A Perfect Price Discrimination Market Model with Production, and a Rational Convex Program for it*

Gagan Goel† *Vijay V. Vazirani*‡

Abstract

Recent results showing PPAD-completeness of the problem of computing an equilibrium for Fisher's market model under additively separable, piecewise-linear, concave utilities (plc utilities) have dealt a serious blow to the program of obtaining efficient algorithms for computing equilibria in "traditional" market models and has prompted a search for alternative models that are realistic as well as amenable to efficient computation. In this paper we show that introducing perfect price discrimination into the Fisher model with plc utilities renders its equilibrium polynomial time computable. Moreover, its set of equilibria are captured by a convex program that generalizes the classical Eisenberg-Gale program, and always admits a rational solution. We also give a combinatorial, polynomial time algorithm for computing an equilibrium.

Next, we introduce production into our model, and again give a (rational) convex program that captures its equilibria. We use this program to obtain surprisingly simple proofs of both welfare theorems or this model. Finally, we also give an application of our price discrimination market model to online display advertising marketplaces.

1 Introduction

The search for efficient algorithms for computing market equilibria started with much interest within theoretical computer science about a decade ago. The goal was not only academic, i.e., providing an algorithmic ratification of Adam Smith's "invisible hand of the market," but was also motivated by potential applications to the plethora of new and highly lucrative markets that have emerged on the Internet.

This study started with the simple case of linear utility functions, for which polynomial time algorithms were obtained [9, 14], and gradually moved on to more general and realistic utility functions. However, the latter program had limited success (most notably, an efficient algorithm for approximating equilibria for the Fisher model under Leontief utilities [8, 27]), and was dealt a serious blow with the recent resolution of the long-standing open problem of finding the complexity of computing an equilibrium under additively separable, piecewise-linear, concave utilities (plc utilities). First, [6] proved PPAD-hardness for the Arrow-Debreu model under plc utilities. Subsequently, PPAD-hardness was established for Fisher's model, independently and concurrently by [7] and [25]. Membership in

*College of Computing, Georgia Institute of Technology, Atlanta, GA 30332–0280. Supported by NSF Grants CCF-0728640 and CCF-0914732, ONR Grant N000140910755, and a Google Research Grant.
†Email: gagang@cc.gatech.edu
‡Email: vazirani@cc.gatech.edu

PPAD was established for both models in [25], hence precisely pinning down the complexity of plc utilities. Assuming P $\neq$ PPAD, this effectively rules out the existence of efficient algorithms for almost all general and interesting classes of "traditional" market models.

On the other hand, markets in the West, based on Adam Smith's free market principle, seem to do a good job of finding prices that maintain parity between supply and demand. This has prompted the question (see [24]) of whether we have failed to capture some essential elements of real markets in our models, and what is the "right" model which is not only realistic and admits equilibria but is also amenable to efficient computation of equilibria.

The purpose of this paper is not to solve the question stated above in its full entirety but to show that there is a natural market model, with buyers having plc utilities, for which the equilibrium computation problem is efficiently solvable, even via a combinatorial, polynomial time algorithm. Our result provides a ray of hope for the above-stated program of changing the traditional market model suitably to make it computationally viable.

Our model was obtained by introducing perfect price discrimination into Fisher markets with plc utilities. This was achieved by adding a middleman to the latter model. The middleman buys goods from sellers according to the prices they set. He then sells these goods to the buyers, but charges them according to the utility they derive. Within the constraints of the model, each buyer is sold goods at an optimal *rate*, i.e., utility accrued per dollar charged.

Price discrimination was first introduced by Pigou [19], who gave the notions of first, second and third degree price discrimination. First degree price discrimination is also called perfect price discrimination and can only be practiced by a monopoly that is able to segregate buyers according to their willingness to pay. The price charged from a buyer is such that her marginal willingness to pay is equal to the marginal cost of the good. Since the monopolist needs to have complete information about the buyers' utility functions, this type of price discrimination has limited applicability. However, in Section 2.4 we do provide a setting, namely online display advertising marketplaces, which satisfies this stringent criterion to a high degree.

An interesting aspect of our market model is that its equilibrium is captured by a convex program; in particular, this gives another way of computing its equilibrium, i.e., is using the ellipsoid algorithm. Let us describe how this convex program was obtained. The classic Eisenberg-Gale convex program [12] captures equilibrium allocations for the linear Fisher market. An obvious way of attacking the question of computing an equilibrium for a Fisher model with plc utilities was to obtain a suitable generalization of the Eisenberg-Gale program. However, the following observation shows that this approach will not go very far: whereas the optimal solutions of a convex program form a convex set, the set of equilibria for a Fisher market with plc utilities can be disconnected.

Introducing price discrimination in the plc Fisher model renders its set of equilibria convex. The situation is somewhat analogous to that of Nash equilibrium. The set of optimal strategies of the pure equilibria of a bimatrix game can be disconnected; however, by introducing mixed strategies, it is rendered a convex set and hence suitable for study with the Kakutani fixed point theorem. The convex program for our model is a natural generalization of the Eisenberg-Gale program.

The notion of a *rational convex program (RCP)* was introduced recently in [23], i.e., a nonlinear convex program that always has a rational solution of polynomial bit-size, if all its parameters are rational numbers. Two classes of RCPs were identified, quadratic and logarithmic. The latter have linear constraints and objective functions of the kind $\sum_i a_i \log f_i(x)$, where a_is are constants and

f_is are linear functions of the variables x. Starting with the celebrated Eisenberg-Gale program, several convex programs arising in mathematical economics and game theory are now known to be logarithmic RCPs. These include convex programs that capture Nash bargaining games [23, 22], and those that capture the Fisher market model under several classes of utility functions: linear [12, 9], utility functions defined via combinatorial problems, including some in Kelly's [17] resource allocation model [15, 5], and spending constraint utilities [24, 4]. We prove that the program capturing our market model is also a logarithmic RCP. In particular, this implies that the ellipsoid method will yield the exact equilibrium in polynomial time [13].

We generalize our basic model along 2 directions. In Section 11, we give a generalization in which buyers have utility for money, given by a piecewise-linear, concave function for each buyer. Now, at equilibrium, a buyer may choose to not spend all of her money. We show how to extend our combinatorial polynomial time algorithm to this case as well. The solution still turns out to be rational; however, we do not know of a convex program that captures this enhanced model (see Section 14). By exploiting the combinatorial structure discovered in obtaining our algorithm, we give a characterization of the entire set of equilibria of this market. This reveals the range of equilibrium prices of each good and the range of profit that the middleman can accrue from each buyer.

Second, in Section 12 we generalize sellers of goods to firms, which besides selling goods also act as suppliers of labor and producers of goods, or any combination of these activities. These firms have initial endowments of goods and labor, and their goal is to maximize profits by optimally producing and selling goods. Traditionally in economics, and in the model studied by Arrow and Debreu [2], production satisfies non-increasing returns to scale. This is the case in our model as well, though it is imposed in a "piecewise-linear manner". As a consequence of linearity, for given prices of goods, the optimal operation of a firm is captured by an LP. The prices of goods are parameters in this LP.

We show that equilibrium production and allocation for this market model is also captured via a single convex program, generalizing our previous program. The optimal dual of this program yields equilibrium prices; moreover, this program is also a logarithmic RCP. The (unique) optimal solution to the dual variables of this RCP gives equilibrium prices. Furthermore, if in the LP of each firm, we set price parameters to these equilibrium prices then an optimal solution of the resulting LP is given by the optimal solution to this convex program. The idea behind this fact goes back to [16]; however, our model of production is considerably more general than that in [16]. Using this convex program, we obtain surprisingly simple proofs of both welfare theorems (simpler even than those for the basic Arrow-Debreu market model) for our model with production. We leave the open problem of extending our combinatorial algorithm to the model with production.

1.1 Outline of the paper

In Section 2, we present our basic Fisher-based price discrimination model, without production. Section 2.4 gives nn application of this model to online display advertising marketplaces. In Section 3, we give a rational convex program for this basic model. We then embark on obtaining a combinatorial algorithm for finding an equilibrium for this model. In Section 5 we give an algorithm that is guaranteed to terminate with an equilibrium. Its polynomial time implementation and proof are given in Sections 6, 7, 8 and 9. Section 10 gives a characterization of all equilibria, and given one equilibrium, a way of finding the entire set. Next, in Section 11, we extend the model so that buyers have utility for money and may spend only a part of their money. We extend our combinatorial algorithm to this model as well. In Section 12 we enhance our model further by adding production into it and we give a rational convex program for this enhanced model in Section 12.2. Finally, we present the First and Second Welfare Theorems for this enhanced model in Section 13.

2 The Basic Market Model

2.1 Perfect price discrimination

Most businesses today charge different prices from different consumers for essentially the same goods or services in order to maximize their revenues. This practice is called *price discrimination*. It is not only widespread but also essential for survival of certain businesses, e.g., in the airline industry. Price discrimination has been extensively studied in economics from many different angles; see [26, 21, 20, 11, 10, 1, 3, 18] for just a small sampling of papers on this topic.

A monopolistic situation in which the business separates the market into individual consumers and charges each one *prices that they are willing and able to pay* is called *perfect price discrimination*. Of course, the business needs to have complete information about each consumer's preferences.

Our market model consists of buyers, seller and a middleman. As stated in the Introduction, buyers have initial endowments of money with which they wish to buy goods and maximize the utility accrued. The sellers have initial endowments of goods.

The middleman buys goods from the sellers, which charge the middleman in accordance with the prices set and the amounts bought. As stated above, buyers decide at what rate they want utility, and the middleman sells to them goods accordingly, with the only condition that he never sells any part of any good at a loss – the fact that the middleman knows the buyer's utility function enables him to do this. We show below that under these circumstances, for any given prices of goods, there is a unique optimal rate for each buyer. This is also the rate that ensures that the marginal utility accrued by the buyer per unit of money spent is equal to the marginal cost of the goods she is receiving, as desired under perfect price discrimination.

In our model we assume that the buyers have plc utilities. If all buyers had linear utility functions for goods, then the middleman will make no profit. In fact, in this case, our model becomes precisely the classical linear Fisher market. Prices for goods are said to be *equilibrium prices* if with optimal rates for each buyer, the market clears, i.e., all the goods get sold to buyers and all of their money gets spent.

The set of all goods is denoted by G, $|G| = n_G$. Each good is assumed to be divisible. Let B denote the set of buyers, $|B| = n_B$. Assume that the buyers are numbered from 1 to n_B and are indexed by i, goods are numbered from 1 to n_G and are indexed by j.

2.2 The buyers and their utility functions

Let $m_i \in \mathbf{Q}^+$ dollars denote the initial amount of money possessed by buyer $i \in B$. For each buyer i and good j we are specified a function $f_j^i : \mathbf{R}_+ \to \mathbf{R}_+$ which gives the utility that i derives as a function of the amount of good j that she receives. Each function f_j^i is a non-negative, non-decreasing, piecewise-linear, concave function. The overall utility of buyer i, $u_i(\boldsymbol{x})$ for a bundle $\boldsymbol{x} = (x_1, \ldots, x_g)$ of goods, is additively separable over the goods, i.e., $u_i(\boldsymbol{x}) = \sum_{j \in G} f_j^i(x_j)$.

