 (4.5′). As in LCP (1), this complementarity condition is not needed for establishing market clearing. However, it does play an important role in proving Corollary 5.3; see also the Remark in section 4.2.

$$(4.5)\qquad \forall j \in \mathcal{G}: \quad \sum_{i,k} q^i_{jk} - p'_j \le 1 \quad \text{and} \quad p'_j\left(\sum_{i,k} q^i_{jk} - p'_j - 1\right) = 0.$$

$$(4.6)\qquad \forall i \in \mathcal{A}: \quad \sum_j w^i_j p'_j - \sum_{j,k} q^i_{jk} \le -\sum_j w^i_j \quad \text{and}$$

$$\lambda_i\left(\sum_j w^i_j(p'_j + 1) - \sum_{j,k} q^i_{jk}\right) = 0.$$

$$(4.7)\qquad \forall(i,j,k): \quad u^i_{jk}\lambda_i - p'_j - \gamma^i_{jk} \le 1 \quad \text{and} \quad q^i_{jk}(u^i_{jk}\lambda_i - p'_j - 1 - \gamma^i_{jk}) = 0.$$

$$(4.8)\qquad \forall(i,j,k): \quad q^i_{jk} - l^i_{jk}p'_j \le l^i_{jk} \quad \text{and} \quad \gamma^i_{jk}(q^i_{jk} - l^i_{jk}p'_j - l^i_{jk}) = 0.$$

It is easy to check that Lemmas 4.1 and 4.2 still hold. Additionally, we prove the following.

LEMMA 4.3. *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 (4.7) will be satisfied as a strict inequality hence forcing, via (4.7′), $q^i_{jk} = 0$ for each segment of i and hence contradicting market clearing (Lemma 4.1).

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

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

The strong connectivity assumption ensures that $\exists(i,j,k) \in Q$ with $u^i_{jk} > 0$ as the agent is nonsatiated for at least one good. Thus, σ_i is well defined. Now, by the constraint (4.7) for this segment, and the nonnegativity of γ^i_{jk}, we get that $u^i_{jk}\lambda_i = p_j + \gamma^i_{jk} \ge p_j = u^i_{jk}\sigma_i \Rightarrow \lambda_i \ge \sigma_i$.

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 (4.7). As a result, in order to satisfy (4.8′), the inequality (4.8) must be satisfied with equality, i.e., this segment is fully allocated. And if (i,j,k) is in an undesirable partition, then by construction it is unallocated and we have $q^i_{jk} = 0$.

Finally, if $(i,j,k) \in Q$, there are two cases. If $\lambda_i > \sigma_i$, then in order to satisfy (4.7), $\gamma^i_{jk} > 0$. Again, in order to satisfy (4.8′), the inequality (4.8) 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 (4.7) for the allocated segments of Q. In turn $\gamma^i_{jk} = 0$ even for unallocated segments of Q. As a result, the only constraints on q^i_{jk} are that

$0 \le q^i_{jk} \le l^i_{jk} p_j$, 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. □

THEOREM 4.4. *The set of solutions of LCP* (2) *captures exactly the set of equilibria of market* $\mathcal{M}$*, up to scaling.*

Proof. From Lemmas 4.1 and 4.3 it follows that every solution of LCP (2) corresponds to an equilibrium of market $\mathcal{M}$. Further, by Lemma 4.2 every equilibrium of $\mathcal{M}$ gives a solution of LCP (1). Since we have assumed that for every good j, desire(j) > 1, $\boldsymbol{p} > 0$ at this solution. Scaling it to ensure $\boldsymbol{p} > 1$ gives a solution of LCP (2). □

Theorem 4.4 settles the appropriate subcase of the open problem posed by Eaves (1975) [19], of formulating an LCP to capture equilibria of markets with piecewise-linear, concave utility functions.

4.1. Our nondegeneracy assumption. Recall that in section 2 while outlining Lemke's algorithm, we had assumed that the polyhedron corresponding to the LCP was nondegenerate. Now, it turns out that the polyhedron corresponding to LCP (2) has an inherent degeneracy, so we need to clarify the nondegeneracy assumption we are making. The degeneracy comes about because of the following fact established in the proof of Lemma 4.1: adding the constraints in (4.5) over all goods and those in (4.6) over all agents yields two identical equations. Henceforth, we will say that the polyhedron corresponding to LCP (2) is nondegenerate if it has no other degeneracies. In other words, if a set of inequalities holding with equality at a vertex are linearly dependent, then this linear dependency has to be between (4.5) $\forall j$, and (4.6) $\forall i$.

THEOREM 4.5. *If polyhedron* $\mathcal{P}$ *corresponding to LCP* (2) *is nondegenerate, then the solutions of LCP* (2) *will be in one-to-one correspondence with the equilibria of* $\mathcal{M}$*, up to scaling.*

Proof. Since the equilibria of $\mathcal{M}$ are scale invariant, we consider only those where the minimum price is one. Let $\mathcal{E}$ be this set. In LCP (2), $p'_j + 1$ represents the price of good j, and therefore it suffices to show one-to-one correspondence between elements of $\mathcal{E}$ and solutions of LCP (2) with $p'_j = 0$ for some good j.

Given $(\boldsymbol{p_*}, \boldsymbol{x_*}) \in \mathcal{E}$, using Theorem 4.4 it maps to a solution of LCP (2) with $\boldsymbol{p'_*} = \boldsymbol{p_*} - \boldsymbol{1}$ and $\boldsymbol{q_*}$ as given in Lemma 4.1. We will show that the set of solutions of LCP (2) with $\boldsymbol{p'} = \boldsymbol{p'_*}$ and $\boldsymbol{q} = \boldsymbol{q_*}$ is a singleton, and the theorem will follow.

