A Simplex-Like Algorithm for Fisher Markets

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

[1] Indian Institute of Technology, Bombay
{adsul,jugal,ruta,sohoni}@cse.iitb.ac.in
[2] Indian Institute of Technology, Hyderabad
sobhan@iith.ac.in

Abstract. 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 non-linear; however, unlike that, 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 pivoting algorithm which is provably strongly polynomial for many special cases.

1 Introduction

Fisher and Arrow-Debreu market models are the two fundamental market models in mathematical economics. In this paper, we focus on the Fisher market model with linear utilities. An instance of this model consists of a set of buyers, a set of divisible goods, initial endowments, also referred to as the money owned by the buyers, quantities of the goods and (linear) utility functions of the buyers. The problem is to determine market equilibrium prices and allocation of the goods to buyers such that the market clears and the utility function for each buyer is maximized. Towards this, Eisenberg and Gale [6,10] 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.

Recently, many algorithmic results [4,5,9,11] pertaining to the computation of market equilibrium prices and allocation for the linear case of Fisher and Arrow-Debreu market models have been obtained. In [4], Deng et al. 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. [5] developed a primal-dual type first polynomial time algorithm to solve the Fisher market model. A polynomial time algorithm for the more general Arrow-Debreu market is also presented in [9]. More recently, a strongly polynomial time algorithm for the Fisher market was given by Orlin [11]. 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 paper is an attempt towards this objective.

S. Kontogiannis, E. Koutsoupias, P.G. Spirakis (Eds.): SAGT 2010, LNCS 6386, pp. 18–29, 2010.

In this paper, we propose a novel convex optimization formulation for the Fisher market problem. In the Eisenberg-Gale formulation [6,10], 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, unlike that, 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 formulation, which maximizes a convex function under flow constraints, obtained by Shmyrev [12] and Birnbaum et al. [2], however this formulation also does not guarantee the optimum to be at a vertex.

We define *special vertices* in our polytope and every such vertex corresponds to the Fisher solution with a different endowment vector. 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 it 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 and show that this algorithm is strongly polynomial for many special cases. Two interesting cases are:

- Either the number of buyers or the goods is fixed.
- All the non-zero utilities are of the type α^k, where $\alpha > 0$ and $0 \leq k \leq M$ (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, firstly, in practice, utilities are hardly exactly known, and secondly, as shown in [1] 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 [5], however one crucial difference is that the prices, DPSV algorithm computes at intermediate stages, may not occur at a vertex in the polytope. The DPSV algorithm may be interpreted as an interior point method in our formulation. Further, the utility of our formulation is also illustrated by its easy extension to incorporate transportation costs as well [8]. There seems no way to modify Eisenberg-Gale or Shmyrev formulations to capture the equilibrium solution for this extended model. Independently, Chakrabarty et al. [3] 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.

Organization. The rest of the paper is organized as follows. In Section 2, we give a precise formulation of the Fisher market problem and introduce the new convex optimization program and analyze it. In Section 3, we discuss the simplex-like algorithm. In Section 4, we show that the algorithm is provably strongly polynomial for many special cases. In Section 5, we summarize the number of pivoting steps taken by the algorithm on random instances of the Fisher market. Finally we conclude in Section 6.

2 New Convex Optimization Formulation

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

2.1 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 good j, q_j is the quantity of good 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 good, *i.e.*, $\forall j \in \mathcal{G}$, $\sum_{i\in\mathcal{B}} x_{ij} = q_j$ and $\forall i \in \mathcal{B}$, $\sum_{j\in\mathcal{G}} p_j x_{ij} = m_i$.
- **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 $\frac{u_{ij}}{p_j} = \max_{k\in\mathcal{G}} \frac{u_{ik}}{p_k}$.

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

2.2 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, $\frac{1}{y_i}$ 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 1 to 4 and the related convex program (with the same cost function) be the auxiliary convex program.

Claim. $Pr(O) = O_{aux}$, where $Pr(O)$ is the projection of O onto the y-p-space.

