

EFFICIENT ALGORITHMS FOR MARKET EQUILIBRIA

A Thesis
Presented to
The Academic Faculty

by

Nikhil R. Devanur

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
College of Computing

Georgia Institute of Technology
August 2007

EFFICIENT ALGORITHMS FOR MARKET EQUILIBRIA

Approved by:

Professor Vijay V. Vazirani, Advisor
College of Computing
Georgia Institute of Technology

Professor Robin Thomas
School of Mathematics
Georgia Institute of Technology

Professor Santosh Vempala
College of Computing
Georgia Institute of Technology

Professor Dana Randall
College of Computing
Georgia Institute of Technology

Professor Subhash Khot
College of Computing
Georgia Institute of Technology

Date Approved: 14 May 2007

To my parents,

Smt. Yashoda and Sri Rangarajan Devanur

and my dear brother,

Nishchal.

ACKNOWLEDGEMENTS

I am indebted to my advisor Vijay V Vazirani; this thesis would not have been possible without him. I thank him for all his encouragement, support and guidance through the years. In fact, meeting him at a conference was what brought me to Georgia Tech for my PhD. His knowledge and experience have been invaluable to me, and I was very fortunate to have such a person to call upon whenever needed.

Also, special thanks goes to Subhash Khot, working with whom has been a privilege. He has been a great teacher, guide, colleague and a friend. I have also had the good fortune of working with Dick Lipton and Milena Mihail. Dick, especially has always bombarded me with his problems and ideas. They have had a great influence on me and my thinking.

I would like to thank H Venkateswaran, Dana Randall, Prasad Tetali, Robin Thomas and Ernie Croot. I enjoyed their classes, and I have recieved a great amount of education and guidance from them. Thanks to my co-authors Christos Papadimitriou, Amin Saberi and Deeparanab Chakrabarty, for letting me include our joint work in my thesis.

The students in the Theory group over the years have made my stay very enjoyable. So many of them have made it possible for me to do so many things, it is not possible to list them all here. Special thanks to Nisheeth Vishnoi and Deeparnab Chakrabarty, for the several (hundreds of?) hours spent with them thinking about various things and managing to prove a few. Equally indebted am I to my friends outside the group, because of whom I have come to love Atlanta and all its hangouts.

No amount of thanks is enough for my parents and my brother. None of this would be meaningful but for them. This thesis is dedicated to them.

Finally, I would like to thank Megha, who made it very difficult for me to concentrate in the last few months, at the same time making it extremely exciting.

TABLE OF CONTENTS

DEDICATION	iii
ACKNOWLEDGEMENTS	iv
LIST OF TABLES	ix
LIST OF FIGURES	x
SUMMARY	xi
I INTRODUCTION	1
1.1 History and Motivation	1
1.2 Contributions of this thesis	2
1.2.1 Linear Utilities	2
1.2.2 Spending Constraint Utilities	3
1.2.3 EG Markets	5
II PRELIMINARIES	9
2.1 The Fisher Model	9
2.2 The Arrow-Debreu Model	10
2.3 Demand functions and Fixed point Theorems	11
2.4 Approximate equilibria	13
2.5 Weak Gross Substitutability and the Tatonnement process	13
III LINEAR UTILITIES, FISHER MODEL	15
3.1 The Eisenberg-Gale Convex Program	15
3.2 High level idea of the algorithm	16
3.3 The enhanced setting and how to deal with it	18
3.4 A simple algorithm	21
3.5 Finding tight sets	25
3.6 Termination with market clearing prices	28
3.7 Establishing polynomial running time	30

3.7.1	Balanced flows and their properties	30
3.7.2	The polynomial time algorithm	35
3.8	Discussion	38
IV	LINEAR UTILITIES, ARROW-DEBREU MODEL	40
4.1	The Algorithm	40
4.2	Analysis of the Algorithm	41
4.3	Improved Running time	43
4.4	Comparison with Related Work	44
V	SPENDING CONSTRAINT UTILITIES	46
5.1	Definition	46
5.2	Existence and Uniqueness of Equilibrium	48
5.2.1	The Fisher Model	48
5.2.2	The Arrow-Debreu Model	53
5.3	Algorithms for Step Functions	55
5.4	Algorithms for Continuous Functions	59
5.5	A Heuristic	62
VI	EG MARKETS	63
6.1	Definition	63
6.2	Rationality of EG[2] Markets	66
6.3	Algorithms for combinatorial EG[2] markets	69
6.3.1	Binary Search Algorithm	69
6.3.2	Combinatorial Markets	72
VII	CONCLUSION	74
7.1	Piecewise-linear utilities	74
7.2	Beyond Weak Gross Substitutibility	74
APPENDIX A	KKT CONDITIONS	76
APPENDIX B	PROJECTION OF POLYTOPES	77

REFERENCES	79
----------------------	----

LIST OF TABLES

- | | | |
|---|---|---|
| 1 | Table of Results about Rationality of Equilibrium in Capacity Allocation markets. The table also notes the existence of max-min theorems. | 7 |
|---|---|---|

LIST OF FIGURES

1	An Example of the network $N(\mathbf{p})$, satisfying the invariant.	18
2	As the prices are increased a set \tilde{S} goes tight.	23
3	Occurrence of Event 2.	25
4	If $\gamma_i(\mathbf{p}, f) > \gamma_j(\mathbf{p}, f)$ and there is a path from j to i in $R(\mathbf{p}, f) \setminus \{s, t\}$, then a circulation would give a more balanced flow.	32
5	Representing a spending constraint utility function.	47

SUMMARY

The mathematical modeling of a market, and the proof of existence of equilibria have been of central importance in mathematical economics. Since the existence proof is non-constructive in general, a natural question is if computation of equilibria can be done efficiently. Moreover, the emergence of Internet and e-commerce has given rise to new markets that have completely changed the traditional notions. Add to this the pervasiveness of computing resources, and an algorithmic theory of market equilibrium becomes highly desirable. The goal of this thesis is to provide polynomial time algorithms for various market models.

Two basic market models are the Fisher model: one in which there is a demarcation between buyers and sellers, buyers are interested in the goods that the sellers possess, and sellers are only interested in the money that the buyers have; and the Arrow-Debreu model: everyone has an endowment of goods, and wants to exchange them for other goods. We give the first polynomial time algorithm for exactly computing an equilibrium in the Fisher model with linear utilities. We also show that the basic ideas in this algorithm can be extended to give a strongly polynomial time approximation scheme in the Arrow-Debreu model.

We also give several existential, algorithmic and structural results for new market models:

- the spending constraint utilities (defined by Vazirani [42]) that captures the “diminishing returns” property while generalizing the algorithm for the linear case.
- the capacity allocation market (defined by Kelly [36]), motivated by the study of fairness and stability of the Transmission Control Protocol (TCP) for the

Internet, and more generally the class of Eisenberg-Gale (EG) markets (defined by Jain and Vazirani [31]).

Finally, this line of research has given insights into the fundamental techniques in algorithm design. The primal-dual schema has been a great success in combinatorial optimization and approximation algorithms. Our algorithms use this paradigm in the enhanced setting of Karush-Kuhn-Tucker (KKT) conditions and convex programs.

CHAPTER I

INTRODUCTION

It is not from the benevolence of the butcher, the brewer or the baker that we expect our dinner, but from their regard to their own interest. We address ourselves not to their humanity but to their self-love, and never talk to them of our necessities but of their advantages.

Adam Smith, The Wealth of Nations.

1.1 History and Motivation

The concepts of supply and demand, and the notion of an equilibrium as the point where the two meet are the most basic to economics. General equilibrium theory, extending these concepts to include more aspects of a real economy such as multiple commodities and production, has been a cornerstone of modern mathematical economics. The formalization of such a model was pioneered by Leon Walras [43] and has been a subject of interest ever since (including an early, simpler model defined by Irving Fisher [5]). A major breakthrough was the proof of existence of equilibrium prizes by Arrow and Debreu [3], for which they later won the Nobel Prize.

However these proofs use Fixed Point Theorems, so they are non constructive, and a major drawback of this model has been the lack of algorithmic results, except a few isolated instances. (See for instance, the book by Scarf [39]. Even here the algorithms are not efficient, that is, they do not run in polynomial time.) The problem of how a market reaches the equilibrium has been attributed to the “invisible hand”, but a natural question to ask is, *how efficient is the invisible hand?* Efficient algorithms to compute market equilibrium would give more justification to the model.

In any case, the goal of the general equilibrium theory has been that it be used as a tool to evaluate the effects of economic policies. The book by Shoven and Whalley [40] gives a good survey of techniques used to make realistic models of actual economies and apply this theory. They also illustrate how this gives fresh insights into the impacts of policies, not obtained by other models. Another such example is the work of Kehoe and Kehoe [35] evaluating the impact of NAFTA on the economic welfare of participating countries. Once again, efficient algorithms to compute market equilibrium would greatly increase the applicability of these techniques. In fact, Kakade et. al. [34] have used one of our algorithms to study the effect of trade agreements on price variation, etc.

The emergence of Internet has greatly affected the study of markets and has led to a fresh look at these concepts. It has given rise to new markets, for instance, involving the trading of bandwidth among different service providers. Also it has enabled e-commerce and online markets, such as the ones due to Amazon, EBay and Google. Finally the availability of massive computational power has naturally led to automatized pricing in many businesses.

All these have resulted in a surge of interest in using the tools from the theory of algorithms to compute equilibrium prices. In this thesis we present several such algorithms.

1.2 Contributions of this thesis

1.2.1 Linear Utilities

We present a *combinatorial* polynomial time algorithm ([15]) when the utility functions are linear, in the Fisher model (See section 2.1 for a definition). For this case, it is natural to seek an algorithmic answer in the theory of linear programming. However, there does not seem to be any natural linear programming formulation for this problem. Instead, a remarkable nonlinear convex program, given by Eisenberg and

Gale [19], captures, as its optimal solutions, equilibrium allocations. Let us outline the advantages of our approach versus simply solving the convex program.

The usual advantages of combinatorial algorithms apply to our work as well, namely such algorithms are easier to adapt, certainly heuristically and sometimes even formally, to related problems and fine tuned for use in special circumstances; in fact, the algorithms in Chapters 4 and 5 build upon this basic algorithm. No convex program formulation is known for many of these cases.

One of the tools for analyzing the stability of a market is the *tatonnement* (or the groping) process. It is a price updating rule that says, “increase the prices of those goods for which the demand is greater than the supply and decrease the prices of those goods for which the supply is greater than the demand”. Many versions of this process have been suggested based upon the exact nature of the update. Fast convergence of this process is an indication of greater stability. Our algorithm can be interpreted as a discrete version of the *tatonnement* process, and what we show is that this version converges to the equilibrium in polynomial time.

Finally, this line of research has given insights into the fundamental techniques in algorithm design. The primal-dual schema has been a great success in combinatorial optimization and approximation algorithms, and is used in the setting of LP-duality theory and complementary slackness conditions. Our algorithm uses the primal-dual paradigm in the enhanced setting of convex programming and the KKT conditions. In Section 3.3 we pinpoint the added difficulty of working in this enhanced setting and the manner in which our algorithm circumvents this difficulty.

1.2.2 Spending Constraint Utilities

A natural generalization of linear utility functions is the piecewise linear and concave (PLC) utility functions. Moreover, concave functions that are additive over goods can be approximated by PLC functions, thus an algorithm for PLC utilities might be

used to obtain a PTAS for separable concave utilities. However attempts at extending the algorithm for linear case to PLC utilities have so far failed. One of the reasons for this is that the algorithm *monotonically* raises prices, and this works because linear utilities satisfy weak gross substitutability (definition in Section 2.5). But the same is not true for PLC utilities, even if each function is restricted to no more than two pieces. This means that an algorithm would be required to raise and lower the prices, and proving fast convergence for such a procedure is in general quite difficult. Motivated by this difficulty, Vazirani [42] generalized linear utilities in a new direction, obtaining what are called the spending constraint utilities: a buyer's total utility function is still linear and additively separable over goods; in addition, a limit is specified on the amount of money he can spend on each good. This can be further generalized to capture the property of decreasing marginal utilities: the buyer has several linear utility functions (each specified by a constant utility rate) for each good, each with a specified spending limit. Such utility functions do satisfy weak gross substitutability. Indeed, [42] extends the algorithm of [15] to get a polynomial time algorithm for spending constraint utilities, in the Fisher model.

[42] considered only finitely many linear utility functions for each buyer and each good (the discrete case). We ([17]) show that this can be generalized so that the utility rates are continuously varied. Such a utility function has the property that given any prices, each buyer has a unique bundle that maximizes his utility. An equilibrium is now specified by the prices alone, unlike the discrete case where one has to specify an equilibrium *allocation* in addition. This entails an efficiency in communication (see [14] for a detailed discussion). However the functions so obtained do not fall under the class of functions for which the Arrow-Debreu Theorem [3] holds. We show existence of equilibrium for such functions by an application of Brouwer's Fixed Point Theorem. We also show that the equilibrium is unique. Uniqueness of equilibrium has been considered desirable, since it indicates stability ([4, 13]). In contrast, PLC

utilities do not have a unique equilibrium. Moreover we show that the algorithm of [42] can be used as a black-box to obtain a PTAS for the continuous case.

[42] considers only the Fisher model, and a natural problem is to extend it to the Arrow-Debreu model (definition in Section 2.2). Since the buyers' incomes are not fixed in the Arrow-Debreu model, the definition of spending constraint utilities needs to be changed slightly: the limit on each linear function is now a fraction of the buyer's income, instead of a specific amount of money. As in the Fisher model, we extend it to the continuous case, and give a proof of existence which now needs the stronger Kakutani's Fixed Point Theorem. We give an example to show that equilibrium is no longer unique. We also show how to extend the algorithm of [42] to get a PTAS, first for the discrete case, which can in turn be used to get a PTAS for the continuous case.

In summary, we show that the spending constraint utilities not only parallel the traditional utilities in existential and uniqueness results, but are also amenable to efficient algorithms. But does this help in our quest for efficient algorithms for the traditional utility functions? We suggest a possible avenue via a heuristic for PLC utilities which uses the algorithm for spending constraint utilities (Section 5.5).

1.2.3 EG Markets

As was mentioned earlier, the Eisenberg-Gale convex program captures, as its optimal solution, equilibrium allocations for the linear case of the Fisher model. Over the years, convex programs with the same basic structure were found for more general utility functions: scalable utilities [20], Leontief utilities [11], Linear Substitution utilities [45] and homothetic utilities with productions [33]. Interestingly enough, a program with the same structure as the Eisenberg-Gale program is used by Kelly [36] in his seminal work giving a mathematical model for understanding TCP congestion control. Resources in these markets are edge capacities and agents want to build

combinatorial objects such as source-sink flow paths or spanning trees or branchings, e.g., for establishing TCP connections or broadcasting messages to all nodes in the network. The following market, called the *capacity allocation market*, is of special significance within Kelly’s framework: Given a network (directed or undirected) with edge capacities specified and a set of source-sink pairs, each with initial endowment of money specified, find equilibrium flow and edge-prices. The equilibrium must satisfy:

- Only saturated edges can have positive prices.
- All flows are sent along a minimum cost path from source to sink.
- The money of each source-sink pair is fully spent.

Jain and Vazirani [31] defined the notion of *Eisenberg-Gale markets* generalizing all of the markets mentioned above, and studied several properties of these markets. They showed that several of these Eisenberg-Gale markets are *rational*: they always have a rational solution if all the input parameters are rational. (The Eisenberg-Gale program, and in turn, the linear case of Fisher’s model also has this property.) However, several other such markets are irrational [24, 31]. Among the markets characterized, an important distinction between the rational and irrational markets was that combinatorial problems underlying the former satisfied max-min theorems, which were used critically to establish rationality, and those for the latter didn’t. Also these max-min theorems helped establish combinatorial strongly polynomial time algorithms to find the equilibrium in these markets.

The markets for which this question was left open do not support max-min theorems, and it was expected that the equilibrium would be irrational. Surprisingly enough, despite this, they turn out to be rational. More generally, we ([6]) show that all markets in EG[2], the class of Eisenberg-Gale markets with two agents, are rational. We also show that whenever the polytope containing the set of feasible utilities of the two agents can be described via a combinatorial LP, the market admits

a strongly polynomial algorithm. Our algorithm circumvents the lack of underlying max-min theorems by using the more general LP-Duality Theory itself; on the flip side, our methods work only for the case of two agents. This difference in the combinatorial structure manifests itself in the algorithmic ideas needed: whereas [31] use the primal-dual schema and their algorithms can be viewed as ascending price auctions, we use a carefully constructed binary search. The algorithms of [31] are combinatorial whereas ours are not, ours require a subroutine for solving combinatorial LP's. The latter can be accomplished in strongly polynomial time using Tardos' algorithm [41].

Table 1 summarizes the results about rationality of equilibrium in the two capacity allocation markets. We also record the existence of max-min theorems in each case.

Table 1: Table of Results about Rationality of Equilibrium in Capacity Allocation markets. The table also notes the existence of max-min theorems.

	One Source Multiple Sink	Two Source Two Sink	Multiple Source Multiple Sink
Directed Networks	<ul style="list-style-type: none"> • Rational [31] • max-flow = min-cut [23] 	<ul style="list-style-type: none"> • Rational (Our result) • max-flow \neq min-cut [25] 	<ul style="list-style-type: none"> • Irrational [24] • max-flow \neq min-cut [12]
Undirected Networks	<ul style="list-style-type: none"> • Rational [31] • max-flow = min-cut [22] 	<ul style="list-style-type: none"> • Rational [31] • max-flow = min-cut [28] 	<ul style="list-style-type: none"> • Irrational [24] • max-flow \neq min-cut [12]

1.2.3.1 Organization

Most of the basic definitions and concepts are presented in Chapter 2. The algorithm for the linear utilities in the Fisher model is in Chapter 3. Chapter 4 extends this algorithm to the Arrow-Debreu model. Chapter 5 contains all the results on the spending constraint utilities. EG Markets and the results therein are in Chapter 6.

Chapter 7 lists related work and some open problems.

CHAPTER II

PRELIMINARIES

In this chapter we define the various mathematical models of a market that we consider in this thesis. We also state the classical theorems from economics showing existence of equilibrium.

2.1 *The Fisher Model*

The following describes the Fisher model of a market. It formalizes the notion of an equilibrium being the state of the market where the demand meets the supply, for all goods simultaneously.

- It consists of a set A of divisible *goods* and a set B of *buyers*. Assume that $A = \{1, 2, \dots, n\}$ and $B = \{1, 2, \dots, n'\}$.
- The available supply of good j is b_j .
- Each buyer i has a specific endowment of money, $e(i)$.
- The *preferences* of buyer i are given by a *utility* function, $U_i : \mathbf{R}_+^n \rightarrow \mathbf{R}_+$. Buyer i prefers \mathbf{x} to \mathbf{x}' if and only if $U_i(\mathbf{x}) > U_i(\mathbf{x}')$. Assume that U_i 's are all concave functions.

Given prices p_1, \dots, p_n of the goods, buyer i wants to buy a basket $\mathbf{x}(i)$ of goods (there could be many) that make him happiest. So $\mathbf{x}(i)$ is a solution to the following convex program.

$$\begin{array}{ll} \text{maximize} & U_i(\mathbf{x}) \\ \text{subject to} & \sum_{j \in A} p_j x_j \leq e(i), \\ & x_j \geq 0, \quad \forall j \in A. \end{array}$$

Notation 1 *The variables/functions in bold face, for instance \mathbf{x} , represent vectors, and the co-ordinates of \mathbf{x} are denoted by x_1, x_2, \dots, x_n . Also \mathbf{x} denotes a general allocation, whereas $\mathbf{x}(i)$ denotes an optimal allocation for buyer i .*

A thing to note is that this program does not depend on the supplies of the goods. If $\sum_i x_j(i) > b_j$, then there is a deficiency of good j and if $\sum_i x_j(i) < b_j$, then there is a surplus. Also, the above model assumes that the buyers don't have any utility for the money they come with, so the optimal solution will exhaust all of the endowment. However, we will later see that the model (and the algorithms) can be extended even when the buyers do value money.

We will say that p_1, \dots, p_n are *market equilibrium* prices if after each buyer is assigned such a basket $\mathbf{x}(i)$, there is no surplus or deficiency of any of the goods, that is, for each good j , $\sum_i x_j(i) = b_j$.

It turns out that there exist market equilibrium prices if the utilities are all concave and each good has a potential buyer (one who derives nonzero utility from this good). Our problem is to compute such prices in polynomial time.

2.2 *The Arrow-Debreu Model*

The *Arrow-Debreu* model (which is also known as the Walrasian model or the exchange model,) differs from the Fisher model in that the endowment of each buyer i is a bundle of goods $\mathbf{e}(i) \in \mathbf{R}^n$ (instead of money, as before). The supply is now $b_j = \sum_{i \in B} e_j(i)$. Given the prices, assume that each buyer gets an income of $e(i) = \sum_j p_j e_j(i)$ by selling all of his endowment. So in the Fisher model, the incomes are fixed whereas in the Arrow-Debreu model they are dependent on the prices. The conditions of equilibrium are as before, that each buyer is allocated an optimal basket such that the market clears.