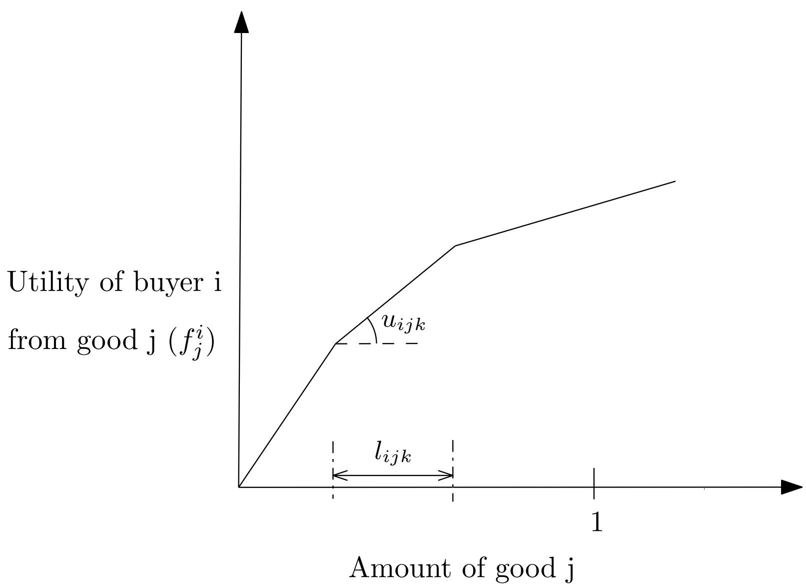

Figure 1: The piecewise-linear, concave utility of buyer i for good j.

We will call each piece of f_j^i a *segment*. Number the segments of each function in order of decreasing slope. Let $s_{ijk},\ k = 1, 2, \ldots$ denote the kth segment of f_j^i. Let x denote the amount of good j given to buyer i and assume that a segment s_{ijk} corresponds to x in the interval $[a, b]$. Then, the amount of good j represented by segment s_{ijk}, $l_{ijk} = b - a$, and u_{ijk} will denote the rate at which buyer i accrues utility per unit of good j received, when she is getting an allocation corresponding to segment s_{ijk}. Clearly, the maximum utility she can receive corresponding to segment s_{ijk} is $u_{ijk} \cdot l_{ijk}$; we will denote this by utility(s_{ijk}). We will assume that u_{ijk} and l_{ijk} are rational numbers. Let S_{ij} denote the set of segments of function f_j^i and let S_i denote the set of all segments of buyer i, i.e., $S_i = \cup_{j=1}^{g} S_{ij}$.

2.3 The middleman and determining buyers' rates

By *price of good j* we mean the price charged by the sellers from the middleman for 1 unit of good j. Assume that the prices of goods are set at $\boldsymbol{p} = (p_1, \ldots, p_{n_G})$, which are the prices at which the sellers sell good to the middleman. Define the *bang-per-buck* of segment s_{ijk} to be u_{ijk}/p_j and denote it by bpb(s_{ijk}); clearly, this is the amount of utility accrued by i per dollar spent for an allocation corresponding to segment s_{ijk}, if buyer i were to buy directly from the sellers, without going through the middleman.

As stated above, in perfect price discrimination, buyers are charged prices that they are willing and able to pay. We first describe informally how this principle leads to a method of charging money for a bundle of goods. Assume that buyer i buys a certain bundle of goods, including good j. How do we determine the price she is willing to pay for j. Clearly, she gets utility at different rates for this good, as a function of the amount of good obtained. Find the least rate in the obtained bundle, say r, and assume that i is willing to pay a price of p_j for r units of utility, i.e., she is willing to obtain goods at the rate of r/p_j units of utility per dollar spent. This enables us to compute the amount to be charged for the entire bundle.

Suppose buyer i fixes her rate at r_i which is the amount of utility she wants per dollar, independent of the good. Then, for an allocation corresponding to segment s_{ijk}, the middleman is effectively charging the buyer $\frac{u_{ijk}}{r_i}$ dollars per unit of j. In particular, if $\text{bpb}(s_{ijk}) < r_i$, then the middleman will be allocating this segment at a loss, i.e., at a price smaller than p_j dollars per unit of j. Moreover, the larger $\text{bpb}(s_{ijk})/r_i$ is, the higher is the profit the middleman can make from allocations corresponding to this segment. Therefore, once i announces her rate, the middleman removes from consideration all segments $s \in S_i$ such that $\text{bpb}(s) < r_i$, and allocates to i goods corresponding to segments that gives him the highest profit, until i exhausts her money.

Now, given how the middleman responds to prices and rates, what rate maximizes the utility of a buyer i? We will define a rate r_i^*, which we will call the *optimal* rate of buyer i, as a function of prices $\boldsymbol{p}$, and show that this rate maximizes buyer i's utility. The overall objective is to find prices for goods such that if the buyers report their optimal rates, the market clears under above transactions, i.e., there is no surplus or deficiency of any good. This is our notion of *equilibrium* for the market.

Rate r_i^* is obtained as follows. Sort all segments in S_i by decreasing bang-per-buck and start with a sufficiently large number α. Consider all segments $s \in S_i$ such that $bpb(s) \geq \alpha$, and add up their total utilities. We will denote this by $t(\alpha)$, i.e.,

$$t(\alpha) = \sum_{s \in S_i:\ \text{bpb}(s) \geq \alpha} \text{utility}(s).$$

Now the cost of buying goods corresponding to all these segments at rate α is $t(\alpha)/\alpha$. When α is very large, this will be less than m_i. Observe that as α is decreased, this number increases monotonically. Now r_i^* is the largest value of α such that this number is $\geq m_i$. Formally,

$$r_i^*(\boldsymbol{p}) \;=\; \arg\max_{\alpha} \left\{ \frac{t(\alpha)}{\alpha} \geq m_i \right\}.$$

We will denote $r_i^*(\boldsymbol{p})$ by simply r_i^* when its meaning is clear from the context.

Let us say that rate r_i is *good* if each segment s such that $\text{bpb}(s) > r_i$ is fully allocated to i, and corresponding to segments s such that $\text{bpb}(s) = r_i$, i is allocated just the right amount of goods so that her total utility adds up to $r_i \cdot m_i$. The following lemma is straightforward.

Lemma 1 *Rate r_i is good iff it equals r_i^*.*

Lemma 2 *For any prices $\boldsymbol{p}$, rate r_i^* maximizes the utility for buyer i.*

Proof : If the rate is fixed at $\alpha < r_i^*$, then $\alpha \cdot m_i < r_i^* \cdot m_i$ and therefore i will be allocated smaller utility. Next consider fixing the rate at $\beta > r_i^*$. Let s be the smallest bang-per-buck of an allocated segment at rate r_i^*. If $\text{bpb}(s) = r_i^*$, then at rate β she will accrue smaller utility. Otherwise, for $\text{bpb}(s) \geq \beta \geq r_i^*$, i will still be allocated $r_i^* \cdot m_i$ utility and for $\beta > \text{bpb}(s)$, she will accrue strictly smaller utility. This proves the lemma. □

2.4 An application to online display advertising marketplaces

Currently, there are companies that sell ad slots on web sites to advertisers. In keeping with our model, we will view such a company as the middleman, the owners of web sites as sellers and the advertisers as buyers. We will view ad slots on different web sites as different items, whose prices will be determined by the market; presumably these will be equilibrium prices in some model (we argue below that our model applies). An advertiser's utility for a particular ad slot is determined by the probability that her ad will get clicked if it is shown on that slot; her total utility is additive over all the slots she is allocated. Advertisers typically pay at fixed rate to the middleman for the expected number of clicks they get, i.e., they are paying at fixed rate for every unit of utility they get, as is the case in our model. Using knowledge of the utility function of buyers, the middleman is able to price discriminate and charge more from buyers who get more clicks (i.e., utility). Clearly, this situation is captured by our model.

3 A Rational Convex Program for the Basic Model

In this section, we give a logarithmic RCP whose optimal primal and dual solutions yield equilibrium allocations, price and rates for the basic market model given in Section 2. Let x_{ijk} denote the amount of good j that is allocated to buyer i from the k^{th} segment s_{ijk} of S_{ij}. W.l.g. assume that the total supply of each good is 1. Now, consider the following convex program:

$$\begin{aligned} \text{maximize} \quad & \sum_{i \in B} m_i \log(u_i) && (1) \\ \text{subject to} \quad & \forall i \in B: \; u_i = \sum_{j \in G} \sum_{k \in S_{ij}} u_{ijk} x_{ijk} \\ & \forall j \in G: \; \sum_{i \in B} \sum_{k \in S_{ij}} x_{ijk} \leq 1 \\ & \forall i \in B, \; \forall j \in G, \forall k \in S_{ij}: \; x_{ijk} \leq l_{ijk} \\ & \forall i \in B, \; \forall j \in G, \forall k \in S_{ij}: \; x_{ijk} \geq 0 \end{aligned}$$

Here the first constraint is ensuring that u_i is the total utility of buyer i, the second constraint is ensuring that for any good the amount sold doesn't exceed its supply, and the third constraint is ensuring that the amount allocated from each segment doesn't exceed its size.

Let p_j be the dual variable corresponding to good j in the second set of constraints above. We will prove the following theorem in this section:

Theorem 3 *Prices* $\mathbf{p}$ *are equilibrium prices if and only if they form an optimal dual solution to convex program (1).*

We will make two mild **assumptions:**

1. For every good j, $\sum_{i,k:\ u_{ijk}>0} l_{ijk} > 1$; that is, the supply of every good is limited w.r.t the total demand of buyers if the prices were zero.

2. Each buyer i desires some good; that is, $u_{ijk} > 0$ for some segment s_{ijk} of every buyer i.

Note that, because of the 2^{nd} assumption, in the optimal solution of the above convex program, $u_i > 0$ for every buyer i.

The KKT conditions of the above convex program are:

(1) $\forall j \in G: \ p_j \geq 0,$

(2) $\forall i \in B, \ \forall j \in G, \ \forall k \in S_{ij}: \ q_{ijk} \geq 0,$

(3) $\forall j \in G: \ p_j > 0 \Rightarrow \sum_{i \in B} \sum_{k \in S_{ij}} x_{ijk} = 1.$

(4) $\forall i \in B, \ \forall j \in G, \ \forall k \in S_{ij}: \ q_{ijk} > 0 \Rightarrow x_{ijk} = l_{ijk},$

(5) $\forall i \in B, \ \forall j \in G \ \forall k \in S_{ij}: p_j + q_{ijk} \geq \frac{m_i . u_{ijk}}{u_i}.$

(6) $\forall i \in B, \ \forall j \in G \ \forall k \in S_{ij}: x_{ijk} > 0 \Rightarrow p_j + q_{ijk} = \frac{m_i . u_{ijk}}{u_i}.$

We will call p_j to be the *price* of good j, and q_{ijk} to be the *price differential*, which is unique for each buyer i, good j, and segment $k \in S_{ij}$. Also define the rate of a buyer i (r_i) to be $\frac{u_i}{m_i}$. Note that from equation (6), for any segment s_{ijk} for which x_{ijk} is positive: $r_i = \frac{u_i}{m_i} = \frac{u_{ijk}}{p_j + q_{ijk}} \leq \frac{u_{ijk}}{p_j}$. Thus if a segment s is allocated, fully or partially, to buyer i, its bang-per-buck value is at least r_i.

For the next two lemmas, assume that the x_{ijk}'s, u_i's, and p_j's satisfy the above KKT conditions. Also define the rate of the buyer i, r_i, to be equal to $\frac{u_i}{m_i}$.

Lemma 4 *The rate r_i is optimal for buyer i.*

Proof : Suppose that for some segment s_{ijk}, $\frac{u_{ijk}}{p_j} > r_i = \frac{u_i}{m_i}$, then from the 5^{th} KKT condition, we get that q_{ijk} is strictly positive, which from the 4^{th} KKT condition implies that $x_{ijk} = l_{ijk}$. Thus, from lemma 1 and lemma 2, we get that the rate $r_i = \frac{u_i}{m_i}$ is optimal for each buyer i. □

Lemma 5 *Under assumption 1, for every good j, p_j is strictly positive and j is exactly sold, i.e.,* $\sum_{i,k} x_{ijk} = 1$.

