A simplex-like algorithm for linear Fisher markets†

Bharat Adsul[1], Ch. Sobhan Babu[2], Jugal Garg[1,]*, Ruta Mehta[1] and Milind Sohoni[1]

[1]Department of Computer Science and Engineering, Indian Institute of Technology, Bombay, Powai, Mumbai 400 076, India
[2]Department of Computer Science and Engineering, Indian Institute of Technology, Hyderabad 502 205, India

We propose a new convex optimization formulation for the Fisher market problem with linear utilities. Like the Eisenberg–Gale formulation, the set of feasible points is a polyhedral convex set while the cost function is nonlinear; however, the optimum is always attained at a vertex of this polytope. The convex cost function depends only on the initial endowments of the buyers. This formulation yields an easy, simplex-like algorithm which is provably strongly polynomial for many special cases. The algorithm can also be interpreted as a complementary pivot algorithm resembling the classical Lemke–Howson algorithm for computing Nash equilibrium of two-person bimatrix games.

Keywords: Convex optimization, Fisher market model, market equilibrium, simplex-like algorithm.

Introduction

FISHER[2] and Arrow–Debreu[3] market models are the two fundamental market models in mathematical economics. In this article, we focus on the Fisher market model with linear utilities. An instance of this model consists of a set of buyers, each with money endowment and (linear) utility functions, and a set of divisible goods. The problem is to determine market equilibrium prices and allocation of the goods to the buyers such that the market clears and each buyer gets her utility maximizing bundle subject to her budget constraints. Towards this, Eisenberg and Gale[4,5] formulated a remarkable convex optimization program whose optimal solution, more precisely, values of the primal and dual variables at an optimal solution, captures equilibrium allocation and prices respectively.

Many algorithmic results pertaining to the computation of market equilibrium prices and allocation for the linear case of Fisher and Arrow–Debreu market models have been obtained[6–9]. Deng *et al.*[6] gave a strongly polynomial time algorithm for the Fisher market with either constant number of goods or constant number of buyers. Building on the Eisenberg–Gale program, Devanur *et al.*[7] developed a primal–dual type polynomial time algorithm to solve the Fisher market model. For the more general Arrow–Debreu market, Eaves[10] gave a linear complementarity problem (LCP) formulation and showed that the Lemke's algorithm[11], when applied to this LCP, converges to a market equilibrium. Later, Jain[8] obtained a polynomial time algorithm for Arrow–Debreu market. More recently, Orlin[9] obtained a strongly polynomial time algorithm for the Fisher market. A tantalizing open question is to formulate a linear program that captures the Fisher solution. A positive resolution of this question would, of course, imply a simplex-like algorithm for computing the same. This article is an attempt towards this objective.

This article is the first in the series of articles on vertex marching algorithms for market equilibrium computation. All are complementary pivot algorithms resembling either Lemke or Lemke–Howson algorithm. The Lemke's scheme can be applied to any LCP, however, with no convergence guarantees. The Lemke–Howson[12] (LH) scheme was known only for computing a Nash equilibrium for two-person bimatrix games (2-Nash). Both Lemke and Lemke–Howson are complementary pivot schemes, but work in two different polytopes and possibly lead to different insights. Lemke's scheme expands the LCP polytope by going one dimension up; follows a path on its one-skeleton and stops whenever it hits the original LCP polytope (a section in the expanded polytope). In contrast, the LH scheme works on the LCP polytope itself; follows a path on its one-skeleton and stops when it hits a solution vertex. None of these schemes requires monotonicity-based argument for convergence.

In this article, we propose a novel convex optimization formulation for the Fisher market problem. In the Eisenberg–Gale formulation[5,6], the set of feasible points is a convex polytope which merely models the packing constraints and is oblivious to the parameters of the problem. Like the Eisenberg–Gale formulation, the set of feasible points in our formulation is also a convex polytope. However, our convex polytope is defined in terms of the input parameters, specifically utilities and money, and is rich enough so as to ensure that the optimum is always attained at a vertex of this polytope. Furthermore, the convex cost function in our formulation depends only on the initial endowments of the buyers. There is another

†A preliminary version of the paper was presented at SAGT 2010 (ref. 1).
*For correspondence. (e-mail: jugal@ces.iitb.ac.in)

formulation, which maximizes a convex function under flow constraints, obtained by Shmyrev[13] and Birnbaum *et al.*[14]. However this formulation also does not guarantee the optimum to be at a vertex.

We notice that the optimum vertex satisfies a set of complementarity conditions. Using these conditions, we define special vertices in our polytope and every such vertex corresponds to the Fisher solution with a different endowment vector. Next, we give a combinatorial characterization of special vertices and show that starting from any special vertex, there is a simplex-like path of special vertices where the cost function monotonically increases and ends at a vertex corresponding to the Fisher solution. There may be many such paths of special vertices in the polytope. Using a simple pivoting rule, we give an algorithm which traces one such path. Since all the vertices on the path followed by the algorithm satisfy all the complementarity conditions, the pivot in each step is complementary, and it resembles Lemke–Howson algorithm because the polytope is unchanged. The primary difference from Lemke–Howson is the monotonicity-based argument for convergence, which is why it is also simplex-like. Experiments indicate that our algorithm is practical – on randomly generated instances, the number of iterations it needs is linear in the total number of non-zero utilities specified in the input. Further, we show that this algorithm is strongly polynomial for many special cases. Two interesting cases are: (i) Either the number of buyers or the goods is fixed. (ii) All the non-zero utilities are of the type α^k, where $\alpha > 0$ and $0 \le k \le M$ (M is bounded by a polynomial in the number of buyers and goods).

This algorithm is conceptually simple, much easier to implement and runs very fast in practice. In fact, these special cases seem sufficient to handle most practical situations. This is because, first, in practice, utilities are hardly exactly known, and secondly, as shown in Adsul *et al.*[15] buyers have every reason to strategize and report fictitious utilities. The events that may occur in the algorithm, while finding the adjacent special vertex, are similar as in the DPSV algorithm[7]. However one crucial difference is that the prices which the DPSV algorithm computes at intermediate stages may not occur at a vertex in the polytope. Further, the utility of our formulation is also illustrated by its easy extension to incorporate transportation costs as well (see Appendix A). There seems no way to modify Eisenberg–Gale or Shmyrev formulations to capture the equilibrium solution for this extended model. Independently, Chakraborty *et al.*[16] also give a similar formulation for this extended model along with an algorithm to compute ε-approximate equilibrium prices and allocations. However, the Fisher market with transportation cost may have irrational solutions, so the optimum solution may not be at a vertex.