Suppose it is not true. In a solution of LCP (2) fixing $\boldsymbol{p'}$, $\boldsymbol{q}$, and $\boldsymbol{\gamma}$ fixes $\boldsymbol{\lambda}$ as well. Therefore, it must be the case that for some agent i, its flexible partition Q is fully allocated, i.e., $q^i_{*_{jk}} = l^i_{jk}(p'_{*_j} + 1)\ \forall (j,k) \in Q$. Then setting $\lambda_i = \frac{u^i_{jk}}{p'_{*_j}+1}$, where $(j,k) \in Q$ and $\gamma^i_{jk} = 0\ \forall (j,k) \in Q$ gives a solution of LCP (2) together with $\boldsymbol{p'_*}$, $\boldsymbol{q_*}$, and other coordinates of $\boldsymbol{\gamma}$ and $\boldsymbol{\lambda}$ set accordingly. For this solution at least $M + 2$ conditions of $\mathcal{P}$ hold with equality, where M is the number of variables. Namely, one each to satisfy complementarity conditions (4.7′) and (4.8′), and market clearing conditions (4.5), $\forall j$, (4.6) $\forall i$. This gives M conditions. One more from $p'_j = 0$ for some j. And finally, $\forall (j,k) \in Q$, (4.8) and $\gamma^i_{jk} \ge 0$ hold with equality which gives at least one more.

Even with inherent degeneracy at most $M + 1$ conditions can be tight at any point of a nondegenerate $\mathcal{P}$, hence we get a contradiction. □

Remark. Observe that in the case that all segments of the flexible partition, Q, of an agent i are fully allocated at an equilibrium, the setting of λ_i in the corresponding solution of LCP (2) 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 4.5 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.

4.2. The augmented LCP. LCP (2) has the same form as the formulation given in (2.1) in section 2. Next, we obtain its augmented LCP, i.e., in the form of the formulation given in (2.3); 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 (4.6). Hence, we make two changes to LCP (2) to obtain the augmented LCP, which we call *LCP* (3). First, we change (4.6) as follows:

$$\forall i \in \mathcal{A}: \quad \sum_j w^i_j p'_j - \sum_{j,k} q^i_{jk} - z \leq -\sum_j w^i_j \quad \text{and}$$
$$\lambda_i \left(\sum_j w^i_j (1 + p'_j) - \sum_{j,k} q^i_{jk} - z \right) = 0. \tag{4.9}$$

Second, we impose nonnegativity on z. If the polyhedron $\mathcal{P}'$ of LCP (3) is nondegenerate, then as described in section 2 its solutions form paths and cycles on the 1-skeleton of $\mathcal{P}'$ and the Lemke's algorithm traverses one of these paths. However, the inherent degeneracy in LCP (2) described in section 4.1 extends to degeneracy in $\mathcal{P}'$ at points with $z = 0$. We need to argue that no other degeneracy is introduced due to this augmentation.

In general, if a polyhedron is degenerate then it has a degenerate vertex.[8] We will show that if nonzero parameters of the market, namely, u^i_{jk}'s, l^i_{jk}'s, and w^i_j's, are in a general position then every vertex solution of LCP (3) with $z > 0$ is nondegenerate, and therefore has a unique double label. Thus the solutions will still form paths and cycles on the 1-skeleton of $\mathcal{P}'$ (see section 2).

LEMMA 4.6. *If nonzero u^i_{jk}'s, l^i_{jk}'s, and w^i_j's do not have any polynomial relation among them, except for $\sum_{i \in \mathcal{A}} w^i_j = 1$ $\forall j$, then every vertex solution of LCP* (3) *with $z > 0$ is nondegenerate.*

Proof. Let $\boldsymbol{v} = (\boldsymbol{\lambda}, \boldsymbol{p'}, \boldsymbol{q}, \boldsymbol{\gamma}, z)$ be a vertex solution of LCP (3) with $z > 0$. To the contrary suppose it is degenerate, then there are at least two double labels at $\boldsymbol{v}$. Let $\mathcal{T}$ be the set of linear conditions of LCP (3) that are holding with equality at $\boldsymbol{v}$. Remove variables that are zero from all the conditions of $\mathcal{T}$, nonnegativity conditions corresponding to these variables, and conditions corresponding to double labels from $\mathcal{T}$. A brief outline of the proof is as follows: using equalities of $\mathcal{T}$, we will first write all the nonzero variables as linear functions of z, where coefficients and constants of the functions are in terms of monomials in market parameters. Finally, replacing these

[8] A d-dimensional face of a polyhedron $\mathcal{P}$ in n dimensions is said to be degenerate if at least $n - d + 1$ conditions defining $\mathcal{P}$ are holding with equality at this face.

in the two conditions corresponding to double labels will give a polynomial relation among the parameters.

For forced segments, i.e., with $\gamma^i_{jk} > 0$, remove (4.7) and (4.8) from $\mathcal{T}$, and replace q^i_{jk} with $l^i_{jk}(p'_j + 1)$. Note that for undesirable segments (4.7) is not tight and q^i_{jk} is zero, and hence they do not have any of (4.7) or (4.8) in $\mathcal{T}$. Thus, none of (4.8) remain in $\mathcal{T}$ and all the remaining (4.7) correspond to flexible segments.

Using the remaining conditions of $\mathcal{T}$ we will write all the nonzero variables, except z, as linear functions of z. The remaining q^i_{jk}'s correspond to flexible segments, which is at most one for any given (i, j). So we will rename it as q^i_j. Also rename the corresponding u^i_{jk} by u^i_j.

Let $\mathcal{E}$ be the set of (i, j) pairs where agent i has a flexible segment for good j, i.e., the remaining equalities of type (4.7) with $\gamma^i_{jk} = 0$. Thus, we have

$$u^i_j \lambda_i - p'_j = 1 \quad \text{for each} \quad (i, j) \in \mathcal{E}. \tag{4.10}$$

Considering the pairs in $\mathcal{E}$ as edges between elements of $\mathcal{A}$ and $\mathcal{G}$ we get a bipartite graph, say H. This graph is acyclic, or else we get a polynomial relation between u^i_j's using (4.10) along the cycle edges, and by eliminating λ_i's and p_j's.