Proof : Suppose that the price of some good j is zero. Then from the 5^{th} condition above, $q_{ijk} > 0$ for every segment s_{ijk} for which $u_{ijk} > 0$. Thus along with 4^{th} condition, this will imply that $\sum_{i,k} x_{ijk} > 1$ which violates the constraint $\sum_{i,k} x_{ijk} \leq 1$ in the convex program. Thus the price of every good is strictly positive. Using 3^{rd} condition, this implies that every good is completely sold, i.e., $\sum_{i,k} x_{ijk} = 1$. □

From the above observations, finding an optimal solution to the above convex program (1) is equivalent to finding a price vector $\boldsymbol{p}$, a rate vector $\boldsymbol{r}$, and allocations of the goods to the buyers (vector $\boldsymbol{x}$) that satisfy the following equilibrium conditions.

1. For prices $\boldsymbol{p}$, the rate $r_i = \frac{u_i}{m_i}$ is optimal for each buyer i. Moreover, since $u_i > 0$, buyer i spends his money completely.

2. No portion of a segment s is sold to a buyer i, if $bpb(s) < r_i$.

3. All goods are sold out completely.

Thus, if prices $\boldsymbol{p}$ are optimal dual variables of the above convex program then these prices are also equilibrium prices.

Now, suppose we are given equilibrium prices $\boldsymbol{p}$; we will show that these prices are also optimal dual variables. Given equilibrium prices, there exists optimal rates $\boldsymbol{r}$ of buyers and allocation of goods $\boldsymbol{x}$ to the buyers, so that a segment s is allocated to buyer i only if $bpb(s) \geq r_i$. Moreover, if $bpb(s) > r_i$, then the segment s is fully allocated to buyer i. We will show the existence of variables q_{ijk}'s such that the KKT conditions are satisfied: if $\frac{u_{ijk}}{p_j} \geq \frac{u_i}{m_i}$, set q_{ijk} so that $\frac{u_{ijk}}{p_j+q_{ijk}} = \frac{u_i}{m_i}$, else set $q_{ijk} = 0$. It is not difficult to see that these q_{ijk}'s along with $\boldsymbol{p}, \boldsymbol{r}$, and $\boldsymbol{x}$ satisfy the KKT conditions. Thus prices $\boldsymbol{p}$ are optimal dual variables. This finishes the proof of theorem 3.

Theorem 6 *Program (1) is a rational convex program. Furthermore, rates r_i's are also rational numbers.*

Proof : The idea of the proof is to use the KKT conditions of convex program (1) to derive an exponential family of LPs such that the optimal solution to one of them is also an optimal solution to program (1). Since all parameters in this LP are rational, it has a rational solution. As such, KKT condition (6) is non-linear. However, by replacing $1/(p_j + q_{ijk})$ by a new variable, we can make it linear.

First observe that for any good j, we can assume that at least one segment s_{ijk} with $x_{ijk} > 0$ is such that $q_{ijk} = 0$. If not, we can raise p_j and lower the q_{ijk}'s without violating any of the KKT conditions, until this holds. Guess the x_{ijk}'s and q_{ijk}'s that are positive in an optimal solution to the convex program; now, each LP in the family corresponds to one such guess.

Corresponding to one such guess, the variables of the LP are the positive x_{ijk}'s and corresponding to each q_{ijk}, it has a variable r_{ijk} which represents $1/(p_j + q_{ijk})$. The LP are the following constraints. It has g equations corresponding to the 3rd KKT condition. Corresponding to each positive q_{ijk}, the LP has an equation corresponding to the $4th$ KKT condition. All inequalities corresponding to the $5th$ KKT condition; in these, $1/(p_j + q_{ijk})$ is written as r_{ijk} and u_i is written as $\sum_{j \in G} \sum_{k \in S_{ij}} u_{ijk} x_{ijk}$. For each positive x_{ijk}, the inequality written for the $5th$ KKT condition is also written as an equality. In addition, it has non-negativity constraints on the variables, x_{ijk}'s and r_{ijk}'s. Clearly, all these constraints are linear.

Now, the LP corresponding to the "correct" guess must be feasible and hence must yield a rational solution. This solution will satisfy all KKT conditions and hence will be an optimal solution to program (1). Hence this program is rational.

Finally, the rationality of the x_{ijk}'s implies rationality of u_i's, which implies rationality of the rates. □

From here on we will look at the design of a combinatorial algorithm to compute an equilibrium solution for the case when the supply of the goods is fixed. In section 5, to give an overview of the algorithm as well as introduce some key components of the main algorithm, we look at a simpler algorithm which converges in finitely many steps but we don't know if it has a polynomial time convergence or not. In section 6, we build upon this simpler algorithm to give an algorithm which converges in polynomial time. Since there could be multiple equilibrium solutions, in section 10, we characterize the set of all equilibrium solutions and show how to compute this set given any one equilibrium solution.

4 Basic Terminology

Let us now define some terminology which will be used throughout the algorithm.

Given prices $\boldsymbol{p} = \{p_1, p_2, \dots p_g\}$ and rates $\boldsymbol{r} = \{r_1, r_2, \dots r_n\}$, we say that a segment s_{ijk} is *forced*, *undesirable*, or *active* if and only if $\frac{u_{ijk}}{p_j} > r_i$, $\frac{u_{ijk}}{p_j} < r_i$, or $\frac{u_{ijk}}{p_j} = r_i$ respectively. We will use $\mathcal{F}^b(i)$, $\mathcal{A}^b(i)$, and $\mathcal{U}^b(i)$ to denote all the forced, active, and undesirable segments, respectively, of buyer i. Similarly for any good j, we define $\mathcal{F}^g(j)$, $\mathcal{A}^g(j)$, and $\mathcal{U}^g(j)$. Notice that for any buyer i and good j, there is at most one *active* segment, i.e., $|\mathcal{A}^b(i) \cap \mathcal{A}^g(j)| \le 1$. We will drop the superscript when it is clear from the context.

We will say *do all the forced allocations* to mean that allocate fully all the *forced* segments to their corresponding buyers, i.e. set $x_{ijk} = l_{ijk}$ for any segment s_{ijk} for which $\frac{u_{ijk}}{p_j} > r_i$. *Undesirable* segments are never allocated by the algorithm. All the *active* allocations are done by doing a max-flow computation in the network $N(\boldsymbol{p}, \boldsymbol{r})$ as described below.

Network $N(\boldsymbol{p}, \boldsymbol{r})$. Construct a directed network $N(\boldsymbol{p}, \boldsymbol{r})$ as follows. First do all the forced allocations. Let $amount(\mathcal{F}^g(j))$ be the quantity of good allocated during the forced allocations, and $amount(\mathcal{F}(i))$ be the amount of money buyer i pays to the middleman for the forced segments (which is the total utility buyer i gets from the forced segments divided by r_i). Network $N(\boldsymbol{p}, \boldsymbol{r})$ has a source s, a sink t and has vertex subsets B and G corresponding to the buyers and goods, respectively. For each good $j \in G$, there is an edge (s, j) of capacity $p_j.(1\text{-}amount(\mathcal{F}^g(j)))$, and for each buyer $i \in B$, there is an edge (i, t) of capacity $m_i - amount(\mathcal{F}(i))$. For each *active* segment s_{ijk}, include a directed edge (j, i) of capacity $l_{ijk}.p_j$.

All the *active* allocations are done by first computing a max-flow $\mathbf{f}$ in the above network, and then allocating amount a_{ij} of good j to buyer i, where $a_{ij} = \frac{f_{ji}}{p_j}$.

The following lemma is an easy observation.

Lemma 7 *Prices* $\boldsymbol{p}$ *and rates* $\boldsymbol{r}$ *are equilibrium prices and rates if and only both* $(s, B \bigcup G \bigcup t)$ *and* $(s \bigcup B \bigcup G, t)$ *are min-cuts in* $N(\boldsymbol{p}, \boldsymbol{r})$ *.*

Proof : First suppose that both $(s, B \bigcup G \bigcup t)$ and $(s \bigcup B \bigcup G, t)$ are min-cuts. This means that there is a way to send flow from s to t which saturates all the outgoing edges of s (meaning all the remaining goods can be sold) and all the incoming edges of t (meaning all the remaining money of the buyers can be spent). Now, by the definition of the network $N(\boldsymbol{p}, \boldsymbol{r})$, any segment s_{ijk} for which $\frac{u_{ijk}}{p_j} > r_i$, that segment is fully allocated to buyer i. Thus, prices $\boldsymbol{p}$ and rates $\boldsymbol{r}$ are equilibrium prices and rates. Now on the contrary, suppose that either $(s, B \bigcup G \bigcup t)$ or $(s \bigcup B \bigcup G, t)$ is not a min-cut. Clearly this implies that after allocating all the $forced$ segments, there is no way to allocate *active* segments that sells all the remaining goods and spends all the money exactly. Hence prices $\boldsymbol{p}$ and rates $\boldsymbol{r}$ doesn't form an equilibrium solution. $\square$

5 An Algorithm that Terminates

In this section, we give a simple algorithm that converges in finitely many steps. We give details for the polynomial time implementation in section 6.

The algorithm maintains a price vector $\boldsymbol{p}$ and a rate vector $\boldsymbol{r}$ at every point of the run of the algorithm. Given prices $\boldsymbol{p}$ and rates $\boldsymbol{r}$, an allocation of the goods to the buyers is determined by first doing a *forced* allocation and then doing allocation of the remaining goods by finding a max flow in the network $N(\boldsymbol{p}, \boldsymbol{r})$, as described in the section 4. We will make sure that the prices, the rates, and the corresponding allocations at any point of time are such that: 1) every good is sold out completely, 2) money spent by buyers is less than or equal to their budgets, 3) each buyer gets optimal utility for the money he spends and at the current prices. As the algorithm progresses, it will adjust the prices and the rates so as to increase the spending of the buyers while making sure that the first and the third condition are not violated.

More formally, we maintain the following invariant throughout the run of the algorithm.

Invariant: w.r.t current prices $\boldsymbol{p}$ and rates $\boldsymbol{r}$,

1. After the forced allocations, no good is sold more than 1 unit, and no buyer spends more money than his budget.
2. $(s, B \bigcup G \bigcup t)$ is a min-cut in the network $N(\boldsymbol{p}, \boldsymbol{r})$.

Using the *initialization* stage, algorithm starts with prices $\boldsymbol{p}$ and rates $\boldsymbol{r}$ which satisfies the above invariant. It then raises prices of the goods and drops the rates of the buyers such that the money spent by the buyers keeps on increasing, while maintaining the above invariant. When money of all the buyers is completely spent, i.e. $(s \bigcup B \bigcup G, t)$ is also a min-cut, we achieve a solution for which all the three equilibrium conditions are satisfied.

5.1 Initialization

In this section we will compute initial prices $\boldsymbol{p}$ and rates $\boldsymbol{r}$ for which the above invariant holds. Pick any buyer, say buyer i. To begin with, let us first find prices $\boldsymbol{p}$ and rate r_i so that buyer i demands all the goods. This can be achieved as follows: Let U be the utility obtained by buyer i if he is allocated all the goods. Set r_i to be equal to $\frac{U}{m_i}$. Now, for any good j, let k_j be the smallest k such that $\sum_{t=1}^{k} l_{ijt} \geq 1$, i.e. s_{ijk_j} is the segment which represents the value at $f_j^i(1)$. Set the price of a good j to be equal to $\frac{u_{ijk_j}}{U}$.

Let J_1 be the set of goods for which $p_j > 0$, i.e., for which $u_{ijk_j} > 0$. Similarly, let J_2 be the set of goods for which $p_j = 0$, i.e., for which $u_{ijk_j} = 0$. Fix the rates of other buyers ($r_{i'}$ $\forall i' \neq i$) high enough so that they don't demand any good from set J_1. This can be achieved by setting $r_{i'} > \frac{u_{i'jk}}{p_j}$ for every $j \in J_1$ and k. Now if $J_2 = \emptyset$, the prices $\boldsymbol{p}$ and rates $\boldsymbol{r}$ will satisfy the invariant. On the other hand, if $J_2 \neq \emptyset$, goods in J_2 will be desired by buyers other than i even if they have very high rates. In this case, we need to raise the prices of the goods in J_2. We give the details below.

