Fall 2009

Algorithmic game theory and the computation of market equilibria

Benton John McCune
University of Iowa

ALGORITHMIC GAME THEORY AND THE COMPUTATION OF MARKET EQUILIBRIA

by

Benton John McCune

An Abstract

Of a thesis submitted in partial fulfillment of the
requirements for the Doctor of Philosophy
degree in Computer Science
in the Graduate College of
The University of Iowa

December 2009

Thesis Supervisor: Associate Professor Kasturi Varadarajan

ABSTRACT

It is demonstrated that for certain markets where traders have constant elasticity of substitution utility (CES) functions, the existence of a price equilibrium can be determined in polynomial time. It is also shown that for a certain range of elasticity of substitution where the CES market does not satisfy gross subsitutability that price equilibira can be computed in polynomial time. It is also shown that for markets satisfying gross substitutability, equilibria can be computed in polynomial time even if the excess demand is a correspondence. On the experimental side, equilibrium computation algorithms from computer science without running time guarantees are shown to be competitive with software packages used in applied microeconomics. Simulations also lend support to the Nash equilibrium solution concept by showing that agents employing heuristics in a restricted form of Texas Holdem converge to an approximate equilibrium. Monte Carlo simulations also indicate the long run preponderance of skill over chance in Holdem tournaments.

Abstract Approved: ______________________________
Thesis Supervisor

____
Title and Department

____
Date

ALGORITHMIC GAME THEORY AND THE COMPUTATION OF MARKET EQUILIBRIA

by

Benton John McCune

A thesis submitted in partial fulfillment of the
requirements for the Doctor of Philosophy
degree in Computer Science
in the Graduate College of
The University of Iowa

December 2009

Thesis Supervisor: Associate Professor Kasturi Varadarajan

Graduate College
The University of Iowa
Iowa City, Iowa

CERTIFICATE OF APPROVAL

PH.D. THESIS

This is to certify that the Ph.D. thesis of

Benton John McCune

has been approved by the Examining Committee for the thesis requirement for the Doctor of Philosophy degree in Computer Science at the December 2009 graduation.

Thesis committee: ______________________________
Kasturi Varadarajan, Thesis Supervisor

____
Samuel Burer

____
Sukumar Ghosh

____
Suely Oliveira

____
Sriram Pemmaraju

To my wife Jennifer, my mother Patricia, my father Daniel, and my sister Julie for their many years of love and support

ACKNOWLEDGEMENTS

Where the thesis falls short, the responsibility is my own. Where it succeeds, it owes a good deal to more people than I can name. I shall however try to name a few.

I would like to thank the members of my committee - Samuel Burer, Sukumar Ghosh, Suely Olieveira, Sriram Pemmaraju, and Kasturi Varadarajan. Their comments and questions were helpful in improving the thesis and their service over the years is appreciated. I would also like to acknowledge Jarkko Kari's service on my qualifying exam committee.

I would like to extend thanks my coauthors Bruno Codenotti, Sriram Pemmaraju, Sriram Penumachta, Rajiv Raman, and Kasturi Varadarajan.

I would like to thank Steve Bowers and Andrew Rinner at ITS for allowing me to acquire some practical experience in software development and participate in a worthwhile project. I would also like to thank Nedim, Vani, and Derek for helping to make my time there well spent. The programming experience acquired there no doubt helped improve my ability to write the code needed for the end of the thesis.

For making the Computer Science department a pleasant place to work, I'd like to thank my fellow students and friends. Particularly those who've moved on helping to provide motivation for me to the same! So many thanks to Juw Won, Imran, Rajiv, Sang-Cheol, Zhihong, and Shouxi. Thanks and best wishes to those who remain in the department - particularly Matt, Gaurav, Erik, and Saurav.

I would be remiss if I didn't acknowledge my many excellent teachers throughout more than a few years of schooling. I must single out a few. I would like to thank my physics teacher, Mr. Gerald Bucklin, for being a terrific teacher and introducing me to the power and excitement of ideas. I'd also like to thank Dr. Clifford Reiter for introducing me to the world of research. His ability to find problems that are both interesting and tractable has only grown more appreciated over the years.

I would like to thank Bruno Codenotti for providing me with an introduction to the field of algorithmic game theory. His impact on my work (not to mention that of other students) was out of all proportion to his length of time at the University of Iowa. His passion and learning and ability to contextualize research areas was both a resource and an inspiration.

Sriram Pemmaraju's lectures in both his courses and the Algorithms Reading Group were some of the finest I have attended - a model of clarity and rigor. His ceaseless interest in new problems always provided an example of what a scholar should be. And, to my mind no less important, one could always count on him for a smile and a kind word.

I am especially grateful to my advisor, Dr. Kasturi Varadajan. Having come to Iowa without having taken a course in computer science, it was Kasturi who first introduced me to the study of algorithms through his excellent course and sparked my first real interest in theoretical computer science. Kasturi was always generous with both time and ideas. This was critical as time after time, many days of confusion would be cleared upon his blackboard. This work would not have been possible without his kind advice, erudition, encouragement and, not least, his patience. And for that, I will remain grateful.

ABSTRACT

It is demonstrated that for certain markets where traders have constant elasticity of substitution utility (CES) functions, the existence of a price equilibrium can be determined in polynomial time. It is also shown that for a certain range of elasticity of substitution where the CES market does not satisfy gross subsitutability that price equilibira can be computed in polynomial time. It is also shown that for markets satisfying gross substitutability, equilibria can be computed in polynomial time even if the excess demand is a correspondence. On the experimental side, equilibrium computation algorithms from computer science without running time guarantees are shown to be competitive with software packages used in applied microeconomics. Simulations also lend support to the Nash equilibrium solution concept by showing that agents employing heuristics in a restricted form of Texas Holdem converge to an approximate equilibrium. Monte Carlo simulations also indicate the long run preponderance of skill over chance in Holdem tournaments.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

The last decade has seen a great deal of growth in the field that has come to be called algorithmic game theory. Computer scientists have made rich contributions to this field with work on mechanism design, the efficency of equilibria, as well as the computation of equilibria. Central to the field is the concept of equilibrium (Nash and correlated equilibria in games, price equilibria in markets) - the predicted outcome of independent rational agents pursuing their own self interest.

The thesis consists of five chapters besides this introduction. The main problem studied in the first three chapters is the computation of market equilibria. Market equilibria are examined both theoretically and experimentally. The final two chapters study the game of poker through simulations.

The study of market equilibria has been central to economic theory for over a century. Though long serving as a cornerstone in the foundation of microeconomic theory, it is only over the past few decades that economists have increasingly come to rely on general equilibrium models to model real world problems [59]. General Equilibrium analysis has been applied to areas such as domestic tax policy and international trade policy.

Efficient algorithms for computing market equilibria could be helpful when analyzing complex models with many variables. The problem of finding efficient algorithms for computing market equilibria has illicited a great deal of interest from computer scientists in recent years. In a short span of time, much progress has been made.

I will begin by providing the definitions and concepts needed to discuss the market equilibrium problem. Then, I proceed to discuss some of the important results from the economics literature. The work from mathematical economics has been used extensively in the computer science results of recent years. I then proceed to discuss some of the recent developments in computer science.

This brief review will largely restrict itself to a discussion of an *exchange economy*. The most important thing missing from this type of model is the production of goods. In the *exchange* setting, all goods are present in the market at the beginning and no goods are produced as outputs from or serve as inputs to a production process.