2.3 Demand functions and Fixed point Theorems

Assume for the rest of this section that the utility function U_i is continuous and strictly concave. Then for every price vector \mathbf{p} , there is a unique optimal basket $\mathbf{x}(i)$. In fact, we also require that $p_j > 0$ for each good j . Otherwise the demand for that good would be unbounded. Then we call $\mathbf{x}(i, \mathbf{p})$ the demand of buyer i at price \mathbf{p} , and $\mathbf{x}(i) : \text{Int}(\mathbf{R}_+^n) \rightarrow \mathbf{R}_+^n$ the demand function of i . ($\text{Int}(\mathbf{R}_+^n)$ is the *interior* of \mathbf{R}_+^n , which is the set of all vectors in \mathbf{R}_+^n with strictly positive co-ordinates.)

Define $\boldsymbol{\xi} : \text{Int}(\mathbf{R}_+^n) \rightarrow \mathbf{R}_+^n$ to be the total demand function in terms of money spent on the goods, i.e.,

$$\xi_j(\mathbf{p}) = \sum_{i \in B} p_j x_j(i, \mathbf{p}).$$

Also define the excess demand function $\boldsymbol{\zeta} : \text{Int}(\mathbf{R}_+^n) \rightarrow \mathbf{R}_+^n$ to be

$$\zeta_j(\mathbf{p}) := \left(\sum_{i \in B} x_j(i, \mathbf{p}) \right) - b_j.$$

The equilibrium condition can be restated as $\boldsymbol{\zeta}(\mathbf{p}) = 0$. Note that in the Arrow-Debreu model, $\boldsymbol{\zeta}$ is invariant on scaling, that is multiplying the prices of all goods by the same number does not change $\boldsymbol{\zeta}$. This is because in that case, all the incomes get multiplied by the same factor, and hence the demands remain the same. So we can confine the domain of $\boldsymbol{\zeta}$ to the interior of the unit simplex,

$$S = \{\mathbf{p} \in \mathbf{R}_+^n : p_1 + p_2 + \cdots + p_n = 1\}.$$

Existence of equilibrium is established by showing that the excess demand function has certain nice properties and then using a fixed point theorem to show that for any such function there exists a \mathbf{p} such that $\boldsymbol{\zeta}(\mathbf{p}) = 0$. In particular, the properties that $\boldsymbol{\zeta}$ is required to satisfy in order to guarantee such a price vector are as follows. We will use this theorem in Section 5.2.2 in order to show existence of equilibrium for a model that does not fit into the description here.

Lemma 2 For a function $\boldsymbol{\zeta}(\cdot) = (\zeta_1(\cdot), \zeta_2(\cdot), \dots, \zeta_n(\cdot))$ from $\text{Int}(S)$ into \mathbf{R}^n , if

1. ζ is continuous and bounded from below.
2. ζ satisfies the Walras' law, i.e., $\mathbf{p} \cdot \zeta(\mathbf{p}) = 0$ holds for all $\mathbf{p} \in S$.
3. If a sequence $\{\mathbf{p}_m\}$ of strictly positive prices

$$\mathbf{p}_m = (p_1^m, p_2^m, \dots, p_n^m) \rightarrow \mathbf{p} = (p_1, p_2, \dots, p_n)$$

and $p_k > 0$ holds for some k , then the sequence $\{\zeta_k(\mathbf{p}_m)\}$ of the k^{th} components of $\{\zeta(\mathbf{p}_m)\}$ is bounded.

4. $\mathbf{p}_m \rightarrow \mathbf{p} \in \partial S$, with $\mathbf{p}_m \in S$ imply $\lim_{m \rightarrow \infty} \|\zeta(\mathbf{p}_m)\| = \infty$.

Then, there exists at least one vector $\mathbf{p} \in S$ such that $\zeta(\mathbf{p}) = 0$.

We first state Brouwer's and then the more general Kakutani's fixed point theorems that are used in establishing the above lemma.

Lemma 3 (Brouwer) Let $\mathbf{g} : S \rightarrow S$ be a continuous function from a non-empty, compact, convex set $S \subset \mathbf{R}^n$ into itself, then there is an $\mathbf{x}^* \in S$ such that $\mathbf{g}(\mathbf{x}^*) = \mathbf{x}^*$.

We now recall a few preliminaries required to state Kakutani's fixed point theorem. A **correspondence** between two sets X and Y is a function ϕ from X to 2^Y , the set of all subsets of Y . The **graph** of a correspondence ϕ is

$$G_\phi = \{(x, y) \in X \times Y : x \in X, y \in \phi(x)\}$$

When X and Y are topological spaces, ϕ is said to have a **closed graph** whenever G_ϕ is a closed subset of $X \times Y$.

Lemma 4 Kakutani's fixed point theorem Let X be a nonempty, compact and convex subset of \mathbf{R}^n . If ϕ is a non-empty and convex valued correspondence from X to itself, and has a closed graph, then ϕ has a fixed point, i.e., $\exists x \in X$ with $x \in \phi(x)$.

The existence of equilibrium prices follows from showing that the excess demand function ζ satisfies all the required properties.

Theorem 5 (*Arrow and Debreu [3, 1]*) *Every market in the Arrow-Debreu model with continuous and strictly concave utility functions has an equilibrium price.*

The Fisher model is a special case of the Arrow-Debreu model, by considering money as one of the goods. Hence the existence of equilibrium in the Fisher model also follows from the above theorem.

2.4 *Approximate equilibria*

We now introduce the notion of an approximate equilibrium, by relaxing the constraint that the demand be exactly equal to supply. However, the deviation from equilibrium is bounded. For any $\epsilon > 0$, a price vector \mathbf{p} is an ϵ -approximate market equilibrium if each buyer can be allocated an optimal basket \mathbf{x}_i and

$$\sum_{j \in A} |p_j b_j - \xi_j(\mathbf{p})| \leq \epsilon \sum_{j \in A} p_j b_j.$$

There are several ways to define an approximate equilibrium. Another way would have been to say that for each good, the deficiency or the surplus is small w.r.t the supply. In fact an approximate equilibrium with this definition would also be an approximate equilibrium with our definition. However our definition is a more general one, since it allows goods with smaller prices to have a larger deficiency or surplus.

Another way to define an approximate equilibrium would be to allow the agents to be allocated approximately optimal basket while keeping the demand equal to the supply.

2.5 *Weak Gross Substitutability and the Tatonnement process*

The *tatonnement process* is a price updating rule that says, “increase the prices of those goods for which the demand is greater than the supply and decrease the prices

of those goods for which the supply is greater than the demand ¹". Many versions of this process have been suggested based upon the exact nature of the update. It was shown in [2] that a continuous time version converges when the utilities satisfy the *weak gross substitutability* (WGS) property.

Definition 6 *A demand function $\mathbf{f} : \text{Int}(\mathbf{R}_+^n) \rightarrow \mathbf{R}_+^n$ satisfies **Weak Gross Substitutability** if $f_j(\mathbf{p})$ does not decrease on increasing the price of any good j' other than j :*

$$\frac{\partial f_j}{\partial p_{j'}} \geq 0, \forall j \neq j' \in A.$$

The algorithms in Chapters 3, 4 and 5 are in fact a particular discrete-step implementation of the tatonnement process, and the running time analysis can also be thought of as showing a fast convergence of this process.

¹Typically keeping the prices normalized

CHAPTER III

LINEAR UTILITIES, FISHER MODEL

3.1 *The Eisenberg-Gale Convex Program*

In this chapter we consider the Fisher model with linear utility functions, of the form $U_{ij}(\mathbf{x}) = \sum_j u_{ij}x_j$. Without loss of generality, we may assume that the supply for each good b_j is 1, by suitably scaling our unit of measurement. Let $\mathbf{p} = (p_1, \dots, p_n)$ denote a vector of prices. If at these prices buyer i is given good j , she derives u_{ij}/p_i amount of utility per unit amount of money spent. Clearly, she will be happiest with goods that maximize this ratio. Define her *bang per buck* to be $\alpha_i = \max_j \{u_{ij}/p_j\}$; clearly, for each $i \in B, j \in A$, $\alpha_i \geq u_{ij}/p_j$. If there are several goods maximizing this ratio, she is equally happy with any combination of these goods. The equilibrium allocation exhausts the endowments of all the buyers, and all the goods are sold out.

The equilibrium allocation in this case is captured by the Eisenberg-Gale convex program, which is as follows:

$$\begin{aligned}
 & \text{maximize} && \sum_{i \in B} e_i \log u_i \\
 & \text{subject to} && u_i = \sum_{j \in A} u_{ij}x_j(i) \quad \forall i \in B \\
 & && \sum_{i \in B} x_j(i) \leq 1 \quad \forall j \in A \\
 & && x_j(i) \geq 0 \quad \forall i \in B, \forall j \in A
 \end{aligned}$$

The price of good j in the equilibrium is equal to the optimum value of the Lagrangian multipliers corresponding to the second set of constraints in the above program. The Lagrangian multiplier corresponding to the first set of constraints are $1/\alpha_i$. By the Karush-Kuhn-Tucker (KKT) conditions (see Appendix A for definition),

optimal solutions to $x_j(i)$'s and p_j 's must satisfy the following conditions:

1. $\forall i \in B, j \in A : p_j \geq \frac{u_{ij}}{\alpha_i} \Leftrightarrow \alpha_i \geq \frac{u_{ij}}{p_j}.$
2. $\forall i \in B, \forall j \in A : x_j(i) > 0 \Rightarrow \alpha_i = \frac{u_{ij}}{p_j}.$
3. $\forall i \in B, \frac{e_i}{u_i} = \frac{1}{\alpha_i}.$
4. $\forall j \in A : p_j > 0 \Rightarrow \sum_{i \in A} x_j(i) = 1.$

Via these conditions, it is easy to see that an optimal solution to the Eisenberg and Gale program gives equilibrium allocations and prices for Fisher's linear case. The first two conditions make sure that the buyers are allocated only those goods that maximize his bang-per-buck. The third condition (given the first two) ensures that the endowment is exhausted, and the last makes sure that the goods are sold out. The Eisenberg-Gale program also helps prove, in a very simple manner, basic properties of the set of equilibria: Equilibrium exists under certain conditions (the mild conditions stated above), the set of equilibria is convex, equilibrium utilities and prices are unique, and if the program has all rational entries then equilibrium allocations and prices are also rational.

3.2 High level idea of the algorithm

As is usual in primal-dual algorithms, our algorithm alternates between primal and dual update steps. The primal variables in the Eisenberg-Gale program are allocations to buyers and the “dual” variables (Lagrangian multipliers) are prices of goods. Throughout the algorithm, the prices are such that buyers have surplus money left over. Each update attempts to decrease this surplus, and when it vanishes, the prices are right for the market to clear exactly.

The difficulty here is that the number of update steps executed needs to be bounded by a polynomial. This requires introducing the notion of balanced flows,

a non-trivial idea that is likely to find future applications (see Section 3.7). We explain briefly the role played by this new notion. The idea behind balanced flows is two-fold – to consider only those buyers who have a lot of surplus money and to use a more sophisticated potential function for measuring progress. Progress is measured by considering the sum of *squares* of the surpluses, instead of simply the sum. The advantage is that this potential function decreases not only when the overall surplus drops but also when the surplus moneys readjust into a more favorable configuration that can lead to a decrease in the total surplus in future iterations.

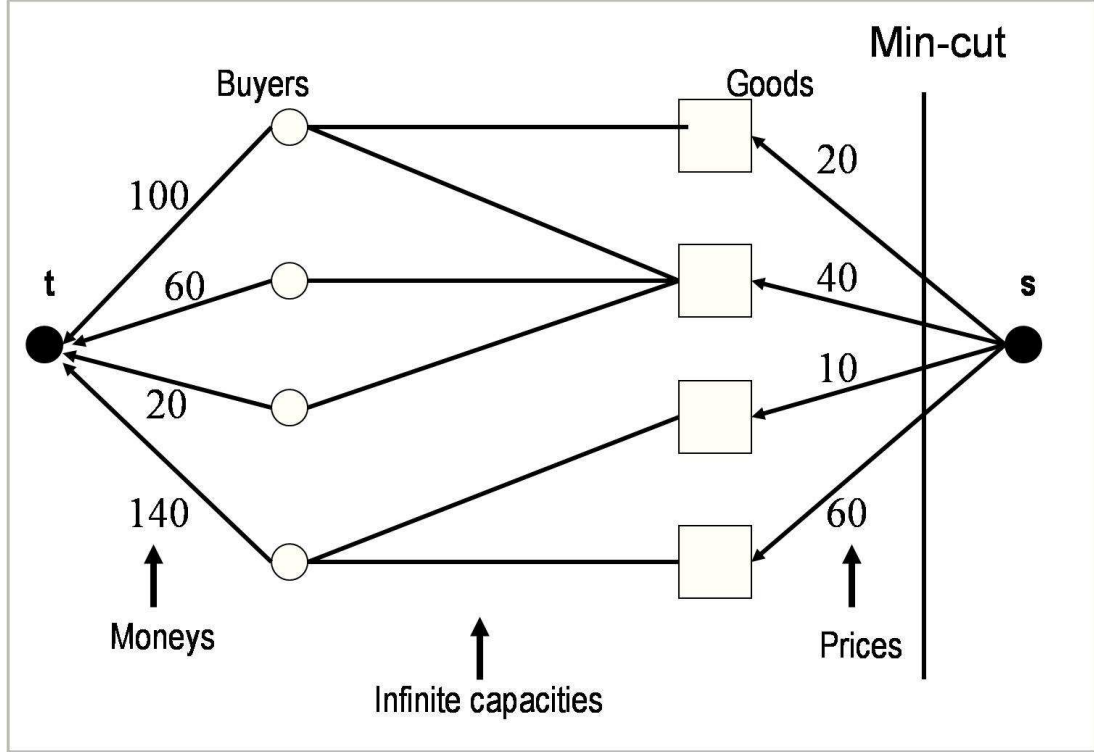
Define a bipartite graph, G , with bipartition (A, B) and for $i \in B, j \in A$, (i, j) is an edge in G iff $\alpha_i = u_{ij}/p_j$. We will call this graph the *equality subgraph* and its edges the *equality edges*.

Any goods sold along the edges of the equality subgraph will make buyers happiest, relative to the current prices. Computing the largest amount of goods that can be sold in this manner, without exceeding the budgets of buyers or the amount of goods available (assumed unit for each good), can be accomplished by computing a *max-flow* in the following network: Direct edges of G from A to B and assign a capacity of infinity to all these edges. Introduce source vertex s and a directed edge from s to each vertex $j \in A$ with a capacity of p_j . Introduce sink vertex t and a directed edge from each vertex $i \in B$ to t with a capacity of e_i . The network is clearly a function of the current prices \mathbf{p} and will be denoted $N(\mathbf{p})$. The algorithm maintains the following throughout:

Invariant 1: The prices \mathbf{p} are such that $(s, A \cup B \cup t)$ is a min-cut in $N(\mathbf{p})$ (See Figure 1.

The Invariant ensures that, at current prices, all goods can be sold. The only eventuality is that buyers may be left with surplus money. The algorithm raises prices systematically, always maintaining the Invariant, so that surplus money with buyers keeps decreasing. When the surplus vanishes, market clearing prices have

Figure 1: An Example of the network $N(\mathbf{p})$, satisfying the invariant.



been attained. This is equivalent to the condition that $(s \cup A \cup B, t)$ is also a min-cut in $N(\mathbf{p})$, i.e., max-flow in $N(\mathbf{p})$ equals the total amount of money possessed by the buyers.

Another ingredient for ensuring polynomial running time is a new combinatorial fact: understanding how the min-cut changes in $N(\mathbf{p})$ as the prices are increased in a systematic manner (see Section 3.5).

Remark 7 *With this setup, we can define our market equilibrium problem as an optimization problem: find prices \mathbf{p} under which network $N(\mathbf{p})$ supports maximum flow.*

3.3 The enhanced setting and how to deal with it

We will use the notation set up in the previous section to pinpoint the difficulties involved in solving the Eisenberg-Gale program combinatorially and the manner in

which these difficulties are circumvented.

As is well known, the primal-dual schema has yielded combinatorial algorithms for obtaining, either optimal or near-optimal, integral solutions to numerous linear programming relaxations. Other than one exception, namely Edmonds' algorithm for maximum weight matching in general graphs [18], all other algorithms raise dual variables via a greedy process.

The disadvantage of a greedy dual growth process is obvious – the fact that a raised dual is “bad”, in the sense that it “obstructs” other duals which could have led to a larger overall dual solution, may become clear only later in the run of the algorithm. In view of this, the issue of using more sophisticated dual growth processes has received a lot of attention, especially in the context of approximation algorithms. Indeed, Edmonds' algorithm is able to find an optimal dual for matching by a process that increases and decreases duals.

The problem with such a process is that it will make primal objects go tight and loose and the number of such reversals will have to be upper bounded in the running time analysis. The impeccable combinatorial structure of matching supports such an accounting and in fact this leads to a strongly polynomial algorithm. However, thus far, all attempts at making such a scheme work out for other problems have failed.

The fundamental difference between complimentary slackness conditions for linear programs and KKT conditions for nonlinear convex programs is that whereas the former do not involve both primal and dual variables simultaneously in an equality constraint (obtained by assuming that one of the variables takes a non-zero value), the latter do.

Now, our dual growth process is greedy – prices of goods are never decreased. Yet, because of the more complex nature of KKT conditions, edges in the equality subgraph appear and disappear as the algorithm proceeds. Hence, we are forced to carry out the difficult accounting process alluded to above for bounding the running

time.

We next point out which KKT conditions our algorithm enforces and which ones it relaxes, as well as the exact mechanism by which it satisfies the latter. Throughout our algorithm, we enforce the first two conditions listed in Section 3.1. As mentioned in Section 3.2, at any point in the algorithm, via a max-flow in the network $N(\mathbf{p})$, all goods can be sold; however, buyers may have surplus money left over. W.r.t. a balanced flow in network $N(\mathbf{p})$ (see Section 3.7 for a definition of such a flow), let m_i be the money spent by buyer i . Thus, buyer i 's surplus money is $\gamma_i = e_i - m_i$. We will relax the third KKT condition to the following:

$$\forall i \in B, \frac{m_i}{u_i} = \frac{1}{\alpha_i}.$$

We consider the following potential function:

$$\Phi = \gamma_1^2 + \gamma_2^2 + \dots + \gamma_{n'}^2,$$

and we give a process by which this potential function decreases by an inverse polynomial fraction in polynomial time (in each phase, as detailed in Lemma 27). When Φ drops all the way to zero, all KKT conditions are exactly satisfied.

There is a marked difference between the way we satisfy KKT conditions and the way primal-dual algorithms for LP's do. The latter satisfy complimentary conditions in *discrete steps*, i.e., in each iteration, the algorithm satisfies at least one new condition. So, if each iteration can be implemented in strongly polynomial time, the entire algorithm has a similar running time. On the other hand, we satisfy KKT conditions *continuously* – as the algorithm proceeds, the KKT conditions corresponding to each buyer get satisfied to a greater extent.

Next, let us consider the special case of Fisher's market in which all u_{ij} 's are 0/1. There is no known LP that captures equilibrium allocations in this case as well and the only recourse seems to be the special case of the Eisenberg-Gale program in

which all u_{ij} 's are restricted to 0/1. Although this is a nonlinear convex program, it is easy to derive a strongly polynomial combinatorial algorithm for solving it. Of course, in this case as well, the KKT conditions involve both primal and dual variables simultaneously. However, the setting is so easy that this difficulty never manifests itself. The algorithm satisfies KKT conditions in discrete steps, much the same way that a primal-dual algorithm for solving an LP does.

In retrospect, [38] (and perhaps other papers in the past) have implicitly given strongly polynomial primal-dual algorithms for solving nonlinear convex programs. Some very recent papers have also done so explicitly, e.g., [32]. However, the problems considered in these papers are so simple (e.g., multi-commodity flow in which there is only one source), that the enhanced difficulty of satisfying KKT conditions is mitigated and the primal-dual algorithms are not much different than those for solving LP's.

3.4 *A simple algorithm*

In this section, we give a simple algorithm (Algorithm 1), without the use of balanced flows. Although we do not know how to establish polynomial running time for it, it still provides valuable insights into the problem and shows clearly exactly where the idea of balanced flows fits in. We pick up the exposition from the end of Section 3.2.

How do we pick prices so that Invariant 1 holds at the start of the algorithm? The following two conditions guarantee this:

- The initial prices are low enough prices that each buyer can afford all the goods. Fixing prices at $1/n$ suffices, since the goods together cost one unit and all e_i 's are integral.
- Each good j has an interested buyer, i.e., has an edge incident at it in the equality subgraph. Compute α_i for each buyer i at the prices fixed in the previous step and compute the equality subgraph. If good j has no edge incident, reduce

its price to

$$p_j = \max_i \left\{ \frac{u_{ij}}{\alpha_i} \right\}.$$

The iterative improvement steps follow the spirit of the primal-dual schema: The “primal” variables are the flows in the edges of $N(\mathbf{p})$ and the “dual” variables are the current prices. The current flow suggests how to improve the prices and vice versa.