Let $amount(\mathcal{F}(j))$ and $amount(\mathcal{A}(j))$ be the total amount of good j corresponding to all the segments in $\mathcal{F}(j)$ and $\mathcal{A}(j)$ respectively w.r.t the prices and rates defined above. Consider a good j in J_2 for which $amount(\mathcal{F}(j)) > 1$. Now as we raise the price p_j of good j, the segments in $\mathcal{A}(j)$ will move out and some of the segments from $\mathcal{F}(j)$ will move to $\mathcal{A}(j)$. We keep on raising the price p_j until the following holds: $amount(\mathcal{F}(j)) \leq 1$, and $amount(\mathcal{F}(j)) + amount(\mathcal{A}(j)) \geq 1$.

We return the final prices $\boldsymbol{p}$ and $\boldsymbol{r}$ as the prices and rates of the initialization stage. It is easy to see that the following claim holds after the initialization stage.

Claim 8 *Prices* $\boldsymbol{p}$ *and rates* $\boldsymbol{r}$ *computed in the initialization stage satisfy the Invariant.*

5.2 Raising Prices

In this stage we will work with network $N(\boldsymbol{p}, \boldsymbol{r})$ for any given prices $\boldsymbol{p}$ and rates $\boldsymbol{r}$. Recall that the network $N(\boldsymbol{p}, \boldsymbol{r})$ is constructed by first allocating all the forced segments. This stage consists of multiple rounds. In each round, we will compute a max flow f and find subsets $I \subseteq B$ and $J \subseteq G$ such that raising the prices of the goods in G and decreasing the rates of the buyers in I doesn't violate the invariant.

More formally: Find a max flow in the network $N(\boldsymbol{p}, \boldsymbol{r})$. Let $(s \bigcup (B - I) \bigcup (G - J), t \bigcup I \bigcup J)$ be a min-cut in $N(\boldsymbol{p}, \boldsymbol{r})$ with the maximum number of nodes on the 's' side of the cut. Here $I \subseteq B$, and $J \subseteq G$. The property of this cut is that buyers in I have strictly positive surplus money, and all the edges from $G - J$ to I are saturated. Now as we multiply the prices of goods in J by a factor x and divide rates of buyers in I by the same factor x, and continuously raise x, then the edges between $(B - I)$ and J will disappear since the corresponding segment will become undesirable. Moreover, the edges between I and $G - J$ will also disappear, as the corresponding segments will become forced. Since all the edges that become forced are already saturated and all the edges that become undesirable have no flow, we can raise the x continuously without violating the invariant until one of the following event happens.

1. An undesirable segment s_{ijk} becomes active for some good $j \in (G - J)$ and $i \in I$. This will happen because for an undesirable segment, the gap between $\frac{u_{ijk}}{p_j}$ and r_i keeps on decreasing as we lower the r_i. Add edge (j,i) to the network $N(\boldsymbol{p}, \boldsymbol{r})$ of capacity l_{ijk}.(current price of j), and start a new round.

2. A forced segment s_{ijk} becomes active for some good $j \in J$ and $i \in (B - I)$. This will happen because for any such segment, the gap between $\frac{u_{ijk}}{p_j}$ and r_i keeps on decreasing as we increase the p_j. Add edge (j,i) to the network $N(\boldsymbol{p}, \boldsymbol{r})$ with appropriate capacity, update the capacity of edges (s, j) and (i, t), and start a new round.

3. When x becomes so big that raising it any further would violate the second invariant (when the network has a min-cut smaller than $(s, B \cup G \cup t)$). This happens when some subsets $S \subseteq J$ and $T \subseteq I$ $(T \supset \emptyset)$ goes *tight*; subsets S and T are said to be *tight* if in the current network, $\Gamma(T) = S$, $\Gamma(S) \subseteq T$, and total capacity of edges (s, S) equals (T, t). Here function $\Gamma(.)$ is the neighborhood function. Note that $\Gamma(S) \subset T$ could happen along with $\Gamma(T) = S$ if some buyer in T does not have any edge to the goods in G in the current network. It could also happen that $\Gamma(T) = S = \emptyset$ when buyers in T exhausted their money when we lowered their rate. In either case, at this point we can no longer increase x as increasing it any further would make the value of cut $(s \bigcup (B - I) \bigcup (G - J) \bigcup T \bigcup S, t \bigcup (I - T) \bigcup (J - S))$ smaller than the value of

cut $(s, B \cup G \cup t)$. Moreover, tightness implies that there exists a min-cut with a strictly smaller set $I - I'$ of buyers on the t side of the cut, which we will find in the next round. Thus we start a new round if this event happens.

Clearly, in the above algorithm, we are always making progress- total unspent money of the buyers keeps decreasing. So if we can prove that the invariant holds throughout the algorithm, after finitely many steps, $(s \bigcup B \bigcup G, t)$ will also be a min-cut. By Lemma 7, this would imply that the prices and rates are in equilibrium.

We give a pseudo code of the algorithm in the box below. It uses the following subroutine:

Find sets(I): Sets $I \subseteq B$ and $J \subseteq G$ are initialized so that $(s \bigcup (B - I) \bigcup (G - J), t \bigcup I \bigcup J)$ forms a min-cut in $N(\boldsymbol{p}, \boldsymbol{r})$ with the maximum number of nodes on the 's' side of the cut. All edges are removed from goods in J to buyers in $B - I$. Also remove the edges from goods in $G - J$ to buyers in I, and *force* allocate the corresponding segments.

Algorithm 9

1. **Initialization**.
2. **(New Round)** Compute a max flow in $N(\boldsymbol{p}, \boldsymbol{r})$.
3. If $\forall i \in B$, $spend(i) = m_i$, then END.
4. **Find sets(I)**.
5. Multiply the prices of the goods in J and divide the rates of the buyers in I by x.

   Initialize $x \leftarrow 1$, and increase x continuously until one of the following happens:

   (i) An undesirable segment s_{ijk} becomes active, for $j \in (G - J)$ and $i \in I$.
   Add an edge (j, i) to $N(\boldsymbol{p}, \boldsymbol{r})$ of capacity l_{ijk}*(current price of good j).
   Go to Step 2.

   OR

   (ii) A forced segment s_{ijk} becomes active, for $j \in J$ and $i \in (B - I)$.
   Add an edge (j, i) to $N(\boldsymbol{p}, \boldsymbol{r})$ of capacity l_{ijk}*(current price of good j), update the capacity of edges (s, j) and (i, t).
   Else, **Update sets(I)** and go to Step 2.

   OR

   (iii) (S, T) goes tight, where $S \subseteq J$ and $T \subseteq I$.
   Go to Step 2.

6 Polynomial Time Implementation

In this section, we give the algorithm which runs in polynomial time. A crucial ingredient for making the algorithm run in polynomial time is *balanced flows*.

Balanced Flows. Balanced flow was first used by [9] for the market equilibrium, and was later by

[24] and [23] for the spending constraint market equilibrium and Nash bargaining, respectively.

For simplicity, denote the current network, $N(\boldsymbol{p}, \boldsymbol{r})$, by simply N. Given a feasible flow f in N, let $R(f)$ denote the residual graph w.r.t. f. Define the *surplus* of buyer i w.r.t. flow f in network N, $\theta_i(N, f)$, to be the residual capacity of the edge (i, t) with respect to flow f in network N, i.e., m_i minus the flow sent through the edge (i, t). The *surplus vector w.r.t. flow* f is defined to be $\theta(N, f) := (\theta_1(N, f), \theta_2(N, f), \ldots, \theta_n(N, f))$. Let $\|v\|$ denote the l_2 norm of vector v. A *balanced flow* in network N is a flow that minimizes $\|\theta(N, f)\|$. A balanced flow must be a max-flow in N because augmenting a given flow can only lead to a decrease in the l_2 norm of the surplus vector.

A balanced flow can be computed in N using at most n max-flow computations. It is easy to see that all balanced flows in N have the same surplus vector. The key property of a balanced flow that our algorithm will rely on is that a maximum flow f in N is balanced iff it satisfies Property 1:

Property 1: For any two buyers i and j, if $\theta_i(N, f) < \theta_j(N, f)$ then there is no path from node i to node j in $R(f) - \{s, t\}$.

We define the following notion of *dominating* buyers.

Dominating Buyers. For a given balanced flow f, a set of buyers I is said to *dominating* if in the residual network there is no path from buyers in $B - I$ to buyers in I.

Dominating buyers satisfy the following crucial property.

Property 2: Let I be a set of dominating buyers, and let j be any good such that j has outgoing edges to both I and $B - I$ in the current network. Now, either all the edges of j to I are saturated or no edge from j to $B - I$ carry any flow.

Now we describe the algorithm which runs in a polynomial time. First step, i.e. initialization, is same as the one described earlier. The rest of the algorithm consists of various rounds. In each round we raise the prices of some goods J and lower the rates of some buyers I; we will do it until one of the events as described in the previous section happens. We then update the network if required, and recompute the balanced flow in the new network. If in the balanced flow, a buyer in the set I spends his money completely, we call the round to be a *phase*. Otherwise we call the round to be an *iteration*.

Each round is of the following form:

Round. Let f be a balanced flow in the current network $N(\boldsymbol{p}, \boldsymbol{r})$. Let I be a set of *dominating* buyers and let J be the set of goods which have edges with residual capacity to buyers in I. Note that due to property 2, none of the edges from J to $B - I$ will carry any flow. Now we will increase the price of goods in J and decrease the $\boldsymbol{r}$'s of buyers in I at the same rate x, i.e., prices get multiplied by x and $\boldsymbol{r}$'s get divided by x, where $x > 1$. Note that the edges between $G - J$ and I are saturated (by definition) in the current network and we can make them forced, as the decrease in $\boldsymbol{r}$ of buyers in I will force them to become forced anyway. Also we can drop the edges between J and $B - I$, as these edges will also disappear from the current network when the rates of the buyers in I will be decreased.

When we raise the price and decrease the rates, the three events as described in section 5 can happen. Each of those events can be computed in polynomial time (see section 8). Full algorithm is given in figure 10. The two subroutines used in the algorithm are:

- **Find sets(I):** Sets $I \subseteq B$ and $J \subseteq G$ are initialized as follows.

$$I \leftarrow \arg\max_{i \in B}\{\theta_i\}, \quad J \leftarrow \Gamma^+(I)$$

  Here $\Gamma^+(.)$ is a function which, for a set of buyer I, returns the set of goods J that have edges with positive residual capacity to buyers in I. All edges are removed from goods in J to buyers in $B - I$. Also remove the edges from goods in $G - J$ to buyers in I, and *force* allocate the corresponding segments. Note that the set I of buyers are dominating buyers.

- **Update sets(I):** Find the set I' of all buyers in $B - I$ that have residual paths to buyers in I. Update

$$I \leftarrow (I \cup I'), \quad J \leftarrow \Gamma^+(I)$$

  All edges are removed from goods in J to buyers in $B - I$. Also remove the edges from goods in $G - J$ to buyers in I, and *force* allocate the corresponding segments. Note that the set I of buyers are dominating buyers.

Algorithm 10

1. **Initialization**.
2. **(New Phase)** Compute a balanced flow in $N(\boldsymbol{p}, \boldsymbol{r})$.
3. If $\forall i \in B$, $spend(i) \ = \ m_i$, then END.
4. **Find sets(I)**.
5. **(New Iteration)**

   Multiply the prices of the goods in J and divide the rates of the buyers in I by x.

   Initialize $x \leftarrow 1$, and increase x continuously until one of the following happens:

   (i) An undesirable segment s_{ijk} becomes active, for $j \in (G-J)$ and $i \in I$.
   Add an edge (j,i) to $N(\boldsymbol{p}, \boldsymbol{r})$ of capacity l_{ijk}∗(current price of good j) and compute a balanced flow in it.
   If for some $i \in I$, $spend(i) = m_i$, go to Step 2.
   Else, **Update sets(I)** and go to Step 5.