1.1 Market Equilibrium Definitions

We consider the exchange model in detail. We are given m economic agents or traders who trade in n divisible goods. Let R^n_+ be the subset of R^n where each vector has only nonnegative components. A vector $x = (x_1, x_2, \ldots x_n) \in R^n_+$ will represent a bundle of goods which consists, for each i, of x_i units of good i. Each trader will have a concave utility function $u_i : \mathbf{R^n_+} \rightarrow \mathbf{R}_+$ that induces a preference ordering on bundles of goods which are represented by vectors in $\mathbf{R^n_+}$. Traders enter the market with an initial endowment of goods represented by $w_i = (w_{i1}, \ldots, w_{in}) \in \mathbf{R^n_+}$. All traders then sell all their goods at the market price and buy the most favorable bundle of goods they can afford. If all traders do this and demand does not exceed supply, an equilibrium is said to exist.

1.1.1 Equilibrium

More formally, a price is represented by a vector $\pi = (\pi_1, \ldots, \pi_n) \in R^n_+$ with π_j signifying the price of the jth good. The bundle of goods purchased by the ith trader is given by $x_i = (x_{i1}, \ldots, x_{in}) \in \mathbf{R^n_+}$. A vector $\pi \in R^n_+$ and vectors $\bar{x}_i$, for each trader i, are said to an equilibrium price and allocation if the following conditions hold.

1) $\bar{x}_i$ is a solution to the following optimization problem, for which π is considered an input:

$$\max u_i(x) \tag{1.1}$$

$$\text{subject to } \pi \cdot x \leq \pi \cdot w_i$$

$$\text{and } x \in \mathbf{R^n_+}$$

A solution, $\bar{x}_i$, to this problem is called the demand of the ith trader.

2) Aggregate Demand does not exceed initial endowments:

$$\text{For each good j, we have } \sum_{i=1}^{m} \bar{x}_{ij} \leq \sum_{i=1}^{m} w_{ij}.$$

1.1.2 Market Example

We consider a simple market where there are two traders that trade in apples and oranges. Each trader has a linear utility function. Trader 1 has $u_1 = A + 2O$

where A represents the number of apples and O the number of oranges. Trader 2 has $u_2 = A + O$. Suppose Trader 1's initial endowment is (3,1) - she has three apples and one orange. Trader 2's initial endowment is (1,2).

I claim that $\pi = (2, 3)$ is an equilibrium price vector - apples are 2 dollars with oranges being 3 dollars. This can be easily verified. Trader 1 sells her bundle of goods for $3 \times 2 + 1 \times 3 = 9$ dollars. She then buys 3 oranges since that maximizes her "bang for the buck" because she gets twice as much utility from oranges as she does from apples and they only cost 50% more. Trader 2 sells her goods for $1 \times 2 + 2 \times 3 = 8$ dollars. Since she is indifferent between apples and oranges and oranges are more expensive, she spends all 8 dollars on 4 apples. The traders entered the market with a combined total of 4 apples and 3 oranges and leave with a combined 4 apples and 3 oranges. Thus, (2,3) must be an equilibrium price.

1.1.3 Demand and Excess Demand

For any price vector π, not necessarily an equilibrium price, we call any $x_i(\pi)$ that satisfies the conditions in optimization problem 1.1 a *demand* of trader i at price π. *Market or Aggregate Demand* of good j at price π is defined to be $X_j(\pi) = \sum_{i=1}^{m} x_{ij}$. We call $Z_j(\pi) = X_j(\pi) - \sum_{i=1}^{m} w_{ij}$ the *market excess demand* of good j at price π. The vectors $X(\pi) = (X_1(\pi), \ldots, X_n(\pi))$ and $Z(\pi) = (Z_1(\pi), \ldots, Z_n(\pi))$ are simply called *market demand* and *market excess demand.*

It should be noted that a trader's demand need not be unique. That is, the demand need not be a function, but could be a multi-valued correspondence. Think of the two trader linear example introduced above. If the price vector was (1,1), then trader 2 would be indifferent between any combination of apples and oranges that added up to her income.

These definitions are quite important as many properties and important results can be stated using excess demand. This allows one to state rather general results without explicit reference to a specific underlying utility function or preference ordering.

1.1.4 Walras' Law and Homogeneity

We say that a market satisifies *Walras' Law* if for every price π, $\pi \cdot Z(\pi) = 0$. If traders satisfy their budget constraints tightly, then Walras' Law follows easily. This will happen whenever preferences satisfy *local nonsatiation* (e.g. 3 apples are better than 2 apples), so in most reasonable markets, Walras' Law will hold.

Another property that is even more general is that the demand function is homogeneous of degree zero in price, that is, for all prices and all $k > 0$, $x_i(\pi) = x_i(k\pi)$. It seems reasonable that in an exchange market, it's the relative price of goods that's important. An obvious corollary is that any constant multiple of an equilibrium price is an equilibrium price.

Any result established for markets where these principles hold is considered extremely general.

1.1.5 Gross Substitutability

The property of *gross substitutability* (GS) is quite important to equilibrium theory and the computation of market equilibria. There are many reasonable markets that satisfies this property, but also many that do not. An excess demand function is said to satisfy gross subsitutability if for any two prices, π^1 and π^2 such that $0 < \pi_j^1 \leq \pi_j^2$, for all j, and $\pi_l^1 < \pi_l^2$ for some l, then for any good k where $\pi_k^1 = \pi_k^2$, $Z_k(\pi_1) < Z_k(\pi_2)$. If we can only guarantee that $Z_k(\pi_1) \leq Z_k(\pi_2)$ then we say that *weak gross substitutability* (WGS) is satisfied. This simply means that if you raise the price on one good, then the demand of the other goods will go down (or at least, not go up under WGS.)

It should be pointed out out that it can be easily shown that if each traders excess demand function satisfies weak gross substitutability, then the market excess demand satisfies weak gross substitutability.

1.1.6 Important Utility Functions

One widely used class of utility functions are the *constant elasticity of substitution* or CES utility functions. CES functions have the following form:

$$u(x_i) = (\sum_{j=1}^{n} (a_{ij}x_{ij})^{\rho})^{1/\rho}.$$

We restrict the coefficients so that $a_{ij} \geq 0$ and $\rho < 1$ and $\rho\ ne 0$. Thus, if we restrict ourselves to nonnegative allocations, we have the convenient property that for all x, $u(x) \geq u(0) = 0$. Also, the *elasticity of substitution* for goods in the market or σ can be defined in terms of theρ exponents. More precisely, $\sigma = \frac{1}{1-\rho}$. The family of CES functions include some significant special functions as limiting cases. As $\sigma \to \infty$, the utility functions become *linear*, that is $u_i(x) = \sum_{j=1}^{n} a_{ij}x_{ij}$. If $\sigma \to 1$, then the utility functions become *Cobb-Douglas*, identical to $u_i(x) = \prod_{j=1}^{n} x_{ij}^{a_{ij}}$. With Cobb-Douglas utilities, at any price whatsoever, a trader will always spend the same proportion of their income on a good. As $\sigma \to 0$, the CES utility function becomes the *Leontief* utility function which has the form $u_i(x) = \min_j a_{ij}x_{ij}$. These special cases of CES are important enough that even results for exchange markets restricted to traders with one of these utility functions have drawn considerable interest.

The CES functions are widely used in economists in part because of their power in modelling many different types of markets as illustrated by the discussion of their special cases. Another reason is that one can easily do an explicit calculation of demand and therefore of excess demand. This can be quite important in practice. An even more flexible utility function is the *nested CES*. A nested CES is a CES at the top level, but the x_{ij} portion may be replaced by another nested CES function. The nested CES are extremely powerful and it is also possible to efficiently compute demands for traders with nested CES utilities.