Proof. Clearly, $Pr(O) \subseteq O_{aux}$, and for $O_{aux} \subseteq Pr(O)$, $Z = [z_{ij}]$ should be constructed for a given $(\boldsymbol{y}, \boldsymbol{p}) \in O_{aux}$. 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. □

Table 1. New Convex Program

$$\text{maximize} \quad \sum_{i\in\mathcal{B}} m_i \log y_i$$

subject to

$$\forall i \in \mathcal{B}, \forall j \in \mathcal{G} \; : \; u_{ij} y_i \le p_j \tag{1}$$

$$\sum_{j\in\mathcal{G}} p_j \le \sum_{i\in\mathcal{B}} m_i \tag{2}$$

$$\forall i \in \mathcal{B} \; : \; y_i \ge 0 \tag{3}$$

$$\forall j \in \mathcal{G} \; : \; p_j \ge 0 \tag{4}$$

$$\forall i \in \mathcal{B} \; : \; \sum_{j\in\mathcal{G}} z_{ij} \le m_i \tag{5}$$

$$\forall j \in \mathcal{G} \; : \; \sum_{i\in\mathcal{B}} z_{ij} = p_j \tag{6}$$

$$\forall i \in \mathcal{B}, \forall j \in \mathcal{G} \; : \; z_{ij} \ge 0 \tag{7}$$

Therefore, in order to understand the optimality conditions, we may as well work with the KKT conditions for the auxiliary convex program. Let $x_{ij}, q, \mu_i, \lambda_j$ be the Lagrangian (dual) variables corresponding to the equations (1-4). An optimal solution must satisfy the KKT conditions in Table 2.

Table 2. KKT conditions

$$\forall i \in \mathcal{B} \; : \; \frac{m_i}{y_i} = \sum_{j\in\mathcal{G}} u_{ij} x_{ij} - \mu_i \tag{8}$$

$$\forall i \in \mathcal{B}, \forall j \in \mathcal{G} \; : \; (u_{ij} y_i - p_j) x_{ij} = 0 \tag{9}$$

$$\forall j \in \mathcal{G} \; : \; -\sum_{i\in\mathcal{B}} x_{ij} - \lambda_j + q = 0 \tag{10}$$

$$(\sum_{j\in\mathcal{G}} p_j - \sum_{i\in\mathcal{B}} m_i) q = 0 \tag{11}$$

$$\forall i \in \mathcal{B}, \forall j \in \mathcal{G} \; : \; x_{ij}, \lambda_j, \mu_i, q \ge 0 \tag{12}$$

$$\forall j \in \mathcal{G} \; : \; -p_j \lambda_j = 0 \tag{13}$$

$$\forall i \in \mathcal{B} \; : \; -y_i \mu_i = 0 \tag{14}$$

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

Proof. $\mu_i \neq 0 \Rightarrow y_i = 0 \Rightarrow$ 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, $\lambda_j \neq 0 \Rightarrow p_j = 0 \Rightarrow y_i = 0$, 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 (8-12), we get,

$$\forall i \in \mathcal{B} \quad : \quad m_i = \sum_{j \in \mathcal{G}} u_{ij} x_{ij} y_i \tag{15}$$

$$\forall i \in \mathcal{B}, \forall j \in \mathcal{G} \quad : \quad (u_{ij} y_i - p_j) x_{ij} = 0 \tag{16}$$

$$\forall j \in \mathcal{G} \quad : \quad \sum_{i \in \mathcal{B}} x_{ij} = q \tag{17}$$

$$: \quad (\sum_{j \in \mathcal{G}} p_j - \sum_{i \in \mathcal{B}} m_i) q = 0 \tag{18}$$

$$\forall i \in \mathcal{B}, \forall j \in \mathcal{G} \quad : \quad x_{ij}, q \geq 0 \tag{19}$$

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

Proof. As $q = 1$, interpreting $X = [x_{ij}]$ as an allocation, we see that conditions (15-17) imply that the market clearing constraint holds at the price vector $\boldsymbol{p}$. Further, using condition 2, we have $x_{ij} > 0 \Rightarrow y_i u_{ij} = p_j$. As $(\boldsymbol{y}, \boldsymbol{p}) \in O_{aux}$, we also have, $\forall i \in \mathcal{B}, \forall j \in \mathcal{G} : u_{ij} y_i \leq p_j$. 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 $(\boldsymbol{y}, \boldsymbol{p}) \in O_{aux}$ be the unique optimum solution to the auxiliary convex program. Let $\mathbb{X} = \{X = [x_{ij}]_{i \in \mathcal{B}, j \in \mathcal{G}} \mid (\boldsymbol{y}, \boldsymbol{p}, X) \text{ satisfies (8-14)}\}$. 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 $G = (\mathcal{B}, \mathcal{G}, E)$, where $E = \{(i, j) \in \mathcal{B} \times \mathcal{G} \mid x^a_{ij} > 0\}$, is acyclic. Let Z^a be the equilibrium money allocation w.r.t. X^a. The next proposition asserts that $(\boldsymbol{y}, \boldsymbol{p}, Z^a)$ is in fact a vertex of O.