   OR

   (ii) A forced segment s_{ijk} becomes active, for $j \in J$ and $i \in (B-I)$.
   Add an edge (j,i) to $N(\boldsymbol{p}, \boldsymbol{r})$ of capacity l_{ijk}∗(current price of good j), update the capacity of edges (s,j) and (i,t), and compute a balanced flow in it.
   If for some $i \in I$, $spend(i) = m_i$, go to Step 2.
   Else, **Update sets(I)** and go to Step 5.

   OR

   (iii) (S, T) goes tight, where $S \subseteq J$ and $T \subseteq I$.
   Go to Step 2.

7 Proof of Correctness and Termination

Let z be the total number of segments, and let U denote the highest value of u_{ijk} over all i, j, and k. Also let L to be the highest value of the numerator or denominator of l_{ijk} over all i, j, and k. Let $\Delta = n(2U)^n L^n$, where $n = n_G + n_B$.

Lemma 11 *The algorithm maintains the Invariants throughout.*

Proof : Suppose that at the beginning of a phase or an iteration, both the invariants are satisfied. It is easy to see that the first invariant is satisfied at the end of a phase or an iteration. Let us show that the second invariant is also satisfied at the end of a phase or an iteration. Firstly, it is easy to see that at the start of any phase or iteration, the set I of buyers are dominating buyers. Thus the set J of goods who price is increased are not allocated to buyers in $B-I$ in the *current* network. Thus when we raise the price of goods in J and lower the rates of I, goods in J can be still be sold to buyers in I until either some set goes tight or an edge appears between J and $B-I$. Segments of goods in $G - J$ that are sold to buyers in I are fully saturated at the start of any phase or iteration. Thus when we lower the rates of buyers in I, these segments get forced allocated, and hence goods in $G - J$ are also fully allocated. Since all the goods can be sold out completely at the end of a phase or an iteration, the second invariant holds.

This, along with the fact that initialization maintains both the invariants, proves the lemma.

□

Lemma 12 *The total number of iterations in a phase is bounded by 2z.*

Proof : Notice that, in the network N, a forced segment becomes active only between sets J and $B - I$. An undesirable segment becomes active only between I and $G - B$. An active segment becomes forced only between sets I and $G - B$, and it becomes undesirable only between sets J and $B - I$. Also notice that once a buyer enters set I, he remains in set I until the end of the phase. Consider a segment s between buyer i and good j. We want to bound the number of times segment s changes its state (forced, active, undesirable) within a given phase. It is not difficult to see that the maximum number of times segment s can change a state is 2. Since there are z segments, there are at most $2z$ iterations in a phase. □

Theorem 13 *The algorithm terminates with an equilibrium.*

Proof : Note that in every phase, the rate of some buyers will go down. Now the claim is that the rates cannot go arbitrarily low. This is because the algorithm maintains the invariant that all the goods are completely sold out, thus very small rates will imply that some buyer has over spent, violating the first invariant and hence contradict lemma 11. Thus the algorithm will terminate in finitely many steps, and since it terminates only when all the buyers have spent their money, we achieve an equilibrium solution. □

8 Computing x^* via Min-Cuts in Parametric Networks

We will show how to compute the value of x^*, the value of x at which a new event (as described in section) occurs. Note that the value of x at which the first two events occur is easy to compute. In this section, we will give an algorithm to compute the value of x at which some subset (S, T) goes tight where $S \neq \emptyset$. The case when $S = \emptyset$ is also easy to compute. Let $S^* \subseteq J$ denote the tight set. Throughout this section, $\boldsymbol{p}$ will denote prices at the beginning of the current phase, i.e., at $x = 1$. Network $W(\boldsymbol{p}, \boldsymbol{r})$ is the subnetwork of $N(\boldsymbol{p}, \boldsymbol{r})$ on $\{s\} \cup J \cup I \cup \{t\}$. At $x = 1$, $(s, J \cup I \cup t)$ is a min-cut in $W(\boldsymbol{p}, \boldsymbol{r})$. Let $m(T)$ (for $T \subseteq B$) be the total initial money of the buyers in set T, and $m'(T)$ be the total money spent on the forced segments by the buyers in set T. Also let $c(G_1, B_1)$ be the capacity of the edges going from set $G_1 \subseteq G$ to set $B_1 \subseteq B$ in the current network. For $S \subseteq G$, define

$$\text{best}(S, x) = \min_{T \subseteq \Gamma(S)} \{\boldsymbol{m}(T) - x \cdot \boldsymbol{m}'(T) + x \cdot c(S; \Gamma(S) - T)\},$$

For the rest of the section, we will abuse the notation of p_j and use it to denote $p_j \cdot (1 - amount(\mathcal{F}^g(j)))$, i.e., price of the remaining good after forced allocations.

Lemma 14 *The smallest value of x at which a subset goes tight in $W(\boldsymbol{p}, \boldsymbol{r})$ is given by*

$$x^* = \min_{x \geq 1, \emptyset \neq S \subseteq J} \frac{\text{best}(S, x)}{\boldsymbol{p}(S)},$$

and the unique maximal set minimizing the above expression is S^*.

Proof : Let $x = \beta$ be the smallest value of x at which a new min-cut appears in $W(\boldsymbol{p}, \boldsymbol{r})$. Let the min-cut maximizing the s side be $(s \cup J_1 \cup I_1, J_2 \cup I_2 \cup t)$. Since $W(\boldsymbol{p}, \boldsymbol{r})$ satisfies Invariant 2 at $x = \beta$, we get that for any subset $S \subseteq J$,

$$\boldsymbol{p}(S) \cdot \beta \leq \text{best}(S, \beta).$$

Since Invariant 2 holds and $(s \cup J_1 \cup I_1, J_2 \cup I_2 \cup t)$ is a min-cut in $W(\boldsymbol{p}, \boldsymbol{r})$ at $x = \beta$, J_1 must be a tight set and therefore,

$$\boldsymbol{p}(J_1) \cdot \beta = \text{best}(J_1).$$

The lemma follows. □

Lemma 15 *The following hold:*

- *If $x \leq x^*$, then $(s, J \cup I \cup t)$ is a min-cut in $W'(\boldsymbol{p}, \boldsymbol{r})$.*
- *If $x > x^*$, then for any min-cut $(s \cup J_1 \cup I_1, J_2 \cup I_2 \cup t)$ in $W'(\boldsymbol{p}, \boldsymbol{r})$, $S^* \subseteq J_1$.*

Proof : By the definition of x^*, if $x \leq x^*$, $\forall S \subseteq J: \ \boldsymbol{p}'(S) \cdot x \leq \text{best}(S)$. Therefore, $(s, J \cup I \cup t)$ is a min-cut in $W(\boldsymbol{p}, \boldsymbol{r})$.

Next, suppose that $x > x^*$, and consider a min-cut $(s \cup J_1 \cup I_1, J_2 \cup I_2 \cup t)$ in $W(\boldsymbol{p}, \boldsymbol{r})$. First observe that $S^* \subseteq J_2$ contradicts the minimality of this cut:

since $\boldsymbol{p}(S^*) \cdot x > \text{best}(S^*, x)$ [1], a smaller cut results if S^* is moved into J_1, and set T that minimizes $\text{best}(S^*, x)$ is moved into I_1.

Let $S^* \cap J_1 = S_1$, $S^* \cap J_2 = S_2$, and suppose that $S_2 \neq \emptyset$. Observe that if $\Gamma(S_1) \cap \Gamma(S_2) = \emptyset$, then $\text{best}(S_1, x^*) + \text{best}(S_2, x^*) \leq \text{best}(S^*, x^*)$. To achieve a similar effect even if $\Gamma(S_1) \cap \Gamma(S_2) \neq \emptyset$ let us define for $S \subseteq J_2$:

$$\text{best}'(S, x) = \min_{T \subseteq \Gamma(S) - I_1} \{\boldsymbol{m}(T) - x \cdot \boldsymbol{m}'(T) + x \cdot c(S; \Gamma(S) - I_1 - T) - x \cdot c(S_1; T)\},$$

and let us define $\text{bestT}'(S, x)$ to be a maximal subset of $\Gamma(S)$ optimizing the above expression. Now observe that

$$\text{best}(S_1, x^*) + \text{best}'(S_2, x^*) \leq \text{best}(S^*, x^*).$$

[1]This follows from the fact that $\boldsymbol{p}(S^*) \cdot x^* = \text{best}(S^*, x^*)$ and $x > x^*$

Hence,

$$\text{best}(S_1, x^*) + \text{best}'(S_2, x^*) \leq x^* \cdot \boldsymbol{p}(S^*).$$

If $\text{best}'(S_2, x) < x \cdot \boldsymbol{p}(S_2)$, then a smaller cut can be found by moving S_2 into J_1, and moving $\text{bestT}'(S_2)$ from I_2 to I_1. Therefore,

$$\text{best}'(S_2, x) \geq x \cdot \boldsymbol{p}(S_2).$$

This along with the fact that $x > x^*$ implies that

$$\text{best}'(S_2, x^*) > x^* \cdot \boldsymbol{p}(S_2).$$

Combining with the previous inequality, we get

$$\text{best}(S_1, x^*) < x^* \cdot \boldsymbol{p}(S_1),$$

which contradicts the definition of x^*. Therefore, $S_2 = \emptyset$ and hence $S^* \subseteq J_1$. □

Define

$$\overline{x} = \min_{T \subseteq \Gamma(J)} \left\{ \frac{\boldsymbol{m}(T)}{\boldsymbol{p}(J) + \boldsymbol{m}'(T) - c(J; \Gamma(S) - T)} \right\}.$$

We will use $\overline{x}$ as an initial guess of x^*. The reason for working with $\overline{x}$ is that in $W(\boldsymbol{p}, \boldsymbol{r})$ if the min cut turns out to be of the form $(s \cup J \cup I_1, I_2 \cup t)$ for $x = \overline{x}$, its capacity is same as the cut $(s \cup J \cup I, t)$. This property will play a critical role in the next lemma. The expression $c(J; \Gamma(S) - T)$ can be re-written as $c(J; \Gamma(S)) - c(J; T)$. Thus computing $\overline{x}$ is same as maximizing

$$\frac{\boldsymbol{p}(J) + \boldsymbol{m}'(T) + c(J; T) - c(J; \Gamma(S))}{\boldsymbol{m}(T)},$$

which can be easily computed.

Lemma 16 *Let $x = \overline{x}$ and let the minimal min-cut in $W(\boldsymbol{p}, \boldsymbol{r})$ (i.e., the unique min-cut minimizing the s side) be $(s \cup J_1 \cup I_1, J_2 \cup I_2 \cup t)$. If $J_1 = I_1 = \emptyset$ then $x = x^*$ and $S^* = J$. Otherwise, $x > x^*$ and J_1 is a proper subset of J.*

Proof : Clearly, $x \geq x^*$. If the min-cut is $(s, J \cup I \cup t)$ then by Lemma 15, $x = x^*$. Let T^* be the set for which $\overline{x}$ is defined. It is not difficult to see that $\text{best}(J, x^*) = \boldsymbol{m}(T^*) - x^* \cdot \boldsymbol{m}'(T^*) + x^* \cdot c(J; \Gamma(S) - T^*)$. Therefore $\text{best}(J, x^*)$ equals $x^* \cdot \boldsymbol{p}(J)$, and hence $S^* = J$.