In a follow-up work[17], we consider an LCP-like formulation, using the complementarity conditions of this article, for the Arrow–Debreu model with linear utilities, and give a Lemke–Howson type algorithm on utility space while market clearing is always enforced. The algorithm starts with an easy-to-handle utilities and moves continuously to the original utilities, while keeping track of the solutions for the intermediate utilities. In this process it keeps expanding the polytope, and again convergence is proved using monotonicity-based argument. An adaptation of this algorithm to scaling gives an efficient algorithm for Fisher markets. After this, we consider Arrow–Debreu markets with more general utility function of separable piecewise-linear concave (SPLC), for which it was known that equilibria are rational but computing one is PPAD-complete[18], a complexity class defined by Papadimitriou[19]. After failed attempts of extending our previous algorithms, we considered Eaves[10] algorithm for the linear case. His LCP formulation and our LCP-like formulation, for capturing market equilibria for the linear case, turned out to be similar. We extended Eaves' formulation to capture market equilibrium for the SPLC case, and proved that Lemke's algorithm when applied to this LCP converges to an equilibrium[20].

Organization. The rest of the article is organized as follows. First we give a precise formulation of the Fisher market problem and introduce the new convex optimization program and analyse it. Then we discuss the simplex-like algorithm. Next we show that the algorithm is provably strongly polynomial for many special cases. The next section summarizes the experimental results on randomly generated instances. Finally we conclude the article.

New convex optimization formulation

We begin with a precise description of the Fisher market model.

Problem formulation

The input to the Fisher market problem is a set of buyers $\mathcal{B}$, a set of goods $\mathcal{G}$, a utility matrix $U = [u_{ij}]_{i \in \mathcal{B}, j \in \mathcal{G}}$, a quantity vector $\boldsymbol{q} = (q_j)_{j \in \mathcal{G}}$ and a money vector $\boldsymbol{m} = (m_i)_{i \in \mathcal{B}}$, where u_{ij} is the utility derived by buyer i from a unit amount of goods j, q_j is the quantity of goods j, and m_i is the money possessed by buyer i. Let $|\mathcal{B}| = m$ and $|\mathcal{G}| = n$. We assume that for every good j, there is a buyer i such that $u_{ij} > 0$ and for every buyer i, there is a good j such that $u_{ij} > 0$; otherwise we may discard those goods and buyers from the market.

The problem is to compute equilibrium prices $\boldsymbol{p} = [p_j]_{j \in \mathcal{G}}$ and allocations $X = [x_{ij}]_{i \in \mathcal{B}, j \in \mathcal{G}}$ such that they satisfy the following two constraints.

Market clearing: The demand equals the supply of each of the goods, i.e., $\forall j \in \mathcal{G}$, $\sum_{i \in \mathcal{B}} x_{ij} = q_j$ and $\forall i \in \mathcal{B}$, $\sum_{j \in \mathcal{G}} p_j x_{ij} = m_i$.

Table 1. The convex program

maximize	:	$\sum_{i\in\mathcal{B}} m_i \log y_i$	
subject to			
$\forall i \in \mathcal{B}, \forall j \in \mathcal{G}$	:	$u_{ij}y_i \le p_j,$	(1)
		$\sum_{j\in\mathcal{G}} p_j \le \sum_{i\in\mathcal{B}} m_i,$	(2)
$\forall i \in \mathcal{B}$	:	$y_i \ge 0,$	(3)
$\forall j \in \mathcal{G}$	:	$p_j \ge 0,$	(4)
$\forall i \in \mathcal{B}$	:	$\sum_{j\in\mathcal{G}} z_{ij} \le m_i,$	(5)
$\forall j \in \mathcal{G}$	:	$\sum_{i\in\mathcal{B}} z_{ij} = p_j,$	(6)
$\forall i \in \mathcal{B}, \forall j \in \mathcal{G}$	:	$z_{ij} \ge 0.$	(7)

Optimal goods: Every buyer buys only those goods, which give her the maximum utility per unit of money (*bang per buck*), i.e. if $x_{ij} > 0$ then $(u_{ij}/p_j) = \max_{k\in\mathcal{G}}(u_{ik}/p_k)$.

Note that, by scaling u_{ij}'s appropriately, we may assume that q_j's are unit.

Convex program

In this section, we introduce the new convex optimization program whose optimal solution captures the Fisher market equilibrium. Our convex program is described in Table 1, where p_j corresponds to the price of good j and z_{ij} corresponds to the money spent by buyer i on good j. At optimum, $(1/y_i)$ is the *bang per buck* of buyer i. We refer to the ambient space as the y–p–z space.

Note that the feasible set O is a convex polytope in y–p–z space and the cost function is independent of the variables z_{ij}. Let O_{aux} be the auxiliary polytope in the y–p space defined by the constraints given by eqs (1)–(4) and the related convex program (with the same cost function) be the auxiliary convex program. Let $Pr(O)$ be the projection of O onto the y–p space.

Claim. $Pr(O) = O_{\text{aux}}$.

Proof. Clearly, $Pr(O) \subseteq O_{\text{aux}}$, and for $O_{\text{aux}} \subseteq Pr(O)$, $Z = [z_{ij}]$ should be constructed for a given $(\boldsymbol{y}, \boldsymbol{p}) \in O_{\text{aux}}$. One way to do this is by constructing a max-flow network, where there is an edge from the source to every good $j \in \mathcal{G}$ with capacity p_j and from every buyer $i \in \mathcal{B}$ to the sink with capacity m_i. Further, there is an edge from every good $j \in \mathcal{G}$ to every buyer $i \in \mathcal{B}$ with ∞ capacity. Clearly, the max-flow gives the required z_{ij}'s. □

Therefore, in order to understand the optimality conditions, we may as well work with the Karush–Kuhn–Tucker (KKT) conditions for the auxiliary convex program. Let x_{ij}, q, μ_i, λ_j be the Lagrangian (dual) variables corresponding to the eqs (1)–(4). An optimal solution must satisfy the KKT conditions in Table 2.

Table 2. Karush–Kuhn–Tucker conditions

$\forall i \in \mathcal{B}$	:	$\frac{m_i}{y_i} = \sum_{j\in\mathcal{G}} u_{ij}x_{ij} - \mu_i,$	(8)
$\forall i \in \mathcal{B}, \forall j \in \mathcal{G}$	:	$(u_{ij}y_i - p_j)x_{ij} = 0,$	(9)
$\forall j \in \mathcal{G}$	:	$\sum_{i\in\mathcal{B}} x_{ij} - \lambda_j + q = 0,$	(10)
		$\left(\sum_{j\in\mathcal{G}} p_j - \sum_{i\in\mathcal{B}} m_i\right)q = 0,$	(11)
$\forall i \in \mathcal{B}, \forall j \in \mathcal{G}$	:	$x_{ij}, \lambda_j, \mu_i, q \ge 0,$	(12)
$\forall j \in \mathcal{G}$	:	$-p_j\lambda_j = 0,$	(13)
$\forall i \in \mathcal{B}$	:	$-y_i\mu_i = 0,$	(14)

Claim. *At any optimum,* $\mu_i = 0$, $\forall i \in \mathcal{B}$ *and* $\lambda_j = 0$, $\forall j \in \mathcal{G}$.