1.1.7 The Fisher Market

One special case of the exchange model is equivalent to *Fisher model*. In this case, we have proportional endowments, that is

$$\frac{w_{ij}}{\sum_{p=1}^{m} w_{pj}} = \frac{w_{ik}}{\sum_{p=1}^{m} w_{pk}}$$

for all traders i and all goods j, k. This means that traders' relative incomes are entirely independent of price. In the standard treatment of the Fisher model, traders are simply consumers endowed with a money income. This money is used to purchase a bundle of n goods just like in the exchange case, but a single seller has all the goods. An equilibrium occurs when each consumer buys the most preferred basket they can afford and the consumers don't demand more in the aggregate than the seller can provide.

1.1.8 Approximate Equilibria

Since equilibrium prices may be irrational, algorithms cannot compute exact equilibria. We therefore need precise definitions of approximate equilibria which can be computed. Roughly speaking, weak approximate equilibria occur when traders get bundles near their optimal utility whenever the traders come close to staying within their budget constraints.

More precisely, we say that a bundle $x_i \in \mathbf{R}_+^\mathbf{n}$ is a μ-approximate demand of trader i at price π if for $\mu \geq 1$ (this restriction on μ holds in all definitions that follow), if $u_i(x_i) \geq \frac{1}{\mu} u_i^*$ and $\pi \cdot x_i \leq \mu \pi \cdot w_i$ where u_i^* is the trader's optimal utility subject to the budget constraint.

Prices π and allocations x_i form a strong μ-approximate equlibrium if x_i is the demand of trader i at prices π and $\sum_{i=1}^m x_{ij} \leq \mu \sum_{i=1}^m w_{ij}$ for each good j.

Prices π and allocations x_i form a weak μ-approximate equlibrium if x_i is a μ-approximate demand of trader i at prices π and $\sum_{i=1}^m x_{ij} \leq \mu \sum_{i=1}^m w_{ij}$ for each good j.

We call an algorithm a polynomial time algorithm if it computes a $(1+\epsilon)$-approximate equilibrium for any $\epsilon > 0$ in time that is polynomial in the input parameters and $\log(\frac{1}{\epsilon})$.

An algorithm is called a polynomial time approximation scheme if it is only polynomial time in $\frac{1}{\epsilon}$ (rather than $\log(\frac{1}{\epsilon})$.)

1.2 Some Relevant History of Market Equilibria in Mathematical Economics

As mentioned earlier, computer scientists have only contributed to the field of market equilibrium computation in recent years. In this section, I discuss some of the important results to be found in the economics literature.

1.2.1 The Existence of Equilibrium

Leon Walras, largely ignored in his time, is now widely considered the father of general equilibrium theory and even the "greatest of all economists" by Schumpeter [55]. Walras formulated a market model in 1874 with his "Elements of Pure Economics" [98]. Walras even attempted a proof of existence and provided the first algorithm to compute this market equilibrium with his *tatonnement* price adjustment mechanism .

It would take a full eighty years before the existence of equilibrium in a relatively unrestricted neo-Walrasian setting with concave utility functions would be shown by Arrow and Debreu [6]. Arrow and Debreu built on the work of Wald who had shown existence in a more restricted setting [97]. (Incidentally, Wald also presented the first versions of gross substitutability and the weak axiom of revealed preferences. [55])

Existence of equilibira theorems have generally relied on fixed point theorems such as Brouwer's and Kakutani's. Arrow and Debreu utilized Nash's famous equilibrium result in game theory which in turn relied on a fixed point theorem from mathematics.

For the exchange market we've been considering, the proof of existence is now quite simplified and takes roughly a page in microeconomics textbooks [95]. Since we know that we have homogeneity of degree zero, we simply normalize prices so that they add up to one, that is $\sum_{j=1}^{n} \pi_j = 1$. Denote the $n-1$ dimensional simplex by S^{n-1} and we know that now $\pi \in S^{n-1}$. If $Z : S^{n-1} \to \mathbf{R^n}$ is simply a continuous function and satisfies Walras' Law then an equilibrium can be shown to exist by an application one of the most basic fixed point theorems, Brouwer's theorem.

If the excess demand satisfies gross substitutability then the normalized price equlibrium is unique [95]. This need not be the case in general, markets could contain many disconnected sets of price equilibria.

1.2.2 Tatonnement

As alluded to above, the first attempt at an algorithm to compute a Walrasian equilibrium was proposed by Walras. It was a simple price update procedure inspired by the Paris Bourse and Walras' economic intuition which he called tatonnement. There is an auctioneer who announces a price vector to the traders. The traders then compute their individual excess demands and report them to the auctioneer. The auctioneer then computes the aggregate excess demand by totalling the individual excess demands. The auctioneer then adjusts the price vector in the direction of the excess demand. That is, those goods for which demand is greater than what is present in the market have their prices increased, those goods with negative excess demand have their prices decreased. The new prices are then reported to the traders and the process repeats until $Z_j(\pi) \leq 0$, that is, until we are at an equilibrium. This accords nicely with intuitions about how the law of supply and demand would operate. Formalized, the update step would look like this:

$$\pi^{k+1} = \pi^k + f(Z(\pi^k))$$

where f is a sign preserving function.

Paul Samuelson formalized a continuous version of the tatonnement process that was important in later analysis [84]. Here the tatonnement process is governed by a system of differential equations:

$$\frac{d\pi_k}{dt} = G_k(Z(\pi)), k = 1, \ldots, n$$

where G_k is a continuous sign preserving function.

After the celebrated existence result was shown in 1954 by Arrow and Debreu, more work went into analyzing the stability of tatonnement [4]. Walras had hoped that his price adjustment mechanism would always converge to equilibria. Arrow, Block, and Hurwicz answered in the affirmative for the restricted case of markets that satisfy weak gross substitutability [4]. Unfortunately for the Walras conjecture, convergence could not be guaranteed in markets without weak gross substitutability. It was soon shown by Scarf with a simple example that tatonnement does not always converge [88]. Scarf had produced a mere three trader market where tatonnement continually oscillates if it does not start at equilibrium. Later, it would be shown by Sonnenschein, Mantel, and Debreu that Scarf's example was not remotely unique (for a recent treatment, see [18]).

1.2.3 Some other Computational Approaches for General Markets

Scarf's counterexample encouraged ongoing efforts towards finding alternative methods to compute equilibria. Scarf himself used the fixed point based existence theorems to produce techniques to approximate fixed points or in our case, equilibrium prices [87]. In these approaches, one follows a path through the decomposed price simplex and eventually reach equilibrium. Others such as Kuhn [62] and Eaves [40] expanded on the work of Scarf. The worst case running time is actually exponential in the number of goods though. There is reason to suspect that this line of research cannot result in a polynomial time algorithm [75].

Smale developed an alternative technique based on Newton's method with guaranteed convergence [90, 91]. This approach also has no polynomial worst case guarantees, but generally performs well in practice.

1.2.4 Approaches for more Specialized Markets

Curtis Eaves did work in a few of the markets mentioned earlier. Eaves was able to find a polynomial time algorithm for the Cobb-Douglas market. He was actually able to transform the computation of the market equilibrium into the relatively easy case of solving a linear system of equations for a nonnegative solution [38]. This will of course give a very nice polynomial bound on running time. Eaves also did work on the exchange market where traders have linear utility functions. Eaves formulated the market equilibrium problem as a linear complemenatarity problem [37]. Lemke's algorithm will then compute the solution to the linear complementarity problem, which then gives equilibrium prices and allocations. Lemke's algorithm does not have polynomial time worst case guarantees though.