If $(s, J \cup I \cup t)$ is not a min-cut, then by Lemma 15, $x > x^*$. Suppose $J_1 = J$ and the min-cut is $(s \cup J \cup I_1, I_2 \cup t)$. By the property stated above, the capacity of this cut is $\boldsymbol{m}(T) - x \cdot \boldsymbol{m}'(T) + x \cdot c(J; \Gamma(S) - T)$. For the chosen value of x, the capacity of $(s, J \cup I \cup t)$ is $x \cdot \boldsymbol{p}(J) = \boldsymbol{m}(T) - x \cdot \boldsymbol{m}'(T) + x \cdot c(J; \Gamma(S) - T)$ contradicting the fact that it is not a min-cut. Hence J_1 is a proper subset of J. □

In the setting of Lemma 16, suppose $x > x^*$ and $(s \cup J_1 \cup I_1, J_2 \cup I_2 \cup t)$ is the min-cut in $W(\boldsymbol{p}, \boldsymbol{r})$. From the network $W(\boldsymbol{p}, \boldsymbol{r})$, we will construct a new network $\overline{W}(\boldsymbol{p}, \boldsymbol{r})$ which is the induced sub-network on the vertex se $\{s\} \cup J_1 \cup \Gamma(J_1) \cup \{t\}$. From the above lemma, we get the following corollary:

Corollary 17 $$x^* = \min_{x \geq 1, \emptyset \neq S \subseteq J_1} \frac{\text{best}(S, x)}{\boldsymbol{p}(S)},$$
and the unique maximal set minimizing the above expression is S^*.

Theorem 18 x^* *and* S^* *can be found using at most* n *max-flow computations.*

Proof : Let
$$x = \min_{T \subseteq \Gamma(J)} \left\{ \frac{\boldsymbol{m}(T)}{\boldsymbol{p}(J) + \boldsymbol{m'}(T) - c(J; \Gamma(S) - T)} \right\}$$
and compute a min-cut in $W(\boldsymbol{p}, \boldsymbol{r})$. If $(s, J \cup I \cup t)$ is a min-cut in $W(\boldsymbol{p}, \boldsymbol{r})$, then by Lemma 16, $x^* = x$ and $S^* = J$. Otherwise, $x > x^*$. Now, by Corollary 17, it is sufficient to recurse on the network $\overline{W}(\boldsymbol{p}, \boldsymbol{r})$, which has fewer goods. □

9 Establishing Polynomial Running Time

Let N be the network obtained after making all changes to edges during an iteration. Let flow f be the flow at the beginning of the iteration and f^* the flow at the end of the iteration after making edge changes and finding a balanced flow. We will use the following lemma from [24].

Lemma 19 *(Lemma 24 in [24]) If* f *and* f^* *are respectively a feasible and a balanced flow in* N *such that* $\theta_i(N, f^*) = \theta_i(N, f) - \delta$, *for some* $i \in B$ *and* $\delta > 0$, *then* $\|\theta(N, f)^{*2}\| \leq \|\theta(N, f)^2\| - \delta^2$.

Let N_0 denote the network at the beginning of a phase. Assume that a phase consists of a total of at most k iterations, and that N_t denotes the network at the end of iteration t. Let f_t be a balanced flow in N_t and let I_t denote the set I in the network N_t, for $0 \leq t \leq k$. Let f'_t be the flow f_t augmented as follows: if in N_{t+1}, a frozen segment becomes active, then add the flow corresponding to this segment to f_t.

Lemma 20 f'_t *is a feasible flow in* N_{t+1}, *for* $0 \leq t < k$.

Proof : Each of the following actions that occur in an iteration can only lead to a network that supports an augmented max-flow:

- Lowering the rates of buyers in I, and raising the prices of goods in J.
- Adding an edge when an undesirable segment between I and $G - J$ becomes active.
- Adding an edge when a frozen segment s_{ijk} between J and $B - I$ becomes active. In this case we update the capacities of edges (s, j) and (i, t) in network N_{t+1} which accommodates the extra flow required on the edge (j, i).

The lemma follows. □

Now,from Lemmas 19 and 20, we get the following corollary:

Corollary 21 $\|\theta(N_t)\|$ *is monotonically decreasing with* t.

Let δ_t denote the minimum surplus of a buyer in I_t in network N_t, for $0 \leq t \leq k$; clearly, $\delta_0 = \delta$, surplus at the beginning of a phase, and $\delta_k = 0$.

Lemma 22 *If* $\delta_{t-1} > \delta_t$ *then there exists an* $i \in I_{t-1}$ *such that* $\theta_i(N_{t-1}, f_{t-1}) - \theta_i(N_t, f_t) \geq \delta_{t-1} - \delta_t$.

Proof : Consider the residual network corresponding to the balanced flow computed at the end of iteration t. By the definition of I_t, every vertex $v \in I_t \setminus I_{t-1}$ can reach a vertex $i \in I_{t-1}$ in the residual network and therefore, by Property 1, $\theta_v(N_t, f_t) \geq \theta_i(N_t, f_t)$. This means that the minimum surplus δ_t is achieved by a vertex i in I_{t-1}. Hence the surplus of vertex i decreases by at least $\delta_{t-1} - \delta_t$ during iteration t. □

Lemma 23 *If* $\delta_t > \delta_{t+1}$ *then* $\|\theta(N_t)\|^2 - \|\theta(N_{t+1})\|^2 \geq (\delta_t - \delta_{t+1})^2$, *for* $0 \leq t < k$.

Proof : By Lemma 22, if $\delta_t > \delta_{t+1}$ then there is a buyer i whose surplus drops by $\delta_t - \delta_{t+1}$ in going from f_t to f_{t+1}. By Lemma 20, f_t' is a feasible flow in N_{t+1}. Finally, Lemma 19 gives the desired conclusion. □

Lemma 24

$$\|\theta(N_0)\|^2 - \|\theta(N_k)\|^2 \geq \frac{\delta^2}{2z}.$$

Proof : The left hand side can be written as a telescoping sum in which each term is of the form $\|\theta(N_t)\|^2 - \|\theta(N_{t+1})\|^2$. By Corollary 21, each of these terms is nonnegative.

Consider only those terms in which the difference $\delta_t - \delta_{t+1} > 0$. The sum of their squares is minimized when all these differences are equal. Using Lemma 23 and the fact that $\delta_0 = \delta$ and $\delta_k = 0$, yields

$$\|\theta(N_0)\|^2 - \|\theta(N_k)\|^2 \geq \frac{\delta^2}{k}.$$

By Lemma 12, $k \leq 2z$. The lemma follows. □

Lemma 25 *In a phase, the* l_2^2*-norm of the surplus vector drops by a factor of*

$$\left(1 - \frac{1}{n_B(2z)}\right).$$

Proof : From Lemma 24 and the fact that $\|\theta(N_0)\|^2 \leq n_B\delta^2$,

$$\|\theta(N_k)\|^2 \leq \|\theta(N_0)\|^2 - \frac{n_B\delta^2}{n_B.2z} \leq \|\theta(N_0)\|^2 - \frac{\|\theta(N_0)\|^2}{n_B.2z}$$

$$\leq \|\theta(N_0)\|^2 \left(1 - \frac{1}{n.2z}\right).$$

The lemma follows. □

Lemma 26 *If a phase terminates with tight set $S \subseteq J$ with $S \neq \emptyset$, then the prices of goods in S are rational numbers with denominators $\leq \Delta$.*

Proof : Suppose set $(S, T \subseteq I)$ goes tight at the end of phase. Without loss of generality assume that (S, T) form a connected component in the current network. Let j, j' be any two goods in J. If j reaches j' with a path of length $2l$, then $r_{j'} = ap_j/b$ where a and b are products of the l rates of the corresponding segments. Since alternate edges of this path contribute to a and b, we can partition the rates in this subgraph into two sets such that a and b use rates from distinct sets. Similarly we can write the rate of a buyer i in T, i.e. r_i, as b/ap_j, where a and b are products of the $l-1$ and l rates respectively of the corresponding segments. For the rest of the proof, let $p_j = \alpha_j p$ for any good j in S, and $r_i = \beta_i/p$ for any buyer i in T. Now by the definition of tight set:

$$p(S) = m(T) - m'(T) + c(S; \Gamma(S) - T)$$

Thus,

$$\sum_{j \in S} \alpha_j p = m(T) - \sum_{i \in T} U_i.p/\beta_i + \sum_{e=(ijk), j \in S, i \in T} l_{ijk}\alpha_j p$$

Here U_i is the total utility allocated to buyer i using frozen segments. Clearly, the denominator of p is $\leq n(U)^n L^n$, and thus denominator of p_j is $\leq \Delta$.

□

Corollary 27 *Consider two phases P and P', not necessarily consecutive, such that good j lies in the newly tight sets at the end of P as well as P'. Then the increase in the price of j, going from P to P', is $\geq 1/\Delta^2$.*

Proof : Let the prices of j at the end of P and P' be p/q and r/s, respectively. Clearly, $r/s > p/q$. Since $q \leq \Delta$ and $s \leq \Delta$, therefore the increase in price of j equals

$$\frac{r}{s} - \frac{p}{q} \geq \frac{1}{\Delta^2}.$$

□

Lemma 28 *Once the square of the surplus vector drops below $1/\Delta^4$, an equilibrium state is reached in at most $n_B \cdot n_G$ phases.*

Proof : To prove this lemma, we will slightly modify the algorithm. When a set (S, T) goes tight with set $S = \emptyset$ and $T \subset I$, then instead of ending the phase there we will define $I = I - T$ and continue with the phase. Therefore a phase either ends with $S \neq \emptyset$ or $T = I$. Note that in any consecutive sequence of n_B phases, there must be a phase in which $S \neq \emptyset$, otherwise all the buyers must have exhausted their money and hence an equilibrium state is achieved. Therefore in a consecutive $n_B \cdot n_G$ phases, two of the phases must have some common good j in the tight set. Now the lemma follows from the Corollary 27. □

Theorem 29 *The algorithm finds equilibrium prices and allocations using*

$$O\left(n_B^2 \cdot z^2(\log n + n \log U + n \log L + \log M)\right)$$

max-flow computations.

Proof : By Lemma 25, the square of the surplus vector drops by a factor of two after $O(zn)$ phases. At the start of the algorithm, the square of the surplus vector is at most M^2. Once its value drops below $1/\Delta^4$, by lemma 28, equilibrium prices are attained in $n_G * n * B$ phases. Therefore the number of phases is bounded by

$$O(n_B^2 \cdot z \log(\Delta^4 M^2)) \;=\; O(n_B^2 \cdot z(\log n + n \log U + n \log L + \log M)).$$

By Lemma 12 each phase consists of at most $2z$ iterations and by Theorem 18 each iteration requires n max-flow computations. The theorem follows. □

10 Characterizing the Set of All Equilibria

In this section, we will show how to find the set of all equilibria given any one equilibrium solution. This, in conjunction with our algorithm, would imply that we can find the set of all equilibria.

Consider an equilibrium price vector $\boldsymbol{p}$ and rate vector $\boldsymbol{r}$. Also let $\mathbf{x}$ to be an equilibrium allocation vector for prices $\boldsymbol{p}$ and rates $\boldsymbol{r}$. Note that since the objective function of the convex program is strictly concave, the optimal utility vector is unique. Hence the equilibrium rate vector $\boldsymbol{r}$ is also unique. Now, consider the active segments of some good j, i.e., the set $\mathcal{A}^g(j)$. If $x_{ijk} \neq l_{ijk}$ for some segment $s_{ijk} \in \mathcal{A}^g(j)$, call good j to be *frozen*. Otherwise call good j to be *flexible* (if $\mathcal{A}^g(j) = \emptyset$, then also we call j to be *flexible*).

For any flexible good j, we can raise its price, without violating any of the KKT conditions, until a forced segment corresponding to j becomes active. Call the price at which this happens to be $upper(j)$. Also if $\mathcal{A}^g(j) = \emptyset$, we can keep lowering the price of j until an undesirable segment corresponding to j becomes active. Call this price $lower(j)$. If $\mathcal{A}^g(j) \neq \emptyset$, then set $lower(j) = p_j$.