Proof. $\mu_i \ne 0 \Rightarrow y_i = 0 \Rightarrow$ the optimal solution has cost $-\infty$. However, we may easily construct a feasible point in the polytope, where the cost is some real value, therefore all μ_i's are zero. Similarly, $\lambda_j \ne 0 \Rightarrow p_j = 0 \Rightarrow y_i = 0$, for some $i \in \mathcal{B}$. Hence, all λ_j's are zero. □

Putting $\mu_i = 0$ and $\lambda_j = 0$ in the KKT conditions, i.e. eqs (8)–(12), we get

$\forall i \in \mathcal{B}$	:	$m_i = \sum_{j\in\mathcal{G}} u_{ij}x_{ij}y_i,$	(15)
$\forall i \in \mathcal{B}, \forall j \in \mathcal{G}$	:	$(u_{ij}y_i - p_j)x_{ij} = 0,$	(16)
$\forall j \in \mathcal{G}$	:	$\sum_{i\in\mathcal{B}} x_{ij} = q,$	(17)
	:	$\left(\sum_{j\in\mathcal{G}} p_j - \sum_{i\in\mathcal{B}} m_i\right)q = 0,$	(18)
$\forall i \in \mathcal{B}, \forall j \in \mathcal{G}$	:	$x_{ij}, q \ge 0,$	

From eqs (15)–(18),

$$\sum_{i\in\mathcal{B}} m_i = \sum_{i\in\mathcal{B}}\sum_{j\in\mathcal{G}} p_j x_{ij} = \sum_{j\in\mathcal{G}}\sum_{i\in\mathcal{B}} p_j x_{ij} = \sum_{j\in\mathcal{G}} p_j q \Rightarrow q = 1.$$

Proposition 1. *Let* $(\boldsymbol{y}, \boldsymbol{p}) \in O_{\text{aux}}$ *be an optimal solution to the auxiliary convex program. Then* $\boldsymbol{p}$ *is a market equilibrium price vector.*

Proof. As $q = 1$, interpreting $X = [x_{ij}]$ as an allocation, we see that the conditions given by eqs (15)–(17) imply that the market clearing constraint holds at the price vector $\boldsymbol{p}$. Further, using eq. (2), we have $x_{ij} > 0 \Rightarrow y_iu_{ij} = p_j$. As $(\boldsymbol{y}, \boldsymbol{p}) \in O_{\text{aux}}$, we also have, $\forall i \in \mathcal{B}, \forall j \in \mathcal{G} : u_{ij}y_i \le p_j$. Putting these two together, it is easily verified that the optimal goods constraint is also satisfied. □

Proposition 2.
(i) *The auxiliary convex program admits a unique optimal solution.*
(ii) *Equilibrium prices are unique and allocations form a polyhedral set.*

Proof. Part (i) follows from the fact that the cost function is strictly concave, and part (ii) follows from the KKT conditions. □

Let $(\boldsymbol{y}, \boldsymbol{p}) \in O_{\text{aux}}$ be the unique optimum solution to the auxiliary convex program. Let $\mathbb{X} = \{X = [x_{ij}]_{i \in \mathcal{B}, j \in \mathcal{G}} | (\boldsymbol{y}, \boldsymbol{p}, X)$ satisfy eqs (8)–(14)$\}$. Note that $\mathbb{X}$ is a convex set. As argued in the proof of Proposition 1, we may think of $X \in \mathbb{X}$ as an equilibrium allocation and $\boldsymbol{p}$ as the equilibrium price. Now, we define $Z = [z_{ij}]$ w.r.t. $X \in \mathbb{X}$ as $z_{ij} = x_{ij}p_j$, $\forall i \in \mathcal{B}$, $\forall j \in \mathcal{G}$. In other words, z_{ij} is the money spent by buyer i on good j at the equilibrium allocation X. We refer to Z as an equilibrium money allocation. It easily follows that $(\boldsymbol{y}, \boldsymbol{p}, Z)$ is an optimum solution to the main convex program. Note that there is an $X^a \in \mathbb{X}$ such that the bipartite graph $\mathcal{G} = (\mathcal{B}, \mathcal{G}, E)$, where $E = \{(i, j) \in \mathcal{B} \times \mathcal{G} | x^a_{ij} > 0\}$, is acyclic. Let Z^a be the equilibrium money allocation w.r.t. X^a. The following corollary follows from the KKT condition, i.e. eq. (16).

Corollary 3. *At the optimum solution* $(\boldsymbol{y}, \boldsymbol{p}, Z^a)$, $z^a_{ij}(u_{ij}y_i - p_j) = 0$, $\forall (i, j) \in (\mathcal{B} \times \mathcal{G})$.

The conditions in Corollary 3 are called complementarity conditions, as mentioned in the Introduction. It says that for every pair of $(i, j) \in (\mathcal{B} \times \mathcal{G})$, either $z_{ij} = 0$ or $u_{ij}y_i - p_j = 0$. In other words, when $z_{ij} > 0$, $u_{ij}y_i - p_j = 0$. The next proposition asserts that $(\boldsymbol{y}, \boldsymbol{p}, Z^a)$ is in fact a vertex of O.

Proposition 4. *The point* $(\boldsymbol{y}, \boldsymbol{p}, Z^a)$ *is a vertex of* O.

Proof. Consider the bipartite graph $G = (\mathcal{B}, \mathcal{G}, E)$, where $E = \{(i, j) \in \mathcal{B} \times \mathcal{G} | z^a_{ij} > 0\}$. Note that G is acyclic, hence it is a forest. Let k be the number of edges in E, and each edge gives a tight constraint of eq. (1), i.e. $u_{ij}y_i - p_j = 0$. Since $(\boldsymbol{y}, \boldsymbol{p}, Z^a)$ is an optimum point, all y_i's and p_j's are non-zero.

There are $mn + m + n$ variables in the convex program, so at a vertex in O, at least $mn + m + n$ linearly independent (*l.i.*) constraints must be tight. Hence, we need to show $mn + m + n$ *l.i.* tight constraints in O at $(\boldsymbol{y}, \boldsymbol{p}, Z^a)$. We show that k constraints of eq. (1), m constraints of eq. (5), n constraints of eq. (6) and $mn - k$ constraints of eq. (7) are tight, resulting in total $mn + m + n$ constraints and we show that they all are *l.i.*, hence proving the proposition.