In every connected component of H pick a representative good. If there is an undersold good in a component, then pick that as its representative good. Let g be the representative good of a component $\mathcal{C}$. Then for any $j \in \mathcal{C}$, using (4.10) for the edges on the path between g and j in $\mathcal{C}$, $(p'_j + 1)$ can be written as $\frac{f_1(U)}{f_2(U)}(p'_g + 1)$, where $f_1(U)$ and $f_2(U)$ are monomials in u^i_j's, and similarly λ_i for each agent $i \in \mathcal{C}$. If g is undersold then $p'_g = 0$ and hence $\forall j \in \mathcal{C}$, p'_j is a constant represented by the ratio of monomials in u^i_j's. Further, no other good can be undersold in $\mathcal{C}$ or else using this fact we will get a polynomial relation in u^i_j's.

Suppose for component $\mathcal{C}$, g is not undersold, i.e., all its goods are fully sold. Consider a leaf node u of $\mathcal{C}$ and remove the edge, say (u, v), incident on it to construct $\mathcal{C}'$. Consider $\mathcal{C}'$ as rooted at v. Using market clearing conditions (4.9) and (4.5), respectively, for agents and goods in $\mathcal{C}'$ from leafs to the root, we can write all the q^i_j's for edges in $\mathcal{C}'$ as linear functions of representative price variables and z. Market clearing conditions of u and v give two different expressions for q^i_j on the missing edge (u, v), and thus we obtain a linear relation among representative prices and z. This has to be a nontrivial relation because exactly one of them will have a w^i_j not present in the other.

Even if g is undersold implying $p'_g = 0$, using a similar approach where g is a root, q^i_j's of component $\mathcal{C}$ can written as linear functions of representative prices and z. This generates a system of linear equations, one for each component: $p'_g = 0$ if g is undersold, and a linear equality in z and representative prices if g is fully sold. From these obtain p'_g's as linear functions of z. Replace these in expressions for λ_i's, q^i_j's, and remaining p'_j's to obtain their linear functions on z.

Now consider the equalities corresponding to double labels that we had removed from $\mathcal{T}$. Replace all the variables by their linear function in z. Use one to get an expression for z in terms of input parameters u^i_{jk}'s, l^i_{jk}'s, and w^i_j's. Replacing z in another gives a polynomial relation among the parameters, even after replacing any $\sum_j w^i_j$ with one. This is because there is no such way to eliminate u^i_{jk}'s and l^i_{jk}'s. A contradiction. $\square$

Remark. Clearly, the set of solutions of LCP (3) in which $z = 0$ enjoy all properties established in Theorem 4.4 and its lemmas. However, if $z > 0$, market clearing, which was shown in Lemma 4.1, does not hold. Despite the nonapplicability of Lemma 4.1, observe that by (4.5′), if $p'_j > 0$, then good j must be fully sold. This fact will be used at multiple places, e.g., for proving Corollary 5.3.

5. Proving nonexistence 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, $\boldsymbol{y_*} = \boldsymbol{y_\circ} = 0$ 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[9] 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 nonnegativity constraint in LCP (3).

The vector $\boldsymbol{y}$ consists of four types of variables, i.e., $\boldsymbol{y} = (\boldsymbol{\lambda}, \boldsymbol{p}', \boldsymbol{q}, \gamma)$. 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 5.1. *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, (4.7) is satisfied with strict inequality. Hence, by (4.7′), $q^i_{jk} = 0$. Hence i does not spend any money and her surplus equals her earnings.

If $\lambda_i > 0$ then by (4.9′), $z = \sum_j w^i_j(1 + p'_j) - \sum_{j,k} q^i_{jk}$. Hence, by (4.9), z represents her unspent money. Since $z \geq 0$, i's surplus is nonnegative. □

LEMMA 5.2. *If in a solution, T, to LCP* (3)*, each good is fully sold then $z = 0$.*

Proof. Since each good j is fully sold, $p_j = 1 + p'_j = \sum_{i,k} q^i_{jk}$. Adding over all goods we get $\sum_{i,j,k} q^i_{jk} = \sum_j p_j$. The left-hand side (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 5.1, each

[9] The exception is the contradiction in Corollary 5.3.

agent has nonnegative surplus. Therefore, each agent must have zero surplus. Hence by Claim 5.1, $\lambda_i > 0\ \forall i$ and $z = 0$. □

Corollary 5.3. $\boldsymbol{p'_\circ} \not> 0$.

Proof. Suppose $\boldsymbol{p'_\circ} > 0$. Then, at every point of R with $\alpha > 0$, $\boldsymbol{p'} > 0$ and therefore by (4.5′) every good is fully sold. Hence, by Lemma 5.2, $z = 0$. 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 $\alpha > 0$, $z > 0$ leads to a contradiction. □

Lemma 5.4. *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 nonempty. Let $A_1 \subseteq A$ be the set of agents who are nonsatiated 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 $\overline{S}$. Note that at every solution of LCP (3), allocation of each agent is as per the decreasing order of bang-per-buck due to (4.7), (4.7′), (4.8), and (4.8′).

By (4.5), 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 5.1 each agent has a nonnegative 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 nonzero amount of a good from $\overline{S}$ must eventually demand more than one unit of some good in S, contradicting (4.5).

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 nonsatiated. 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 (4.5).

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 nonsatiated. 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 (4.5), since desire(j) > 1. □

Lemma 5.5. *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 (4.5) 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 nonsatiated 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 (4.7), 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 (4.7′) 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.

Table 1
Complementary pivot algorithm for SPLC utilities.

```
Initialization: Let $T \leftarrow T_0$
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. $T \leftarrow T'$.
Endwhile
Output solution $T$.
```

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, (4.7) is satisfied as a strict inequality, which forces $\boldsymbol{q_*} = 0$ by ($4.7'$). Now, by ($4.5'$), this forces $\boldsymbol{p'_*} = 0$, and in turn ($4.8'$) forces $\boldsymbol{\gamma_*} = 0$. Altogether we get $\boldsymbol{y_*} = 0$. ◻

Corollary 5.3, Lemmas 5.4 and 5.5 give the following theorem.