For $T \subseteq B$, define its money $m(T) = \sum_{i \in T} e_i$. W.r.t. prices \mathbf{p} , for set $S \subseteq A$, define its money $m(S) = \sum_{j \in A} p_j$; the context will clarify the price vector \mathbf{p} . For $S \subseteq A$, define its *neighborhood in $N(\mathbf{p})$*

$$\Gamma(S) = \{j \in B \mid \exists i \in S \text{ with } (i, j) \in G\}.$$

By the assumption that each good has a potential buyer, $\Gamma(A) = B$. The Invariant can now be more clearly stated.

Lemma 8 *For given prices \mathbf{p} network $N(\mathbf{p})$ satisfies the Invariant iff*

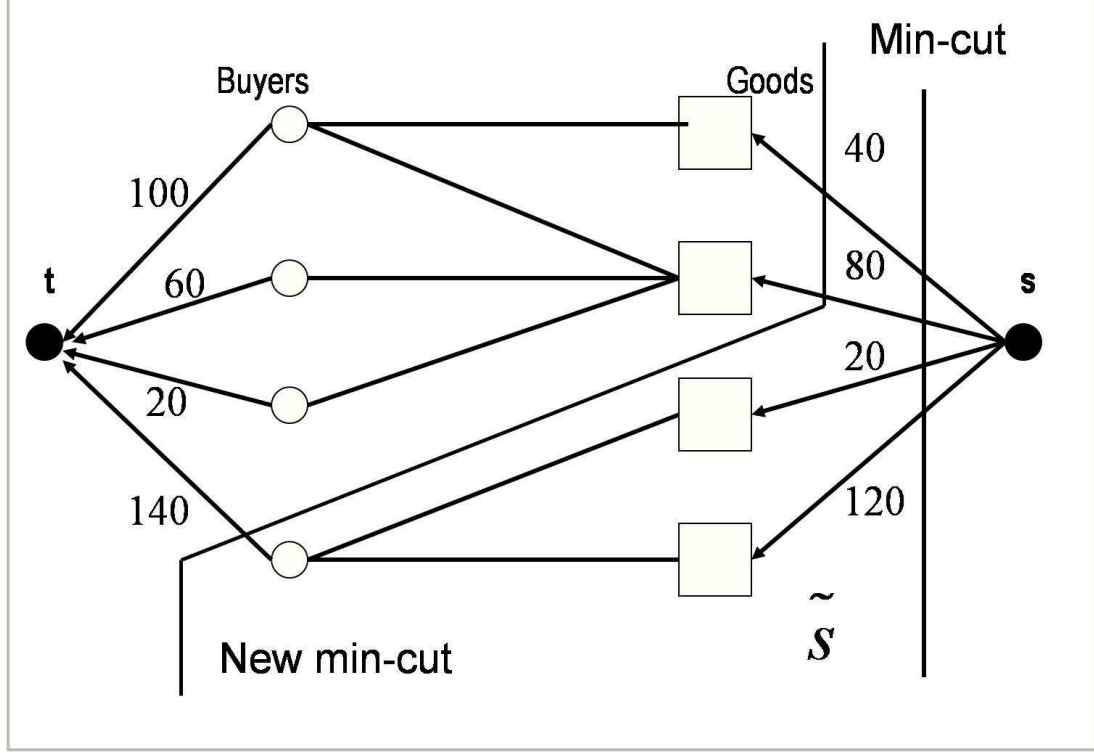
$$\forall S \subseteq A : m(S) \leq m(\Gamma(S)).$$

Proof: The forward direction is trivial, since under max-flow (of value $m(A)$) every set $S \subseteq A$ must be sending $m(S)$ amount of flow to its neighborhood.

Let’s prove the reverse direction. Assume $(s \cup A_1 \cup B_1, A_2 \cup B_2 \cup t)$ is a min-cut in $N(\mathbf{p})$, with $A_1, A_2 \subseteq A$ and $B_1, B_2 \subseteq B$. The capacity of this cut is $m(A_2) + m(B_1)$. Now, $\Gamma(A_1) \subseteq B_1$, since otherwise the cut will have infinite capacity. Moving A_1 and $\Gamma(A_1)$ to the t side also results in a cut. By the condition stated in the Lemma, the capacity of this cut is no larger than the previous one. Therefore this is also a min-cut in $N(\mathbf{p})$. Hence the Invariant holds. \square

If the Invariant holds, it is easy to see that there is a unique maximal set $\tilde{S} \subseteq A$ such that $m(\tilde{S}) = m(\Gamma(\tilde{S}))$. Say that this is the *tight set*. w.r.t. prices \mathbf{p} . When it is not clear, from the context, the prices w.r.t which \tilde{S} is defined, we write $\tilde{S}(\mathbf{p})$.

Figure 2: As the prices are increased a set \tilde{S} goes tight.



Clearly the prices of goods in the tight set cannot be increased without violating the Invariant. Hence our algorithm only raises prices of goods in the *active subgraph* consisting of the bipartition $(A - \tilde{S}, B - \Gamma(\tilde{S}))$. We will say that the algorithm *freezes* the subgraph $(\tilde{S}, \Gamma(\tilde{S}))$. Observe that in general, the bipartite graph $(\tilde{S}, \Gamma(\tilde{S}))$ may consist of several connected components (w.r.t. equality edges). Let these be $(S_1, T_1), \dots, (S_k, T_k)$.

Clearly, as soon as prices of goods in $A - \tilde{S}$ are raised, edges (i, j) with $i \in \Gamma(\tilde{S})$ and $j \in (A - \tilde{S})$ will not remain in the equality subgraph anymore. We will assume that these edges are dropped. Before proceeding further, we must be sure that these changes do not violate the Invariant. This follows from:

Lemma 9 *If the Invariant holds and $S \subseteq A$ is the tight set, then each good $j \in (A - S)$ has an edge, in the equality subgraph, to some buyer $i \in (B - \Gamma(S))$.*

Proof: Since the Invariant holds, $j \in (A - S)$ must have an equality graph edge

incident at it. If all such edges are incidents at buyers in $\Gamma(S)$, then $\Gamma(S \cup j) = \Gamma(S)$ and therefore

$$m(S \cup j) > m(S) = m(\Gamma(S)) = m(\Gamma(S \cup j)).$$

This contradicts the fact that the Invariant holds. \square

We would like to raise prices of goods in the active subgraph in such a way that the equality edges in it are retained. This is ensured by multiplying prices of all these goods by x and gradually increasing x , starting with $x = 1$. To see that this has the desired effect, observe that (i, j) and (i, l) are both equality edges iff

$$\frac{p_j}{p_i} = \frac{u_{ij}}{u_{il}}.$$

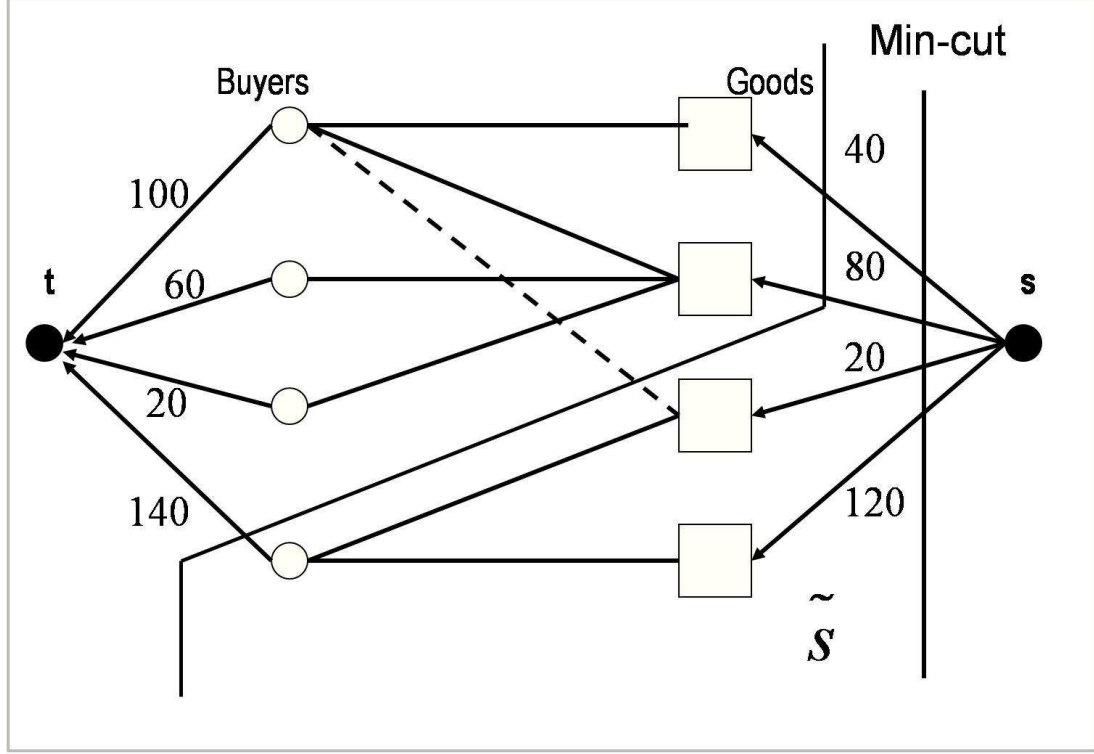
The algorithm raises x , starting with $x = 1$, until one of the following happens:

- **Event 1:** Some set $R \neq \emptyset$ goes tight in the active subgraph.
- **Event 2:** An edge (i, j) with $i \in (B - \Gamma(\tilde{S}))$ and $j \in \tilde{S}$ becomes an equality edge. (Observe that as prices of goods in $A - \tilde{S}$ are increasing, goods in \tilde{S} are becoming more and more desirable to buyers in $B - \Gamma(\tilde{S})$, which is the reason for this event.)

If Event 1 happens, we redefine the active subgraph to be $(A - (\tilde{S} \cup R), B - \Gamma(\tilde{S} \cup R))$, and proceed with the next iteration. Suppose Event 2 happens and that $j \in S_l$, for some $1 \leq l \leq k$. (Recall that $\tilde{S} = S_1 \cup S_2 \cup \dots \cup S_k$.) Because of the new equality edge (i, j) , $\Gamma(S_l) = T_l \cup i$. Therefore S_l is not tight anymore. Hence we move (S_l, T_l) into the active subgraph.

To complete the algorithm, we simply need to compute the smallest values of x at which Event 1 and Event 2 happen, and consider only the smaller of these. For Event 2, this is straightforward. Below we build an algorithm for Event 1.

Figure 3: Occurrence of Event 2.



3.5 Finding tight sets

Let \mathbf{p} denote the current price vector (i.e. at $x = 1$). We first present a lemma that describes how the min-cut changes in $N(x \cdot \mathbf{p})$ as x increases. Throughout this section, we will use the function m to denote money w.r.t. prices \mathbf{p} . W.l.o.g. assume that w.r.t. prices \mathbf{p} the tight set in G is empty (since we can always restrict attention to the active subgraph, for the purposes of finding the next tight set). Define

$$x^* = \min_{\emptyset \neq S \subseteq A} \frac{m(\Gamma(S))}{m(S)},$$

the value of x at which a nonempty set goes tight. Let S^* denote the tight set at prices $x^* \cdot \mathbf{p}$, i.e., $S^* = \tilde{S}(x^* \cdot \mathbf{p})$. If $(s \cup A_1 \cup B_1, A_2 \cup B_2 \cup t)$ is a cut in the network, we will assume that $A_1, A_2 \subseteq A$ and $B_1, B_2 \subseteq B$.

Lemma 10 *W.r.t. prices $x \cdot \mathbf{p}$:*

- if $x \leq x^*$ then $(s, A \cup B \cup t)$ is a min-cut.

- if $x > x^*$ then $(s, A \cup B \cup t)$ is not a min-cut. Moreover, if $(s \cup A_1 \cup B_1, A_2 \cup B_2 \cup t)$ is a min-cut in $N(x \cdot \mathbf{p})$ then $S^* \subseteq A_1$.

Proof: Suppose $x \leq x^*$. By definition of x^* ,

$$\forall S \subseteq A : x \cdot m(S) \leq m(\Gamma(S)).$$

Therefore by Lemma 8, w.r.t. prices $x \cdot \mathbf{p}$, the Invariant holds. Hence $(s, A \cup B \cup t)$ is a min-cut.

Next suppose that $x > x^*$. Since $x \cdot m(S^*) > x^* \cdot m(S^*) = m(\Gamma(S^*))$, w.r.t. prices $x \cdot \mathbf{p}$, the cut $(s \cup S^* \cup \Gamma(S^*), t)$ has strictly smaller capacity than the cut $(s, A \cup B \cup t)$. Therefore the latter cannot be a min-cut.

Now consider the min-cut $(s \cup A_1 \cup B_1, A_2 \cup B_2 \cup t)$. Let $S^* \cap A_2 = S_2$ and $S^* - S_2 = S_1$. Suppose $S_2 \neq \emptyset$. Clearly $\Gamma(S_1) \subseteq B_1$ (otherwise the cut will have infinite capacity). If $m(\Gamma(S_2) \cap B_2) < x \cdot m(S_2)$, then by moving S_2 and $\Gamma(S_2)$ to the s side of this cut, we can get a smaller cut, contradicting the minimality of the cut picked. In particular, $m(\Gamma(S^*) \cap B_2) \leq m(\Gamma(S^*)) = x^* \cdot m(S^*) < x \cdot m(S^*)$. Therefore $S_2 \neq S^*$, and hence, $S_1 \neq \emptyset$. Furthermore,

$$m(\Gamma(S_2) \cap B_2) \geq x \cdot m(S_2) > x^* m(S_2).$$

On the other hand,

$$m(\Gamma(S_2) \cap B_2) + m(\Gamma(S_1)) \leq x^* (m(S_2) + m(S_1)).$$

The two imply that

$$\frac{m(\Gamma(S_1))}{m(S_1)} < x^*,$$

contradicting the definition of x^* . Hence $S_2 = \emptyset$ and $S^* \subseteq A_1$. \square

Remark 11 *A more complete statement for the first part of Lemma 10, which is not essential for our purposes, is: If $x < x^*$, then $(s, A \cup B \cup t)$ is the unique min-cut in $N(x \cdot \mathbf{p})$. If $x = x^*$, then the min-cuts are obtained by moving a bunch of connected components of $(S^*, \Gamma(S^*))$ to the s -side of the cut $(s, A \cup B \cup t)$.*

Lemma 12 *Let $x = m(B)/m(A)$ and suppose that $x > x^*$. If $(s \cup A_1 \cup B_1, A_2 \cup B_2 \cup t)$ is a min-cut in $N(x \cdot \mathbf{p})$ then A_1 must be a proper subset of A .*

Proof: If $A_1 = A$, then $B_1 = B$ (otherwise this cut has ∞ capacity), and $(s \cup A \cup B, t)$ is a min-cut. But for the chosen value of x , this cut has the same capacity as $(s, A \cup B \cup t)$. Since $x > x^*$, the latter is not a min-cut by Lemma 10. Hence, A_1 is a proper subset of A . \square

Lemma 13 *x^* and S^* can be found using n max-flow computations.*

Proof: Let $x = m(B)/m(A)$. Clearly, $x \geq x^*$. If $(s, A \cup B \cup t)$ is a min-cut in $N(x \cdot \mathbf{p})$, then by Lemma 10, $x^* = x$. If so, $S^* = A$.

Otherwise, let $(s \cup A_1 \cup B_1, A_2 \cup B_2 \cup t)$ be a min-cut in $N(x \cdot \mathbf{p})$. By Lemmas 10 and 12, $S^* \subseteq A_1 \subset A$. Therefore, it is sufficient to recurse on the smaller graph $(A_1, \Gamma(A_1))$. \square

Initialization:

$\forall j \in A, p_j \leftarrow 1/n; \quad \forall i \in B, \alpha_i \leftarrow \min_j u_{ij}/p_j;$

Compute equality subgraph G ;

$\forall j \in A$ **if** $\text{degree}_G(j) = 0$ **then** $p_j \leftarrow \max_i u_{ij}/\alpha_i$;

Recompute G ;

$(F, F') \leftarrow (\emptyset, \emptyset)$ (The frozen subgraph); $(H, H') \leftarrow (A, B)$ (The active subgraph);

while $H \neq \emptyset$ **do**

$x \leftarrow 1$;

Define $\forall j \in H$, price of j to be $p_j x$;

Raise x continuously until one of two events happens:

if $S \subseteq H$ *becomes tight* **then**

Move $(S, \Gamma(S))$ from (H, H') to (F, F') ;

Remove all edges from F' to H ;

if an edge $(i, j), i \in H', j \in F$ attains equality, $\alpha_i = u_{ij}/p_j$, **then**

Add (i, j) to G ;

Move connected component of j from (F, F') to (H, H') ;

Algorithm 1: The Basic Algorithm

3.6 Termination with market clearing prices

Let M be the total money possessed by the buyers and let f be the max-flow computed in network $N(\mathbf{p})$ at current prices \mathbf{p} . Thus $M - f$ is the *surplus money* with the buyers. Let us partition the running of the algorithm into *phases*, each phase terminates with the occurrence of Event 1. Each phase is partitioned into *iterations* which conclude with a new edge entering the equality subgraph. We will show that f must be proportional to the number of phases executed so far, hence showing that the surplus must vanish in bounded time.

Let $U = \max_{i \in B, j \in A} \{u_{ij}\}$ and let $\Delta = nU^n$.

Lemma 14 *At the termination of a phase, the prices of goods in the newly tight set must be rational numbers with denominator $\leq \Delta$.*

Proof: Let S be the newly tight set and consider the equality subgraph induced on the bipartition $(S, \Gamma(S))$. Assume w.l.o.g. that this graph is connected (otherwise we prove the lemma for each connected component of this graph). Let $j \in S$. Pick a subgraph in which j can reach all other vertices $j' \in S$. Clearly, at most $2|S| \leq 2n$ edges suffice. If j reaches j' with a path of length $2l$, then $p_{j'} = ap_j/b$ where a and b are products of l utility parameters (u_{ik} 's) each. Since alternate edges of this path contribute to a and b , we can partition the u_{ik} 's in this subgraph into two sets such that a and b use u_{ik} 's from distinct sets. These considerations lead easily to showing that $m(S) = p_j c/d$ where $c \leq \Delta$. Now,

$$p_j = m(\Gamma(S))d/c,$$

hence proving the lemma. \square

Lemma 15 *Each phase consists of at most n iterations.*

Proof: Each iteration brings goods from the tight set to the active subgraph. Clearly this cannot happen more than n times without some set going tight. \square

Lemma 16 *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, $r/s > p/q$. By Lemma 14, $q \leq \Delta$ and $r \leq \Delta$. Therefore the increase in price of j ,

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

□

Lemma 17 *After k phases, $f \geq k/\Delta^2$.*

Proof: Consider phase P and let j be a good that lies in the newly tight set at the end of this phase (clearly, there is at least one such good). Let P' be the last phase, earlier than P , such that j lies in the newly tight set at the end of P' as well. If there is no such phase (because P is the first phase in which j appears in a tight set), then let P' be the start of the algorithm. Let us charge to P the entire increase in the price of j , going from P' to P (even though this increase takes place gradually over all the intermediate phases). This increase could not have already been charged to any of the earlier phases, since P is the first phase since P' that j appears in a tight set. By Lemma 16, the increase in the price of j during this period is $\geq 1/\Delta^2$. In this manner, each phase can be charged $1/\Delta^2$. The lemma follows. □

Corollary 18 *Algorithm 1 terminates with market clearing prices in at most $M\Delta^2$ phases, and executes $O(Mn^2\Delta^2)$ max-flow computations.*

3.7 Establishing polynomial running time

In this section we speed up Algorithm 1 by increasing the prices of goods adjacent only to “high-surplus” buyers. However, the surplus of a buyer might be different for two different max-flows in the same network. Therefore, we restrict ourselves to a special kind of flow called a *balanced flow* so that the surplus of a buyer is well defined.

3.7.1 Balanced flows and their properties

For a given flow f in the network $N(\mathbf{p})$, define the *surplus* of buyer $i \in B$, $\gamma_i(\mathbf{p}, f)$, to be the residual capacity of the edge (i, t) with respect to f , which is equal to m_i minus the flow sent through the edge (i, t) . Define the surplus vector $\boldsymbol{\gamma}(\mathbf{p}, f) := (\gamma_1(\mathbf{p}, f), \gamma_2(\mathbf{p}, f), \dots, \gamma_n(\mathbf{p}, f))$. Let $\|v\|$ denote the l_2 norm of vector v .

Definition 19 Balanced flow For any given \mathbf{p} , a flow that minimizes $\|\boldsymbol{\gamma}(\mathbf{p}, f)\|$ over all choices of f is called a balanced flow.

If $\|\boldsymbol{\gamma}(\mathbf{p}, f)\| < \|\boldsymbol{\gamma}(\mathbf{p}, f')\|$, then we say f is more balanced than f' .

A balanced flow has to be a max-flow, since otherwise sending a positive flow along an augmenting path from s to t clearly decreases $\|\boldsymbol{\gamma}(\mathbf{p}, f)\|$.

Lemma 20 For any given \mathbf{p} , all balanced flows in $N(\mathbf{p})$ have the same surplus vector.