Since $mn - k$ z_{ij}'s are zero at $(\boldsymbol{y}, \boldsymbol{p}, Z^a)$, we have $mn - k$ *l.i.* tight constraints of eq. (7). Next, we eliminate all the z_{ij}'s, which are zero, from the constraints, i.e. eqs (5) and (6). It is easy to see that we have k constraints of eq. (1), m constraints of eq. (5) and n constraints of eq. (6), which are tight. The only thing remaining is that they are all *l.i.* The table below describes all the $m + n + k$ tight constraints. There are $m + n + k$ columns and last k columns correspond to the k non-zero z_{ij}'s. We show all the columns are *l.i.*, i.e. there is no linear relation among them.

	y_1	y_2	…	y_m	p_1	p_2	…	p_n	$z_{i_1j_1}$	$z_{i_2j_2}$	…	$z_{i_kj_k}$
R_1	u_{11}				-1							
	u_{12}					-1						
		u_{21}			-1							
			⋮				⋮					
R_2					-1				1		…	
						-1				1	…	
							⋮				⋮	
R_3									1		…	
										1	…	
											⋮	

If there is a linear relation (*lr*) among these columns, then clearly it has to contain some columns of y_i's, some columns of p_j's and some columns of z_{ij}'s. Let column y_k be in *lr* and α_k be its coefficient. Now, consider the connected component C of G, which contains buyer k. To kill the coordinate of y_k in *lr*, the columns corresponding to all the buyers (y_i's), and goods (p_j's) in C have to be present and the coefficients are positive multiple of α_k. These positive multiples are some fraction of product of utilities. This results in a vector v, which has zero coordinates in R_1 as shown in the above table.

Next, to kill the coordinates of R_2 in v, we get l linear equations in the coefficients α_e of z_e's ($e \in C$), where l is the number of goods in C. Each α_e appears exactly once in one of the linear equations. The sum of all these linear equations is $\sum_{e \in C} \alpha_e = c \cdot \alpha_k$, where c is some positive constant. By choosing any α_e's such that all the l linear equations are satisfied will kill the coordinates of R_2 in v.

Now, to kill the coordinates of R_3 in v, we get r linear equations in the coefficients α_e of z_e's ($e \in C$), where r is the number of buyers in C. Each α_e appears exactly once in one of the linear equations. The sum of all these r linear equations is $\sum_{e \in C} \alpha_e = 0$. This implies $\alpha_k = 0$, and hence the contradiction. □

Remark 5. The auxiliary program itself captures the equilibrium prices at the optimal solution, though not necessarily at one of its vertices. Since there are $m + n$ variables in O_{aux}, every vertex requires $m + n - 1$ tight inequalities of eq. (1) (one comes from eq. (2)). Therefore, only those instances for which the bipartite graph G (at optimum), as defined above, is a single connected component can be at a vertex of O_{aux}.

A simplex-like algorithm

We begin with some notation. Henceforth, we denote the input to the Fisher market problem by $(U, \boldsymbol{m})$. The set of buyers and the set of goods are implicit. We use g_j and b_i to denote the good j and buyer i respectively. For convenience, we assume that $u_{ij} > 0$, $\forall i \in \mathcal{B}$, $\forall j \in \mathcal{G}$.

Now, we turn our attention to the polytope O defined in the previous section. We have shown that there exists a vertex $v = (\boldsymbol{y}, \boldsymbol{p}, Z)$ of the polytope O which captures the equilibrium prices and an equilibrium money allocation. An important property of v is that it satisfies the complementarity conditions $\forall i \in \mathcal{B}$, $\forall j \in \mathcal{G}$, $z_{ij}(u_{ij}y_i - p_j) = 0$. In other words, every buyer spends money only on her optimal goods.

Definition 6. *A vertex $v = (\boldsymbol{y}, \boldsymbol{p}, Z)$ of O is called **special** if $z_{ij}(u_{ij}y_i - p_j) = 0$, $\forall i \in \mathcal{B}$, $\forall j \in \mathcal{G}$.*

It is easy to see that if $v = (\boldsymbol{y}, \boldsymbol{p}, Z)$ is a special vertex, then it corresponds to a solution for an instance of the Fisher market problem. Namely, let $\mathcal{B}' = \{i \in \mathcal{B} | y_i \neq 0\}$, $\mathcal{G}' = \mathcal{G}$ and U' be U restricted to $\mathcal{B}' \times \mathcal{G}'$. Further, for $i \in \mathcal{B}'$, let $m'_i = \sum_{j \in \mathcal{G}} z_{ij}$. Clearly, v corresponds to a solution of $(U', \boldsymbol{m}')$. Our goal is to find a special vertex for which $\boldsymbol{m}' = \boldsymbol{m}$.

Characterization of special vertices

Let $v = (\boldsymbol{y}, \boldsymbol{p}, Z)$ be a special vertex of O. Without loss of generality, we may assume that all y_i's and all p_j's are non-zero at v, because if $p_j = 0$ for some $j \in \mathcal{G}$ at v, then v is a trivial point, i.e. all coordinates are zero, and if $y_k = 0$ for some $k \in \mathcal{B}$ at v, then there is an adjacent vertex $v' = (\boldsymbol{y}', \boldsymbol{p}', Z')$ to v, where $\boldsymbol{p}' = p$, $Z' = Z$, $y'_i = y_i$, $\forall i \neq k$, and $y'_k = \min_{j \in \mathcal{G}}(p_j/u_{ij})$.

Now we describe a combinatorial characterization of v. Towards this, we define $E(v)$ and $F(v)$ as follows

$$E(v) = \{(i, j) \in \mathcal{B} \times \mathcal{G} \mid u_{ij}y_i = p_j\} \text{ and}$$

$$F(v) = \{(i, j) \in \mathcal{B} \times \mathcal{G} \mid z_{ij} > 0\}.$$

The elements in $E(v)$ and $F(v)$ are called *tight* and *non-zero* edges respectively. By definition, $F(v) \subseteq E(v)$. Let $G(E(v), F(v))$ be the graph, whose vertices are the connected components $C_1, C_2, \ldots$ of the bipartite graph $(\mathcal{B}, \mathcal{G}, F(v))$, and there is an edge between C_i and C_j in $G(E(v), F(v))$, if there is at least one edge in $E(v) - F(v)$ between the corresponding components of $(\mathcal{B}, \mathcal{G}, F(v))$.

We say that buyer i belongs to a vertex C of $G(E(v), F(v))$, if buyer i lies in the corresponding component of $(\mathcal{B}, \mathcal{G}, F(v))$. We call a connected component of G as simply a component of G.

Definition 7. *W.r.t. $v = (\boldsymbol{y}, \boldsymbol{p}, Z)$,*
- ***surplus** of buyer i is defined to be the non-negative value $m_i - \sum_{j \in \mathcal{G}} z_{ij}$.*
- *a buyer is called a **zero surplus** buyer if his surplus is zero; otherwise he is called a **positive surplus** buyer.*
- *a component of $(\mathcal{B}, \mathcal{G}, F(v))$ is called **saturated** if all buyers in that component are zero surplus buyers; otherwise it is called **unsaturated**.*
- *a vertex of $G(E(v), F(v))$ is called **saturated** if the corresponding component of $(\mathcal{B}, \mathcal{G}, F(v))$ is saturated; otherwise it is called **unsaturated**.*

Theorem 8. *v has following properties:*
- *Every component of $(\mathcal{B}, \mathcal{G}, F(v))$ contains at most one positive surplus buyer.*
- *Every component of $G(E(v), F(v))$ has at least one saturated vertex.*

Proof. If a component of $(\mathcal{B}, \mathcal{G}, F(v))$ contains more than one positive surplus buyer, then we can perturb z_{ij}'s on the path connecting the two buyers while maintaining the same set of inequalities tight. In that case, v is not a vertex.