Theorem 5.6. *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 ath constraint is v_a. This gives us an LCP in the form of the formulation given in (2.3); 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. If M is the matrix of this LCP and $\boldsymbol{d}$ is the coefficient vector of variable z, then $T_0 = (\boldsymbol{y}^0, \boldsymbol{v}^0, z^0)$, where $\boldsymbol{y}^0 = 0$, $z^0 = \max_i \sum_j w^i_j$, and $\boldsymbol{v}^0 = \boldsymbol{q} - M\boldsymbol{y} + \boldsymbol{d}z$. Clearly, for $i' \in \operatorname{argmax}_i \sum_j w^i_j$, we have $v^0_{i'} = y^0_{i'} = 0$.

Assuming that nonzero market parameters are in a general position (except for one w^i_j per good so that $\sum_i w^i_j = 1$), Lemma 4.6 ensures that vertices with $z > 0$ are nondegenerate. Therefore the algorithm will never encounter degeneracy and hence will have a unique double label in each step.

In the absence of a nondegeneracy assumption, there are standard ways to handle degeneracy in Lemke's scheme, namely, the *lexico-minimum ratio test* (see section 4.3 of [48] and also [8]) to ensure termination in a finite number of steps. Using similar techniques, the algorithm of Table 1 can be modified to handle degeneracies as well. Thus, Theorem 5.6 directly yields the following.

Theorem 6.1. *If an SPLC market $\mathcal{M}$ satisfies strong connectivity and the desire of each good exceeds one, then $\mathcal{M}$ admits an equilibrium and the algorithm in Table* 1 *terminates with one.*

Theorem 6.1 settles the appropriate case of the open problem, posed by Eaves (1975) [19], as described in the introduction. The algorithm also gives a constructive proof of the existence of an equilibrium for SPLC markets that satisfies strong connectivity.

Theorem 6.2. *Assuming strong connectivity, the problem of computing an equilibrium of a market with SPLC utilities is in PPAD.*

Proof. By Theorem 5.6, the algorithm in Table 1 must converge to an equilibrium. Let v be a vertex on the Lemke path found by this algorithm. To prove membership of SPLC markets in PPAD, we need to show that the unique predecessor and successor of v on this path can be found efficiently. Clearly, these two vertices, say u and w, can be found simply by pivoting. To determine which vertex leads to the start of the path, i.e., the primary ray, and which leads to the end, we use Todd's result [57] on the orientability of the path followed by a complementary pivot algorithm. Todd shows that the signs of the subdeterminants of tight constraints satisfied by the vertices u, v, and w give this information. Hence we get a direct proof of membership of the problem in PPAD. □

This settles the question posed by Vazirani and Yannakakis in [59] 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 at $z = 0$ as that of LCP (2), and is 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 4.1 holds. Once again, we will say that the polyhedron corresponding to LCP (3) is nondegenerate if it has no other degeneracies at $z = 0$, and conditions of Lemma 4.6 are satisfied.

LEMMA 6.3. *Let v be a vertex solution to LCP* (3) *with $z = 0$. Then there is exactly one $j \in \mathcal{G}$ with $p'_j = 0$ at v assuming nondegeneracy.*

Proof. The nondegeneracy assumption ensures that exactly one extra inequality is tight at v, implying that for exactly one complementarity condition the variable is zero and inequality is also tight. The surplus of agents is upper bounded by z (using (4.9)), and hence is at most zero. This together with (4.5) implies that all the goods are sold completely, i.e., all of (4.5) hold with equality. Therefore there cannot be more than one $p'_j = 0$ at v.

Suppose, there is no $p'_j = 0$ at v. Consider a point w in the interior of the edge preceding v. At w we have $\boldsymbol{p}' > 0$ and $z > 0$, a contradiction due to Lemma 5.2. □

Let v be a vertex solution to LCP (3) with $z = 0$. By Lemma 6.3, 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 6.4. *If the polyhedron $\mathcal{P}'$ corresponding to LCP* (3) *of an SPLC market is nondegenerate, 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 (using Lemma 4.6). The solutions of LCP (3) satisfying $z = 0$ are precisely the solutions to LCP (2). By Theorem 4.5, 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. One of the paths starts with the primary ray and ends with an equilibrium. Since by Theorem 5.6 $\mathcal{P}'$ has no secondary rays, the rest of the equilibria must be paired up. Hence there are an odd number of equilibria. □

7. Strongly polynomial bound.

Devanur and Kannan [14] 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 polyhedron $\mathcal{P}'$ of LCP (3), when the 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 $(p_1, \dots, p_n, z)$-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 $p_1, \ldots, p_n$, z. For each 5-tuple (i, j, j', k, k'), where $i \in \mathcal{A}$, $j \neq j' \in \mathcal{G}$, $k \leq |f^i_j|$, and $k' \leq |f^i_{j'}|$, 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 $Q^i_1 \cup Q^i_2 \cup \cdots \cup Q^i_{l-1}$. 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 a 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 rightmost 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 7.1. *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. Each fully labeled vertex and each cell have their 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 we have the following:

- 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 consideration. If it does then the 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 $\mathcal{P}'$. □

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 $\lambda_1, \dots, \lambda_m$. Every fully labeled vertex $(\boldsymbol{\lambda}, \boldsymbol{p}', \boldsymbol{q}, \boldsymbol{\gamma}, z)$ naturally gets mapped to this space by taking a 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 $Q^j_1, Q^j_2, \dots$. 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 a flexible partition, all the partitions before it forced partitions and all partitions after it undesirable partitions for good j. Further, let (i, j, k) be a segment in the 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 $i \neq i' \in \mathcal{A},\ j \in \mathcal{G},\ k \leq |f^i_j|$, and $k' \leq |f^{i'}_j|$. Given a cell, the signs of these hyperplanes in the cell give a 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 $Q^j_1, Q^j_2, \dots$.

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 7.1. 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 7.2. *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 hyperplanes introduced is strongly polynomial in both cases.