Proposition 3. *The point $(\boldsymbol{y}, \boldsymbol{p}, Z^a)$ is a vertex of O.*

Proof. There are $mn + m + n$ variables in the convex program, and we show that there are $mn + m + n$ linearly independent tight constraints at $(\boldsymbol{y}, \boldsymbol{p}, Z^a)$ (refer to Theorem 3.3.3 in [7] for details).

Remark 4. The auxiliary program itself captures the equilibrium prices at the optimal solution, though not necessarily at one of its vertices. [7] has the detailed analysis of both the polytopes.

3 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 $\forall i \in \mathcal{B},\ \forall j \in \mathcal{G},\ z_{ij}(u_{ij}y_i - p_j) = 0$. In other words, every buyer spends money only on her optimal goods.

Definition 5. *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 $\mathcal{B}' = \{i \in \mathcal{B} \mid y_i \neq 0\}, \mathcal{G}' = \mathcal{G}$ 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 $(U', \boldsymbol{m'})$.

3.1 Characterization of Special Vertices

Let $v = (\boldsymbol{y}, \boldsymbol{p}, Z)$ be a special vertex of O. W.l.o.g., 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 $\boldsymbol{p'} = \boldsymbol{p}$, $Z' = Z$, $y'_i = y_i,\ \forall i \neq k$, and $y'_k = \min_{j \in \mathcal{G}} \frac{p_j}{u_{ij}}$.

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

$$E(v) = \{(i,j) \in \mathcal{B} \times \mathcal{G} \mid u_{ij}y_i = p_j\} \quad \text{and} \quad F(v) = \{(i,j) \in \mathcal{B} \times \mathcal{G} \mid z_{ij} > 0\}$$

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 $C_1, C_2, \dots$ 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 6. *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 its surplus is zero, otherwise it 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 7. *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 buyers, then the z_{ij}'s in that component may be modified such that the same set of inequalities are tight before and after the modification, *i.e.*, 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 are tight before and after the scaling, hence a contradiction. □

Corollary 8. *If $(U, \boldsymbol{m})$ are algebraically independent, 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 9. *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 7 and Corollary 8. □

3.2 Algorithm

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 AdjacentVertex procedure for the main convex program.

We assume that $(U, \boldsymbol{m})$ are algebraically independent[1]. The AdjacentVertex procedure, given in Table 3, 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 8). 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. There are two cases depending on where b_i belongs: C_s (Case 1) or C_u (Case 2).

Case 1: We get a new vertex v', adjacent to v in O, by relaxing the inequality $u_{ij} y_i \le p_j$, which is tight 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

[1] For the general $(U, \boldsymbol{m})$, AdjacentVertex may be easily modified.

Table 3. AdjacentVertex Procedure

```
AdjacentVertex(v)
   v' ← v;
   if v is optimum then
      return v';
   endif
   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;
   endif
   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.

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 $z_{ij} \geq 0$, which is tight 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 lists the three possible cases for the new inequality.

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 10. *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.*

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 ← AdjacentVertex(v);
    endwhile
    i ← i + 1;
endwhile
```

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).

Remark 11. Algorithm 1 provides a polyhedral interpretation to a sequential run of the so called *Basic Algorithm* in [5], where buyers are added one at a time.

Lemma 12. *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 $u_{kl}y_k \leq p_l$ for some (b_k, g_l). Let C_s^j be the component containing buyer k in the j^{th} 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 are clearly bounded by 2^{m+n}, and in every n iterations of 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 13. A more refined bound is 2^{m+n+1}.

4 Analysis

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

Main Idea of Algorithm 1. 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 $z_{kl} = 0$. Now, consider the tree T in $(\mathcal{B}, \mathcal{G}, E(v))$ 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.

Theorem 14. *Algorithm 1 is strongly polynomial when either the number of buyers or goods is constant.*

Proof. W.l.o.g., we assume that $(U, \boldsymbol{m})$ are algebraically independent[2].

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 j^{th} 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^j_s, and one unsaturated vertex, say C^j_u. Note that buyer i belongs to C^j_u.