Similarly, if a component of $G(E(v), F(v))$ does not have a saturated vertex, then the p_j's in that component may be scaled uniformly such that the same set of inequalities is tight before and after the scaling, hence a contradiction. □

Corollary 9. *If $(U, \boldsymbol{m})$ are algebraically independent (i.e. no algebraic relation among them), then*
- *the bipartite graph $(\mathcal{B}, \mathcal{G}, E(v))$ is a forest. Hence there is at most one edge in $E(v) - F(v)$ between any two components of $(\mathcal{B}, \mathcal{G}, F(v))$.*
- *every component of $G(E(v), F(v))$ has exactly one saturated vertex.*

Lemma 10. *Let v be a special vertex of O. Then*

(i) $(\mathcal{B}, \mathcal{G}, F(v))$ is acyclic.
(ii) If $(U, \boldsymbol{m})$ are algebraically independent, then $(\mathcal{B}, \mathcal{G}, E(v))$ is acyclic and the number of positive surplus buyers is $|E(v) - F(v)|$.

Proof. Since v is a vertex of O, therefore $(\mathcal{B}, \mathcal{G}, F(v))$ is acyclic. Part (ii) follows from Theorem 8 and Corollary 9. □

In general, a simplex-like pivoting algorithm moves from a vertex to an adjacent vertex such that the cost function increases. Therefore, we first describe the adjacent vertex procedure which does the same for the main convex program.

Adjacent vertex procedure

For convenience, we assume that $(U, \boldsymbol{m})$ are algebraically independent. The adjacent vertex procedure, given in

Table 3, can be easily extended for the general $(U, \boldsymbol{m})$ (see Appendix B). The procedure takes a special vertex v and outputs another special vertex v' adjacent to v, such that the cost function increases. If v is optimum, then it outputs $v' = v$.

Otherwise, there is a component C of $G(E(v), F(v))$ containing an unsaturated vertex. Clearly C is a tree and there is exactly one saturated vertex, say C_s, in C (Corollary 9). We consider C as the rooted tree with root C_s. We pick an edge e between C_s and an unsaturated vertex, say C_u, in C. Let (b_i, g_j) be the edge in $E(v) - F(v)$ corresponding to e. Note that, at vertex v both $z_{ij} \geq 0$ and $u_{ij}y_i - p_j \leq 0$ are tight. There are two cases depending on where b_i belongs: C_s (Case 1) or C_u (Case 2). In both the cases the basic idea for the pivot is to increase prices of goods in C_u so that more money of unsaturated buyers is spent, and thereby C_u moves towards saturation. This in turn also increases the cost.

Case 1: We get a new vertex v', adjacent to v in O, by relaxing the inequality $u_{ij}y_i = p_j$ at v. Let T_u be the subtree of C rooted at C_u and J_u be the set of goods in the components of $(\mathcal{B}, \mathcal{G}, F(v))$ corresponding to the vertices of T_u. v' may also be obtained by increasing the prices of the goods in J_u uniformly and by modifying y_i's and z_{ij}'s accordingly till a new inequality becomes tight. Table 4 lists the three possible cases for the new inequality.

Case 2: We get a new vertex v', adjacent to v in O, by relaxing the inequality $z_{ij} = 0$ at v. Let J be the set of goods in the components of $(\mathcal{B}, \mathcal{G}, F(v))$ corresponding to the vertices of C. v' may also be obtained by increasing the prices of the goods in J uniformly and by modifying the y_i's and z_{ij}'s accordingly till a new inequality becomes tight (Table 4).

Table 3. Adjacent vertex procedure

```
Adjacent vertex (v)
  v′ ← v;
  if v is optimum then
   └ return v′;
  C ← component of G(E(v), F(v)) containing an unsaturated vertex;
  C_s ← saturated vertex in C;
  C_u ← unsaturated vertex, adjacent to C_s, in C;
  e ← edge between C_s and C_u;
  (b_i, g_j) ← edge in E(v) − F(v) corresponding to e;
  if (b_i, g_j) is from C_s to C_u then
   └ v′ ← adjacent vertex obtained by relaxing u_ij y_i ≤ p_j;
  else v′ ← adjacent vertex obtained by relaxing z_ij ≥ 0;
  return v′;
```

Table 4. Different cases for the new tight inequality

1. A non-zero edge (b_k, g_l) becomes zero, i.e. $z_{kl} \geq 0$ becomes tight.
2. A non-tight edge (b_k, g_l) becomes tight, i.e. $u_{kl}y_k \leq p_l$ becomes tight.
3. An unsaturated vertex in C becomes saturated, i.e. $\sum_{l \in \mathcal{G}} z_{kl} \leq m_k$ becomes tight, where buyer k is a positive surplus buyer w.r.t. v.

Both the cases result in the new vertex v' adjacent to v in O, where $\boldsymbol{p}$ as well as $\boldsymbol{y}$ increase monotonically and $\sum_{j \in \mathcal{G}} p_j$ as well as $\sum_{i \in \mathcal{B}} y_i$ increase strictly going from v to v'. Hence the cost function value increases strictly going from v to v'. Note that v' is also a special vertex of O. From the above discussion, the following lemma is straightforward.

Lemma 11. *If a special vertex v is not optimum, then there exists an adjacent special vertex v' such that the value of cost function is more at v' than v.*

The algorithm

There may be many simplex-like paths in O to reach at the optimum vertex using different pivoting rules. Algorithm 1 traces a particular simplex-like path in O, where the pivoting rule is such that there is at most one buyer with a positive surplus at every vertex on the path. In this algorithm, we do not consider the components, which contain only a single buyer.

Algorithm 1. A simplex-like pivoting algorithm

```
U′ ← ⟨u_11,..., u_1n⟩; m′ ← ⟨m_1⟩;
v ← special vertex corresponds to the solution of (U′, m′);
i ← 2;
while i ≤ m do
  /* Note that the inequality y_i ≥ 0 is tight at v */
  v ← vertex adjacent to v obtained by relaxing y_i ≥ 0;
  while surplus of buyer i w.r.t. v is non-zero do
   └ v ← Adjacent vertex (v);
  i ← i + 1;
```

There are two types of iterations of the inner while loop, one in which we relax the inequality $z_{kl} \geq 0$ (type 1) and the other in which we relax the inequality $u_{kl}y_k \leq p_l$ (type 2) for some (b_k, g_l). Consider the inner while loop for buyer i and let v be the current special vertex. The component C of $G(E(v), F(v))$ containing buyer i has exactly two vertices, one saturated (C_s) and one unsaturated (C_u), and an edge (b_k, g_l) between them. Note that buyer i belongs to C_u, and $z_{kl} = 0$ and $u_{kl}y_k - p_l = 0$ at v. Now, consider the tree T in $(\mathcal{B}, \mathcal{G}, F(v))$ rooted at buyer i. The edges are directed downwards, i.e. away from the root. We increase the prices of the goods uniformly in T in order to decrease the surplus of buyer i. This increases the flow on the edges, which are from a buyer to a good (forward) and decreases the flow on the edges, which are from a good to a buyer (backward).