Proof: It is easy to see that if γ_1 and γ_2 are the surplus vectors w.r.t flows f_1 and f_2 , then $(\gamma_1 + \gamma_2)/2$ is the surplus vector w.r.t the flow $(f_1 + f_2)/2$. So the set of all feasible surplus vectors is a convex region. A balanced flow minimizes a strictly concave function of the surplus vector, and so the surplus vector is unique. \square

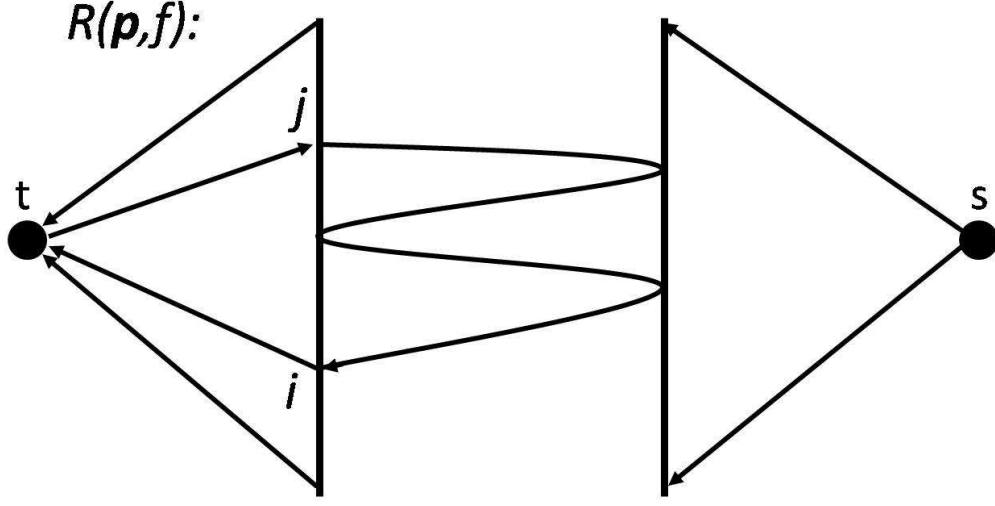
As a result, one can define the surplus vector for a given price as $\boldsymbol{\gamma}(\mathbf{p}) := \boldsymbol{\gamma}(\mathbf{p}, f)$ where f is a balanced flow in $N(\mathbf{p})$. For a given \mathbf{p} and a flow f in $N(\mathbf{p})$, let $R(\mathbf{p}, f)$ be the residual network of $N(\mathbf{p})$ with respect to the flow f . The following property characterizes a balanced flow among all max-flows.

Property 1 If $\gamma_j(\mathbf{p}, f) < \gamma_i(\mathbf{p}, f)$, then there is no path from node j to node i in $R(\mathbf{p}, f) \setminus \{s, t\}$.

Theorem 21 A maximum-flow f is balanced iff it has Property 1.

Proof: Suppose f is a balanced flow. Let $\gamma_i(\mathbf{p}, f) > \gamma_j(\mathbf{p}, f)$ for some i and $j \in B$, and suppose for the sake of contradiction, that there is a path from j to i in $R(\mathbf{p}, f) \setminus \{s, t\}$. The capacity of (t, j) is positive in $R(\mathbf{p}, f)$ since otherwise the only edge going out of j in $R(\mathbf{p}, f)$ would be (j, t) . Also $\gamma_i(\mathbf{p}, f) > 0$, so (i, t) has a positive capacity in $R(\mathbf{p}, f)$. Hence one can send a circulation of positive value along $t \rightarrow j \rightarrow i \rightarrow t$ in $R(\mathbf{p}, f)$, decreasing γ_i and increasing γ_j . The resulting flow is more balanced than f , contradicting the fact that f is a balanced flow (See Figure 4).

Figure 4: If $\gamma_i(\mathbf{p}, f) > \gamma_j(\mathbf{p}, f)$ and there is a path from j to i in $R(\mathbf{p}, f) \setminus \{s, t\}$, then a circulation would give a more balanced flow.



To prove the other direction, we show that a flow f satisfying Property 1 is locally optimum w.r.t the l_2 norm of the surplus vector as the objective function. In fact, any circulation in $R(\mathbf{p}, f)$ can only send flow from a high surplus buyer to a low surplus buyer resulting in a less balanced flow. Since the l_2 norm is a strictly concave function, a locally optimum solution is also globally optimum.

□

Theorem 22 *Given a network $N(\mathbf{p})$ a balanced flow can be computed using at most n max-flow computations.*

Proof: One iteration of the algorithm is as follows: From the given network N reduce the capacities of all edges that go from B to t continuously at the same rate, except for those edges whose capacity becomes zero. Stop when the capacity of the cut $(\{s\} \cup A \cup B, \{t\})$ is equal to that of the cut $(\{s\}, A \cup B \cup \{t\})$. Call this new network N' .

Let (S, T) be the maximal min-cut in N' ; $s \in S, t \in T$. If $T = \{t\}$, then $(\{s\}, A \cup B \cup \{t\})$ is also a min-cut in N' , and the corresponding max-flow is a balanced flow in the original network N . Otherwise, let N_1 and N_2 be the subnetworks of N induced

by $S \cup \{t\}$ and $T \cup \{s\}$ respectively. We claim that the union of balanced flows in N_1 and N_2 is a balanced flow in N . The networks N_1 and N_2 are vertex disjoint, except for s and t , and hence in the next iteration, a max-flow in both N_1 and N_2 can be computed simultaneously and will be counted as just one max-flow computation. Since in each iteration, the size of the biggest partition decreases by at least 1, n max-flow computations suffice.

In order to prove the claim, we show that such a flow has Property 1. Recall that Property 1 says that for a pair of nodes $i, j \in B$, if $\gamma_i > \gamma_j$ then there is no path from j to i in $R(\mathbf{p}, f) \setminus \{s, t\}$. Note that there is no edge from $N_1 \cap A$ to $N_2 \cap B$. So Property 1 can be violated only when $i \in N_1$, $j \in N_2$ and $\gamma_i > \gamma_j$. We show that this cannot happen; we show that the surplus of all buyers in N_1 (in a balanced flow) is smaller than that of all buyers in N_2 .

First of all, all nodes $\hat{i} \in B$ such that the capacity of (\hat{i}, t) is zero in N' , are in N_1 , since S is the maximal min-cut in N' . So for all nodes i in $B \cap N_2$, the same amount, say X , was subtracted from the capacity of (i, t) in N . Clearly, the average surplus in N_2 is greater than X . In fact, for all $K \subseteq A \cap N_2$, the average surplus in $L := \Gamma(K)$ is $> X$. Now let L be the set of all buyers in N_2 with the lowest surplus. Let K be the set of goods reachable from L in the residual graph w.r.t a balanced flow in N_2 . From Property 1, it follows that $L = \Gamma(K)$, and hence the average surplus in L which is also the lowest surplus in N_2 is $> X$. A similar argument shows that the biggest surplus in N_1 is at most X .

□

Lemma 23 *If f and f^* are respectively a feasible and a balanced flow in $N(\mathbf{p})$ such that $\gamma_i(\mathbf{p}, f^*) = \gamma_i(\mathbf{p}, f) - \delta$, for some $i \in B$ and $\delta > 0$, then $\|\gamma(\mathbf{p}, f)^*\|^2 \leq \|\gamma(\mathbf{p}, f)\|^2 - \delta^2$.*

Proof: Suppose we start with f and get a new flow f' by decreasing the surplus of i by δ , and increasing the surpluses of some other buyers in the process. We show that this already decreases the l_2 norm of the surplus vector by δ^2 and so the lemma follows.

Consider the flow $f^* - f$. Decompose this flow into flow paths and circulations. Among these, augment f with only those that go through the edge (i, t) , to get f' . These are either paths that go from s to i to t , or circulations that go from i to t to some i_l and back to i . Then $\gamma_i(f') = \gamma_i(f^*) = \gamma_i(f) - \delta$ and for a set of vertices i_1, i_2, \dots, i_k , we have $\gamma_{i_l}(f') = \gamma_{i_l}(f) + \delta_l$, s.t. $\delta_1, \delta_2, \dots, \delta_k > 0$ and $\sum_{l=1}^k \delta_l \leq \delta$. Moreover, for all l , there is a path from i to i_l in $R(\mathbf{p}, f^*)$. Since f^* is balanced, and satisfies Property 1, $\gamma_i(f') = \gamma_i(f^*) \geq \gamma_{i_l}(f^*) \geq \gamma_{i_l}(f')$.

By Lemma 24, $\|\gamma(\mathbf{p}, f')\|^2 \leq \|\gamma(\mathbf{p}, f)\|^2 - \delta^2$ and since f^* is a balanced flow in $N(\mathbf{p})$, $\|\gamma(\mathbf{p}, f^*)\|^2 \leq \|\gamma(\mathbf{p}, f')\|^2$.

□

Lemma 24 *If $a \geq b_i \geq 0, i = 1, 2, \dots, n$ and $\delta \geq \sum_{j=1}^n \delta_j$ where $\delta, \delta_j \geq 0, j = 1, 2, \dots, n$, then $\|(a, b_1, b_2, \dots, b_n)\|^2 \leq \|(a + \delta, b_1 - \delta_1, b_2 - \delta_2, \dots, b_n - \delta_n)\|^2 - \delta^2$.*

Proof:

$$(a + \delta)^2 + \sum_{i=1}^n (b_i - \delta_i)^2 - a^2 - \sum_{i=1}^n b_i^2 \geq \delta^2 + 2a(\delta - \sum_{i=1}^n \delta_i) \geq 0$$

□

Remark 25 *In a set of feasible vectors, a vector v is called min-max fair iff for every feasible vector u and an index i such that $u_i < v_i$ there is a j for which $u_j < v_j$ and $v_j < v_i$. Similarly, v is max-min fair iff $u_i > v_i$ implies that there is a j for which $u_j < v_j$ and $v_j > v_i$. The surplus vector of a balanced flow is both min-max and max-min fair. Even though we don't use this property for our results, we note it since it could be of independent interest.*

3.7.2 The polynomial time algorithm

The main idea of Algorithm 2 is that it tries to reduce $\|\gamma(\mathbf{p}, f)\|$ in every phase. Intuitively, this goal is achieved by finding a set of high-surplus buyers in the balanced flow and increasing the prices of goods in which they are interested. If a subset becomes tight as a result of this increase, we have reduced $\|\gamma(\mathbf{p}, f)\|$ because the surplus of a formerly high-surplus buyer is dropped to zero. The other event that can happen is that a new edge is added to the equality subgraph. In that case, this edge will help us to make the surplus vector more balanced: we can reduce the surplus of high-surplus buyers and increase the surplus of low-surplus ones. This operation will result in the reduction of $\|\gamma(\mathbf{p}, f)\|$.

Initialization:

$\forall j \in A, p_j \leftarrow 1/n;$

$\forall i \in B, \alpha_i \leftarrow \min_j u_{ij}/p_j;$

Define $G(A, B, E)$ with $(i, j) \in E$ iff $\alpha_i = u_{ij}/p_j$;

$\forall j \in A$ **if** $\text{degree}_G(j) = 0$ **then** $p_j \leftarrow \max_i u_{ij}/\alpha_i$;

Recompute G ; $\delta = M$;

repeat

 Compute a balanced flow f in G ;

 Define δ to be the maximum surplus in B ;

 Define H to be the set of buyers with surplus δ ;

repeat

 Let H' be the set of neighbors of H in A ;

 Remove all edges from $B \setminus H$ to H' ;

$x \leftarrow 1$; Define $\forall j \in H'$, price of j to be $p_j x$;

 Raise x continuously until one of the two events happens:

Event 1: An edge $(i, j), i \in H, j \in A \setminus H'$ attains equality, $\alpha_i = u_{ij}/p_j$;

 Add (i, j) to G ;

 Recompute a balanced flow f ;

 In the residual network corresponding to f in G , define I to be the set of buyers that can reach H ; $H \leftarrow H \cup I$;

Event 2: $S \subseteq H$ becomes tight;

until some subset $S \subseteq H$ is tight;

until A is tight ;

Algorithm 2: A Polynomial Time Algorithm

The algorithm starts with finding a price vector that does not violate the invariant. The rest of the algorithm is partitioned into *phases*. In each phase, we have an active graph (H, H') with $H \subset B$ and $H' \subset A$ and we increase the prices of goods in H' like Algorithm 1. Let δ be the maximum surplus in B . The subset H is initially the set of buyers whose surplus is equal to δ . H' is the set of goods adjacent to buyers in H . A phase ends when the surplus of some buyer in H becomes zero.

Each phase is divided into *iterations*. In each iteration, we increase the prices of goods in H' . An iteration ends when either a new edge joins the equality subgraph or a subset becomes tight. The active subgraph (H, H') changes between iterations. Let \mathbf{p}_t and H_t be the price vector and the set of nodes in H at the end of the t 'th iteration in that phase. Let \mathbf{p}_0 and H_0 denote the prices and the set of nodes in H before the first iteration.

If a new edge is added to the equality subgraph at the end of an iteration, we recompute the balanced flow f . Then we define H_t by adding to H_{t-1} all vertices that can reach some vertex of H_{t-1} in $R(\mathbf{p}_t, f) \setminus \{s, t\}$. If a subset becomes tight as a result of increase of the prices, then the phase terminates.

Let k denote the number of iterations in the phase. Every time an edge is added to the equality subgraph, $|H'|$ is increased by at least one. Therefore k is at most n .

Define $\delta_t = \min_{i \in H_t}(\gamma_i(\mathbf{p}_t))$, for $0 < t \leq k$. $\delta_0 = \delta$ and the phase ends when some subset goes tight, which means that the surplus of some buyer in H becomes zero; so $\delta_k = 0$.

Lemma 26 *If $\delta_{t-1} - \delta_t \geq 0$ then there exists an $i \in H$ such that $\gamma_i(\mathbf{p}_{t-1}) - \gamma_i(\mathbf{p}_t) \geq \delta_{t-1} - \delta_t$.*

Proof: Consider the residual network $R(\mathbf{p}_t, f)$ corresponding to the balanced flow computed at the end of iteration t . By definition of H_t , every vertex $v \in H_t \setminus H_{t-1}$ can reach a vertex $i \in H_{t-1}$ in $R(\mathbf{p}_t, f)$ and therefore, by Theorem 21, $\gamma_v(\mathbf{p}_t) \geq \gamma_i(\mathbf{p}_t)$.

This means that minimum surplus δ_t is achieved by a vertex i in H_{t-1} . Hence, the surplus of vertex i is decreased by at least $\delta_{t-1} - \delta_t$ during iteration t . \square

Lemma 27 *If \mathbf{p}_0 and \mathbf{p}_k are price vectors before and after a phase, $\|\gamma(\mathbf{p}_k)\|^2 \leq \|\gamma(\mathbf{p}_0)\|^2(1 - \frac{1}{n^2})$.*

Proof: In every iteration we increase prices of goods in H or add new edges to the equality subgraph. Moreover, all the edges of the network that are deleted in the beginning of a phase have zero flow. Therefore, the balanced flow computed at iteration $t-1$ is a feasible flow for $N(\mathbf{p}_t)$. So if $\delta_{t-1} - \delta_t < 0$, then $\|\gamma(\mathbf{p}_t)\| \leq \|\gamma(\mathbf{p}_{t-1})\|$. Otherwise, by Lemmas 26 and 3.7.1 $\|\gamma(\mathbf{p}_t)\| \leq \|\gamma(\mathbf{p}_{t-1})\| - (\delta_{t-1} - \delta_t)^2$. Since $\delta_0 = \delta$ and $\delta_k = 0$,

$$\|\gamma(\mathbf{p}_k)\|^2 \leq \|\gamma(\mathbf{p}_0)\|^2 - \frac{\delta^2}{n}.$$

Now $\|\gamma(\mathbf{p}_0)\|^2 \leq \delta^2 n$ so

$$\|\gamma(\mathbf{p}_k)\|^2 \leq \|\gamma(\mathbf{p}_0)\|^2(1 - \frac{1}{n^2}).$$

\square

By the bound given in the above, it is easy to see that after $O(n^2)$ phases, $\|\gamma(\mathbf{p})\|^2$ is reduced to at most half of its previous value. In the beginning, $\|\gamma(\mathbf{p})\|^2 \leq M^2$. Once the value of $\|\gamma(\mathbf{p})\|^2 \leq \frac{1}{\Delta^4}$, the algorithm takes at most one more step. This is because Lemma 14, and consequently, Lemma 16 holds for Algorithm 2 as well. Hence, the number of phases is at most

$$O(n^2 \log(\Delta^4 M^2)) = O(n^2(\log n + n \log U + \log M))$$

As noted before, the number of iterations in each phase is at most n . Each iteration requires at most $O(n)$ max-flow computations.

Hence we get:

Theorem 28 *Algorithm 2 executes at most*

$$O\left(n^3(\log n + n \log U + \log M)\right)$$

max-flow computations and finds market clearing prices.

3.8 Discussion

Our algorithm is not strongly polynomial. Indeed, obtaining such an algorithm is an important open question remaining. It will require a qualitatively different approach, perhaps one which satisfies KKT conditions in discrete steps, as is the rule with all other primal-dual algorithms known today (as pointed out in Section 3.3, we start by suitably relaxing the KKT conditions and our algorithm satisfies these conditions continuously rather than in discrete steps). In fact it is not even known whether the machinery developed in Section 3.7 is necessary for obtaining a polynomial time algorithm, i.e., does the algorithm given in Sections 3.4 and 3.5 have a polynomial running time? If not, it would be nice to find a family of instances on which it takes super-polynomial time.

The primal-dual schema first introduced by Kuhn [37] in the context of solving bipartite matching, and over the years it led to the most efficient known algorithms for many fundamental problems in \mathbf{P} , including matching, flows, shortest paths and branchings. The mechanism of relaxing complementary slackness conditions, first identified and formalized in [44], gave an adaptation of this schema to the setting of approximation algorithms, where again it yielded algorithms with good approximation factors and running times for several fundamental problems. Is there a suitable mechanism that yields an adaptation of this paradigm to finding approximation algorithms for solving nonlinear convex programs? This question is particularly significant for nonlinear programs since other than rare exceptions, such programs will have only irrational solutions on some inputs and so a combinatorial algorithm (e.g., one that does not output its solutions in radicals) will necessarily have to find an approximate

solution.

CHAPTER IV

LINEAR UTILITIES, ARROW-DEBREU MODEL

In this chapter we show that the basic machinery of Algorithm 2 can be extended to compute an approximate equilibrium in the Arrow-Debreu model. The main difference between the Fisher and the Arrow-Debreu model is that in the Fisher model, each buyer i has a specific endowment of money, $e(i)$, where as in the Arrow-Debreu model, the endowment of each buyer i is a bundle of goods $\mathbf{e}(i) \in \mathbf{R}^n$.

4.1 *The Algorithm*

Given prices \mathbf{p} , the income $e(i)$ of buyer i in the Arrow-Debreu model is $\sum_j p_j e_j(i)$. Most of the machinery in Chapter 3 is based on the assumption that the income is fixed as we change the prices, so it cannot be used as is. We get around this difficulty by letting the incomes not reflect the changing prices all the time, and instead by updating them periodically.

It was natural in the Fisher model that we start with low prices so that $(s, A \cup B \cup t)$ is a min-cut in $N(\mathbf{p})$ and maintain this as an invariant while increasing prices, until $(s \cup A \cup B, t)$ is also a min-cut. The progress of the algorithm is measured by the surplus.

But the Arrow-Debreu model calls for a different approach. This is because, in the Arrow-Debreu model prices are scale-invariant: multiplying all the prices by the same value has no net effect on the market, so there are no “low” prices. Instead, we start with the price vector $\mathbf{p} = \mathbf{1}^n$. Also whenever the incomes are updated, the capacity of the cuts $(s, A \cup B \cup t)$ and $(s \cup A \cup B, t)$ are equal to each other. So clearly the invariant as before cannot be maintained. However, this is not critical to the rest of the algorithm, so we can do without it. The algorithm increases prices continuously

until the cuts $(s, A \cup B \cup t)$ and $(s \cup A \cup B, t)$ are approximately min-cuts in $N(\mathbf{p})$. As a result we get an approximate market equilibrium. Progress here is measured as the relative surplus, that is, the ratio of the surplus to the sum¹ of all prices.

The Algorithm is organized into *epochs*; the incomes of the buyers are fixed during an epoch and they are updated between two epochs. An epoch involves running several phases, identical to Algorithm 2.

Let $P = \sum_j p_j$ denote the total prices of all goods, $M = \sum_i e(i)$ denote the total income of all the buyers, and f denote the value of a max-flow in $N(\mathbf{p})$.

The algorithm is as follows: start with the price vector $\mathbf{p} = \mathbf{1}^n$ and compute the incomes $e(i) = \sum_j p_j e_j(i)$. Run an epoch until $|\gamma(\mathbf{p})| \leq n\epsilon$, where ϵ is the desired approximation. If, at the end of an epoch, either $P - M \leq n\epsilon$ or $P \geq \frac{n}{\epsilon}$, then end the algorithm. Otherwise update the incomes and run the next epoch.