THEOREM 7.3. *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 the 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

$$\max_{j \in \mathcal{G}} \frac{u^i_j}{p_j}.$$

Second, we don't need the variables γ^i_{jk} and the constraints and complementarity conditions given in (4.8). This was precisely the LCP derived by Eaves [20].

$$\text{(8.1)} \qquad \forall j \in \mathcal{G}: \qquad \sum_i q^i_j - p'_j \leq 1 \quad \text{and} \quad p'_j \left(\sum_i q^i_j - p'_j - 1 \right) = 0.$$

$$\text{(8.2)} \qquad \forall i \in \mathcal{A}: \qquad \sum_j w^i_j p'_j - \sum_j q^i_j - z \leq - \sum_j w^i_j \quad \text{and}$$

$$\lambda_i \left(\sum_j w^i_j (p'_j + 1) - \sum_j q^i_j - z \right) = 0.$$

$$\text{(8.3)} \qquad \forall (i,j): \qquad u^i_j \lambda_i - p'_j \leq 1 \quad \text{and} \quad q^i_j (u^i_j \lambda_i - p'_j - 1) = 0.$$

Construct a bipartite graph whose vertices are $\mathcal{A} \cup \mathcal{G}$ and (i,j) is an edge iff inequality (8.3) is satisfied as an equality. Call this the *tight graph* and its edges the *tight edges*. Observe that tight edges correspond to the maximum bang-per-buck (agent, good) pairs. By (8.3$'$), goods can only be sold along tight edges. 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 corresponding to the tight inequalities on e; singleton agents are not considered as components.

By (8.1$'$), if $p'_j > 0$, j must be fully sold. If all goods are fully sold, all of the agents' money must be spent, making $z = 0$ (using Lemma 5.2). However, since $z > 0$, there is at least one undersold good, say j and $p'_j = 0$ at all points of edge e. Furthermore, the tight graph too remains unchanged on all points of e implying that in a component $(p'_j + 1)$'s and λ_i's have to change by the same multiplicative factor in order to maintain equalities of type (8.3) corresponding to tight edges. Hence, the prices of goods and the λ_i's of agents must remain unchanged, in any component containing a good with $p'_j = 0$, while moving along e.

Next consider the remaining components. Note that all goods in these components are fully sold and all agents have the same surplus, namely, z (using Claim 5.1). 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. As noted earlier, in such a component, $(1 + p'_j)$'s and λ_i's change by the same multiplicative factor. Hence prices and λ_i's in a component either monotonically increase or decrease. Next we show that it is in fact true across all the components.

LEMMA 8.1. *While tracing an edge e of polyhedron $\mathcal{P}^l$*

- *all prices and λ_i's either monotonically increase or monotonically decrease;*
- *z monotonically decreases iff prices monotonically increase.*

Proof. For the first part, we know that all $(1+p'_j)$'s and λ_i's either monotonically increase or decrease in a component of a *tight graph*. We only need to show that it is true across the components as well. To the contrary, let S_1 and S_2 be the sets of components whose prices strictly decrease and increase, respectively. Consider the market clearing conditions for all the agents of S_1. The total income of these agents increases, while the total spending decreases. Therefore, the total surplus of these agents increases. Since the surplus of all these agent is the same, namely, z, we get that z is increasing on e.

On the other hand, consider the market clearing conditions for all the agents of S_2. The total income of these agents decreases and the total spending increases, hence the total surplus of these agents decreases. This implies that z has to decrease on e, a contradiction.

For the second part, suppose z decreases and prices also decrease. Consider the subset S of components of a tight graph for which prices strictly decrease. As prices of goods outside S remain constant, the decrease in total spending of agents in S is strictly more than the decrease in their earnings, hence the surplus, i.e., z, increases. The other direction follows using a similar argument. □

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 $v = (\boldsymbol{\lambda}, \boldsymbol{p}, \boldsymbol{q}, z)$. While moving on π suppose the algorithm enters v through e_1 and leaves it through e_2.

LEMMA 8.2. *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 at v be as follows:

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 (using Lemma 8.1).
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 otherwise the prices should decrease on e_1. A contradiction to z decreasing on e_1 using Lemma 8.1.
4. $\lambda_i \geq 0$: This case never arises otherwise λ_i's should decrease on e_1. This in turn implies prices are decreasing and z is increasing on e_1 (Lemma 8.1), a contradiction.
5. $u^k_l \lambda_k - p'_l \leq 1$: Then we relax $q^k_l = 0$. This case is a 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. Then C_1 merges with one of the components in C_0 and therefore gets connected with a good j whose $p'_j = 0$. Hence prices of goods in C_1 are fixed on e_2; if the rest of the prices decrease then the income and in turn surplus of agents in C_1 decreases, a contradiction (using Lemma 8.1).
   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_i's change by a multiplicative factor α_x on e_1. Since an agent

of C_1 got interested in a good of C_2 at v, we have $\alpha_1 > \alpha_2 \geq 1$. Let z change by a multiplicative factor α on e_1.

Consider all α's relative to α_1, or in other words all the price changes relative to price changes in C_1. Using the market clearing conditions for the h components we can write $\alpha_x = c_x\alpha_1 + d_x$ and $\alpha = c\alpha_1 + d$. On e_2, C_1 and C_2 merge into a single component C_{12}. Let α''s be the factors on e_2, and similarly express each α'_x as a linear expression $c'_x\alpha'_{12}+d'_x$ and $\alpha' = c'\alpha'_{12}+d'$. Since all the prices either increase or decrease on an edge (Lemma 8.1), we get $c_x, c'_x > 0 \ \forall x \geq 3$.