When Arrow, Block and Hurwicz established their important results on the stability of tatonnement [4], they also proved a lemma that would turn out to be useful for future work. It is as follows:

Lemma 1.2.1 Separation Lemma *If a market satisfies positive homogeneity, gross substitutability and Walras' law and possesses an equilibrium price vector* π^* *satisfying* $\pi^*_j > 0$ *for each good j, then for all non-equilibrium price vectors satisfying* $\pi_j > 0$*, we have* $\pi^* \cdot Z(\pi) > 0$*.*

This lemma can be generalized to hold when only weak gross substitutability is present. This implies that the set of equilibrium prices is convex. This also provides a *separating hyperplane* that separates any non equilibrium price vector $\pi \in \mathbf{R}^{\mathbf{n}}_{+}$ from the convex set of equilibrium prices. We even have $\pi^* \cdot Z(\pi) > 0$ from the theorem and $\pi \cdot Z(\pi) = 0$ from Walras' law. We only need to be able to compute the individual excess demands $Z_j(\pi)$ in order to produce the separating hyperplane.

Polterovich, Spivak, Primak, Newman, and Nenakov expanded on the Separation Lemma (see [73, 74, 78, 79]). Nenakov and Primak rewrote the conditions for market equilibrium when traders have linear utilities as a convex feasibility problem with a finite number of constraints [73]. They also wrote a convex program for a linear utility model that included some production, but that program had an infinite number of inequalities. Nenakov and Primak also used the same approach for traders with Cobb-Douglas utility functions.

Gale used his joint work with Eisenberg [42] to show that the equilibrium for a linear utility Fisher market can be derived from the solution to a convex program [47]. This approach was then expanded by Eisenberg to cover homogeneous utility functions ($u(\alpha x) = \alpha u(x)$) [41]. If we let e_i signify the income of trader i and q_j represent the amount of good j present in the market, then this is the Eisenberg program:

$$\text{Maximize } \sum_{i=1}^{m} \log u_i(x_i)$$

$$\text{subject to } \sum_{i=1}^{m} x_{ij} \leq q_j \;\forall j, 1 \leq j \leq n$$

$$x_{ij} \geq 0 \;\forall i, j, 1 \leq i \leq m, 1 \leq j \leq n$$

The solutions x_i are equilibrium allocations from which one can obtain the equilibrium price vector.

1.3 Recent Computer Science approaches

In 2001, Christos Papadimitriou gave a lecture that initiated the recent computer science research on computing market equilibria [76]. In the first few years, this research proceeded unaware of some important results from the mathematical economics literature. In some cases, this resulted in reproduction of earlier results. More recently, work has utilized more concepts from economics and expanded upon earlier results. Computer scientists have not only turned economists' work on the structure of equilibria into polynomial time algorithms, but have also expanded our knowledge of the structure of equilibria.

Devanur, Papadimitriou, Saberi, and Vazirani produced a novel polynomial time algorithm for the special case of the Fisher Market where every buyer has a linear utility function [69]. They modelled their algorithm after Kuhn's primal-dual algorithm for bipartite matching.

Codenotti and Varadarajan [29] formulated a solution for Leontief utility Fisher Markets in terms of convex programming. This was done before computer scientists were made aware of Eisenberg's convex program discussed earlier [41]. In the linear case, solutions will always be rational if the parameters of the problem are. Codenotti and Varadarajan demonstrated that this is not the case for Leontief utility functions or homogeneous utility functions generally. If prices can be irrational, then an approximation algorithm is the best one can do.

Ye would go on to develop finely tuned interior point algorithms to solve convex programs for both Fisher and Arrow-Debreu linear utility models in the most efficient ways known [100]. Before Ye's paper, the best known time complexity for these problems were $O(n^8 log(1/\epsilon))$ using two different methods. Ye uses a modified primal-dual path following algorithm similar to those used in linear programming to achieve a new time complexity bound of $O(n^4 log(1/\epsilon))$.

With computer scientists now having absorbed more of the established theory from economics, much progress has been accomplished in expanding the regions where we have polynomial time algorithms.

Codenotti, Pemmaraju, and Varadarajan [23] were able to expand upon the Separation Lemma of Arrow, Block, Hurwicz [4] to compute a polynomial time algorithm for markets where the aggregate excess demand satisfies weak gross substitutability and you have the ability to efficiently compute an approximate demand. This result is nice because it includes many of the important special cases such as the linear, Cobb-Douglas, and CES functions with $\sigma \geq 1$. The framework can be used to generate polynomial time results without assuming anything about the precise form of the utility functions other than that the excess demand function will satisfy weak gross substitutability (and a few other weak assumptions). The approach utilized in their paper is extended in Chapter 2.

It has recently been shown that in general, the problem of computing an equilibrium is quite thorny. Codenotti, Saberi, Varadarajan and Ye [27] show that it is NP-hard to decide whether a Leontief exchange economy even has an equilibrium.

Chen and Deng [16] made a major breakthrough in algorithmic game theory when they demonstrated that finding a Nash equilibrium in a two player game is PPAD complete (for a precise definition of PPAD, see [31, 75].) When this result is combined with the result from [27] that reduces two-player games to a special type of Leontief Exchange economy, we see that it is PPAD-complete to compute an equilibrium for markets even when they are known to exist. Huang and Teng [57] show that the polynomial time computation of an approximate equilibrium is not possible unless $PPAD \subset P$.

Once it was shown that the continuous version of tatonnement converged for markets satisfying weak gross substitutability, a natural question for the computer scientist would be whether the discrete tatonnement could serve as an efficient polynomial time algorithm in those markets. Codenotti, Pemmaraju, and Varadarajan [23] had already presented a polynomial time algorithm for practically the entire range of markets with demand functions satisfying WGS, but the simplicity of Walras' 19^{th} century algorithm remained alluring and Codenotti, McCune, and Varadarajan's paper [20] was able to achieve a positive result on a discrete version of tatonnement.

It turns out that a simple discrete version of tatonnement does converge to an approximate equilibrium in polynomial time for markets satisfying WGS. Recalling that the aggregate excess demand may be irrational, the discrete version of tatonnement is of the form:

$$\pi_j^{k+1} = \pi_j^k + \beta Y_j^k,$$

where π_j^k denotes the price of good j at the k-th iteration, and Y_j^k is an approximation of market demand for good j at price vector π^k. β is a carefully chosen positive, rational parameter. The technique of market transformation used in the paper is also utilized in chapter 2 so we save a discussion of it until then.

There have been some very recent expansion of polynomial time computability results into settings that do not satisfy gross substituability. Devanur and Kannan [34] show that an equilibrium for a market where traders have piece-wise linear concave (PLC) utility functions and there are a constant number of goods can be computed in polynomial time. However, Chen et al. [15] show that the problem of computing an equilibrium for markets where traders have additively separable PLC utility function is PPAD-complete.

1.4 Thesis Contributions to the Computation of Market Equilibria

Chapter 2 extends polynomial time computability of market equilibria to markets satisfying gross substitutability where the demand is a correspondence. This result was previously established for markets where the demand is a function of the price [23]. The technical lynchpin for the algorithm is a strong separation lemma that allows the use of the ellipsoid method. This lemma stregthens the lemma from [77] in a way that is analagous to how [23] strengthens Lemma 1.2.1.

Devanur and Vazirani [35] haved introduced a spending constraint market model that is described in the chapter. The chapter also provides an exact polynomial time algorithm for the spending constraint market model. The chapter gives the widest, general framework for computing for markets satisfying gross subsitutability.