Initialization:

$\forall j \in A, p_j \leftarrow 1;$

$\forall i \in B, e(i) \leftarrow \sum_j p_j e_j(i);$

$P = \sum_j p_j, M = \sum_i e(i), f = \text{value of a max-flow in } N(\mathbf{p});$

repeat

$\forall i \in B, e(i) \leftarrow \sum_j p_j e_j(i);$

repeat

 Run a phase as in Algorithm 2;

until $|\gamma(\mathbf{p})| \leq n\epsilon;$

until $P - M \leq n\epsilon$ or $P \geq \frac{n}{\epsilon} ;$

Algorithm 3: Algorithm for Linear Utilities: Arrow-Debreu Model

4.2 Analysis of the Algorithm

Theorem 29 ([16]) *For all $\epsilon > 0$ Algorithm 3 gives a 4ϵ -approximate market equilibrium and needs $O\left(\frac{n^4}{\epsilon^2} \log \frac{n}{\epsilon}\right)$ max-flow computations.*

The first important observation is that throughout the algorithm, the value of $P - f$ is non-increasing, even though P is always increasing. Given this observation,

¹Recall that we assumed $b_j = 1$, so this is the total worth of all goods.

if the algorithm is run long enough (until $P \geq \frac{n}{\epsilon}$) then $P - f$ would be $\leq P\epsilon$ and this is a sufficient condition for an approximate equilibrium (Lemma 31). The question is, how many epochs does this require. The answer is that in every epoch, we may assume that P increases by at least $n\epsilon$, and hence it requires $O\left(\frac{1}{\epsilon^2}\right)$ epochs.

Lemma 30 *Throughout a run of Algorithm 3, $P - f$ never increases.*

Proof: During an epoch, the algorithm always increases the price of a good j for which the edge (s, j) is saturated. Therefore any increase in P always results in an equal increase in f .

At the end of an epoch, when the $e(i)$'s are updated, P does not change, but f can only increase. This is because $e(i)$'s can only increase as a result of the update.

Hence $P - f$ never increases. \square

Lemma 31 *For a price vector \mathbf{p} , if $P - f \leq \epsilon P$, then \mathbf{p} is a 2ϵ -approximate market equilibrium.*

Proof: It follows from the observation that there exists an allocation with $|\xi(\mathbf{p}) - \mathbf{p}| = 2(P - f)$.

Suppose $(s \cup A_1 \cup B_1, A_2 \cup B_2 \cup t)$ is a min-cut in $N(\mathbf{p})$, with $A_1, A_2 \subseteq A$ and $B_1, B_2 \subseteq B$.

Consider the buyers in B_1 . Their optimal allocation is given by a max-flow since the edges (i, t) are saturated for all $i \in B_1$. For every good in A_1 , the residual capacity of the edge $(s, j) = \xi_j(\mathbf{p}) - p_j$. Hence $\sum_{j \in A_1} |\xi_j(\mathbf{p}) - p_j| = P - f$.

Consider the buyers in B_2 . A max-flow may not saturate the edges (i, t) for all $i \in B_2$. In any case, we can allocate optimal baskets to buyers in B_2 by augmenting a max-flow in order to exhaust all the surplus. In any such allocation, for all $j \in A_2$, $\xi_j(\mathbf{p}) \geq p_j$ and $\sum_{j \in A_2} |\xi_j(\mathbf{p}) - p_j| = \text{total surplus of all buyers} = M - f = P - f$.

\square

Lemma 32 *If \mathbf{p}_0 and \mathbf{p}^* are price vectors before and after an epoch, then the number of phases in the epoch is $O\left(n^2 \log\left(\frac{|\gamma(\mathbf{p}_0)|}{|\gamma(\mathbf{p}^*)|}n\right)\right)$.*

Proof: This follows almost immediately from Lemma 27. \square

Proof:[of Theorem 29] **Correctness:** Note that $P \geq n$, $P \geq f$ and $P - f \leq n$. $|\gamma(\mathbf{p})| = M - f$, so at the end of each epoch $M - f \leq n\epsilon$.

If at the end of any epoch, $P - M \leq n\epsilon$, then

$$P - f = (P - M) + (M - f) \leq 2n\epsilon \leq 2P\epsilon.$$

On the other hand, if $P \geq \frac{n}{\epsilon}$, then again

$$P - f \leq n \leq P\epsilon.$$

Running time: Note that the increase in P during an epoch is exactly equal to $P - M$ at the end of that epoch (since at the beginning, they were both equal). If in each epoch $P - M > n\epsilon$, then after $\frac{1}{\epsilon^2}$ epochs $P \geq \frac{n}{\epsilon}$.

Since at the beginning of each epoch $|\gamma(\mathbf{p})| \leq P - f \leq n$ and the epoch ends if $|\gamma(\mathbf{p})| \leq n\epsilon$, there are $O(n^2 \log \frac{n}{\epsilon})$ phases in each epoch.

Moreover, from Lemma 15, each phase needs $O(n^2)$ max-flow computations.

Hence the algorithm needs $O\left(\frac{n^4}{\epsilon^2} \log \frac{n}{\epsilon}\right)$ max-flow computations.

\square

4.3 Improved Running time

The running time can be brought down by a more complicated rule to end epochs (and update incomes). In the earlier algorithm, we only used the fact that $P - f \leq n$. However as the algorithm progresses $P - f$ keeps getting smaller and smaller. In general, suppose $P - f \leq n\alpha$. Then it is enough to stop when $P \geq \frac{n\alpha}{\epsilon}$. The improvement in the running time comes from showing that in every epoch we can guarantee that either P increases by $\frac{n\alpha}{4}$, or α decreases geometrically.

Initialization: $\forall j \in A, p_j \leftarrow 1;$ $\forall i \in B, e(i) \leftarrow \sum_j p_j e_j(i);$ $\alpha \leftarrow 1;$ $P = \sum_j p_j, M = \sum_i e(i), f = \text{value of a max-flow in } N(\mathbf{p});$ **repeat** $\quad \forall i \in B, e(i) \leftarrow \sum_j p_j e_j(i);$ **repeat** $\quad \quad \text{Run a phase as in Algorithm 2;}$ $\quad \text{until } |\gamma(\mathbf{p})| \leq \frac{n\alpha}{4};$ $\quad \text{if } P - M \leq \frac{n\alpha}{4} \text{ then}$ $\quad \quad \alpha \leftarrow \alpha/2;$ $\text{until } \alpha \leq \epsilon \text{ or } P \geq \frac{n\alpha}{\epsilon};$ **Algorithm 4:** Algorithm with an improved running time

Theorem 33 *For all $\epsilon > 0$ Algorithm 4 gives an ϵ -approximate market equilibrium and needs $O\left(\frac{n^4}{\epsilon} \log^2 \frac{n}{\epsilon}\right)$ max-flow computations.*

Proof: We show that at any point in the algorithm, $P - f \leq n\alpha$. The proof is by induction. Initially $\alpha = 1$ and the inequality holds. It continues to hold while α remains unchanged. $\alpha \leftarrow \alpha/2$ is executed only when $P - M \leq \frac{n\alpha}{4}$, and $M - f \leq \frac{n\alpha}{4}$. Hence at this point $P - f \leq \frac{n\alpha}{2}$.

If at the end of an epoch, $\alpha \leq \epsilon$, then $P - f \leq n\epsilon \leq P\epsilon$. Otherwise if $P \geq \frac{n\alpha}{\epsilon}$, then too $P - f \leq n\alpha \leq P\epsilon$.

At the end of each epoch either $\alpha \leftarrow \alpha/2$ or P increases by at least $\frac{n\alpha}{4}$. The former can happen $O(\log \frac{1}{\epsilon})$ times before $\alpha \leq \epsilon$, and the latter $O(\frac{1}{\epsilon})$ times before $P \geq \frac{n\alpha}{\epsilon}$. The theorem follows. \square

4.4 Comparison with Related Work

Deng, Papadimitriou and Safra[14] gave a polynomial time algorithm for the Arrow-Debreu model when the number of goods is bounded. Jain, Mahdian and Saberi [30] gave the first PTAS for the case considered in this chapter. In particular they get

an ϵ -approximate approximation that requires $O\left(\frac{1}{\epsilon}\right)$ iterations of solving the Fisher case exactly. Their algorithm, in general, depends on the size of the numbers giving the utility rates and endowments of the buyers. Note that the running time of our algorithm depends only on n, n' and ϵ . This is analogous to the standard notion of strongly polynomial time algorithms where the running time is independent of the *size of the numbers* occurring in the instance. The improvement comes about because [30] use the algorithm in [15] as a black box, whereas we open it up and build upon the main ideas in [15]. Garg and Kapoor [26] gave some very interesting approximate equilibrium algorithms for the linear case of both models using an auction based approach. The running time of this algorithm has a smaller dependency on the n and n' , but it is not strongly polynomial time either. Finally, Jain [29] gave an exact algorithm based on solving a convex program using the ellipsoid method and Diophantine approximation. Ye [45] gave an interior point based method to solve the same convex program.

CHAPTER V

SPENDING CONSTRAINT UTILITIES

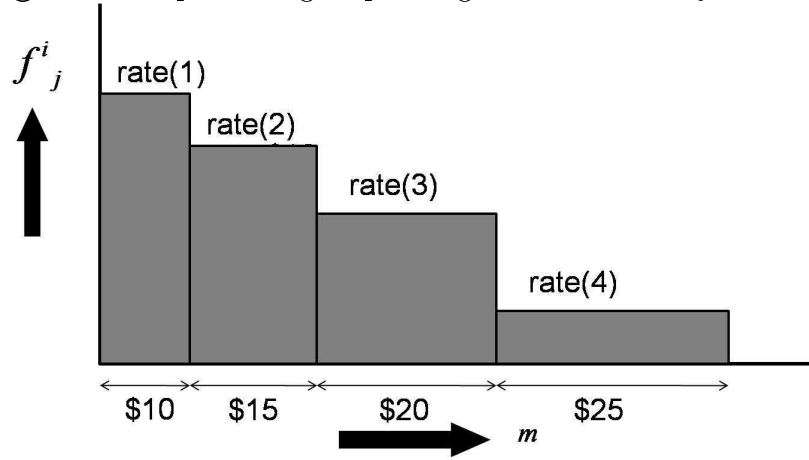
5.1 *Definition*

Most naturally occurring utility functions have decreasing marginal utilities. A natural candidate to model such utilities are piecewise linear concave utilities. However, no algorithm to compute equilibrium for such utilities is known. One reason why an algorithm such as Algorithm 2 does not generalize is that they do not satisfy WGS. The spending constraint utilities (defined by Vazirani [42]) have the best of both worlds: they capture the property of decreasing marginal utilities, while at the same time, a generalization of Algorithm 2 computes the equilibrium in polynomial time. The spending constraint utilities differ from the classical utility functions in that they depend¹ on the prices of the goods. We will first define a very special case of the spending constraint utilities, and progressively consider more general versions.

Spending constraint utilities are additively separable, that is they are of the form $U_i(\mathbf{x}) = \sum_{j \in A} U_{ij}(x_j)$. Suppose that the utility of buyer i for good j is linear, but in addition to that, he specifies a constraint on the amount of money spent on good j . Such constraints are natural, we typically have budgets on how much we spend on food, rent and entertainment, for instance. In other words the optimization problem that the buyer solves now has an additional constraint of the form $p_j x_j \leq \text{budget}(j)$. Yet another way to state the same thing is that the utility is linear up to the point when $p_j x_j = \text{budget}(j)$, and beyond that the utility does not increase any more. The important new aspect is that this transition is in terms of the amount of money spent

¹Quasi-linear utility functions, which are of the form $U(\mathbf{x}) = v(\mathbf{x}) - \sum_j p_j x_j$ also depend on the prices. However, the dependency there arises simply because buyers value money as well. The dependency here is more intricate.

Figure 5: Representing a spending constraint utility function.



(and hence depends on the price of the good) and not in terms of the quantity of good consumed.

More generally, having spent a certain amount of money on good j , the buyer may decide that his rate² of utility for that good transitions to something lower (instead of transitioning to zero, as before). In fact there may be several such transitions, each occurring when he spends different amounts of money at that rate. Such a utility function can be represented by a decreasing step function that gives the rate of utility as a function of the amount of money spent. Contrast this with the representation of a piecewise linear and concave utility function by specifying its derivative, which gives the rate of utility as a function of the quantity of good consumed.

Let f_j^i be a decreasing step function representing such a utility. Each step of f_j^i is called a *segment*. Suppose that the transitions happen at b_1, b_2, \dots and so on. Let segment k have range $[b_{k-1}, b_k]$ and let $rate(k)$ be the value of f_j^i in this segment. Given the price p_j and x_j , suppose that $p_j x_j \in [b_{k-1}, b_k]$. Then the buyer derives utility at rate $rate(1)$ for the first b_1 dollars, or equivalently, the first $\frac{b_1}{p_j}$ units of good j . Similarly, he derives utility at rate $rate(2)$ for the next $b_2 - b_1$ dollars, or

²This is the utility derived per unit of the good. For linear utility functions the rate is a constant.

equivalently, the next $\frac{b_2-b_1}{p_j}$ units of good j , and so on. Therefore

$$U_{ij}(x_j) = \frac{\text{rate}(1)b_1}{p_j} + \frac{\text{rate}(2)(b_2 - b_1)}{p_j} + \dots + \frac{\text{rate}(k)(p_j x_j - b_{k-1})}{p_j}.$$

We will use a more succinct way of representing the same:

$$U_{ij}(x_j) = \frac{\int_0^{p_j x_j} f_j^i(m) dm}{p_j}.$$

This representation naturally leads us to the most general definition of spending constraint utilities. We simply allow f_j^i to be any decreasing function. The utility is still defined by the same equation as above.

In the Arrow-Debreu model, the buyers' income itself is dependent on the prices. So now the utility is represented by functions $g_j^i : [0, 1] \rightarrow \mathbf{R}_+$. The argument to the function is now the *fraction* of the income spent, and the value of the function gives the rate of utility as before. Once the prices are known, g_j^i 's can be appropriately scaled to get the f_j^i 's. More precisely, let $f_j^i(m) = g_j^i(ye(i))$ and the utility $U_{ij}(x_j)$ is as before equal to $\int_0^{p_j x_j} f_j^i(m) dm / p_j$.

5.2 Existence and Uniqueness of Equilibrium

5.2.1 The Fisher Model

5.2.1.1 Characterizing the optimal bundle

Suppose that each f_j^i is either continuous and strictly decreasing in $[0, e(i)]$, or zero. Further, if for each j there is at least one i such that f_j^i is non-zero, then say that the rate functions are *nice* for this instance. For the rest of this section, we will restrict ourselves to nice rate functions.

Recall that for a linear utility function, $\sum_j u_{ij} x_j$, the optimal bundle of goods contains only those that maximize the “bang per buck”. That is, if $\alpha_i := \max_{j \in A} \left\{ \frac{u_{ij}}{p_j} \right\}$, then either $\frac{u_{ij}}{p_j} = \alpha_i$ or $x_j(i) = 0$. Among the goods that maximize the bang per buck, it does not matter in what proportion each of them is allocated. A similar characterization can be obtained for these utilities. But now the rate of utility decreases as

you spend more and more money. Therefore the money spent on the goods is spread in such a way that the bang-per-buck is equalized across all goods. Unless, of course, the initial rate of utility for some good is so low that we never spend any money on it.

Suppose that i spends M_j^i amount of money on good j when he buys his optimum bundle. Let

$$\alpha_i := \max_{j \in A} \left\{ \frac{f_j^i(M_j^i)}{p_j} \right\}.$$

Then either $\frac{f_j^i(M_j^i)}{p_j} = \alpha_i$ or $M_j^i = 0$. Further, since the buyers don't have any utility for money, $\sum_{j \in A} M_j^i = e(i)$.

But how do we find such a bundle? The idea is that given an α_i , it is easy to find the M_j^i 's such that either $\frac{f_j^i(M_j^i)}{p_j} = \alpha_i$ or $M_j^i = 0$. So the question really is to find the right α_i . And the one that we need is so that the resulting M_j^i 's satisfy $\sum_{j \in A} M_j^i = e(i)$. We formalize this idea below.

Suppose that $f : [a, b] \rightarrow \mathbf{R}_+$ is continuous and strictly decreasing. Then f is invertible in $[f(b), f(a)]$. Note that $f^{-1} : [f(b), f(a)] \rightarrow [a, b]$ is also continuous and strictly decreasing. If f is identically zero, then define f^{-1} to be identically zero as well. Let f_j^{-i} be the inverse of f_j^i .

Given a target rate of bang per buck value α_i for buyer i , the money that he should spend on good j is given by $M_j^i = f_j^{-i}(\alpha_i p_j)$.³ Note that $\alpha_i p_j$ could be greater than $f_j^i(0)$, in which case the inverse does not exist. This happens when the initial rate of utility for the good is too small to achieve a bang per buck of α_i . We can fix this by defining $f_j^{-i}(x) = 0$ for all $x \geq f_j^i(0)$. This preserves the continuity of f_j^{-i} since f_j^{-i} is zero at $f_j^i(0)$. Similarly, the inverse may not exist for $x \leq f_j^i(e(i))$. This happens when the rate of utility is so high that even after spending all the money, the bang per buck is still greater than α_i . We fix this by extending f_j^{-i} to all $x \leq f_j^i(e(i))$

³Another way to think about $f_j^{-i}(\alpha_i p_j)$ is that it is the length of the line segment $y = \alpha_i$ between the y -axis and the curve $y = f_j^i(x)/p_j$.

by defining $f_j^{-i}(x)$ to be $e(i)$ at all these points.

This suggests a way to compute α_i . Recall that $\sum_{j \in A} M_j^i = e(i)$. Therefore α_i is a solution to the equation:

$$\sum_{j \in A} f_j^{-i}(\alpha_i p_j) = e(i). \quad (1)$$

Since the demand for buyer i is more or less determined by α_i , proving that α_i is a well-behaved function of \mathbf{p} translates to similar properties about the demand.

Lemma 34 *If the rate functions are nice and $\mathbf{p} \in \text{Int}(\mathbf{R}_+^n)$, then α_i is uniquely determined by (1). Moreover, α_i is a continuous and non-increasing function of \mathbf{p} .*

Proof: Let $h^i(x, \mathbf{p}) := \sum_{j \in A} f_j^{-i}(x p_j)$. For a given \mathbf{p} , h^i is a continuous and strictly decreasing function in x as long as the rate functions are nice. Therefore there is a unique α_i such that $h^i(\alpha_i, \mathbf{p}) = e(i)$. (With some abuse of notation, one may write $\alpha_i = h^{-i}(e(i), \mathbf{p})$.) Similarly, h^i is a continuous and non-increasing function in \mathbf{p} . Therefore, α_i is a continuous and non-increasing function of \mathbf{p} .

□

Since α_i is unique, there is a unique bundle of goods that maximizes buyer i 's utility. Therefore, we can talk of *the* demand function, $\xi_j(\mathbf{p}) = \sum_{i \in B} M_j^i = \sum_{i \in B} f_j^{-i}(\alpha_i p_j)$. Since we already established that f_j^{-i} is continuous, and that α_i is a continuous function of \mathbf{p} (Lemma 34), $\xi(\mathbf{p})$ is a continuous function of \mathbf{p} . Since we assumed that there is a unit amount of each good, the market equilibrium condition can be restated as $\xi(\mathbf{p}) = \mathbf{p}$.

5.2.1.2 Existence of Equilibria

We will show the existence of equilibria by appealing to Brouwer's Fixed Point Theorem. It requires a continuous function \mathbf{g} defined on a non-empty, compact, convex subset S of \mathbf{R}^n into itself, so that the fixed point corresponds to the equilibrium. A natural candidate is to let $\mathbf{g} = \xi$ and $S = \{\mathbf{p} : p_j \geq 0, \sum_{j \in A} p_j = \sum_{i \in B} e(i)\}$, the

scaled simplex of price vectors. This almost works, except that ξ is defined only in the interior of S .

To fix this, we need to extend the function to the boundary of S . That is, when one or more of the prices is zero. In fact note that as p_j tends to 0, $\xi_j(\mathbf{p})$ tends to infinity. So we show that if p_j is small enough then $\xi_j(\mathbf{p}) > p_j$, no matter what the other prices are, and hence it cannot be an equilibrium. In summary, the equilibrium point is guaranteed to be bounded away from the boundary. So what we do is let g be equal to ξ in the interesting region, Sin , and simply extend it to the rest of S in a continuous manner.

Theorem 35 *For all markets in the Fisher model with spending constraint utilities with nice rate functions, there exists an equilibrium price vector.*

Proof: Consider any $j \in A$. We first find a lower bound l_j on the price p_j in any equilibrium. Let $i \in B$ be such that f_j^i is non-zero. Let l_j be such that $e(i) \geq l_j > 0$ and

$$\frac{f_j^i(l_j)}{l_j} \geq \frac{f_{j'}^i(e(i)/n)}{e(i)/n}, \forall j' \neq j.$$

There exists such an l_j since $\frac{f_j^i(l_j)}{l_j}$ tends to infinity as l_j tends to zero. If $p_j = l_j$, then $M_j^i \geq l_j$.