CLAIM 8.3. *$c_x \leq c'_x \ \forall x \geq 3$.*

Proof. Suppose, on e_1 we get $\alpha_x = c_{x1}\alpha_1 + c_{x2}\alpha_2 + d_x \ \forall x \geq 3$, if we do not eliminate α_2. In addition, we have $\alpha_2 = c_2\alpha_1 + d_2 \Rightarrow \alpha_1 = \frac{1}{c_2}\alpha_2 - \frac{d_2}{c_2}$. Replacing these two in α_x gives, $\alpha_x = \frac{c_{x1}}{c_2}\alpha_2 + c_{x2}c_2\alpha_1 - \frac{d_2}{c_2}c_{x1} + d_2c_{x2} + d_x$. This gives $c_{x1} = c_{x2}c_2$ implying that $c_{x1} > 0 \Leftrightarrow c_{x2} > 0$ because $c_2 > 0$. Further we get $c_x = c_{x1} + c_{x2}c_2$, and then $c_x > 0$ implies that $c_{x1}, c_{x2} > 0$. Since $C_3, \ldots, C_k$ are unchanged when we move from e_1 to e_2, and since C_1 and C_2 merge and form C_{12} on e_2, we have the same expression hold on e_2 with α_1 and α_2 replaced with α_{12}. Thus, we have $c'_x = c_{x1} + c_{x2}$. Finally, $\alpha_1 > \alpha_2$ gives $c_2 < 1$, and in turn we get $c_x \leq c'_x \ \forall x \geq 3$. □

At v we have $\alpha'_{12} = 1$. When we relax $q^k_l = 0$ at vertex v and move on e_2, exactly one of the following two happens: either $\alpha'_{12} > 1$ or $\alpha'_{12} < 1$. Clearly $\alpha'_{12} > 1$ will imply that z decreases on e_2 as was required (Lemma 8.1).

To the contrary suppose $\alpha'_{12} < 1$ on e_2. Then $c_x \leq c'_x \ \forall x \geq 3$ implies that the total surplus of agents in subcomponent C_1 of C_{12} decreases at a faster rate on e_2. This is possible only if C_1 gets money from C_2 through the (k, l) edge implying $q^k_l < 0$, a contradiction.

6. $q^i_j = 0$: Then we relax $u^i_jy_i - p_j = 0$. A similar analysis to 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 8.2, 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 6.4. Therefore, $z = 0$ at both the endpoints and it must increase monotonically while following the path backwards. Thus, both should lead to the primary ray, a contradiction. Hence we get the following.

LEMMA 8.4. *If the polyhedron $\mathcal{P}^l$ of LCP (4) for a linear market is nondegenerate, 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 8.5. Consider a simple market with 2 agents, 3 goods, and 2 segments for every pair of agent and good, where

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

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

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
z	2.2	1.9	1.9	1	1	0.8	0.8	0.7	0.725	0.275	0.05	0
$1+p'_1$	1	1	1	1	1	1	1	1	1	1.5625	1.375	1.5
$1+p'_2$	1	1	1	1	1	1	1	1.125	1.125	1.125	1.5	1.5
$1+p'_3$	1	1	1	1	1	1	1	1	1	1	1	1

In U^i's and L^i's above, the rows correspond to goods and columns to segments, and in W rows correspond to agents and columns to goods. 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'_j's 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 nondegenerate; this can be achieved by a small perturbation of u^i_j's. We continue with the terminology set in the previous section.

Let $\mathcal{A}_0$ be the set of agents with surplus less than z, and let $\mathcal{A}_1 = \mathcal{A}\backslash\mathcal{A}_0$. Consider the bipartite graph between agents of $\mathcal{A}_1$ and all the goods, and the tight edges. Recall that edge (i, j) is said to be tight if agent i gets maximum bang-per-buck from good j at current prices. We say that edge (i, j) is *nonzero* if agent i is spending money on good j.[10]

Let E be the set of tight 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 multiplicative factor with which prices of goods in C_i change (p_j changes to $\alpha_i p_j$). Note that $\alpha_0 = 1$. The market clearing conditions for each C_i gives l equalities in $\alpha_1, \ldots, \alpha_l$ and z. Using these 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.

Starting with z set to the maximum surplus and prices set to one, we will decrease z and accordingly increase prices (Lemma 8.2) so that the following conditions are maintained. These are exactly the conditions (8.1), (8.2), and (8.3), respectively, forming the Eave's LCP.

- Goods are never oversold, and the price of an undersold good is set to one.
- Surplus of an agent is at most z, and only agents with surplus z are in the market (can spend money).
- Agents spend money only on goods giving highest bang-per-buck, i.e., the set of nonzero edges is a subset of E.

Note that as the prices and the surplus change, the money spent by agents on goods also changes. Therefore maintaining the feasible money allocation (i.e., q^i_j's) is a challenging task. To handle this we 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 $g_j \in C_k$. We maintain the