This work was published in [67].

Chapter 3 provides a polynomial time algorithm that decides existence of market equilibia in a CES economy. The constant elasticity of subsitution (CES) utility functions are quite flexible and widely used in economics. This existence problem is solved by checking the bi-connectivity of a digraph associated with the input of the market.

The chapter also presents Codenotti and Varadarajan's convex program for a CES economy that does not satisfy gross subsitutability. The chapter also shows that the ellipsoid method can be used to compute equilibria in polynomial time for a CES market that does not satisfy gross subsitutability. The overwhelming majority of efficient market equilibrium computation algorithms have been for markets satisfying gross subsitutability. Most of the results of this chapter were based on joint work with Bruno Codenotti, Sriram Penumachta, and Kasturi Varadarajan and appeared in [19].

Chapter 4 consists of an experimental study of recently developed algorithms to compute market equilibria and compares their performance against the commonly used packages from applied microeconomics. The experiments find that many markets for which there are no polynomial time guarantees, the recent algorithms from computer science still fare well. This chapter is based on joint work with Bruno Codenotti, Sriram Pemmaraju, Rajiv Raman, and Kasturi Varadarajan. This work appeared in [25, 26], with the final journal version in [24].

1.5 Game Theory and the Nash Equilibrium

Actors in game theoretic models are similar to actors in the exchange market setting. They have outcome preferences that we choose to model with utility functions. They are *rational* decision makers that then makes choices to maximize their utility. In the market setting, individual actors have no need to consider the preferences or endowments of other traders. Any influence those factors may hold are entirely mediated by the prices of the goods and the actor simply maximizes utility subject to their budget constraint. In noncooperative game theory, the actor must consider the interests and anticipate the actions of their opponents.

The Nash equilibrium has long been considered the primary solution concept in game theory. Formally, the Nash equilibrium is only a state of the game where no player has incentive to unilaterally deviate from her current strategy, but the Nash equilibrium has, at least implicitly, been considered a prediction of behavior as well. This aspect of the Nash equilibrium has recently been called into question by computer science.

Let us illustrate the concept of Nash equilibrium with the well known Prisoner's Dilemma. A pair of criminals are accused of jointly participating in a crime. They are interrogated separately and are offered incentives to cooperate by confessing and implicating their fellow prisoner. If both prisoners maintain silence, they will both receive a light sentence of 1 year on lesser charges. If they implicate each other, they both receive a moderate sentence of five years. If Prisoner A implicates Prisoner B, but Prisoner B remains silent then Prisoner A will go free while Prisoner B receives a harsh sentence of ten years. There is one Nash equilibrium and that is where both prisoners defect and implicate each other since regardlesss of what the other prisoner does, you are better off defecting, it is a dominant strategy. This is illustrated by Table 1.1. We assume each prisoner has a utility function of $u(x) = 10 - x$ with x being the number of years in prison. The ordered couplets in the table have the Prisoner A's utility followed by prisoner B's.

	Prisoner B Remains Silent	Prisoner B Defects
Prisoner A Remains Silent	9, 9	0, 10
Prisoner A Defects	10, 0	**5, 5**

Table 1.1: Prisoner's Dilemma Payoff Matrix

A *mixed strategy* is where a player may randomize their behavior and play a probability distribution over the action space. In the Prisoner's Dilemma example, there is a *pure* Nash equilibrium, but these don't exist for every game. Take for example, the Matching Pennies game. In the matching pennies game, each players chooses heads or tails. If both Players make the same choice then player A gets 1 point. If players make opposite choices, then player B receives one.

Not every game has a pure Nash equilibrium, but every game (subject to some very mild assumptions regarding actions and preferences) does have a mixed Nash equilibrium [70, 71]. In the Matching Pennies example, when each player plays Heads

	Head	Tail
Head	1, 0	0, 1
Tail	0, 1	1, 0

Table 1.2: Matching Pennies Payoff Matrix

or Tails with .5 probability, then we are at a Nash equilibrium. This is easy to see - let A equal the probability that player A plays heads, so A's mixed strategy is represented by $(A, 1-A)$. Let B be the probability that player B plays heads. The expected utility of player A is given by $E(u_A) = AB + (1-A)(1-B)$. The expected utility function of player B is given by $E(u_B) = A(1-B) + B(1-A)$. Suppose player B is playing (.5, .5) - that is, she is playing Heads and Tails with probability .5. Then we wish to determine what a best response by A would be. The expected value for player A is $E(u_A) = .5A + .5(1-A) = .5$. Thus, regardless of what player A does, she will have the same expected utility so $(.5, .5)$ is a best response for A. Similarly, if A is playing $(.5, 5)$, then B's expected utility is given by $E(u_B) = .5(1-B) + B(.5) = .5$. Thus, any response by B is a best response so she will experience no regret playing $(.5, .5)$. Since neither player would wish to deviate from playing the mixed strategy (.5, .5), it is a Nash equilibirum.

There has recently been a great deal work investigating the complexity of computing Nash equilibria. Christos Papadimitriou [68] articulates the view that a solution concept must not only be intuitively compelling, but also tractable computationally. Even though the concept of Nash equilibrium is not inherently a computational one, if rational utility maximizing agents are expected to arrive at the equilibrium, one might expect a computer could compute the equilibrium efficiently. In the words of Papadimitriou, “Efficient computability is an important modeling prerequisite for solution concepts.”

One could go even further than Papadimitriou since a solution could potentially be computed efficiently, but only with information unavailable to agents within the game. If a solution cannot be computed based on the information available to agents, it might be thought unlikely to be a valid prediction of behavior. In general, one problem with the concept of Nash equilibrium is that it is not clear how agents will arrive at it.

In 2001, Papadimitriou placed the computation of Nash Equilibrium as one of the central problems in theoretical computer science. It has been shown by Chen and Deng [16] that even in the two player case, the problem of computing a Nash equilibrium is PPAD-complete. Some problems [61] in PPAD(“Polynomial Parity Arguments on Directed Graphs”) [75], such as computing Brouwer fixed points and finding an n-player Nash equilibrium are widely thought to be intractable. Therefore, it is strongly suspected that $PPAD \neq P$. For a precise definition of PPAD, see [31,75].

The Nash equilibrium concept has been central to the burgeoning field of algorithmic game theory. In addition to the work on computation of equilibria, there has been a good deal of work in algorithmic mechanism design, particularly with respect to auctions. Another critical concept that has been examined in great detail, particularly in the area of selfish routing, is the *price of anarchy* - the ratio between the social utility of the optimal outcome and the worst Nash equilibrium. As can be seen from our Prisoner's dilemma example, the predicted outcome, the Nash equilibrium, can be worse for all players than another game outcome. For a review of the field, see [68].

1.6 Additional Thesis Contributions

Chapter 5 addresses the question of how independent agents might arrive at a Nash equilibrium, especially in a game when determining the equilibrium is nontrivial. If we are in a situation with "bounded rationality" where agents have no hope of initially computing and then employing the equilibrium strategy, can they adjust their strategy based on the experience of repeated play and eventually arrive at the equilibrium strategy? This question is even more relevant in a game of imperfect information where players do not have full information about the game even after it has been played.

Chapter 5 examines the behavior of individual agents using heuristic decision procedures and its relation to convergence to Nash equilibirum. This is done by examining certain situations in Texas Holdem poker. I conduct repeated simulations of short stacked two player no limit games with players using a variety of adaptive strategies. It is shown experimentally that player using relatively simple heuristics can converge to an approximate Nash equilibrium. This provides some support for the Nash equilibrium as a solution concept even though equilibria might be difficult to efficently compute in general.