Now we use these lower bounds to divide S into two parts: $Sin := \{\mathbf{p} \in S : p_j \geq l_j, \forall j \in A\}$ and $Sout := S \setminus Sin$. Let $\mathbf{g}(\mathbf{x}) = \xi(\mathbf{x})$ for all $\mathbf{x} \in Sin$. If $\mathbf{x} \in Sout$, then let $\mathbf{g}(\mathbf{x}) = \xi(\bar{\mathbf{x}})$ where $\bar{\mathbf{x}}$ is the point in Sin that is closest to \mathbf{x} .

It is easy to show that an equilibrium price vector cannot lie in $Sout$. If $\mathbf{x} \in Sout$, then for some $j, x_j < l_j$. $\bar{x}_j = l_j$ and $\xi_j(\bar{\mathbf{x}}) \geq l_j$. Hence, $\mathbf{g}(\mathbf{x}) \neq \mathbf{x}$. Therefore any fixed point of \mathbf{g} must lie in Sin .

Finally, note that $\bar{\mathbf{x}}$ is continuous in \mathbf{x} , and ξ is also continuous, which implies that \mathbf{g} is continuous. Therefore $\exists \mathbf{x}^*$ such that $\mathbf{g}(\mathbf{x}^*) = \mathbf{x}^* = \xi(\mathbf{x}^*)$. \square

5.2.1.3 Uniqueness

The uniqueness of equilibrium follows from the fact that the function ξ satisfies Weak Gross Substitutability, and the following curious property.

Definition 36 *A demand function $f : \text{Int}(\mathbf{R}_+^n) \rightarrow \mathbf{R}_+^n$ satisfies **Scale Invariance** if f does not change when all the prices are multiplied by the same non-zero scalar:*

$$f(\mathbf{p}) = f(\theta \mathbf{p}), \forall \theta > 0.$$

Scale invariance is not a property typical of the Fisher model, since buyers have a fixed endowment of money. However, ξ measures the amount of money spent, which turns out to be scale invariant. Notice for instance, that in the case of linear utilities, multiplying all the prices the same amount should leave the *amount of money spent* on the goods invariant.

Lemma 37 *If the demand function ξ of a market satisfies Weak Gross Substitutability, and Scale Invariance then the equilibrium prices are unique.*

Proof: Suppose that there are two equilibrium price vectors, \mathbf{p} and \mathbf{q} . Consider

$$\theta := \max_{j \in A} \left(\frac{p_j}{q_j} \right).$$

Let the maximum be attained for good 1, without loss of generality. That is $p_1 = \theta q_1$. If \mathbf{p} and \mathbf{q} were different, then $p_1 > q_1$. We will prove the theorem by showing a contradiction to this. In particular we show that $\xi_1(\mathbf{p}) \leq \xi_1(\mathbf{q})$. This is sufficient since \mathbf{p} and \mathbf{q} are equilibrium price vectors, and hence $\xi_1(\mathbf{p}) = p_1$ and $\xi_1(\mathbf{q}) = q_1$.

By definition of θ , for all j , $\theta q_j \geq p_j$, i.e., $\theta \mathbf{q}$ is component-wise bigger than or equal to \mathbf{p} . Now consider the process that starts with \mathbf{p} and raises the price of each good until it is $\theta \mathbf{q}$. Since we only increase the prices of goods other than 1, ξ_1 does not decrease during this process (by Weak Gross Substitutability of ξ), i.e., $\xi_1(\theta \mathbf{q}) \geq \xi_1(\mathbf{p})$. However, by Scale Invariance of ξ , we know that $\xi_1(\theta \mathbf{q}) = \xi_1(\mathbf{q})$ and we are done. \square

Lemma 38 *The demand function ξ satisfies Weak Gross Substitutability.*

Proof: It is enough to prove that $f_j^{-i}(\alpha_i p_j)$ is a non-decreasing function of p_j . But the only dependence on p_j comes via α_i . Since f_j^{-i} is non-increasing, it is enough to prove that α_i is a non-increasing function of p_j . But this has already been established in Lemma 34. \square

Lemma 39 *The demand function ξ satisfies Scale Invariance.*

Proof: Suppose α_i is the solution to (1) at the price vector \mathbf{p} , and $M_j^i = f_j^{-i}(\alpha_i p_j)$. Note that at the price vector $\theta \mathbf{p}$, α_i/θ is the solution to (1) and the money spent is still $M_j^i = f_j^{-i}(\frac{\alpha_i}{\theta} \theta p_j)$. Therefore M_j^i is Scale Invariant, and in turn, ξ is Scale Invariant. \square

From Lemmas 38, 39 and 37, the following theorem follows.

Theorem 40 *The equilibrium price vector for any market in the Fisher model, with spending constraint utilities with nice rate functions is unique.*

5.2.2 The Arrow-Debreu Model

As in Section 5.2.1, we assume that the rate functions are nice, that is, each g_j^i is either continuous and strictly decreasing in $[0, 1]$, or zero; and for each good j there is at least one buyer i such that g_j^i is non-zero. The characterization of an optimal bundle is as before, and translating it in terms of g_j^i 's gives us that $M_j^i = e(i)g_j^{-i}(\alpha_i p_j)$ and

$$\sum_{j \in A} g_j^{-i}(\alpha_i p_j) = 1. \quad (2)$$

As in Lemma 34, we have that α_i is uniquely determined by (2) and α_i is a continuous and non-increasing function of \mathbf{p} . And so there is a unique bundle that maximizes buyer i 's utility. Therefore the demand function is given by $x_j(i) = M_j^i/p_j$, and the excess demand function is given by $\zeta_j(\mathbf{p}) = \sum_i x_j(i) - 1$. The following lemma is immediate.

Lemma 41 ζ is continuous.

We show that the excess demand in our case has essentially all the properties that the classic model has and that are sufficient to guarantee the existence of equilibria (Lemma 2). The proofs in this subsection follow the approach in [1] very closely.

Theorem 42 *Equilibrium prices exist for all markets in the Arrow-Debreu model with nice rate functions.*

Proof: First of all, $\zeta(\cdot)$ is scale invariant. It follows from the fact that the budget set, $\{\mathbf{x} : \mathbf{p} \cdot \mathbf{x} \leq \mathbf{p} \cdot \mathbf{e}(i)\}$ does not change when the prices are scaled.

Now we prove that the excess demand function in our case satisfies the hypothesis of Lemma 2.

1. By Lemma 41, ζ is continuous. Since the demand functions are all positive, it is bounded from below.
2. This follows from the fact that each buyer spends all the money he has. Which implies that the total money spent is equal to the total money earned, i.e., $\mathbf{x}(i, \mathbf{p}) \cdot \mathbf{p} = \mathbf{e}(i) \cdot \mathbf{p}$. Since $\zeta(\cdot) = \sum_{i \in B} \mathbf{x}(i, \cdot) - \mathbf{e}(i)$, it implies $\mathbf{p} \cdot \zeta(\mathbf{p}) = 0$.
3. Since $\mathbf{p}_m \rightarrow \mathbf{p}$, \exists a \mathbf{q} such that $\mathbf{p}_m \leq \mathbf{q}$, $\forall m$. Now for any \mathbf{p}_m , and sufficiently large m , $p_k^m \geq \delta$, for some $\delta > 0$. Now $\{\zeta_k(\mathbf{p}_m)\} \leq (\sum_i \mathbf{e}(i)) \cdot \mathbf{p}_m / p_k^m \leq (\sum_i \mathbf{e}(i)) \cdot \mathbf{q} / \delta$.
4. Suppose not, i.e, there exists a bounded subsequence of $\zeta(\mathbf{p}_m)$. Then for each i , \exists a bounded subsequence of $\mathbf{x}(i, \mathbf{p}_m)$. This in turn implies that there exists a convergent subsequence. It follows from Lemma 43 that $\mathbf{p} \gg 0$, which is a contradiction.

Hence there exists a price vector with $\zeta(\mathbf{p}) = 0$, which is precisely the market equilibrium condition. \square

Lemma 43 *Suppose that the sequence $\{\mathbf{p}_m\}$, $\mathbf{p}_m \in \text{Int}(\mathbf{R}_+^n)$ is such that $\mathbf{p}_m \rightarrow \mathbf{p}$ and $\mathbf{x}(i, \mathbf{p}_m) \rightarrow \mathbf{x}$ Then*

1. $\mathbf{x} = \mathbf{x}(i, \mathbf{p})$.

2. $\mathbf{p} \in \text{Int}(\mathbf{R}_+^n)$.

Proof:

Suppose $\mathbf{y} \cdot \mathbf{p} \leq e(i)$, i.e, \mathbf{y} is in the budget set of buyer i . Then we will show that $\mathbf{x} \succeq \mathbf{y}$, i.e, \mathbf{x} is preferred to \mathbf{y} by buyer i . Let $0 < \lambda < 1$. $(\lambda \mathbf{y}) \cdot \mathbf{p} < e(i)$. So $\exists m_0$ such that $\forall m \geq m_0$, $(\lambda \mathbf{y}) \cdot \mathbf{p}_m < e(i)$. Since $\mathbf{x}(i, \mathbf{p}_m)$ was the demand at \mathbf{p}_m , $\mathbf{x}(i, \mathbf{p}_m) \succeq \lambda \mathbf{y}$. So $\mathbf{x} \succeq \lambda \mathbf{y}$. Since this is true for all $0 < \lambda < 1$, by the continuity of the demand function, $\mathbf{x} \succeq \mathbf{y}$.

If $p_j = 0$ for some j , then the budget set is unbounded. Since the demand at \mathbf{p} is bounded, it follows that $\mathbf{p} \in \text{Int}(\mathbf{R}_+^n)$. \square

In the Arrow-Debreu model, equilibrium prices may not be unique. Consider two agents, with endowments $(1, 0)$ and $(0, 1)$ respectively. Suppose that for each agent, the utility for his good far outweighs the utility for the other good. Then the market clears for many different prices, in which each agent buys only what he has.

5.3 Algorithms for Step Functions

In this subsection, we summarize the main ideas from [42] that we use here. We assume that all buyers have (discrete) spending constraint utilities, i.e., each f_j^i is a decreasing step function.

Given non-zero prices $\mathbf{p} = (p_1, \dots, p_n)$, define the *bang per buck* relative to \mathbf{p} for segment $s \in \text{seg}(i, j)$, to be $\text{rate}(s)/p_j$. Sort all segments $s \in \text{segments}(i)$ by decreasing bang per buck, and partition by equality into classes: Q_1, Q_2, \dots . At prices \mathbf{p} , goods corresponding to any segment in Q_l make i equally happy, and those in Q_l make i strictly happier than those in Q_{l+1} .

At any intermediate point in the algorithm, certain segments are already allocated. By *allocating segment* s , $s \in \text{seg}(i, j)$, we mean allocating $\text{value}(s)$ worth of good j to buyer i . The exact quantity of good j allocated will only be determined at termination, when prices are finalized. The algorithm will maintain that for each buyer i , there is an integer t_i such that the set of segments allocated to i correspond exactly to all segments in partitions Q_1, \dots, Q_{t_i-1} and a subset of Q_{t_i} . We will say that the *current partition* for buyer i , denoted by $Q^{(i)}$, is $Q_{t_i} \setminus \text{allocated segments}$. Clearly, there is a unique t_i such that $Q^{(i)}$ is non-empty, unless all the segments have been allocated.

Define the *current bang per buck* of buyer i , $\alpha(i)$, to be the bang per buck of partition $Q^{(i)}$. This is the rate at which i derives utility, per dollar spent, for allocations from $Q^{(i)}$ at current prices. Next, we define the *equality subgraph* $G = (A, B, E)$ on bipartition A, B and containing edges E . Corresponding to each buyer i and each segment $s \in Q^{(i)}$, E contains the edge (i, j) , where $\text{good}(s) = j$.

Let $\text{allocated}(j)$ denote the total value of good j already allocated and let $\text{spent}(i)$ denote the sum of the amount spent by buyer i on allocated segments. Thus, when segment s is allocated, $\text{value}(s)$ is added to $\text{allocated}(j)$ and to $\text{spent}(i)$. Also, define the *money left over* with buyer i , $m(i) = e(i) - \text{spent}(i)$. Denote by \mathbf{a} , \mathbf{s} and \mathbf{m} the vectors of current allocations, amounts spent and left over money, i.e., $(\text{allocated}(j), j \in A)$, $(\text{spent}(i), i \in B)$ and $(m(i), i \in B)$, respectively. We will carry over all these definitions to sets, e.g. for a set $S \subseteq A$, $\mathbf{m}(S)$ will denote $\sum_{j \in S} m(j)$.

We next define network $N(\mathbf{p}, \mathbf{a}, \mathbf{s})$, which is a function of the current prices, allocations and amounts spent. Direct all edges of the equality subgraph, G , from A to B . Add a source vertex s , and directed edges (s, j) , for each $j \in A$ and having capacity $p_j - \text{allocated}(j)$. Add a sink vertex t , and directed edges (i, t) , for each $i \in B$ and having capacity $m(i)$. The capacity of edge (j, i) is $c_{ji} = \text{value}(s)$.

For $S \subseteq A$, define its *neighborhood in the equality subgraph* to be

$$\Gamma(S) = \{i \in B \mid \exists j \in S \text{ with } (i, j) \in G\}.$$

For $A' \subseteq A$ and $B' \subseteq B$, define $c(A'; B')$ to be the sum of capacities of all edges from A' to B' in $N(\mathbf{p}, \mathbf{a}, \mathbf{s})$. For $S \subseteq A$, define

$$\text{best}(S) = \min_{T \subseteq \Gamma(S)} \{\mathbf{m}(T) + c(S; \Gamma(S) - T)\},$$

and define $\text{bestT}(S)$ to be a maximal subset of $\Gamma(S)$ that optimizes the above expression. Observe that $\text{best}(S)$ is the capacity of the min-cut separating t from S in $N(\mathbf{p}, \mathbf{a}, \mathbf{s})$. Also observe that if T_1 and T_2 optimize the above expression, then $i \in T_1 - T_2$ must satisfy $m(i) = c(S; i)$. Hence $\text{bestT}(S)$ is unique. A set $S \subseteq A$ that satisfies $\mathbf{p}(S) - \mathbf{a}(S) = \text{best}(S)$ will be called a *tight set*.

Given flow f in the network $N(\mathbf{p}, \mathbf{a}, \mathbf{s})$ let $R(\mathbf{p}, \mathbf{a}, \mathbf{s}, f)$ denote the residual graph w.r.t. f . Define the *surplus* of buyer i , $\gamma_i(\mathbf{p}, f)$, to be the residual capacity of the edge (i, t) with respect to f , i.e., $m(i)$ minus the flow sent through the edge (i, t) . The *surplus vector* is defined to be $\boldsymbol{\gamma}(\mathbf{p}, f) := (\gamma_1(\mathbf{p}, f), \gamma_2(\mathbf{p}, f), \dots, \gamma_n(\mathbf{p}, f))$. Let $\|v\|$ denote the l_2 norm of vector v . A *balanced flow* in network $N(\mathbf{p}, \mathbf{a}, \mathbf{s})$ is a maximum flow that minimizes $\|\boldsymbol{\gamma}(\mathbf{p}, f)\|$.

We next define subroutine **freeze** which is used in the main algorithm. Subroutine **freeze** computes a balanced flow f in network $N(\mathbf{p}, \mathbf{a}, \mathbf{s})$. Let δ be the maximum surplus of a buyer in this flow and let $B_2 \subseteq B$ be the set of buyers having this surplus. Let $A_2 \subseteq A$ be the set of goods that are adjacent to these buyers in the equality subgraph. Let $A_1 = A \setminus A_2$ and $B_1 = B \setminus B_2$. For each saturated edge (j, i) , i.e., $f(j, i) = c_{ji}$, with $i \in B_2$ and $j \in A_1$, allocate the corresponding segment to i . As a result of these new allocations, there may be buyers that do not have equality edges incident at them. If each buyer has equality subgraph edges incident at it, subroutine **freeze** halts. Otherwise, for each buyer not having such edges,

it computes the current partition of this buyer and adds edges corresponding to it. **freeze** goes back to recomputing a balanced flow in the resulting network.

We next describe one *phase* of the algorithm. In a single phase, the incomes of the buyers, $e(i)$'s are assumed to be fixed. So the description of a phase is identical in both the Fisher and the Arrow-Debreu model. At the beginning of a phase, assume that prices \mathbf{p} and endowments $e(i)$'s are given and run **freeze**. Let the *active subgraph* be the subgraph of G induced by (A_2, B_2) . Multiply the current price, p_j , of each good j in the active subgraph by x . Initialize $x = 1$, and start raising x continuously. As x is raised, one of three events could take place :

- **Event 1:** As prices increase, a subset of A_2 may become tight. If so, the current phase comes to an end.
- **Event 2:** For buyers in B_2 , goods in A_1 are becoming more and more desirable (since their prices are not changing, whereas prices of goods in A_2 are increasing). As a result, a segment $s \in \text{seg}(i, j)$, $i \in B_2$, $j \in A_1$ may enter into the current partition of buyer i , $Q^{(i)}$. When this happens, edge (j, i) is added to the equality subgraph with capacity $\text{value}(s)$ and call subroutine **freeze**. The new active subgraph consists of all buyers and goods that have a residual path in $R(\mathbf{p}, \mathbf{a}, \mathbf{s}, f) - \{s, t\}$ to the current active subgraph (and contains the current active subgraph).
- **Event 3:** Suppose $i \in B_1$ has a segment $s \in \text{seg}(i, j)$ allocated to it, where $j \in A_2$. Because the price of j is increasing, at some point the bang per buck of this segment may equal α_i , i.e., segment s enters i 's current partition. When this happens, we will *deallocate* segment s , i.e., subtract $\text{value}(s)$ from $\text{allocated}(j)$ and from $\text{spent}(i)$. The action taken is same as Event 2.

Let's analyze the running time of one phase. Event 2 can happen at most n times, because each time a new good enters the equality subgraph. Event 3 can happen at

most Z times, since any segment can be deallocated at most once in a phase. The total number of executions of subroutine **freeze** in a phase is $O(Z + n)$. [42] show that each execution of a phase requires $O(n)$ max-flow computations. Moreover, in each phase, the l_2 norm of the surplus vector is reduced by a polynomial fraction.

Lemma 44 ([42]) *If \mathbf{p}_0 and \mathbf{p}^* are price vectors before and after a phase, $\|\boldsymbol{\gamma}(\mathbf{p}^*)\|^2 \leq \|\boldsymbol{\gamma}(\mathbf{p}_0)\|^2(1 - \frac{1}{n(n+Z)})$.*

Note that this lemma is analogous to Lemma 27, which was used to bound the running times of Algorithms 2 and 4. By similar arguments, we get the following theorems.

Theorem 45 ([42]) *There exists an algorithm that for any Fisher market with spending constraint utilities with step functions, finds equilibrium using $O(n^2(n+Z)^2(\log n + n \log U + \log \frac{1}{\epsilon}))$ max-flow computations.*

Theorem 46 *There exists an algorithm that for any Arrow-Debreu market with spending constraint utilities with step functions, finds an ϵ -approximate equilibrium using $O\left(\frac{n^2(n+Z)^2}{\epsilon} \log \frac{n}{\epsilon}\right)$ max-flow computations.*

5.4 Algorithms for Continuous Functions

We now give an algorithm for the Fisher model when the rate functions are nice. Assume that the algorithm is given oracle access to the f_j^i 's. The algorithm is simple: approximate the given functions with step functions where all the segments are of length ϵ . More precisely, $F_j^i(x) := f_j^i(\lceil \frac{x}{\epsilon} \rceil \epsilon)$ is the required approximation. Now run the algorithm of [42] with F_j^i 's as input, and return the price vector thus obtained, say \mathbf{p} . Let M_j^i be the money that buyer i spends on good j when he buys the optimal bundle at prices \mathbf{p} (w.r.t the functions f_j^i 's), and \mathcal{M}_j^i be the money that i spends on j according to the algorithm. (Note that the F_j^i 's are step functions, so there need not be a unique optimal bundle. Hence we consider the allocation given by the algorithm

of [42].) We show that \mathcal{M}_j^i is in fact a good approximation to M_j^i as in the following lemma:

Lemma 47 *Let M_j^i and \mathcal{M}_j^i be as defined above. Let $n = |A|$. Then,*

$$\forall i \in B, j \in A, \mathcal{M}_j^i - n\epsilon \leq M_j^i \leq \mathcal{M}_j^i + \epsilon. \quad (3)$$

Proof: Let $n_j^i := \lfloor M_j^i / \epsilon \rfloor$, i.e., $n_j^i \epsilon \leq M_j^i < (n_j^i + 1)\epsilon$. If we show that $n_j^i \epsilon \leq \mathcal{M}_j^i$, then we get that

$M_j^i \leq \mathcal{M}_j^i + \epsilon$. We may assume that $0 < n_j^i$, since otherwise the inequality trivially follows.