Therefore, when (b_k, g_l) be such that $g_l \in C_u$ and $b_k \in C_s$, we need to relax $u_{kl}y_k \leq p_l$, and when $g_l \in C_s$ and $b_k \in C_u$, we need to relax $z_{kl} \geq 0$ in order to increase the prices. It is also clear that during the price increase, only backward edges may be deleted. Moreover, since the prices of the goods in T increase, buyers in T may become interested in the goods outside T, and it implies that only forward edges may be added. Note that in each of the two cases the pivoting turns out to be the complementary pivot, i.e. relax $z_{kl} = 0$ if $u_{kl}y_k - p_l = 0$ just became tight at v, else relax $z_{kl} = 0$, resembling Lemke–Howson scheme.

Remark 12. Algorithm 1 provides a polyhedral interpretation to a sequential run of the so-called basic algorithm in ref. 7, where buyers are added one at a time.

Running time analysis

In this section, we describe the time analysis of Algorithm 1 and show that it is strongly polynomial for many special cases.

Lemma 13. *Algorithm 1 takes at most* $(m + n * 2^{m+n})$ *iterations.*

Proof. Consider the iterations of type 2 of the inner while loop, where we relax the tight inequality $u_{kl}y_k \le p_l$ for some (b_k, g_l). Let C_s^j be the component containing buyer k in the jth such iteration. Note that C_s^j is a saturated component. Let B^j be the set of buyers and G^j be the set of goods in C_s^j, and $S^j = B^j \cup G^j$. Since prices monotonically increase, therefore all S^j's are distinct. The total number of distinct S^j's is clearly bounded by 2^{m+n}, and in every n iterations of the inner while loop, one iteration has to be of type 2, therefore the number of iterations of the algorithm is bounded by $(m + n * 2^{m+n})$. □

Remark 14. A more refined bound is 2^{m+n+1}.

Theorem 15. *Algorithm 1 is strongly polynomial when either the number of buyers or goods is constant.*

Proof. Without loss of generality, we assume that $(U, \boldsymbol{m})$ are algebraically independent. For the general $(U, \boldsymbol{m})$, a similar proof may be worked out.

It is enough to show that the inner while loop takes a strongly polynomial number of iterations for every buyer i. Let C^j be the component of $G(E(v), F(v))$, which contains buyer i in the jth iteration of the inner while loop for buyer i. If surplus of buyer i is not zero, then C^j contains exactly one saturated vertex, say C_s^j, and one unsaturated vertex, say C_u^j. Note that buyer i belongs to C_u^j.

Let (b_k, g_l) be the edge between C_u^j and C_s^j, and P_j be the path starting from buyer i and ending with the edge (b_k, g_l) in $(\mathcal{B}, \mathcal{G}, E(v))$.

Claim. *All P_j's are distinct.*

Proof. Recall that when the edge (b_k, g_l) is such that buyer k belongs to C_s^j, we relax the inequality $u_{kl}y_k \le p_l$, and when buyer k belongs to C_u^j, we relax the inequality $z_{kl} \ge 0$. In other words, we add the edge (b_k, g_l) when buyer k belongs to C_u^j and delete it when buyer k belongs to C_s^j.

We show that all P_j's, which end in a good, are distinct, and a similar argument may be worked out for the case when they end in a buyer. A path P_j may repeat only when the last edge, say e, is deleted and added again, and this is possible only if some other edge more near to buyer i than e in P_j is deleted. The induction on the length of P_j proves the claim, because the edges between buyer i and the goods never break (buyer i always lies in C_u^j). □

Since the length of any P_j is at most $2 * \min(m, n)$, therefore it is a constant when either m or n is constant. Hence the total number of distinct P_j's is bounded by a polynomial in either m (if n is constant) or n (if m is constant). Hence the length of the simplex-like path in the Algorithm 1 is strongly polynomial when either the number of buyers or goods is constant. □

Theorem 16. *Algorithm 1 is strongly polynomial when* $\forall i \in \mathcal{B}$, $\forall j \in \mathcal{G}$, $u_{ij} = \alpha^{k_{ij}}$, *where* $0 \le k_{ij} \le \text{poly}(m, n)$ *and* $\alpha > 0$.

Proof. We only need to show that for every buyer i, the inner while loop takes a strongly polynomial number of iterations. Consider the iterations of inner while loop for a buyer a. We monitor the values of y_a/p_b, $\forall b \in \mathcal{G}$. Note that y_a/p_b for a good b remains same until both buyer a and good b are in the same component of $G(E(v), F(v))$, otherwise it strictly increases. Let C^j be the component of $G(E(v), F(v))$, which contains buyer a in the jth iteration. If surplus of buyer a is not zero, then C^j contains exactly one saturated vertex, say C_s^j, and one unsaturated vertex, say C_u^j. Note that buyer a belongs to C_u^j.

Let (b_k, g_l) be the edge between C_u^j and C_s^j. There are two types of iterations, one in which we relax the inequality $z_{kl} \ge 0$ (type 1) and the other in which we relax the inequality $u_{kl}y_k \le p_l$ (type 2). Let $z_{kl} \ge 0$ is relaxed in the jth iteration, and $b_a, g_{j_1}, b_{i_1}, \ldots, g_{j_k}, b_k, g_l$ be the path from b_a to g_l in C^j. Clearly,

$$\frac{y_a}{p_l} = \frac{u_{i_1 j_1} \ldots u_{k j_k}}{u_{a j_1} \ldots u_{i_{k-1} j_k} u_{kl}}$$

(using the tight inequalities $u_{ij}y_i \le p_j$), and the value of y_a/p_l strictly increases when iteration of type 1 occurs. Now, we consider the values of $\log_\alpha (y_a/p_j)$, $\forall j \in \mathcal{G}$. Clearly, these values monotonically increase when an iteration of type 1 occurs. Since for every $j \in \mathcal{G}$, the value of $\log_\alpha (y_a/p_j)$ might be at most $n * \text{poly}(m, n)$, therefore for every buyer i, the number of iterations of the inner while loop is bounded by $n^2 * \text{poly}(m, n)$. □

Table 5. Number of pivoting steps taken by Algorithm 1

No. of buyers/goods	4	8	12	16	20	24
Minimum iteration	6	31	84	136	245	223
Mean iteration	12.5	50.9	113.1	186.9	279.8	408.9
Maximum iteration	24	80	168	235	320	514

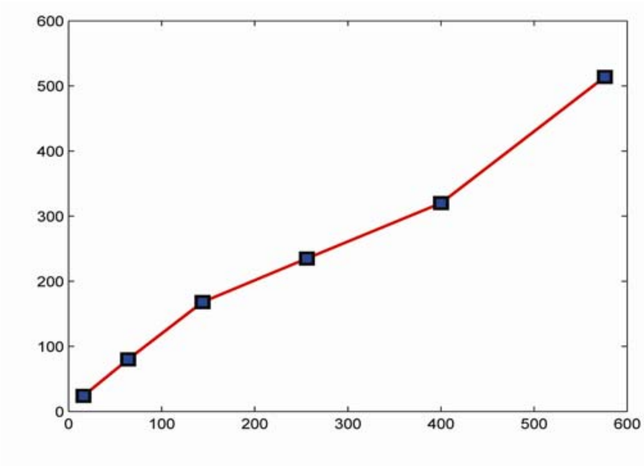

Figure 1. Number of buyers × number of goods versus maximum iteration.