Chapter 6 is less directly connected to the previously discussed literature. Chapter 6 uses Monte Carlo simulations to determine the relative contributions of skill and chance in tournament poker. There have been many arguments demonstrating the presence of skill in poker, but no studies attempting to quantify the relative contributions of skill and luck. Tournament poker is considered by most to have a greater element of luck than typical cash game poker.

The chapter conducts simulations of individual tournaments using a simple, yet realistic model of player skill. As players bankrolls grow and shrink when they win or lose, they choose tournaments with entry fees or buyins appropriately. The simulation of tournaments leads to a simulation of the emergent poker economy. It is then shown that a great deal of player profitability in the poker tournament economy can be explained by skill when a sufficient amount of time has passed and many tournaments have been played.

Players may different in their risk tolerance and bankroll management. This will affect the buyin at which they are willing to play a tournament at with a given bankroll. It is shown that even skilled players can sacrifice profitability if they take on too much risk through aggressive bankroll management - playing tournaments with buyins that are high relative to their bankrolls.

CHAPTER 2
EXTENDING POLYNOMIAL TIME COMPUTABILITY TO MARKETS WITH DEMAND CORRESPONDENCES

2.1 Introduction

This chapter[1] presents a polynomial time algorithm that computes an approximate equilibrium for any exchange economy with a demand correspondence satisfying gross substitutability. Such a result was previously known only for the case where the demand is a function, that is, at any price, there is only one demand vector. The case of multi valued demands that is dealt with here arises in many settings, notably when the traders have linear utilities.

We also show that exchange markets in the spending constraint model have demand correspodences satisfying gross substitutability and that they always have an equilibrium price vector with rational numbers. As a consequence the framework considered here leads to the first exact polynomial time algorithm for this model.

In order to outline the contributions of the chapter to the computation of market equilibria, I will begin by providing the necessary definitions and concepts needed to discuss the market equilibrium problem in the context of demand correspondeces. For the basic market equilibrium definitions, see the thesis introduction.

[1]This work appeared in [67]

2.2 Definitions

We consider the exchange model in detail. We are given m economic agents or traders who trade in n goods. Let R^n_+ be the subset of R^n where each vector has only nonnegative components. Each trader will have a concave, typically continuous, utility function $u_i : \mathbf{R^n_+} \rightarrow \mathbf{R}_+$ that induces a preference ordering on bundles of goods which are represented by vectors in $\mathbf{R^n_+}$. Traders enter the market with an initial endowment of goods represented by $w_i = (w_{i1}, \ldots, w_{in}) \in \mathbf{R^n_+}$. All traders then sell all their goods at the market price and buy the most favorable bundle of goods they can afford. If all traders do this and demand does not exceed supply, we are at an equilibrium.

2.2.1 Demand and Excess Demand

For any price vector π, not necessarily an equilibrium price, we call an $x_i(\pi)$ that satisfies (1.1) a *demand* of trader i at price π. *Market or Aggregate Demand* of good j at price π is defined to be $X_j(\pi) = \sum_{i=1}^m x_{ij}$. We call $Z_j(\pi) = X_j(\pi) - \sum_{i=1}^m w_{ij}$ the *market excess demand* of good j at price π. The collections $\mathcal{X}(\pi) = \{X(\pi)|X(\pi) = (X_1(\pi), \ldots, X_n(\pi))\}$ and $\mathcal{Z}(\pi) = \{Z(\pi) = Z(\pi) = (Z_1(\pi), \ldots, Z_n(\pi))\}$ are simply called *market demand* and *market excess demand.* Note that $\mathcal{X}$ and $\mathcal{Z}$ are both mappings from $\mathbf{R^n_+}$ to $2^{\mathbf{R^n}}$.

We can now simply express what it means for a price π to be an equilibrium for a market with excess demand $\mathcal{Z}$. π is an equilibrium if there exist $z \in \mathcal{Z}(\pi)$ such that $z \leq 0$.

2.2.2 Gross Substitutability Correspondences

The property of gross substitutability has an important effect on the structure of price equilibria and the possibility of computing market equilibria. Roughly speaking, a market possesses the gross substitutability property if when the prices on one set of goods are raised, demand does not decrease for the other goods. A formal definition is provided below.

Following Polterovich and Spivak [77], we define gross substitutability (GS) correspondences. Let π^1 and π^2 be price vectors for a market with n goods. We denote $I(\pi^1, \pi^2) = \{i|\pi_i^1 = \pi_i^2\}$.

We say that gross substitutability prevails for $\mathcal{Z}$, or $\mathcal{Z}$ is a GS correspondence, if for all π^1, π^2 such that $\pi^1 \leq \pi^2$ and $I(\pi^1, \pi^2) \neq \emptyset$, and for any $z \in \mathcal{Z}(\pi^1), y \in \mathcal{Z}(\pi^2)$, the following relation holds

$$min_{i \in I(\pi^1,\pi^2)}(z_i - y_i) \leq 0.$$

That is, at least one good that has its price unchanged does not have its demand decreased. This is an extremely mild definition for gross subsitutability. Other definitions that one might come across (typically in cases where the demand is assumed to be a function) are stronger and imply this one.

2.2.3 Homogeneity and Walras' Law

A market is said to satisfy *positive homogeneity* if for any π and any $\lambda > 0$, $\mathcal{Z}(\pi) = \mathcal{Z}(\lambda\pi)$. Walras' law states that for any price π and any $z \in \mathcal{Z}(\pi)$, $\pi \cdot z = 0$.

These will typically hold under very mild assumptions such as when the the utility function $u(x)$ has the property that for any $x \in \mathbf{R}_+^\mathbf{n}$, there exists a $y \in \mathbf{R}_+^\mathbf{n}$ such that $u(y) > u(x)$.

2.2.4 Approximate Equilibria

When equilibrium prices are irrational, algorithms cannot compute exact equilibria. We therefore need precise definitions of approximate equilibria which can be computed. Roughly speaking, weak approximate equilibria occur when traders get bundles near their optimal utility whenever the traders come close to staying within their budget constraints.

More precisely, we say that a bundle $x_i \in \mathbf{R}_+^\mathbf{n}$ is a μ-approximate demand of trader i at price π if for $\mu \geq 1$ (this restriction on μ holds in all definitions that follow), if $u_i(x_i) \geq \frac{1}{\mu} u_i^*$ and $\pi \cdot x_i \leq \mu \pi \cdot w_i$ where u_i^* is the trader's optimal utility subject to the budget constraint.

Prices π and allocations x form a weak μ-approximate equlibrium if x_i is a μ-approximate demand of trader i at prices π and $\sum_{i=1}^m x_{ij} \leq \mu \sum_{i=1}^m w_{ij}$ for each good j. A price π is considered a weak μ-approximate equlibrium price if if there exists x such that π and x form a weak μ-approximate equlibrium.

We call an algorithm a polynomial time algorithm if it computes a $(1 + \epsilon)$-approximate equilibrium for any $\epsilon > 0$ in time that is polynomial in the input parameters and $\log(\frac{1}{\epsilon})$.

2.3 Results

In this paper, it is shown via a strong separation lemma that when an excess demand correspondence satisfies gross substitutability, a weak approximate equilibrium can be computed in polynomial time using the ellipsoid method. This had been previously established only when the demand was single-valued [23]. The exchange market where traders have linear utilities is the most prominent market where the demand need not be single-valued. Previously, this linear utilities market had to be treated as a special case [58], but in the framework provided by this paper it is solved naturally as merely one case of a market with a demand correspondence satisfying gross substitutability.