Note that we chose our approximation F_j^i of f_j^i such that $F_j^i(x) \leq f_j^i(x)$ for all x . Therefore $\max_{j \in A} F_j^i(\mathcal{M}_j^i) / p_j \leq \alpha_i = f_j^i(M_j^i) / p_j$ since $0 < n_j^i$. Therefore $F_j^i(\mathcal{M}_j^i) \leq f_j^i(M_j^i) \leq f_j^i(n_j^i \epsilon) = F_j^i(n_j^i \epsilon)$, by the definition of F_j^i . Since F_j^i is non-increasing, we get that $n_j^i \epsilon \leq \mathcal{M}_j^i$.

Recall that $\sum_{j \in A} \mathcal{M}_j^i = \sum_{j \in A} M_j^i = e(i)$. Therefore, for any j , $\mathcal{M}_j^i - M_j^i = \sum_{j' \in A, j' \neq j} (\mathcal{M}_{j'}^i - M_{j'}^i) \leq n\epsilon$. \square

Note that $\sum_{i \in B} M_j^i = \xi_j(\mathbf{p})$ (by definition) and $\sum_{i \in B} \mathcal{M}_j^i = p_j$ (since \mathbf{p} is market clearing for the F_j^i 's). Now summing (3) over all $i \in B$, we get that $\forall j \in A$, $p_j - n'n\epsilon \leq \xi_j(\mathbf{p}) \leq p_j + n'\epsilon$, where $n' = |B|$. Therefore,

$$|\xi_j(\mathbf{p}) - p_j| \leq n'n\epsilon. \quad (4)$$

Summing over all $j \in A$, we get that \mathbf{p} is indeed an $n'n^2\epsilon$ -approximate market equilibrium, since we may assume w.l.o.g that the sum of all prices is at least 1.

We have actually proved a stronger version of approximation, i.e., ξ is component-wise close to \mathbf{p} , and that the error is absolute (additive). The definition only needed that the respective sums be close, and the error be relative (multiplicative). In fact, more is true: that the allocation returned by the algorithm (i.e., i spends \mathcal{M}_j^i on j), is almost optimal w.r.t the f_j^i 's.

Let ∇U_j^i be the optimum utility of i at \mathbf{p} , the prices returned by the algorithm, minus the utility i derives from the allocation returned by the algorithm.

$$\nabla U_j^i = \int_{\mathcal{M}_j^i}^{M_j^i} \frac{f_j^i(y)}{p_j} dy \leq (M_j^i - \mathcal{M}_j^i) \frac{f_j^i(0)}{l_j},$$

where l_j is as defined in the proof of Theorem 35. We already proved that $M_j^i - \mathcal{M}_j^i \leq \epsilon$. Moreover, l_j is independent of ϵ and is only polynomial in $1/n$. Hence $\nabla U_j^i \leq \epsilon \text{poly}(n)$.

We can extend this algorithm to the continuous case of the Arrow-Debreu model, the only difference being that the algorithm for step functions only gives an ϵ -approximate equilibrium. It can be shown that the composition of the two algorithms is still ϵ -approximate. As in the Fisher model, we approximate the given functions g_j^i 's by step functions. Since $f_j^i(y) = g_j^i(ye(i))$, and $e(i) \leq n/\epsilon$, we sample g_j^i 's at steps of length ϵ^2/n , ensuring that the steps of f_j^i are always smaller than ϵ . Let $G_j^i(x) := g_j^i(\lceil \frac{xn}{\epsilon^2} \rceil \epsilon^2/n)$. Run the algorithm with G_j^i 's as input, and return the price vector thus obtained, say \mathbf{p} . Let M_j^i be the money that buyer i spends on good j when he buys the optimal bundle at prices \mathbf{p} (w.r.t the functions g_j^i 's), and \mathcal{M}_j^i be the money that i spends on j according to the algorithm. Lemma 47 still holds. But now

$$\begin{aligned} \sum_{j \in A} \left| \sum_{i \in B} \mathcal{M}_j^i - p_j \right| &\leq \epsilon \sum_{j \in A} p_j \\ \text{Therefore } \sum_{j \in A} \left| \sum_{i \in B} M_j^i - p_j \right| &\leq \sum_{j \in A} \left| \sum_{i \in B} \mathcal{M}_j^i - p_j \right| + \sum_{j \in A} \left| \sum_{i \in B} (\mathcal{M}_j^i - M_j^i) \right| \\ &\leq \epsilon \sum_{j \in A} p_j + n^2 n' \epsilon \\ &\leq (nn' + 1) \epsilon \sum_{j \in A} p_j \end{aligned}$$

since $p_j \geq 1$ for all j .

Here we present a heuristic that uses the algorithm for spending constraint utilities as a sub-routine.

5.5 A Heuristic

One of the major open problems is to find a polynomial time algorithm for piecewise linear and concave utilities. Here we present a heuristic for this case, which uses an algorithm for the spending constraint utilities as a subroutine.

Let f_{ij} be the piecewise-linear utility function of buyer i for good j and let g_{ij} be its derivative. Observe that g_{ij} is a decreasing step function. Observe that if the price of good j is known, say p_j , then the function $g_{ij}(x_{ij}p_j)$ gives the rate at which i derives utility per unit of j received as a function of the amount of money spent on j , which defines an instance of the spending constraint utilities. Now consider the following procedure. Start with an initial price vector so that the sum of prices of all goods adds up to the total money possessed by buyers. Using these prices, convert the given piecewise-linear utility functions into spending constraint utility functions and run the algorithm of [42] on this instance to obtain a new price vector. Repeat until the price vector does not change, i.e., a fixed point is obtained. It is easy to see that prices at a fixed point are equilibrium prices for the given piecewise-linear utility functions. An interesting open question is how fast does this procedure converge.

CHAPTER VI

EG MARKETS

6.1 *Definition*

We begin by recalling Kelly's capacity allocation market [36]. Given a network (directed or undirected) with edge capacities specified and a set of source-sink pairs, each with initial endowment of money specified, find equilibrium flow and edge-prices such that

- Only saturated edges can have positive prices.
- All flows are sent along a minimum cost path from source to sink.
- The money of each source-sink pair is fully spent.

As was the case with the Fisher market with linear utilities, the equilibrium flow and prices are given by a convex program (and the Lagrangian multipliers) that maximizes a similar objective function, subject to flow feasibility conditions. It is easy to see that the KKT conditions are equivalent to the equilibrium conditions.

$$\begin{aligned}
 & \text{maximize} && \sum_{i=1}^n m_i \log f_i \\
 & \text{subject to} && \forall i \in I, f_i = \sum_{P \in \mathcal{P}(s_i, t_i)} f_i(P), \\
 & && \forall e \in E, \sum_{P: e \in P} f_i(P) \leq c(e),
 \end{aligned}$$

Definition 48 EG Markets ([31]) *An EG Market \mathcal{M} with the set of buyers (agents) $[n]$ is such that the equilibrium utility allocation of an EG market is captured by the*

following convex program similar to the one considered by Eisenberg and Gale [21] for the Fisher market with linear utilities.

$$\begin{aligned}
& \text{maximize} && \sum_{i=1}^n m_i \log u_i \\
& \text{subject to} && \forall j \in J, \sum_{i \in [n]} a_{ij} u_i + \sum_{k \in K} a_{kj} t_k \leq b_j, \\
& && \forall i \in [n], k \in K, u_i, t_k \geq 0.
\end{aligned}$$

Also the constraints defining the set of feasible utilities should satisfy the following two conditions:

- Free disposal: if \mathbf{u} is feasible, then so is any other \mathbf{u}' dominated by \mathbf{u} .
- Utility Homogeneity: for all $j \in J$, if for some $i \in [n]$, $a_{ij} > 0$ then $b_j = 0$.

The auxiliary variables t_k might be used for instance, to give a more efficient representation of the feasible region, or as a means to provide semantics for the market. For example, in the Fisher model of a market where there are buyers and divisible goods, the auxiliary variables denote the amount of each good every buyer gets.

We emphasize that the notion of a *good* being sold or bought in traditional markets has been subsumed by the various constraints on the utilities of agents. This is useful because in many markets the notion of a good is not clear. For instance, in the capacity allocation market described in Section 1.2.3, the capacity of each edge raises a constraint on the maximum flow (which is the utility in this case) the agents can send across it. Since there are no goods in EG markets, each agent instead pays for the constraints influencing his utility. Thus, each constraint has a *price*. Interpreting the prices as Lagrangian variables and applying the KKT conditions, we get the following equivalent definition of an equilibrium allocation in EG markets.

Definition 49 A feasible utility \mathbf{u} is an equilibrium allocation if there exist witness $\mathbf{t} \in \mathbf{R}_+^K$ and prices $\mathbf{p} \in \mathbf{R}_+^J$ such that

- $\forall i \in [n], m_i = \text{rate}(i)u_i$, where $\text{rate}(i) = \sum_j a_{ij}p_j$.
- $\forall j \in J, p_j > 0 \Rightarrow \sum_{i \in [n]} a_{ij}u_i + \sum_{k \in K} a_{kj}t_k = b_j$.
- $\forall t \in K, t_k > 0 \Rightarrow \sum_{j \in J} a_{kj}p_j = 0$, and $\sum_{j \in J} a_{kj}p_j \geq 0$ otherwise.

In an equilibrium allocation, all money of each agent must be exhausted. This is captured by the first requirement above. Moreover, if a constraint is priced, then it must not be “under utilized” and the second requirement above implies this.

The third condition above arises due to the auxiliary variables. In concrete instances of markets, this condition normally translates to the premise that in equilibrium an agent chooses the *best* basket of goods. For example, in the Fisher market, the third condition would imply that each buyer buys goods of maximum “bang-per-buck”; in the capacity allocation market of Section 1.2.3, the condition corresponds to the fact that each agent chooses the cheapest source-sink path.

Remark: Since the equilibrium of an EG market is captured by a convex program, the equilibrium always exists (even if the constraints are not finite). Given proper separation oracles, the equilibrium could also be approximated to arbitrary small additive error via the ellipsoid method. Moreover, since the objective function above is strictly concave, the equilibrium is unique.

Another Example of an EG Market: The Network Coding Market.

We are given a directed graph $G = (V, E)$; E is the set of resources, with capacities $c : E \rightarrow \mathbf{R}_+$. The set V is partitioned into two sets, terminals and Steiner nodes, denoted T and R , respectively. A set $S \subseteq T$ is the set of sources with money m_v , $v \in S$ specified. Source v broadcasts messages to all terminals at rate r by picking a *generalized branching rooted at v* : a fractional subgraph of G specified via a function $b : E \rightarrow \mathbf{R}_+$ such that $b(e) \leq c(e)$ for all edges e and a flow of r units is possible in the subgraph from v to every terminal u . Generalized branchings rooted

at vertices of S , b_1, \dots, b_k are said to form a *feasible packing* for G if

$$\forall e \in E, b_1(e) + \dots + b_k(e) \leq c(e).$$

Edge e is said to be *saturated* if this inequality holds with equality. Given prices p_e for $e \in E$, the price of generalized branching b is defined to be $\sum_{e \in E} b(e)p_e$.

The network coding market asks for a feasible packing of generalized branchings and prices on edges such that

- The generalized branchings rooted at each source are cheapest possible.
- Only saturated edges have positive prices.
- The money of each source is fully used up.

Definition 50 We denote the class of EG markets with k buyers as $EG[k]$.

Definition 51 If a markets has rational equilibrium prices and allocations whenever the input parameters are all rational, then it is called a rational market.

Definition 52 The polytope of feasible utilities can be described by a linear program. If this linear program is combinatorial¹, then we call the EG market corresponding to it a combinatorial market.

6.2 Rationality of $EG[2]$ Markets

The main results of this section are that EG markets with 2 agents are rational.

Theorem 53 $EG[2]$ markets are rational.

Let the polytope of feasible utilities be

$$\mathbf{P} = \{\mathbf{x} : A\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq 0\},$$

¹An LP of the form $\max\{cx : Ax \leq b, x \geq 0\}$ is combinatorial if the entries in A have binary encoding length polynomial in the dimension of A .

with $u_1 = x_1$ and $u_2 = x_2$ being the utilities of agents 1 and 2 respectively and the rest being auxiliary variables. Let the projection of \mathbf{P} on (u_1, u_2) be

$$\mathcal{P}_u = \{(u_1, u_2) : u_2 \leq \beta_0, u_1 + \alpha_l u_2 \leq \beta_l, 1 \leq l \leq m, u_1 \leq \beta_{m+1}\}.$$

We may assume that we only consider facet inducing inequalities: for all $1 \leq l \leq m$, $u_1 + \alpha_l u_2 = \beta_l$ is a facet of \mathcal{P}_u . Call it facet l . Without loss of generality, assume that the α_l 's and β_l 's are strictly decreasing.

We assert that \mathcal{P}_u defines the same market as \mathbf{P} : when we price the constraints (facets) in \mathcal{P}_u , these prices can be used to get the prices for constraints of \mathbf{P} . Moreover if the prices of the facets are rational, then so are the prices of constraints in \mathbf{P} . For more details, refer Appendix B. Thus in the remaining of the chapter, we discuss methods of pricing the facets.

In the remaining of the section we show that no matter what the moneys of the two agents are, at most two facets need to be priced. Indeed these prices appear as variables in simultaneous linear equations and thus are rational.

Let the facets l and $l + 1$ intersect at the point (u_1^l, u_2^l) . Thus the endpoints of facet l are (u_1^{l-1}, u_2^{l-1}) and (u_1^l, u_2^l) . Associate subintervals of $[0, 1]$ to the facets as follows.

Definition 54

$$\forall 1 \leq l \leq m, I_l := \left[\frac{u_1^{l-1}}{\beta_l}, \frac{u_1^l}{\beta_l} \right], I_{l,l+1} := \left[\frac{u_1^l}{\beta_l}, \frac{u_1^l}{\beta_{l+1}} \right].$$

$$I_{0,1} := \left[0, 1 - \frac{\alpha_1 \beta_0}{\beta_1} \right].$$

The main idea is that if m_1, m_2 are the moneys of the two agents, then $\frac{m_1}{m_1 + m_2}$ falls in exactly one of the intervals I_l or $I_{l,l+1}$. In the first case, we price only the facet l , while in the second we price only the facets l and $l + 1$.

Lemma 55 *If $\frac{m_1}{m_1 + m_2} \in I_l, 1 \leq l \leq m$, then $p_l = \frac{m_1 + m_2}{\beta_l}$ (and 0 otherwise) is an equilibrium price.*

Proof: Its not too hard to check that the utilities $u_1^* := m_1/p_l$ and $u_2^* := m_2/(\alpha_l p_l)$ are equilibrium utilities and lie on facet l . \square

Lemma 56 *If $\frac{m_1}{m_1+m_2} \in I_{l,l+1}$, $1 \leq l \leq m$, then there exists an equilibrium price with only p_{l+1} and p_l having non-zero prices.*

Proof: The equilibrium utility allocation is (u_1^l, u_2^l) . We want p_l and p_{l+1} that satisfy the following two equations. $m_1 = u_1^l(p_l + p_{l+1})$, and $m_2 = u_2^l(\alpha_l p_l + \alpha_{l+1} p_{l+1})$. Note that this system of two equations in two unknowns has a unique solution since they are linearly independent:

$$p_l = \frac{u_1^l m_2 - \alpha_{l+1} u_2^l m_1}{u_1^l u_2^l (\alpha_l - \alpha_{l+1})}, p_{l+1} = \frac{\alpha_l u_2^l m_1 - u_1^l m_2}{u_1^l u_2^l (\alpha_l - \alpha_{l+1})}.$$

However the prices are positive exactly when $\frac{m_1}{m_2} \in \left[\frac{u_1^l}{\alpha_l u_2^l}, \frac{u_1^l}{\alpha_{l+1} u_2^l} \right]$, which happens when $\frac{m_1}{m_1+m_2}$ is in the interval $I_{l,l+1}$. \square

$I \leq I'$ means interval I ends where I' begins. $I < I'$ means interval I ends before I' begins. $I \leq x$ means interval I ends before or at x . $x \leq I$ means interval I starts after or at x . We note the following for future reference.

Observation 57

$$I_l \leq I_{l,l+1} \leq I_{l+1}.$$

Proof:(Proof of Theorem 53) Proof follows from noting that the intervals I_l , for $1 \leq l \leq m$, and $I_{l,l+1}$, for $0 \leq l \leq m$, cover the entire unit interval (Observation 57). Thus for any instance of moneys, the equilibrium prices are rational.

Note we did not say how to price the facet 0, which we need to do when $\frac{m_1}{m_1+m_2}$ falls in $I_{0,1}$. But by symmetry of choice between u_1 and u_2 it follows that we can price it accordingly. \square

6.3 Algorithms for combinatorial EG[2] markets

6.3.1 Binary Search Algorithm

In this section we give a binary search algorithm for finding equilibrium prices. We also give a strongly polynomial time algorithm for finding the equilibrium prices in EG[2] markets that are combinatorial. The algorithm takes as input, the moneys of the buyers, m_1 and m_2 , a description of the polytope \mathbf{P} , and two parameters, M and ϵ such that we are guaranteed that $M \geq \alpha_1$, and $\alpha_l - \alpha_{l+1} \geq 2\epsilon$ for all l .

We now describe the algorithm at a high level. The algorithm does a binary search on α . First, it finds the facets adjacent to α , say l and $l + 1$ such that $\alpha \in [\alpha_l, \alpha_{l+1}]$, and their endpoints. Now, it checks if the equilibrium can be attained by pricing these two facets, using Lemmas 55 and 56. If yes, the algorithm outputs those prices and halts. Otherwise, the monotonicity of the intervals (Observation 57) allows us to restrict our attention to a smaller range.

```

input  :  $m_1, m_2, \mathbf{P}, M, \epsilon$ .
 $U \leftarrow M$ ;
 $L \leftarrow 0$ ;
 $\rho \leftarrow \frac{m_1}{m_1 + m_2}$ ;
1 repeat
     $\alpha \leftarrow (U + L)/2$ ;
2 Find  $l$  such that  $\alpha \in [\alpha_l, \alpha_{l+1}]$ ;
3 Find the endpoints of the facets  $l$  and  $l + 1$  ;
    if  $\rho \in I_l \cup I_{l,l+1} \cup I_{l+1}$  then
        Assign prices to the facets  $l$  and  $l + 1$  as in Lemmas 55 and 56, and
        halt;
    else if  $\rho < I_l$  then
         $L \leftarrow \alpha_l$ ;
    else
         $U \leftarrow \alpha_{l+1}$ ;
    end
until  $U - L < \epsilon$ ;

```

Algorithm 5: The Binary Search Algorithm

The rest of the section describes how to implement Lines 2 and 3 in Algorithm 5.

Let the entries of the matrix A be $A_{(i,j)} = a_{ij}$. Recall that $\mathbf{P} = \{\mathbf{x} : A\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq 0\}$. Let \mathbf{c} be a vector such that $c_1 = 1, c_2 = \alpha$, and $c_i = 0$ otherwise. This is defined so that $\mathbf{c} \cdot \mathbf{x} = u_1 + \alpha u_2$. Let $\mathcal{L}(\alpha) = \max\{\mathbf{c} \cdot \mathbf{x} : \mathbf{x} \in \mathbf{P}\} = \min\{\mathbf{b} \cdot \mathbf{y} : \mathbf{y} \in \mathcal{D}\}$, where \mathcal{D} is the dual polytope $\{\mathbf{y} : A^T \mathbf{y} \geq \mathbf{c}, \mathbf{y} \geq 0\}$. In particular, $\mathcal{L}(0) = \max\{u_1 : \mathbf{x} \in \mathbf{P}\}$ and $\mathcal{L}(\infty) = \max\{u_2 : \mathbf{x} \in \mathbf{P}\}$.

Observe that $\beta_l = \mathcal{L}(\alpha_l)$ for all $0 \leq l \leq m+1$ if we define $\alpha_0 = \infty$ and $\alpha_{m+1} = 0$. Given any $\mathbf{x} \in \mathbf{P}$, define the polytope $\mathcal{Q}(\mathbf{x})$ as the set of all vectors (\mathbf{y}, α) that satisfy

$$\begin{aligned} \forall i, \quad \sum_j a_{ij} y_j &\leq c_i, & \forall j, y_j &\geq 0. \\ \forall i : x_i > 0, \quad \sum_j a_{ij} y_j &= c_i, \\ \forall j : \sum_i a_{ij} x_i &< b_j, y_j &= 0. \end{aligned}$$

Note that the first two constraints imply that $\mathbf{y} \in \mathcal{D}$. The last two constraints imply that \mathbf{x} and \mathbf{y} satisfy the complementary slackness conditions. However, in $\mathcal{Q}(\mathbf{x})$, α is treated as a variable. The algorithm to find the facets adjacent to any given α makes use of the following Lemmas 58 and 59.

Lemma 58 *Let x be any feasible extension of (u_1^l, u_2^l) , that is $x \in \mathbf{P}$, $x_1 = u_1^l$ and $x_2 = u_2^l$. Then $\alpha_l = \min\{\alpha : (y, \alpha) \in \mathcal{Q}(x)\}$, and $\alpha_{l+1} = \max\{\alpha : (y, \alpha) \in \mathcal{Q}(x)\}$.*

Lemma 59 $\mathcal{L}(\alpha) = u_1^l + \alpha u_2^l$ if and only if $\alpha \in [\alpha_l, \alpha_{l+1}]$.