Theorem 16 may be easily generalized to handle the case when some u_{ij}'s are zero. Many easy cases like all utilities are 0/1, non-zero utilities form a tree, etc. may also be easily shown to be strongly polynomial in Algorithm 1.

Experimental results

In this section, we report the experimental results of Algorithm 1. We ran Algorithm 1 on random instances of the Fisher market (i.e. $(U, \boldsymbol{m})$ are generated uniformly at random), while keeping the number of buyers and goods the same (i.e. $m = n$). For each value of $m \in \{4, 8, 12, 16, 20, 24\}$, we ran 100 experiments. Table 5 summarizes the results in terms of the minimum (best), maximum (worst) and mean (average) number of pivoting steps taken by Algorithm 1.

Figure 1 plots the number of buyers × the number of goods versus maximum iteration for a comparative analysis. Clearly, the number of steps seem to increase linearly with the size of instances. Even the worst case instance for each value of m (no. of buyers) requires fewer than $2m^2$ steps. Therefore, Algorithm 1 should have a much better bound.

Conclusion

We have presented a novel convex optimization formulation for the Fisher market problem whose feasible set is a polytope, and it is guaranteed that there is a vertex of this polytope which is an optimal solution. Exploiting this, we have developed a simplex-like vertex-marching algorithm which runs in strongly polynomial time for many special cases.

We have also elaborated on the connection of our algorithm with an LCP formulation and the Lemke/Lemke–Howson scheme. We have argued that perhaps this formulation is more rewarding, both practically and conceptually. The variables y_i's are the key to the formulation and have led to a generalization to Arrow–Debreu markets as well.

We feel that the strongly polynomial algorithm by Orlin[9] is neither polytopal nor very intuitive. The algorithms, which are polytopal and simplex-like are generally easier to understand, simpler to implement using standard math libraries, and run faster in practice. Therefore, an obvious open problem is to give a strongly polynomial, simplex-like algorithm; even a polynomial bound will be interesting. Another open problem is to give a linear programming formulation that captures the equilibrium prices for the Fisher market. Therefore, it will be interesting to construct a linear cost function on our polytope so that optimum vertex gives the equilibrium prices.

Appendix A. Fisher market with transportation costs

We generalize the Fisher market model to incorporate transportation costs as well, where in addition to all the inputs, a transportation cost matrix $C = [c_{ij}]$, where c_{ij} is the cost of transporting unit amount of goods j to buyer i, is also given. Independently, Chakraborty *et al.*[16] also give a similar formulation for this generalization along with an algorithm to compute ε-approximate equilibrium.

Note that, by scaling u_{ij}'s and c_{ij}'s appropriately, we may assume that q_j's are unit. The problem is again to find the equilibrium prices ($\boldsymbol{p} = (p_j)_{j\in\mathcal{G}}$) and allocation ($X = [x_{ij}]_{i\in\mathcal{B}, j\in\mathcal{G}}$), such that the market clears and every buyer receives a maximum utility bundle subject to her budget constraint. Note that c_{ij}'s also have to be taken into account in this model in the calculation of optimal goods. Formally, the equilibrium prices and allocation must satisfy the following two conditions.

Market clearing: There is neither deficiency nor surplus of any good and the money of all the buyers is exhausted, i.e. $\forall j \in \mathcal{G}, \sum_{i\in\mathcal{B}} x_{ij} = 1$ and $\forall i \in \mathcal{B}, \sum_{j\in\mathcal{G}} x_{ij}(p_j + c_{ij}) = m_i$.

Optimal goods: Every buyer buys only those goods which give her the maximum utility per unit of money, i.e. if $x_{ij} > 0$ then $(u_{ij}/(p_j + c_{ij})) = \max_{k\in\mathcal{G}}(u_{ik}/(p_k + c_{ik}))$.

The optimal solution of the Eisenberg–Gale convex program[4] captures equilibrium solution for the Fisher market, in turn also ensures existence and uniqueness of the

Table 6. Convex program

maximize	:	$\sum_{i\in\mathcal{B}} m_i \log y_i - e,$	
subject to			
$\forall i \in \mathcal{B}, \forall j \in \mathcal{G}$	:	$u_{ij}y_i \le p_j + c_{ij},$	(19)
		$\sum_{j\in\mathcal{G}} p_j \le e,$	(20)
$\forall i \in \mathcal{B}$	:	$y_i \ge 0,$	(21)
$\forall j \in \mathcal{G}$	:	$p_j \ge 0,$	(22)
	:	$e \ge 0.$	(23)

Table 7. KKT conditions

$\forall i \in \mathcal{B}$	:	$\frac{m_i}{y_i} = \sum_{j\in\mathcal{G}} u_{ij}x_{ij} - \mu_i,$	(24)
$\forall i \in \mathcal{B}, \forall j \in \mathcal{G}$	:	$(u_{ij}y_i - p_j - c_{ij})x_{ij} = 0,$	(25)
$\forall j \in \mathcal{G}$	:	$-p_j\lambda_j = 0,$	(26)
	:	$-e\gamma = 0,$	(27)
$\forall i \in \mathcal{B}, \forall j \in \mathcal{G}$	:	$-1 = -q - \gamma,$	(28)
$\forall j \in \mathcal{G}$	:	$-\sum_{i\in\mathcal{B}} x_{ij} - \lambda_j + q = 0,$	(29)
		$\left(\sum_{j\in\mathcal{G}} p_j - e\right) q = 0,$	(30)
$\forall i \in \mathcal{B}$	:	$-y_i\mu_i = 0,$	(31)
$\forall i \in \mathcal{B}, \forall j \in \mathcal{G}$	:	$x_{ij}, \lambda_j, \mu_i, q, \gamma \ge 0.$	(32)

equilibrium prices. However, it does not seem possible to extend the Eisenberg–Gale convex program to capture equilibrium solution for the Fisher market with transportation costs.

Interestingly, the optimal solution of the convex formulation, given in Table 6, captures the equilibrium prices (if they exist) for this model, and it is a simple extension of the convex program in Table 6. It also gives the conditions for the existence of equilibrium solution and proves the uniqueness of the equilibrium prices.

Let x_{ij}, q, μ_i, λ_j, γ be the Lagrangian (dual) variables corresponding to the equations (19)–(23). As before, p_j is interpreted as the price of good j, and $[x_{ij}]$ may be interpreted as the allocation. An optimal solution must satisfy the KKT conditions given in Table 7.