The Spending Constraint Model of Vazirani and Devanur [35] is introduced and it is shown that the demand in this model is a GS correspondence. This gives a prominent example of a market that did not naturally fit into any other general framework. It is also shown that price equilibria for the spending constraint model are rational and can be computed exactly in polynomial time.

2.3.1 Related Work

Codenotti, Pemmaraju, and Varadarajan [23] were able to expand upon the Separation Lemma of Arrow, Block, Hurwicz [4] to compute a polynomial time algorithm for markets where the aggregate excess demand function satisfies weak gross subsitutability and you have the ability to efficiently compute an approximate demand. This result thus includes many of the important special cases such as the

Cobb-Douglas, and CES functions with elasticity $\sigma \geq 1$. The framework can be used to generate polynomial time algorithms without assuming anything about the precise form of the utility functions other than that the excess demand function will satisfy weak gross substitutability (and a few other weak assumptions). Codenotti, McCune, and Varadarajan [20] show that a simple Walrasian price adjustment technique can compute approximate equilibrium prices if the demand function satisfies Weak Gross Substitutability. Codenotti et al. developed a convex program for an exchange market for a range of CES utility functions including some that do not satisfy gross subsitutability [19].

2.3.2 Preliminaries

Some more extensive definitions and basic lemmas are needed in order to proceed to the main results of the paper.

2.3.2.1 Correspondences

Following Polterovich and Spivak [77], there are some mild, elementary assumptions regarding the excess demand correspondences in this paper. They are as follows.

- The correspondence is defined on $\mathbf{R}^n_{++}$ (each component is positive), convex-valued, closed and such that every compact set lying in $\mathbf{R}^n_{++}$has a non-empty bounded image in $\mathbf{R}^n$.

- The correspondence is positive homogeneous.

- The correspondence satisfies Walras' law. That is, for all $z \in \mathcal{Z}(\pi)$ and all π, $\pi \cdot z = 0$.

It should be noted that these assumptions are quite natural and extremely mild.

Some more notation is necessary. For vectors π^1 and π^2, let $max(\pi^1, \pi^2)$ be a vector where each component is the maximum of the corresponding component of π^1 and π^2. Similarly, we define $min(\pi^1, \pi^2)$.

We also denote

$$H_1(\pi^1, \pi^2) = \{i|\pi_i^1 < \pi_i^2\}, H_2(\pi^1, \pi^2) = \{i|\pi_i^1 \geq \pi_i^2\}$$

For any $a \in \mathbf{R^n}$ and $H \subset N = \{1, 2, \ldots, n\}$, we define $a[H]$ as a vector with compononents

$$a_i[H] = a_i, i \in H,$$

$$a_i[H] = 0, i \notin H.$$

2.3.2.2 Combining Lemma

Polterovich and Spivak [77] prove a useful "combining" lemma that we restate here.

Lemma 2.3.1 *Let a GS correspondence $\mathcal{Z}$, $x \in \mathcal{Z}(\pi^1), y \in \mathcal{Z}(\pi^2)$ be given. $\bar{\pi} = min(\pi^1, \pi^2)$, and $\bar{\bar{\pi}} = max(\pi^1, \pi^2)$. Then, there exist vectors $\bar{a} \in \mathcal{D}(\bar{\pi})$ and $\bar{\bar{a}} \in \mathcal{D}(\bar{\bar{\pi}})$*

such that

$$\bar{a} \leq x[H_1] + y[H_2] \text{ and } \bar{\bar{a}} \geq x[H_2] + y[H_1] \text{ where}$$

$$H_1 = H_1(\pi^1, \pi^2) \text{ and } H_2 = H_2(\pi^1, \pi^2).$$

2.3.2.3 Polterovich-Spivak Separation Lemma

Polterovich and Spivak [77] prove a separation lemma for correspondences (that satisfy the assumptions listed above) that generalizes the important lemma from Arrow, Block and Hurwicz [4]. The lemma is as follows

Lemma 2.3.2 *Let $\mathcal{Z}$ be a GS correspondence. If $\hat{\pi}$ is an equilibrium, π a price, and $z \in \mathcal{Z}(\pi)$, then $\hat{\pi} \cdot z \geq 0$. If, moreover, π is not an equilibrium price, then $\hat{\pi} \cdot z > 0$.*

2.3.2.4 Demand Oracle

We say that an exchange market M is equipped with a *demand oracle* if there is an algorithm that takes a rational price vector π and returns a vector $Y \in \mathbf{Q^n}$ such that there is a $Z(\pi) \in \mathcal{Z}(\pi)$, with $|Y_j - Z_j(\pi)| \leq \sigma$ for all j. The algorithm is required to run in polynomial time in the input size and $log(1/\sigma)$.

2.3.2.5 A Market Transformation

Let M be an exchange market with m traders and n goods. We then transform market M into market $\hat{M}$ by adding a phantom trader that will give us an equilibrium price vector with a reasonably bounded price ratio. Let $0 < \eta \leq 1$ be a parameter.

For each trader i, the new utility functions and initial endowments are the same as in M' except that there is one additional trader $m+1$. We set $\hat{w}_{m+1} = (\eta W_1, \ldots, \eta W_n)$ for the initial endowment while the trader's utility function is the Cobb-Douglas function $\hat{u}_{m+1}(x) = \prod_{j=1}^{n} x_j^{1/n}$. This trader will spend $1/n$-th of her budget on each good. Notice that the total amount of each good j in the market $\hat{M}$ is now $\hat{W}_j = \sum_{i=1}^{m+1} \hat{w}_{ij} = W_j(1+\eta)$.

The following lemma contains various useful results:

Lemma 2.3.3

- *The market $\hat{M}$ has an equilibrium.*
- *Every equilibrium π of $\hat{M}$ satisfies the condition $\frac{max_j \pi_j}{min_j \pi_j} \leq 2^L$, where L is bounded by a polynomial in the input size of M and $log(\frac{1}{\epsilon})$.*
- *For any $\mu \geq 1$, a weak μ-approx equilibrium for $\hat{M}$ is a a weak $\mu(1+\eta)$-approx equilibrium for M.*
- *$\hat{M}$ has a demand oracle if M does.*
- *Let π and π' be two sets of prices in $\mathbf{R}_+^n$ such that $|\pi_j - \pi_j'| \leq \epsilon \cdot min\{\pi_j, \pi_j'\}$ for each j, where $\epsilon > 0$. Let x_i be a $(1+\delta)$-approximate demand for trader i at prices π. Then x_i is a $(1+\epsilon)^2(1+\delta)$-approximate demand for trader i at prices π'.*

The ratios of largest price to smallest price must be bounded and we define some regions where this is the case. We define the region $\Delta = \{\pi \in \mathbf{R}_+^n | 2^{-L} \leq \pi_j \leq 1\}$. Here, L is given by the second item in lemma 2.3.3 and bounded by a polynomial in the input size of M and $log(1/\epsilon)$. We note that a normalized equilibrium price for $\hat{M}$ lies in Δ. Also, $\Delta^+ = \{\pi \in \mathbf{R}^n_+ | 2^{-L} - \frac{2^{-L}}{2} \leq \pi_j \leq 1 + \frac{2^{-L}}{2}\}$.

2.3.3 Strong Separation Lemma for Correspondences

In this section, we present a strong separation lemma for correspondences. This lemma strengthens Theorem 3 from [77] in a way that is similar to how the separation lemma 3.2 in [23] strengthens the celebrated lemma from [4]. Once this strong separation lemma is established, the ellipsoid method will be able to produce an approximate equilibrium in polynomial time whenever the demand is a GS correspondence.