Theorem 30 *Given any equilibrium prices* $\boldsymbol{p}$ *and rates* $\boldsymbol{r}$*, the set* $\mathcal{P} = \{(\boldsymbol{p}', \boldsymbol{r}) \mid$ *for each frozen good* $j, p'_j = p_j$ *and for each flexible good* $j, p'_j \in [lower(j), upper(j)]\}$ *characterizes the set of all equilibrium prices and rates.*

Proof : It is not difficult to see that an allocation of goods w.r.t prices $\boldsymbol{p}$ is also a valid allocation w.r.t any other prices $\boldsymbol{p}' \in \mathcal{P}$, i.e., these allocations and prices $\boldsymbol{p}'$ are still feasible for the KKT conditions. Hence the above set of prices are equilibrium prices. Now let us argue that no other prices are equilibrium prices. Suppose $\boldsymbol{p'} \notin \mathcal{P}$ is an equilibrium price vector. That means that for some good j, one of the following holds:

- Good j is frozen, and $p'_j \neq p_j$. If $p'_j > p_j$, clearly in that case good j cannot be fully sold. If $p'_j < p_j$ then the total size of the segments in the forced set $\mathcal{F}^g(j)$ is more than 1, hence good j is over sold.
- Good j is flexible, and $p'_j > upper(j)$. Again in this case, good j cannot be fully sold.
- Good j is flexible, and $p'_j < lower(j)$. In this case, the total size of the segments in the forced set $\mathcal{F}^g(j)$ is more than 1, hence good j is over sold.

Therefore, a price vector $\boldsymbol{p'} \notin \mathcal{P}$ is not an equilibrium price vector. □

11 Extension to Buyers with Utility for Money

In this section, we allow buyers to have utility for money, and we show how to extend our algorithm for the fixed supply model to this case. For each buyer i we are specified a function $f_0^i : \mathbf{R}_+ \rightarrow \mathbf{R}_+$ which gives the utility that i derives as a function of the amount of money that she receives. Each function f_0^i is a non-negative, non-decreasing, piecewise-linear, concave function. Even though we consider money as good 0, segments corresponding to these utility functions will not appear as edges in the network $N(\boldsymbol{p}, \boldsymbol{r})$.

In the algorithm, as prices of goods are raised and rates of buyers are decreased, if for some buyer $i \in I$, r_i decreases to the point where she is equally happy leaving with money corresponding to segment $s \in seg(f_0^i)$, then we need to return money corresponding to s before we can raise the prices of goods any more.

For a segment $s \in seg(f_0^i)$, let $value(s)$ and $returned(s)$ be the total money and the total amount of money returned to i respectively w.r.t. segment s.

The following event is executed as the first event of the algorithm for a phase.

- **Event 0:** There is a buyer $i \in I$ with $\text{bpb}(s) = r_i$ for $s \in seg(f_0^i)$.

  In the *residual* network of $N(\boldsymbol{p}, \boldsymbol{r})$ add a vertex v and an edge (v, i) of capacity $value(s)$. Now find a max flow from v to t.

Lemma 31 *In the max-flow computed above, either a subset I' ($\emptyset \subset I' \subseteq I$) and a subset $J' \subseteq J$ goes* tight *or $returned(s) = value(s)$.*

Proof : Suppose no subset (I', J') goes tight. This means that for every subset (I', J'), the total capacity of edges (s, J') is less than the total capacity of (I', t) after $returned(s)$ amount of money is subtracted from i. Thus, if $returned(s) < value(s)$, there must exist an augmenting path from v to t contradicting the fact that we computed a max flow from v to t. □

Now in the algorithm, if a set goes tight, terminate the current phase and start the next phase. Otherwise start a new iteration.

The next lemma is analogous to Lemma 26 for the enhanced model.

Lemma 32 *If a subset (I', J') goes tight with $J' \neq \emptyset$, the prices of goods in the tight set are rational numbers with denominators $\leq \Delta$.*

Proof : Let $T = \text{bestT}(J')$. Let s be the segment that was being returned to buyer i when this happened. Clearly, $i \in I'$ and the subgraph induced on (I', J') by the current edges, after making them undirected, must be a single connected component (otherwise the component not containing i would contain a tight set even before s was returned). Pick a spanning tree, τ, in (I', J').

Observe that when $bpb(s)$ became equal to r_i, for any edge (i, j) incident at i,

$$p_j = \frac{u_{i,j}}{bpb(s)},$$

where, by a slight abuse of notation, we are using $u_{i,j}$ to denote the rate of the segment represented by the edge connecting i to j. Similarly, if j' is reached via the path i, j, i', j' in τ, then

$$p_{j'} = \frac{u_{i,j} \cdot u_{i',j'}}{bpb(s) \cdot u_{i',j}}.$$

Therefore, the denominator of $p_j, \ j \in J'$ is the product of rates of at most n segments and hence is bounded by U^n, which in turn is bounded by Δ. □

If in the max-flow, $returned(s) = value(s)$ then this segment will never be considered again, since the bang-per-buck of s remains unchanged but rate r_i can only decrease as the algorithm proceeds. Hence the total number of occurrences of such iterations is bounded by the number of segments in functions f_0^i, for all i, which in turn is bounded by z. Now, it is easy to see that the running time bound established in Theorem 29 holds for the enhanced model as well, as long as z is taken to be the total number of segments having positive rate in all utility functions specified in the input, including those for money.

12 Introducing Production into our Model

In this section, instead of assuming that there is a fixed supply of goods, we will assume that the goods are produced by firms as mentioned in Section 1.

12.1 The firms and their production capabilities

The firms have initial endowments of goods and labor[2], and their goal is to maximize profits by optimally producing and selling goods at current prices of goods. Our model allows firms to have

[2] We mention these two separately because for simplicity, in the basic model we did not mention labor. However, in the presence of production, it is important to assume the presence of labor. Notice however, that as far as market clearing is concerned mathematically, goods and labor play the same role.

a rich set of production capabilities, in particular, allowing them to model non-increasing returns to scale as was assumed in the Arrow-Debreu model. For the sake of clarity, we first present the model assuming constant returns to scale. In this model, each firm $f \in F$ has variables y_{jf} corresponding to each good $j \in G$ which represent the amount of this good that it sells or buys in the market; y_{jf} is positive if the firm sells good j, negative if it buys it, and zero otherwise. The objective of the firm is to maximize its profit, which at prices $\boldsymbol{p} = (p_1, \ldots, p_g)$ will be

$$\sum_{j \in G} p_j \cdot y_{jf}.$$

Let c_{jf} denote this firm's initial endowment of good j. In our model there is no need to partition the goods into raw materials and manufactured goods or to differentiate between goods and labor.

In order to formally state the various production processes of this firm, we will use auxiliary variables which are local to this firm. We will denoted these by z_{lf}, i.e., they are indexed by l. The constraints imposed on production are all assumed to be linear and are indexed by m. Thus the set of constraints for firm f are:

$$\forall m \ : \quad \sum_{j \in G} a_{jf}^m \cdot y_{jf} + \sum_l b_{lf}^m \cdot z_{lf} \leq d_f^m,$$

where a_{jf}^m, b_{lf}^m and d_f^m are constants determined by the production processes of firm f. In particular, some of the d_f^m's may be the initial endowments, i.e., c_{jf}'s. Clearly, the optimal operation of a firm can be stated as a linear program.

Next, we give some illustrative examples. First, consider a firm that does not produce anything but only acts as a seller. It sells its initial endowment of goods in the market at the going prices. Clearly, its LP only needs constraints of the form $y_{jf} \leq c_{jf}$.

Second, consider a firm that has an initial endowment, c_{1f} of good 1, and is able to produce goods 8 and 9. However, for this, it will need to buy goods 2 and 3. The production also requires good 1. If the amount of good 1 needed for production is less than c_{1f}, firm f sells the excess in the market, and if it is greater than c_{1f}, firm f will need to buy additional amounts of good 1 from the market. Assume that good 8 is produced using goods 1 and 2, and that good 9 is produced using goods 1 and 3. However, goods 8 and 9 are produced via qualitatively different processes. To produce a unit of good 8, the firm uses up α units of good 1 and β units of good 2. On the other hand, good 9 can be produced using either good 1 or good 3, with a unit of good 1 producing γ units of good 9 and a unit of good 3 producing δ units of good 9. These production constraints are captured by the following linear constraints, using auxiliary variables z_{1f} and z_{2f}.

$$y_{1f} + z_{1f} + z_{2f} \leq c_{1f}$$

$$y_{8f} \leq \alpha \cdot z_{1f} \quad \text{and} \quad y_{8f} \leq \beta \cdot y_{2f}$$

$$y_{9f} \leq \gamma \cdot z_{2f} + \delta \cdot y_{3f}$$

The objective of this firm is to maximize $p_1 \cdot y_{1f} + p_2 \cdot y_{2f} + p_3 \cdot y_{3f} + p_8 \cdot y_{8f} + p_9 \cdot y_{9f}$.

Next, we introduce non-increasing returns to scale in our model, though in a "piecewise-linear" manner. Thus the production of good j by firm f is partitioned into schedules, as a function of the amount of j produced. The schedules are indexed by r. Let y_{jfr} denote the amount of good j produced in the rth schedule and let ρ_{fj} denote the total number of schedules for producing good j in firm f. For each schedule, possibly other than the last one, there is a bound on the amount of good that can be produced in that schedule, i.e., a constraint of the form $y_{jfr} \le \alpha$, for some constant α. The total amount of good j produced is given by the equality

$$y_{jf} = \sum_{r=1}^{\rho_{fj}} y_{jfr}.$$

Each raw material and labor required is non-decreasing as a function of the schedule, so that the earlier schedules produce goods at higher profits. The enhanced constraints now required are:

$$\forall m \ : \ \sum_{j \in G} a_{jf}^m \cdot y_{jf} + \sum_{j \in G, r \le \rho_{jf}} e_{jfr}^m \cdot y_{jfr} + \sum_l b_{lf}^m \cdot z_{lf} \le d_f^m,$$

where e_{jfr}^m's are constants. The reason for the first term on the l.h.s. is that production in a certain segment may depend on the total amount of some other good produced. The form of the objective function remains unchanged, since it only deals with the total amounts of each good produced, i.e. y_{jf}. Since the overall goal of the firm is to maximize profit, it will produce good j up to capacity in earlier schedules before starting production in the next schedule.

Prices for goods and labor are said to be *equilibrium prices* if with optimal operation of each firm at these prices and optimal rates for each buyer, the market clears, i.e., all the goods get sold to buyers and all of their money gets spent.

12.2 A logarithmic RCP for the enhanced model

We will now give a convex program that captures the equilibrium solution while simultaneously optimizing for each firm's objective. For the ease of exposition, we will assume constant returns to scale. All the results can easily be extended for non-increasing returns to scale.

Suppose given prices $\boldsymbol{c}$, firm f is optimizing the following linear program:

$$\begin{aligned} \text{maximize} \quad & \sum_{j \in G} c_j \cdot y_{jf} && (2) \\ \text{subject to} \quad & \forall m \ : \ \sum_{j \in G} a_{jf}^m \cdot y_{jf} + \sum_l b_{lf}^m \cdot z_{lf} \le d_f^m \end{aligned}$$

Let x_{ijk} denote the amount of good j which is allocated to buyer i corresponding to the k^{th} segment s_{ijk} of S_{ij}. Consider the following convex program:

$$\begin{aligned} \text{maximize} \quad & \sum_{i \in B} m_i \log(u_i) && (3) \\ \text{subject to} \quad & \forall i \in B: \ u_i = \sum_{j \in G} \sum_{k \in S_{ij}} u_{ijk} x_{ijk} \\ & \forall j \in G: \ \sum_{i \in B} \sum_{k \in S_{ij}} x_{ijk} \le \sum_{f \in F} y_{jf} \end{aligned}$$

$$\forall f \in F,\ \forall m:\ \sum_{j \in G} a_{jf}^{m} \cdot y_{jf} + \sum_{l} b_{lf}^{m} \cdot z_{lf} \leq d_{f}^{m}$$
$$\forall i \in B,\ \forall j \in G,\ \forall k \in S_{ij}:\ x_{ijk} \leq l_{ijk}$$
$$\forall i \in B,\ \forall j \in G,\ \forall k \in S_{ij}:\ x_{ijk} \geq 0$$
$$\forall f \in F,\ \forall l:\ z_{lf} \geq 0$$

Here the first constraint is ensuring that u_i is the total utility of buyer i, the second constraint is ensuring that the total amount of any good sold to the buyers should not be more than what is produced by the firms, the third constraint is capturing the production constraints of the firms, and the fourth constraint is saying that the amount allocated in each segment should not exceed its size.