Proof: Suppose $\alpha \in [\alpha_l, \alpha_{l+1}]$. By definition we have $u_1^l + \alpha_l u_2^l = \beta_l$ and $u_1^l + \alpha_{l+1} u_2^l = \beta_{l+1}$. Say $\alpha = \mu \alpha_l + (1 - \mu) \alpha_{l+1}$, for some $0 \leq \mu \leq 1$. Let $\beta = \mu \beta_l + (1 - \mu) \beta_{l+1}$. Thus, $u_1^l + \alpha u_2^l = \beta$.

We know that for all $(u_1, u_2) \in \mathcal{P}_u$, $u_1 + \alpha_l u_2 \leq \beta_l$ and $u_1 + \alpha_{l+1} u_2 \leq \beta_{l+1}$. By adding μ times the first equation to $1 - \mu$ times the second one, we get $u_1 + \alpha u_2 \leq \beta$. Hence $\beta \geq \mathcal{L}(\alpha) \geq u_1^l + \alpha u_2^l = \beta$.

Suppose $\alpha \in [\alpha_k, \alpha_{k+1}]$, for some $k \neq l$. Let (v_1^k, v_2^k) be the intersection of facets k and $k+1$. Then by the first part, $\mathcal{L}(\alpha) = v_1^k + \alpha v_2^k$. If $\mathcal{L}(\alpha) = u_1^l + \alpha u_2^l$, then there are two distinct points maximizing $\mathcal{L}(\alpha)$ and by Definition 54, we get that $u_1 + \alpha u_2 \leq \mathcal{L}(\alpha)$ itself is a facet. In that case, either $\alpha = \alpha_l = \alpha_{k+1}$ or $\alpha = \alpha_k = \alpha_{l+1}$ and we are done. \square

Lemma 60 *Let \mathbf{x} be any feasible extension of (u_1^l, u_2^l) , that is $\mathbf{x} \in \mathbf{P}$, $x_1 = u_1^l$ and $x_2 = u_2^l$. Then $(\mathbf{y}, \alpha) \in \mathcal{Q}(\mathbf{x})$ if and only if $\alpha \in [\alpha_l, \alpha_{l+1}]$.*

Proof: Suppose $(\mathbf{y}, \alpha) \in \mathcal{Q}(\mathbf{x})$. Then $\mathcal{L}(\alpha) \geq u_1^l + \alpha u_2^l = \sum_i c_i x_i = \sum_i x_i \sum_j a_{ij} y_j = \sum_j y_j \sum_i a_{ij} x_i = \sum_j y_j b_j \geq \mathcal{L}(\alpha)$. So by Lemma 59, $\alpha \in [\alpha_l, \alpha_{l+1}]$.

Suppose $\alpha \in [\alpha_l, \alpha_{l+1}]$. Then by Lemma 59, $\mathcal{L}(\alpha) = u_1^l + \alpha u_2^l$. So x is an optimal primal solution satisfying $Ax \geq b, x \geq 0$, and $cx = \mathcal{L}(\alpha)$. Consider an optimal dual solution y such that $A^T y \geq c, y \geq 0$ and $by = \mathcal{L}(\alpha)$. Apply complementary slackness conditions to x and y to conclude that $(\mathbf{y}, \alpha) \in \mathcal{Q}(\mathbf{x})$. \square

Lemma 58 is an immediate corollary of this lemma.

Now given α , one can find the facets adjacent to it, that is, l such that $\alpha \in [\alpha_l, \alpha_{l+1}]$. First find x that maximizes $cx = u_1 + \alpha u_2$ such that $x \in \mathbf{P}$. Then find $\alpha_l = \min\{\alpha : (\mathbf{y}, \alpha) \in \mathcal{Q}(\mathbf{x})\}$, and $\alpha_{l+1} = \max\{\alpha : (\mathbf{y}, \alpha) \in \mathcal{Q}(\mathbf{x})\}$. We now give a lemma that enables us to find the endpoints of a facet.

Lemma 61 $\mathcal{L}(\alpha_l + \epsilon) = u_1^{l-1} + (\alpha_l + \epsilon)u_2^{l-1}$ and $\mathcal{L}(\alpha_l - \epsilon) = u_1^l + (\alpha_l - \epsilon)u_2^l$.

Let T be the time required to optimize any linear objective function over the polytopes \mathbf{P} and $\mathcal{Q}(\mathbf{x})$. The following theorems characterize the running time and the correctness of the algorithm.

Theorem 62 *The running time of the algorithm is $O(T \log(\frac{M}{\epsilon}))$.*

Proof: The number of iterations of the repeat loop in Line 1 is bounded by $O(\log(\frac{M}{\epsilon}))$.

Line 2 can be done in $O(T)$ time as follows: first find x that maximizes $cx = u_1 + \alpha u_2$

such that $x \in \mathbf{P}$. Then find $\alpha_l = \min\{\alpha : (y, \alpha) \in \mathcal{Q}(x)\}$, and $\alpha_{l+1} = \max\{\alpha : (y, \alpha) \in \mathcal{Q}(x)\}$ (Lemma 58). Each of this takes time T . From Lemma 61, Line 3 can be done in $O(T)$ time too. Hence we are done. \square

Theorem 63 *The algorithm always outputs the equilibrium prices.*

Proof: Suppose $\frac{m_1}{m_1+m_2} \in I_k \cup I_{k-1,k}$. Then $L \leq \alpha_k \leq U$ throughout the algorithm. This is true initially, since $M \geq \alpha_k \geq 0$. Suppose this is true at the beginning of an iteration. If in the iteration, $\rho < I_l$, then from Observation 57, $\alpha_k < \alpha_l$. Similarly, if $\rho > I_{l+1}$, then $\alpha_k > \alpha_{l+1}$. Hence the assertion is true at the end of the iteration too.

Suppose that at the end of an iteration, $U - L < \epsilon$. Note that after each iteration, either both U and L have a value equal to one of the α_l 's, or one of them is 0 or M and the other has a value equal to an α_l . In either case, $U - L < \epsilon \Rightarrow U = L$, which should equal α_k by the first part. Hence we must have found the equilibrium prices in this iteration. \square

6.3.2 Combinatorial Markets

In this section, we show that for combinatorial EG[2] markets, the equilibrium price can be found in strongly polynomial time.

Theorem 64 *If an EG[2] market is combinatorial, then the equilibrium prices can be found in strongly polynomial time.*

Let $\nu(\cdot)$ denote the binary encoding length.

Lemma 65 $\forall l, \nu(\alpha_l) = \nu(A)^{O(1)}$. *That is, the size of the α_l 's is polynomially bounded in the size of the matrix entries..*

Proof: Note that $\mathcal{Q}(x)$ is described by the a_{ij} 's. Theorem follows from Lemma 58 and standard application of Cramer's rule. \square

Lemma 66 *One can find M and ϵ such that $\log\left(\frac{M}{\epsilon}\right) = \nu(A)^{O(1)}$.*

Proof: Let c be the constant in the $O(1)$ in Lemma 65. M can be chosen to be the largest integer with a binary encoding length $\nu(A)^c$. Clearly $\alpha_1 \leq M$. ϵ can then be chosen to be $1/(2M)$. α_l 's have their denominators at most M and hence $\alpha_l - \alpha_{l+1} \geq 1/M = 2\epsilon$. \square

Theorem 64 follows from this lemma and Theorem 62. As a corollary, we get that there is a strongly polynomial time algorithm for the capacity allocation market in directed graphs with two source-sink pairs and the network coding market in a directed network with two sources.

CHAPTER VII

CONCLUSION

In this thesis we have mostly covered combinatorial algorithms for computing market equilibrium. For a good survey of algorithmic results based on solving convex programs, we refer the reader to [9]. We mention here two promising directions for future research.

7.1 Piecewise-linear utilities

One of the major open problems is to find a polynomial time algorithm for piecewise linear and concave utilities. We mention several special cases that are still open.

1. The utility is linear upto some point and then does not increase any more.
2. There are a constant number of buyers.
3. There are a constant number of goods.
4. *Supply aware* buyers: Here, the utilities are quasi-linear, that is buyers value money, so their net utility is the utility they derive minus the price they pay. The supplies are fixed and are universal knowledge. This is a special case of Case 1.

In the other direction, one may try to prove that this case is PPAD-Hard.

7.2 Beyond Weak Gross Substitutibility

Utilities satisfying Weak Gross Substitutibility are easily the most interesting class of utilities for which efficient algorithms are known. Several algorithms [27, 10, 8] are known that compute algorithms for any utility that satisfies WGS. In fact, very few

algorithms are known (for example [7]) for utilities outside this class. An interesting direction is to define an appropriate notion of *approximate WGS* and find algorithms for these utilities.

APPENDIX A

KKT CONDITIONS

Consider a maximization problem such as

$$\begin{aligned} & \text{maximize} && f(x) \\ & \text{subject to} && g_i(x) \leq 0 \quad \forall i, \\ & && x_j \geq 0 \quad \forall j, \end{aligned}$$

where f is a concave function. A feasible solution to this program is an optimal solution, if and only if there exist (Lagrangian multipliers) $\lambda_i \geq 0$, $\forall i$ such that

1. $\sum_i \lambda_i \frac{\partial g_i}{\partial x_j} \geq \frac{\partial f}{\partial x_j}, \quad \forall j.$
2. $x_j > 0 \Rightarrow$ the above holds with equality.
3. $\lambda_i > 0 \Rightarrow g_i(x) = 0.$

These are called the KKT conditions. The above can be extended to programs in which some variable x_j is unconstrained (that is x_j need not be non-negative), in which case the corresponding constraint for j in 1 always holds with equality. Similarly, some constraint i in the program may be an equality, $g_i(x) = 0$, in which case the corresponding multiplier λ_i is unconstrained.

APPENDIX B

PROJECTION OF POLYTOPES

Suppose we eliminate the auxiliary variables \mathbf{t} from the equations to get an equivalent formulation for the feasible region of utilities as

$$\mathcal{P}_u = \left\{ \mathbf{u} : \forall l \in L, \sum_{i \in [n]} \alpha_{il} u_i \leq \beta_l \right\}.$$

This should define the same market as before. However, the prices now correspond to the new constraints, which correspond to the facets of \mathcal{P}_u , indexed by L . We show that given the prices on the facets in L , one can find prices for the original constraints in J that form an equilibrium. Suppose the equilibrium price of facet $l \in L$ is q_l . Let \mathbf{u} be the equilibrium utility, with \mathbf{t} being its witness. Then $rate(i) = \sum_l \alpha_{il} q_l$. At equilibrium, $m_i = rate(i) u_i$ and $q_l > 0 \Rightarrow \sum_i \alpha_{il} u_i = \beta_l$. Now consider the following LP:

$$\begin{aligned} & \text{maximize} && \sum_i \alpha_{il} u_i \\ & \text{subject to} && \forall j \in J, \sum_{i \in [n]} a_{ij} u_i + \sum_{k \in K} a_{kj} t_k \leq b_j. \\ & && \forall i \in [n], k \in K, u_i, t_k \geq 0. \end{aligned}$$

For any l with $q_l > 0$ the optimal value of this LPs has to be β_l . In fact, (\mathbf{u}, \mathbf{t}) is an optimal solution. For each such l , consider any optimal solution \mathbf{y}^l to the dual:

$$\begin{aligned} & \text{minimize} && \sum_j b_j y_j^l \\ & \text{subject to} && \forall i \in [n], \sum_j a_{ij} y_j^l \geq \alpha_{il}, \\ & && \forall k \in K, \sum_j a'_{kj} y_j^l \geq 0, \\ & && \forall j \in J, y_j^l \geq 0. \end{aligned}$$

(\mathbf{u}, \mathbf{t}) and \mathbf{y}^l satisfy the complementary slackness conditions for the above pair of primal-dual programs:

$$\begin{aligned} y_j^l > 0 & \Rightarrow \sum_{i \in [n]} a_{ij} u_i + \sum_{k \in K} a_{kj} t_k \leq b_j. \\ u_i > 0 & \Rightarrow \sum_j a_{ij} y_j^l = \alpha_{il}. \\ t_k > 0 & \Rightarrow \sum_j a'_{kj} y_j^l = 0. \end{aligned}$$

Let $p_j = \sum_l y_j^l q_l$. Using the feasibility and complementary slackness conditions above, one can show that p_j and (\mathbf{u}, \mathbf{t}) indeed satisfy the equilibrium conditions.

REFERENCES

- [1] ALIPRANTIS, C. D., BROWN, D. J., and BURKINSHAW, O., *Existence and Optimality of Competitive Equilibria*. Springer-Verlag, 1990.
- [2] ARROW, K., BLOCK, H., and HURWICZ, L., “On the stability of the competitive equilibrium: II,” *Econometrica*, vol. 27, no. 1, pp. 82–109, 1959.
- [3] ARROW, K. and DEBREU, G., “Existence of an equilibrium for a competitive economy,” *Econometrica*, vol. 22, pp. 265–290, 1954.
- [4] ARROW, K. and HURWICZ, L., “Weak gross substitutability and the existence of competitive equilibria,” *International Economic Review*, vol. 33, pp. 38–49, 1965.
- [5] BRAINARD, W. C. and SCARF, H. E., “How to compute equilibrium prices in 1891,” *Cowles Foundation Discussion Paper*, no. 1270, 2000.
- [6] CHAKRABARTY, D., DEVANUR, N. R., and VAZIRANI, V. V., “New results on rationality and strongly polynomial time solvability in eisenberg-gale markets,” in *Proceedings of the Second Workshop on Internet and Network Economics (WINE)*, 2006.
- [7] CODENOTTI, B., MCCUNE, B., PENUMATCHA, S., and VARADARAJAN, K., “Existence, multiplicity and computation of equilibria for CES exchange economies,” in *Proceedings of FSTTCS*, 2005.
- [8] CODENOTTI, B., MCCUNE, B., and VARADARAJAN, K., “Market equilibrium via the excess demand function,” in *Proceedings of ACM Symposium on Theory of Computing*, 2005.
- [9] CODENOTTI, B., PEMMARAJU, S., and VARADARAJAN, K., “Algorithms column: The computation of market equilibria,” in *ACM SIGACT News* 35(4), 2004.
- [10] CODENOTTI, B., PEMMARAJU, S., and VARADARAJAN, K., “Equilibrium for exchange markets satisfying WGS,” in *Proceedings of ACM Symposium on Discrete Algorithms*, 2005.
- [11] CODENOTTI, B. and VARADARAJAN, K., “Efficient computation of equilibrium prices for markets with Leontief utilities,” in *Proc. 31st International Colloquium on Automata, Languages, and Programming*, Lecture Notes in Computer Science, 2004.

- [12] DAHLHAUS, E., JOHNSON, D., PAPADIMITRIOU, C., SEYMOUR, P., and YANNAKAKIS, M., "The complexity of multiterminal cuts," *SIAM Journal on Computing*, vol. 23, pp. 864–894, 1994.
- [13] DEBREU, G., "Economic theory in a mathematical mode: the Nobel Lecture," 1984. AER.
- [14] DENG, X., PAPADIMITRIOU, C., and SAFRA, S., "On the complexity of equilibria," in *Proceedings of ACM Symposium on Theory of Computing*, 2002.
- [15] DEVANUR, N. R., PAPADIMITRIOU, C. H., SABERI, A., and VAZIRANI, V. V., "Market equilibrium via a primal-dual-type algorithm," in *Proceedings of IEEE Annual Symposium on Foundations of Computer Science*, 2002. Journal version available at <http://www-static.cc.gatech.edu/~nikhil/pubs/AD/market-full.ps>.
- [16] DEVANUR, N. R. and VAZIRANI, V., "An improved approximation scheme for computing Arrow-Debreu prices for the linear case," in *Proceedings of the 23rd Foundations of Software Technologies and Theoretical Computer Science, FSTTCS*, 2003.
- [17] DEVANUR, N. R. and VAZIRANI, V. V., "The spending constraint model for market equilibrium: Algorithmic, existence and uniqueness results," in *Proceedings of 36th STOC*, 2004.
- [18] EDMONDS, J., "Maximum matching and a polyhedron with 0,1-vertices," *Journal of Research of the National Bureau of Standards. Section B*, vol. 69, pp. 125–130, 1965.
- [19] EISENBERG, E. and GALE, D., "Consensus of subjective probabilities: the Pari-Mutuel method," *The Annals of Mathematical Statistics*, vol. 30, pp. 165–168, 1959.
- [20] EISENBERG, E., "Aggregation of utility functions," *Management Sciences*, vol. 7, pp. 337–350, 1961.
- [21] EISENBERG, E. and GALE, D., "Consensus of subjective probabilities: the Pari-Mutuel method," *The Annals of Mathematical Statistics*, vol. 30, pp. 165–168, 1959.
- [22] ELIAS, P., FEINSTEIN, A., and SHANNON, C. E., "Note on maximum flow through a network," *IRE Transactions on Information Theory IT-2*, pp. 117–119, 1956.
- [23] FORD, L. R. J. and FULKERSON, D. R., "Maximal flow through a network," *Canadian Journal of Mathematics*, pp. 99–404, 1956.
- [24] GARG, D., JAIN, K., TALWAR, K., and VAZIRANI, V., "A primal-dual algorithm for computing Fisher equilibrium in absence of gross substitutability property," in *Proceedings of the First Workshop on Internet and Network Economics (WINE)*, 2005.

- [25] GARG, N., VAZIRANI, V., and YANNAKAKIS, M., “Multiway cuts in directed and node weighted graphs,” in *Proc. 21st International Colloquium on Automata, Languages, and Programming*, vol. 820 of *Lecture Notes in Computer Science*, pp. 487–498, Springer-Verlag, Berlin, 1994.
- [26] GARG, R. and KAPOOR, S., “Auction algorithms for market equilibrium,” in *Proceedings of 36th STOC*, 2004.
- [27] GARG, R., KAPOOR, S., and VAZIRANI, V. V., “An auction-based market equilibrium algorithm for the separable gross substitutability case,” in *Proceedings, APPROX*, 2004.
- [28] HU, T., “Multicommodity network flows,” *Operations Research*, vol. 14, pp. 344–360, 1963.
- [29] JAIN, K., “A polynomial time algorithm for computing the Arrow-Debreu market equilibrium for linear utilities,” in *In Proc. of the IEEE Annual Symposium on Foundations of Computer Science FOCS*, 2004.
- [30] JAIN, K., MAHDIAN, M., and SABERI, A., “Approximating market equilibrium,” in *Proceedings of 6th APPROX*, 2003.
- [31] JAIN, K. and VAZIRANI, V. V., “Eisenberg-gale markets: Algorithms and structural properties.” Manuscript, 2006 available from <http://www.cc.gatech.edu/~vazirani/>.
- [32] JAIN, K. and VAZIRANI, V. V., “Eisenberg-gale markets: Combinatorial solvability, efficiency, fairness, and competition monotonicity.” Submitted, 2006.
- [33] JAIN, K., VAZIRANI, V. V., and YE, Y., “Market equilibrium for homothetic, quasi-concave utilities and economies of scale in production,” in *Proceedings, SODA*, 2005.
- [34] KAKADE, S. M., KEARNS, M., ORTIZ, L., PEMANTLE, R., and SURI, S., “The economic properties of social networks,” in *Proceedings of 19th Annual Conference on Neural Information Processing Systems (NIPS)*, 2005.
- [35] KEHOE, P. J. and KEHOE, T. J., “Capturing NAFTA’s impact with applied general equilibrium models,” *Federal Reserve Bank of Minneapolis Quarterly Review*, no. 18:2, 1994.
- [36] KELLY, F. P., “Charging and rate control for elastic traffic,” *European Transactions on Telecommunications*, vol. 8, pp. 33–37, 1997.
- [37] KUHN, H. W., “The Hungarian method for the assignment problem,” *Naval Research Logistics Quarterly*, vol. 2, pp. 83–97, 1955.
- [38] MEGIDDO, N., “Optimal flows in networks with multiple sources and sinks,” *Mathematical Programming*, pp. 97–107, 1974.

- [39] SCARF, H., *The Computation of Economic Equilibria (with collaboration of T. Hansen)*. New Haven: Yale University Press: Cowles Foundation Monograph No. 24., 1973.
- [40] SHOVEN, J. B. and WHALLEY, J., *Applying General Equilibrium*. Cambridge University Press, 1992.
- [41] TARDOS, E., “A strongly polynomial algorithm to solve combinatorial linear programs,” *Oper. Res.*, vol. 34, no. 2, pp. 250–256, 1986.
- [42] VAZIRANI, V. V., “spending constraint utilities, with applications to the ad-words market.” Submitted, 2006.
- [43] WALRAS, L., *Éléments d’économie politique pure ou théorie de la richesse sociale (Elements of Pure Economics, or the theory of social wealth)*. 1874. (1899, 4th ed.; 1926, rev ed., 1954, Engl. transl.).
- [44] WILLIAMSON, D., GOEMANS, M., MIHAIL, M., and VAZIRANI, V., “A primal–dual approximation algorithm for generalized Steiner network problems,” *Combinatorica*, vol. 15, pp. 435–454, 1995.
- [45] YE, Y., “Note on exchange market equilibria with leontief’s utility: Freedom of pricing leads to rationality,” in *Proceedings of the First Workshop on Internet and Network Economics (WINE)*, 2005.