[10] If the given market is nondegenerate, 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:
    $p_j \leftarrow 1, \forall j \in \mathcal{G}$;    $e_i \leftarrow \sum_j w^i_j p_j, \forall i \in \mathcal{A}$;    $z \leftarrow \max_{i \in \mathcal{A}} e_i$;
    $\mathcal{A}_0 \leftarrow \{i \in \mathcal{A} \mid e_i < z\}$;    $\mathcal{A}_1 = \mathcal{A} \setminus \mathcal{A}_0$;    $C_0 \leftarrow \mathcal{G}$;
    $E \leftarrow \{(i,j) \in (\mathcal{A}_1 \times \mathcal{G}) \mid bpb^i_j = \max_{k \in \mathcal{G}} bpb^i_k\}$;
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 nonzero 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
```

following invariant in the algorithm by ensuring that $cap(s,i)$ is the money spent by agent i, and that goods are never oversold.

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.

In the proof of Lemma 8.2 we showed that out of six cases, two cases never arise. The remaining four correspond to the four events of the algorithm. 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 components 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 the case of C_0 we observe that each of its components cannot have more than one undersold good, i.e., goods with price 1. Otherwise, the bang-per-buck conditions for the tight edges on the path between unit price goods give an algebraic relation between u^i_j's, contradicting the assumption that u^i_j's are generic. Further, since prices in C_0 are constant and z is decreasing, only the undersold good can consume the extra money spent. Therefore, if flow on (i,j) decreases to zero, then the undersold good should be in the subcomponent containing i, when (i,j) is removed. In other words, after removal of (i,j) from E the component containing j has only fully sold goods, and hence the same procedure applies.

Event 4 may be easily calculated using the fact that every component of C_0 has exactly one undersold good: for each of its components calculate z that clears the market from the goods side, and pick the maximum z among them.

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^i_{jk}'s, l^i_{jk}'s, and w^i_j's were drawn uniformly at random from the intervals $[0,1]$, $[0, \frac{1}{\#\text{Seg}}]$, and $[0,1]$, respectively. The w^i_j values were scaled so that the total amount of each good is unit. Finally, for each agent i and good j, corresponding u^i_{jk}'s were sorted in decreasing order to get the SPLC utility function f^i_j.

Table 4
Experimental results over random instances.

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

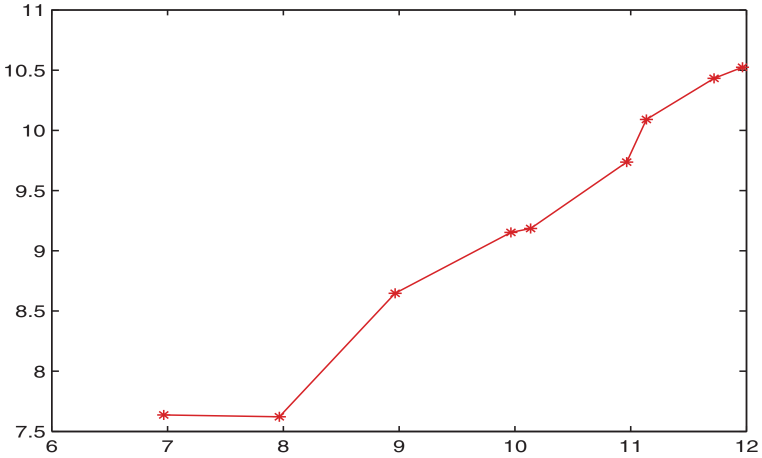

Fig. 1. *Plot of* $\log_2(\sum_{i,j} |f_j^i|)$ *versus* $\log_2$(*Max iters*).

Note that, even in the worst case the number of iterations is of the order of the total number of segments of utility functions, i.e., $\sum_{i,j} |f_j^i| = |\mathcal{A}| \times |\mathcal{G}| \times \#\text{Seg}$. Figure 1 plots $\log_2(\sum_{i,j} |f_j^i|)$ versus $\log_2$(Max iters) for a comparative analysis.

10. Discussion. Besides parity [54], the Lemke–Howson algorithm has yielded numerous insights into structural properties of 2-Nash equilibria, such as index, degree, and stability [28, 61, 54]. It has been the subject of much other work, e.g., [6] determine its smoothed complexity, [56] and [49] 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 [35, 42]; 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 [58, 29, 31, 30, 38, 39, 44] can locate all the Nash equilibria of a game has been studied extensively [3, 32, 45]. 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 nonseparable, PLC 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 [59], is to prove that this case is FIXP-complete; it is known to have algebraic equilibria.

An obvious approach to answering the question of [59] was to build on the work of [15], 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 [15]. 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 [26], 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 theoretical computer science, 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 [19], we restate the question of studying such markets.

REFERENCES

[1] I. Adler, *private communication*, 2014.

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

[3] A. Balthasar, *Equilibrium tracing in strategic-form games*, Econom. Theory, 42 (2010), pp. 39–54.

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

[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, IEEE Computer Society, Los Alamitos, CA, 2009.

[6] X. Chen, X. Deng, and S.-H. Teng, *Settling the complexity of computing two-player Nash equilibria*, J. ACM, 56 (2009), 14.

[7] X. Chen and S.-H. Teng, *Spending is not easier than trading: On the computational equivalence of Fisher and Arrow-Debreu equilibria*, in Algorithms and Computation, Lecture Notes in Comput. Sci. 5878, Springer, Berlin, 2009, pp. 647–656.

[8] V. Ćhvatal, *Linear Programming*, Freeman, New York, 1983.

[9] B. Codenotti, S. De Rossi, and M. Pagan, *An Experimental Analysis of Lemke-Howson Algorithm*, preprint, arXiv:0811.3247, 2008.

[10] R. Cottle, J. Pang, and R. Stone, *The Linear Complementarity Problem*, Academic Press, Boston, 1992.

[11] C. W. Cryer, *The method of Christopherson for solving the free boundary problems for infinite journal bearing by means of finite differences*, Math. Comp., 25 (1971), pp. 435–443.

[12] G. B. Dantzig, *Linear Programming and Extensions*, Princeton University Press, Princeton, NJ, 1963.

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

[14] N. Devanur and R. Kannan, *Market equilibria in polynomial time for fixed number of goods or agents*, in IEEE Annual Symposium on Foundations of Computer Science, IEEE Computer Society, Los Alamitos, CA, 2008, pp. 45–53.

[15] N. Devanur, C. H. Papadimitriou, A. Saberi, and V. V. Vazirani, *Market equilibrium via a primal-dual algorithm for a convex program*, J. ACM, 55 (2008), 22.

[16] N. Devanur and V. Vazirani, *An improved approximation scheme for computing Arrow-Debreu prices for the linear case*, in Proceedings of 23rd Foundations of Software Tech-

nology and Theoretical Computer Science, Lecture Notes in Comput. Sci. 2914, Springer, Berlin, 2003, pp. 149–155.

[17] E. Dierker, *Two remarks on the number of equilibria of an economy*, Econometrica, 40 (1972), pp. 951–953.