Our main theorem is the following:

Theorem 33 *Prices* $\mathbf{p}$ *are equilibrium prices if and only if they form an optimal dual solution to convex program (3). Moreover, the equilibrium production is captured by an optimal solution to primal variables* y_{jk}*'s.*

To prove the above theorem, we will again use the KKT equations. We will show that the KKT equations have two different components, one which corresponds to optimization of the buyers as was shown in previous section and other which corresponds to optimization of the firms. Following are the KKT conditions of convex program (3).

(1) $\forall j \in G:\ p_j \geq 0,$

(2) $\forall f \in F, \forall m:\ \alpha_f^m \geq 0,$

(3) $\forall i \in B,\ \forall j \in G,\ \forall k \in S_{ij}:\ q_{ijk} \geq 0,$

(4) $\forall j \in G:\ p_j > 0 \Rightarrow\ \sum_{i \in B} \sum_{k \in S_{ij}} x_{ijk} = \sum_{f \in F} y_{jf}.$

(5) $\forall i \in B,\ \forall j \in G,\ \forall k \in S_{ij}:\ q_{ijk} > 0 \Rightarrow x_{ijk} = l_{ijk},$

(6) $\forall f \in F, \forall m:\ \alpha_f^m > 0 \Rightarrow\ \sum_{j \in G} a_{jf}^m \cdot y_{jf} + \sum_l b_{lf}^m \cdot z_{lf} = d_f^m,$

(7) $\forall i \in B,\ \forall j \in G\ \forall k \in S_{ij} : p_j + q_{ijk} \geq \frac{m_i . u_{ijk}}{u_i}.$

(8) $\forall i \in B,\ \forall j \in G\ \forall k \in S_{ij} : x_{ijk} > 0 \Rightarrow p_j + q_{ijk} = \frac{m_i . u_{ijk}}{u_i}.$

(9) $\forall f \in F, \forall l : \sum_m b_{lf}^m . \alpha_f^m \geq 0$

(10) $\forall f \in F, \forall l : z_{lf} > 0 \Rightarrow\ \sum_m b_{lf}^m . \alpha_f^m = 0$

(11) $\forall j \in G, \forall f \in F : \sum_m a_{jf}^m . \alpha_f^m = p_j$

As earlier, we will call p_j to be the *price* of good j, and q_{ijk} to be the *price differential* which is unique for each buyer i, good j, and segment $k \in S_{ij}$.

Lemma 34 *If convex program (3) has a finite optimal solution, then such a solution optimizes each firm's profit at equilibrium.*

Proof : First consider the equations (2), (6), (9), (10), and (11). Note that none of these equations have variables x_{ijk}'s or q_{ijk}'s. Also, by setting $\boldsymbol{c} = \boldsymbol{p}$, it is not difficult to see that these equations restricted to any firm f are precisely the complementary slackness conditions of the LP corresponding to that firm. Thus an optimal solution of the convex program also optimizes for the profit of all the firms simultaneously. $\square$

The following fact can be proven along the lines of Theorem 6; its proof is omitted.

Theorem 35 *Program (3) is a rational convex program.*

Since the supply of goods is not fixed, we won't make an assumption similar to the first assumption in the previous section. Instead, we will work with a slightly different, but standard, definition of equilibrium that if a good is not sold completely, then its price must be zero. The following lemma easily follows from the 4^{th} KKT condition.

Lemma 36 *For every good j, either price p_j is zero or good j is exactly sold, i.e., $\sum_{i \in B} \sum_{k \in S_{ij}} x_{ijk} = \sum_{f \in F} y_{jf}$.*

Also using proof similar to lemma 4, one can prove the following lemma:

Lemma 37 *For prices $\boldsymbol{p}$ given by the KKT conditions, the rate $r_i = \frac{u_i}{m_i}$ is optimal for each buyer i.*

Thus we have shown that optimal solution of the convex program satisfies the following equilibrium conditions:

1. For prices $\boldsymbol{p}$, the rate $r_i = \frac{u_i}{m_i}$ is optimal for each buyer i. Moreover, since $u_i > 0$, buyer i spends his money completely.
2. No portion of a segment s is sold to a buyer i, if $bpb(s) < r_i$.
3. If the price of some good is strictly positive, it is sold out completely.
4. Each firm's production optimizes its profit.

The proof of the other direction, that any equilibrium solution is also a solution to the convex program, is similar to that in Section 3. This completes the proof of Theorem 33.

13 First and Second Welfare Theorems

Theorem 38 (First Welfare) *The utilities accrued by buyers at equilibrium prices $\boldsymbol{p}$ and rates $\boldsymbol{r}$ are Pareto efficient.*

This follows from the convex program (3) since a non Pareto-efficient solution cannot be an optimal solution to the convex program.

Theorem 39 (Second Welfare) *For any Pareto optimal utilities $\boldsymbol{u}^*$, there exists a choice of money vector of buyers under which equilibrium utilities are $\boldsymbol{u}^*$.*

Proof : This also follows from the convex program. Let S be the set of all feasible utility vectors. Since any feasible utility vector is a solution to the linear equations in the convex program 1, set S is a convex set. Thus, there must exist a hyperplane $\sum_i a_i u_i = c$ which is tangent to S and passes through $\boldsymbol{u}^*$. Moreover, since $\boldsymbol{u}^*$ is Pareto optimal, $a_i \geq 0$, for each $i \in B$.

Now given such a hyperplane, all we need to do is to find a curve $\sum_i m_i.log(u_i) = c'$, such that its derivative at $\boldsymbol{u}^*$ is same as that of $\sum_i a_i.u_i^* = c$, and it passes through $\boldsymbol{u}^*$. This can be achieved by setting $\frac{m_i}{u_i^*} = a_i$, and setting c' such that $\sum_i m_i.log(u_i^*) = c'$. Since $a_i \geq 0$, we get $m_i \geq 0$. Thus, from the convex program we get that, if we run the market with money of buyers equal to m_i's, equilibrium utilities will be $\boldsymbol{u}^*$. □

14 Discussion

The model described in this paper was obtained in the process of attempting the open problem, posed in [23], of obtaining a combinatorial algorithm for solving the extension of game **ADNB** to plc utilities; linear utilities were assumed in **ADNB**. Combinatorial insights obtained in the process led to the model.

For our basic model, i.e., without production, we have given a combinatorial polynomial time algorithm for computing an equilibrium. In Section 11, we generalize this model, and our algorithm, to include utility for money among buyers. Again, it turns out that equilibrium is rational. However, we do not see how to obtain a convex program for this extension. In contrast, for the extension of Fisher's linear case when buyers have utility for money, given by a linear function, a convex program was recently found by [4]; in fact, this convex program even captures spending constraint utilities defined in [24], together with buyers having utility for money.

Finally, we also leave the open problem of extending our combinatorial algorithm to the entire model, with production included.

References

[1] Simon P. Anderson and Andr de Palma. Spatial price discrimination with heterogeneous products. *The Review of Economic Studies*, 55(4):573–92, 1988.

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

[3] V. Bhaskar and Ted To. Is perfect price discrimination really efficient? An analysis of free entry. *The RAND journal of economics*, 35(4):762–76, 2004.

[4] B. Birnbaum, N. Devanur, and L. Xiao. New convex programs and distributed algorithms for Fisher markets with linear and spending constraint utilities. Unpublished manuscript, 2010.

[5] D. Chakrabarty, N. Devanur, and V. V. Vazirani. New results on rationality and strongly polynomial solvability in Eisenberg-Gale markets. *SIAM Journal on Discrete Mathematics*, 24(3), 2010.

[6] X. Chen, D. Dai, Y. Du, and S.-H. Teng. Settling the complexity of Arrow-Debreu equilibria in markets with additively separable utilities. In *IEEE Annual Symposium on Foundations of Computer Science*, 2009.

[7] X. Chen and S.-H. Teng. Spending is not easier than trading: on the computational equivalence of Fisher and Arrow-Debreu equilibria. *Journal of the ACM*, 56(3), 2009.

[8] B. Codenotti and K. Varadarajan. Efficient computation of equilibrium prices for markets with Leontief utilities. In *ICALP*, 2004.

[9] N. Devanur, C. H. Papadimitriou, A. Saberi, and V. V. Vazirani. Market equilibrium via a primal-dual-type algorithm. *Journal of the ACM*, 55(5), 2008.

[10] A. Edlin, M. Epelbaum, and W. Heller. Surplus maximization and price discrimination in general equilibrium: Part I. Technical Report 9405, Centro de Investigacion Economica, ITAM, 1994.

[11] M. Heller W. P Edlin, A. S. Epelbaum. Is perfect price discrimination really efficient?: Welfare and existence in general equilibrium. *Econometrica*, 66(4):897–922, 1998.

[12] E. Eisenberg and D. Gale. Consensus of subjective probabilities: the Pari-Mutuel method. *The Annals of Mathematical Statistics*, 30:165–168, 1959.

[13] M. Grotschel, L. Lovasz, and A. Schirjver. *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag, 1988.

[14] K. Jain. A polynomial time algorithm for computing the Arrow-Debreu market equilibrium for linear utilities. *SIAM Journal on Computing*, 37(1):303–318, 2007.

[15] K. Jain and V. V. Vazirani. Eisenberg-Gale markets: Algorithms and game-theoretic properties. *Games and Economic Behavior*, 70(1), 2010.

[16] K. Jain, V. V. Vazirani, and Y. Ye. Market equilibrium for homothetic, quasi-concave utilities and economies of scale in production. In *ACM-SIAM Annual Symposium on Discrete Algorithms*, 2005.

[17] F. P. Kelly. Charging and rate control for elastic traffic. *European Transactions on Telecommunications*, 8:33–37, 1997.

[18] Qihong Liu and Konstantinos Serfes. Imperfect price discrimination, market structure, and efficiency. *Canadian Journal of Economics*, 38(4):1191–1203, 2005.

[19] A. C. Pigou. *The Economics of Welfare*. MacMillan and Company, 1900.

[20] Hal Varian. Differential pricing and efficiency. *First Monday*, 1(2), 1996.

[21] Hal R. Varian. Price discrimination and social welfare. *American Economic Review*, 75(4):870–75, 1985.

[22] V. V. Vazirani. 2-player Nash and nonsymmetric bargaining via market equilibria. In *SAGT*, 2009.

[23] V. V. Vazirani. Rational convex programs, testing feasibility, and the Arrow-Debreu Nash bargaining game. Unpublished manuscript, 2009.

[24] V. V. Vazirani. Spending constraint utilities, with applications to the Adwords market. *Mathematics of Operations Research*, 35(2), 2010.

[25] V. V. Vazirani and M. Yannakakis. Market equilibria under separable, piecewise-linear, concave utilities. *Journal of the ACM*, 58(3), 2011.

[26] G. Norman W.B. MacLeod and J.-F. Thisse. Price discrimination and equilibrium in monopolistic competition. *International Journal of Industrial Organization*, 6(4):429–446, 1988.

[27] Y. Ye. Exchange market equilibria with Leontief's utility: Freedom of pricing leads to rationality. *Theoretical Computer Science*, 378:134–142, 2007.