At optimum, clearly $y_i > 0$, $\forall i \in \mathcal{B} \Rightarrow \mu_i = 0$, $\forall i \in \mathcal{B}$. Putting $\mu_i = 0$ in eq. (24) and from eqs (24) and (25), we get

$$\forall i \in \mathcal{B} : \sum_{j\in\mathcal{G}} x_{ij}(p_j + c_{ij}) = m_i,$$

$$\forall i \in \mathcal{B}, \forall j \in \mathcal{G} : x_{ij} > 0 \Rightarrow \frac{u_{ij}}{p_j + c_{ij}} = \frac{\sum_{j\in\mathcal{G}} u_{ij}x_{ij}}{m_i}.$$

The above two conditions say that at optimum, all buyers exhaust their money and they buy only the optimal goods.

Further, from eq. (20) and eqs (26)–(29), we get

$$\forall j \in \mathcal{G}:\ p_j > 0 \Rightarrow \lambda_j = 0,\ e > 0 \Rightarrow \gamma = 0$$
$$\Rightarrow q = 1 \Rightarrow \textstyle\sum_{i\in\mathcal{B}} x_{ij} = 1.$$

The above condition says that if the price of a good is non-zero at optimum, then that good is completely sold.

Now, it is easy to check that any equilibrium solution of the Fisher market with transportation costs will satisfy all the conditions, i.e. eqs (19)–(32) by taking appropriate values of the remaining variables; hence it must be the optimum solution of the convex program. Since the objective cost function is strictly concave, therefore equilibrium prices are unique (if they exist). The equilibrium does not exist when the price of some good is zero and it is not completely sold at the optimum.

Appendix B. Adjacent vertex procedure for general (U, m)

Table 8. Adjacent vertex procedure

```
Adjacent vertex (v)
  v' ← v;
  if v is optimum then
    └ return v';
  C ← component of G(E(v), F(v)) containing an unsaturated vertex;
  L ← set of tight inequalities at v;
  B' ← set of buyers in the component of (B, G, E(v)) corresponding to
       C;
  G' ← set of goods in the component of (B, G, E(v)) corresponding to
       C;
  F'(v) = {(b_i, g_j) ∈ F(v)|i ∈ B', j ∈ G'};
  E ← {(b_i, g_j) ∈ E(v)|b_i and g_j are in different components of (B', G',
       F'(v))};
  /* rank(L): maximal number of linearly independent inequalities in
       L. */
  while rank(L) = mn + m + n and E ≠ ∅ do
     G ← (B', G', F'(v));
     (b_i, g_j) ← an edge in E between C_s and C_u of G;
     if (b_i, g_j) is from C_s to C_u then
        Relax the inequality u_ij y_i ≤ p_j, which is tight at v;
        L ← L\{u_ij y_i ≤ p_j};
     else Relax the inequality z_ij ≥ 0, which is tight at v;
        L ← L\{z_ij ≥ 0};
        /* C_s and C_u are merged into a new unsaturated component. */
        F'(v) ← F'(v) ∪ {(b_i, g_j)};
     E ← E\{(b_i, g_j)};
  /* Now we are on an edge of polytope O */
  v' ← vertex adjacent to v on this edge in O;
  return v';
```


1. Adsul, B., Sobhan Babu, Ch., Garg, J., Mehta, R. and Sohoni, M., A simple-like algorithm for Fisher markets. In Symposium on Algorithmic Game Theory, 2010, pp. 18–29.
2. Brainard, W. C. and Scarf, H. E., How to compute equilibrium prices in 1891. Cowles Foundation Discussion Paper, 2000, 1270.

3. Arrow, K. and Debreu, G., Existence of an equilibrium for a competitive economy. *Econometrica*, 1954, **22**, 265–290.
4. Eisenberg, E. and Gale, D., Consensus of subjective probabilities: the Pari–Mutuel method. *Ann. Math. Stat.*, 1959, **30**, 165–168.
5. Vazirani, V. V., Combinatorial algorithms for market equilibria. In *Algorithmic Game Theory* (eds Nisan, N. *et al.*), Cambridge University Press, 2007, pp. 53–78.
6. Deng, X., Papadimitriou, C. and Safra, S., On the complexity of equilibria. In Proceedings of ACM Symposium on Theory of Computing, Montreal, Quebec, Canada, 2002.
7. Devanur, N., Papadimitriou, C. H., Saberi, A. and Vazirani, V. V., Market equilibrium via a primal-dual-type algorithm. *J. ACM*, 2008, **55**(5): 22.
8. Jain, K., A polynomial time algorithm for computing the Arrow-Debreu market equilibrium for linear utilities. *SIAM J. Comput.*, 2007, **37**, 306–318.
9. Orlin, J. B., Improved algorithms for computing fisher's market clearing prices. In ACM Symposium on the Theory of Computing, Cambridge, Massachusetts, USA, 2010.
10. Eaves, B. C., A finite algorithm for the linear exchange model. *J. Math. Econ.*, 1976, **3**, 197–203.
11. Lemke, C. E., Bimatrix equilibrium points and mathematical programming. *Manage. Sci.*, 1965, **11**, 681–689.
12. Lemke, C. E. and Howson Jr, J. T., Equilibrium points of bimatrix games. *SIAM J. Appl. Math.*, 1964, **12**, 413–423.
13. Shmyrev, V. I., An algorithm for finding equilibrium in the linear exchange model with fixed budgets. *J. Appl. Ind. Math.*, 2009, **3**, 505–518.
14. Birnbaum, B., Devanur, N. and Xiao, L., Distributed algorithms via gradient descent for Fisher markets. In ACM Conference on Electronic Commerce, San Jose, USA, 2011.
15. Adsul, B., Sobhan Babu, Ch., Garg, J., Mehta, R. and Sohoni, M., Nash equilibria in Fisher market. In Symposium on Algorithmic Game Theory, Athens, Greece, 2010, pp. 30–41.
16. Chakraborty, S., Devanur, N. and Karande, C., Market equilibrium with transaction costs. In Workshop on Internet and Network Economics, Stanford, California, USA, 2010.
17. Garg, J., Mehta, R., Sohoni, M. and Vishoni, N., Towards polynomial simplex-like algorithms for market equilibria. In ACM–SIAM Symposium on Discrete Algorithms, New Orleans, USA, 2003.
18. Vazirani, V. V. and Yannakakis, M., Market equilibrium under separable, piecewise-linear, concave utilities. *J. ACM*, 2011, **58**, 10:1–10:25.
19. Papadimitriou, C. H., On the complexity of the parity argument and other inefficient proofs of existence. *J. Comput. Syst. Sci.*, 1992, **48**, 498–532.
20. Garg, J., Mehta, R., Sohoni, M. and Vazirani, V. V., A complementary pivot algorithm for market equilibrium under separable piecewise-linear concave utilities. In ACM Symposium on the Theory of Computing, New York, USA, 2012.