[18] R. Duan and K. Mehlhorn, *A combinatorial polynomial algorithm for the linear Arrow-Debreu market*, in Automata, Languages, and Programming, Lecture Notes in Comput. Sci. 7965, Springer, Berlin, 2013, pp. 425–536.

[19] B. C. Eaves, *A finite algorithm for the linear exchange model*, Cowles Foundation Discussion Paper No. 389, Yale University, New Haven, CT, 1975.

[20] B. C. Eaves, *A finite algorithm for the linear exchange model*, J. Math. Econom., 3 (1976), pp. 197–203.

[21] B. C. Eaves and K. Schmedders, *General equilibrium models and homotopy methods*, J. Econom. Dynam. Control, 23 (1999), pp. 1249–1279.

[22] B. Ellickson, *Competitive Equilibrium: Theory and Applications*, Cambridge University Press, Cambridge, 1994.

[23] M. Esteban-Bravo, *An interior-point algorithm for computing equilibria in economies with incomplete asset markets*, J. Econom. Dynam. Control, 32 (2008), pp. 677–694.

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

[25] C. Garcia and W. Zangwill, *Pathways to Solutions, Fixed Points, and Equilibria*, Prentice-Hall, Englewood Cliffs, NJ, 1981.

[26] J. Garg, R. Mehta, M. Sohoni, and N. K. Vishnoi, *Towards polynomial simplex-like algorithms for market equilibria*, in Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, Philadelphia, 2013, pp. 1226–1242.

[27] R. Garg and S. Kapoor, *Auction algorithms for market equilibrium*, in ACM Symposium on the Theory of Computing, ACM, New York, 2004.

[28] S. Govindan and R. Wilson, *Equivalence and invariance of the index and degree of Nash equilibria*, Games Econom. Behav., 21 (1997), pp. 56–61.

[29] S. Govindan and R. Wilson, *A global Newton method to compute Nash equilibria*, J. Econom. Theory, 110 (2003), pp. 65–86.

[30] P. J. J. Herings and R. Peeters, *A differentiable homotopy to compute Nash equilibria of n-person games*, Econom. Theory, 18 (2001), pp. 159–185.

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

[32] J. Hofbauer, *Some Thoughts on Sustainable/Learnable Equilibria*, http://www.econ.uniurb.it/imgta/PlenaryLecture/Hofbauer.pdf (2003).

[33] K. Jain, *A polynomial time algorithm for computing an Arrow–Debreu market equilibrium for linear utilities*, SIAM J. Comput., 37 (2007), pp. 303–318.

[34] K. Jain, M. Mahdian, and A. Saberi, *Approximating market equilibrium*, in Approximation, Randomization, and Combinatorial Optimization: Algorithms and Techniques, Lecture Notes in Comput. Sci. 2764, Springer, Berlin, 2003, pp. 98–108.

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

[36] V. Klee and G. J. Minty, *How good is the simplex algorithm?*, in Inequalities, III, O. Shisha, ed., Academic Press, New York, 1972, pp. 159–175.

[37] D. Koller, N. Megiddo, and B. Von Stengel, *Efficient computation of equilibria for extensive two-person games*, Games Econom. Behav., 14 (1996), pp. 247–259.

[38] C. E. Lemke, *Bimatrix equilibrium points and mathematical programming*, Manage. Sci., 11 (1965), pp. 681–689.

[39] C. E. Lemke and J. T. Howson, Jr., *Equilibrium points of bimatrix games*, J. Soc. Ind. Appl. Math., 12 (1964), pp. 413–423.

[40] G. Maier, *A matrix structural theory of piecewise linear elastoplasticity with interacting yield planes*, Meccanica, 5 (1970), pp. 54–66.

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

[42] A. Mas-Colell, *The Theory of General Economic Equilibrium: A Differentiable Approach*, Econometric Soc. Monogr. 9, Cambridge University Press, Cambridge, 1985.

[43] R. R. Maxfield, *General equilibrium and the theory of directed graphs*, J. Math. Econom., 27 (1997), pp. 23–51.

[44] R. D. McKelvey and T. R. Palfrey, *Quantal response equilibria for normal form games*, Games Econom. Behav., 10 (1995), pp. 6–38.

[45] R. B. Myerson, *Sustainable equilibria in culturally familiar games*, in Understanding Strategic Interaction: Essays in Honor of Reinhard Selten, W. Albers and R. Selten, eds., Springer, Heidelberg, 1997, pp. 111–121.

[46] J. B. Orlin, *Improved algorithms for computing Fisher's market clearing prices*, in ACM Symposium on the Theory of Computing, ACM, New York, 2010, pp. 291–300.

[47] C. H. Papadimitriou, *On the complexity of the parity argument and other inefficient proofs of existence*, J. Comput. System Sci., 48 (1992), pp. 498–532.

[48] R. Savani, *Finding Nash Equilibria of Bimatrix Games*, Ph.D. thesis, The London School of Economics and Political Science, London, 2006.

[49] R. Savani and B. von Stengel, *Hard-to-solve bimatrix games*, Econometrica, 74 (2006), pp. 397–429.

[50] H. Scarf, *An elementary proof of a theorem on the core of an n person game*, in Herbert Scarf's Contributions to Economics, Game Theory, and Operations Research, Palgrave Macmillan, Basingstoke, England, 1965.

[51] H. Scarf, *The approximation of fixed points of a continuous mapping*, SIAM J. Appl. Math., 15 (1967), pp. 1328–1343.

[52] H. Scarf, *The Computation of Economic Equilibria*, Yale University Press, New Haven, CT, 1973.

[53] K. Schmedders, *A homotopy algorithm and an index theorem for the general equilibrium model with incomplete asset markets*, J. Math. Econom., 32 (1999), pp. 225–241.

[54] L. S. Shapley, *A note on the Lemke-Howson algorithm*, in Pivoting and Extension, Math. Program. Stud. 1, Springer, Berlin, 1974, pp. 175–189.

[55] J. B. Shoven and J. Whalley, *Applying General Equilibrium*, Cambridge University Press, Cambridge, 1992.

[56] D. A. Spielman and S.-H. Teng, *Smoothed analysis: Why the simplex algorithm usually takes polynomial time*, J. ACM, 51 (2004), pp. 385–463.

[57] M. J. Todd, *Orientation in complementary pivot algorithms*, Math. Oper. Res., 1 (1976), pp. 54–66.

[58] A. H. van den Elzen and A. J. J. Talman, *A procedure for finding Nash equilibria in bimatrix games*, ZOR–Methods Models Oper. Res., 35 (1991), pp. 27–43.

[59] V. V. Vazirani and M. Yannakakis, *Market equilibrium under separable, piecewise-linear, concave utilities*, J. ACM, 58 (2011), 10.

[60] 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, ACM, New York, 2012.

[61] A. von Schemde, *Index and Stability in Bimatrix Games: A Geometric-Combinatorial Approach*, Springer, Berlin, 2005.

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

[63] Y. Ye, *A path to the Arrow-Debreu competitive market equilibrium*, Math. Program., 111 (2008), pp. 315–348.