Let (b_k, g_l) be the edge between C^j_u and C^j_s, 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^j_s, we relax the inequality $u_{kl}y_k \leq p_l$, and when buyer k belongs to C^j_u, we relax the inequality $z_{kl} \geq 0$. In other words, we add the edge (b_k, g_l) when buyer k belongs to C^j_u and delete it when buyer k belongs to C^j_s.

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^j_u). □

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 are 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 15. *Algorithm 1 is strongly polynomial when* $\forall i \in \mathcal{B},\ \forall j \in \mathcal{G},\ u_{ij} = \alpha^{k_{ij}}$*, where* $0 \leq k_{ij} \leq poly(m, n)$ *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 $\frac{y_a}{p_b}, \forall b \in \mathcal{G}$. Note that $\frac{y_a}{p_b}$ for a good b remains same until both buyer a and good b are in the same component

[2] For the general $(U, \boldsymbol{m})$, a similar proof may be worked out.

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 j^{th} iteration. If surplus of buyer a is not zero, then C^j contains exactly one saturated vertex, say C^j_s, and one unsaturated vertex, say C^j_u. Note that buyer a belongs to C^j_u.

Let (b_k, g_l) be the edge between C^j_u and C^j_s. There are two types of iterations, 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). Let $z_{kl} \geq 0$ is relaxed in the j^{th} 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 $u_{ij}y_i \leq p_j$), and the value of $\frac{y_a}{p_l}$ strictly increases when iteration of Type 1 occurs. Now, we consider the values of $\log_\alpha \frac{y_a}{p_j}$, $\forall j \in \mathcal{G}$. Clearly, these values monotonically increase when an iteration of Type 1 occurs. Since for every $j \in \mathcal{G}$, the value of $\log_\alpha \frac{y_a}{p_j}$ might be at most $n * poly(m, n)$, therefore for every buyer i, the number of iterations of inner while loop is bounded by $n^2 * poly(m, n)$. □

Theorem 15 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.

5 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 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.

Table 5. Number of Pivoting Steps Taken by Algorithm 1

# buyers/goods	4	8	12	16	20	24
min	6	31	84	136	245	223
max	24	80	168	235	320	514
mean	12.5	50.9	113.1	186.9	279.8	408.9

Clearly, the number of steps seem to increase quadratically with the size of instances, and even the worst case instance for each value of m requires fewer than $2m^2$ steps. Therefore, Algorithm 1 should have a much better bound.

6 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 feel that the strongly polynomial algorithm by Orlin [11] 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.

References

1. Adsul, B., Babu, C.S., Garg, J., Mehta, R., Sohoni, M.: Nash equilibria in Fisher market. arXiv:1002.4832 (2010)
2. Birnbaum, B., Devanur, N.R., Xiao, L.: New convex programs and distributed algorithms for Fisher markets with linear and spending constraint utilities. Working Paper (2010)
3. Chakraborty, S., Devanur, N.R., Karande, C.: Market equilibrium with transaction costs. arXiv:1001.0393 (2010)
4. Deng, X., Papadimitriou, C.H., Safra, S.: On the complexity of equilibria. In: STOC 2002 (2002)
5. Devanur, N.R., Papadimitriou, C.H., Saberi, A., Vazirani, V.V.: Market equilibrium via a primal-dual type algorithm. JACM 55(5) (2008)
6. Eisenberg, E., Gale, D.: Consensus of subjective probabilities: The pari-mutuel method. Annals Math. Stat. 30, 165–168 (1959)
7. Garg, J.: Algorithms for market equilibrium. First APS Report (2008), `http://www.cse.iitb.ac.in/~jugal/aps1.pdf`
8. Garg, J.: Fisher markets with transportation costs. Technical Report (2009), `http://www.cse.iitb.ac.in/~jugal/techRep.pdf`
9. Jain, K.: A polynomial time algorithm for computing the Arrow-Debreu market equilibrium for linear utilities. In: FOCS 2004 (2004)
10. Nisan, N., Roughgarden, T., Tardos, E., Vazirani, V.V.: Algorithmic Game Theory. Cambridge University Press, Cambridge (2007)
11. Orlin, J.B.: Improved algorithms for computing Fisher's market clearing prices. In: STOC 2010 (2010)
12. Shmyrev, V.I.: An algorithm for finding equilibrium in the linear exchange model with fixed budgets. Journal of Applied and Industrial Mathematics 3(4), 505–518 (2009)