First, we prove the following straightforward lemma:

Lemma 2.3.4 *If M is an exchange market with an excess demand $\mathcal{Z}$ that is a GS correspondence and we let $\hat{M}$ with excess demand $\mathcal{Z}'$ be the market M with the special Cobb-Douglas trader added then $\mathcal{Z}'$ is also a GS correspondence.*

Proof:

Let π^1, π^2 such that $\pi^1 \leq \pi^2$ and $I(\pi^1, \pi^2) \neq \emptyset$ be given. Let $g' \in \mathcal{Z}'(\pi^1)$ and $z' \in \mathcal{Z}'(\pi^2)$ be given. Note that $g' = g + z^{CD}(\pi^1)$ and $z' = z + z^{CD}(\pi^2)$ with $g \in \mathcal{Z}(\pi^1)$, $z \in \mathcal{Z}(\pi^2)$ and z^{CD} representing the excess demand function of the special Cobb-Douglas trader.

Since $\mathcal{Z}$ is a GS correspondence, $min_{j \in I(\pi^1,\pi^2)}(g_j - z_j) \leq 0$. Thus, we can choose $j \in I(\pi^1, \pi^2)$ such that $g_j \leq z_j$. Fix this j. Clearly, by GS of Cobb-Douglas utilities, $z_j^{CD}(\pi^1) \leq z_j^{CD}(\pi^2)$. Therefore,$g'_j \leq z'_j$ and $min_{i \in I(\pi^1,\pi^2)}(g'_i - z'_i) \leq 0$. Thus,

$\mathcal{Z}'$ is a GS correspondence.

□

This lemma will let us now prove the main lemma for the paper.

Lemma 2.3.5 *Let M be an exchange market with an excess demand $\mathcal{Z}$ that is a GS correspondence and let $\hat{M}$ with excess demand $\mathcal{Z}'$ be the market M with the special Cobb-Douglas trader added. If $\hat{\pi}$ is an equilibrium for $\hat{M}$ and $\hat{\pi} \in \Delta$, $z' \in \mathcal{Z}'(\pi)$, $\pi \in \Delta^+$, and π is not a $(1+\epsilon)$-approximate equilibrium price for $\hat{M}$. then $\hat{\pi} \cdot z' \geq \delta$ where $\delta \geq 2^{-E}$ and E is bounded by a polynomial in the input size of $\hat{M}$ and $log(\frac{1}{\epsilon})$.*

Proof:

Let $\Omega(\pi, \hat{\pi}) = \{N_1, \ldots, N_t, \ldots, N_l\}$ where $N_t = \{i|\pi_i/\hat{\pi}_i = \gamma_t\}$, with $\gamma_t > \gamma_{t+1}$. Let $\pi^k = min\{\pi, \gamma_k\hat{\pi}\}$ and $H^k = \{i|\pi_i < \gamma_k\hat{\pi}\} = \cup_{t<k}N_t$.

By homogeneity, $0 \in \mathcal{Z}'(\gamma_k\hat{\pi})$. By the previous lemma, $\mathcal{Z}'$ is a GS correspondence. Then, by the Poltervitch-Spivak combining lemma, there exist vectors g'^k such that $g'^k \in \mathcal{Z}'(\pi^k)$, and $g'^k \leq z'[H^k \cup N_k]$.

Note that $g'^k = g^k + z^{CD}(\pi^k)$ where $g^k \in \mathcal{Z}(\pi^k)$ and z^{CD} represents the excess demand of the special Cobb-Douglas trader. Also, $z' = z + z_{CD}(\pi)$ where $z \in \mathcal{Z}(\pi)$.

Also note that $I(\pi^k, \pi) = H^k \cup N_k$. Since $\mathcal{Z}$ is a GS correspondence, $min_{j \in I(\pi^k,\pi)}(g_j^k - z_j) \leq 0$. Thus, we can choose $j(k) \in H^k \cup N_k$ such that $g_{j(k)}^k \leq z_{j(k)}$.

We claim that for some $k \geq 2$ and some j', $\pi_{j'}^{k-1} - \pi_{j'}^k \geq \frac{\epsilon\pi_{j'}^k}{3n}$. Note that this immediately implies $\pi_{j'} - \pi_{j'}^k \geq \frac{\epsilon\pi_{j'}^k}{3n}$. If this weren't the case, then by lemma 2.3.3, z' would be a $(1+\epsilon)$-approximate demand at $\hat{\pi}$, which would make π a weak $(1+\epsilon)$-approximate equilibrium. Fix both k and j'.

The income of the Cobb-Douglas trader at π is given by $\pi \cdot w_{CD}$. Therefore:

$$\pi \cdot w_{CD} - \pi^k \cdot w_{CD} \geq \pi_{j'}\ w_{CD,j'} - \pi^k_{j'} \cdot w_{CD,j'} = (\pi_{j'} - \pi^k_{j'})w_{CD,j'} \geq \frac{\epsilon \pi^k_{j'} w_{CD,j'}}{3n}$$

This trader spends an equal amount of her income on each good. Note that for any good $j \in N^k$, $\pi_j = \gamma_k \hat{\pi}_j = \pi^k_j$. Also note that for any good $j \in H^k$, $\pi_j = \pi^k$. Thus for $j \in \cup_{t \leq k} N_t$,

$$z_{CD,j}(\pi) - z_{CD,j}(\pi^k) = x_{CD,j}(\pi) - x'_{CD,j}(\pi^k) = (\pi \cdot w_{CD})/(n\pi_j) - (\pi^k \cdot w_{CD})/(n\pi^k_j) =$$

$$\frac{\pi \cdot w_{CD} - \pi^k \cdot w_{CD}}{n\pi_j} \geq \frac{\pi^k_{j'} \epsilon w_{CD,j}}{3\pi_j n^2}.$$

Note that this holds for $j(k)$.

We can therefore conclude that $z'_{j(k)} - g'^k_{j(k)} \geq \frac{\pi^k_{j'} \epsilon w_{CD,j}}{3\pi_{j(k)} n^2}$. We then define $\delta' = \frac{\pi^k_{j'} \epsilon w_{CD,j}}{3n^2}$.

Since $\pi^j[\cup_{t \geq j} N_t] = \pi[\cup_{t \geq j} N_t]$, Walras Law and the inequality $g'^j \leq z'[H^j \cup N_j]$ implies

$$0 = \pi^j g'^j \leq \pi z'[\cup_{t \geq j} N_t].$$

Let $\beta_t = \pi z'[N_t]$ which gives us

$$\sum_{t \geq j} \beta_t \geq 0, j = 1, 2, \ldots l.$$

Thus,

$$0 \le \gamma_1^{-1} \sum_{t=1}^{l} \beta_t \le \gamma_1^{-1}\beta_1 + \gamma_2^{-1} \sum_{t=2}^{l} \beta_t \le \ldots < \sum_{t=1}^{l} \gamma_t^{-1}\beta_t = \hat{\pi} z'.$$

The last sum can also be written as $\gamma_1^{-1} \sum_{t=1}^{l} \beta_t + (\gamma_2^{-1} - \gamma_1^{-1}) \sum_{t=2}^{l} \beta_t + \ldots + (\gamma_l^{-1} - \gamma_{l-1}^{-1})\beta_l$. Notice that each term in this sum is nonnegative.

We know that $j(k) \in N^k$ or $j(k) \in H^k$. We also know that $\pi_{j(k)} \cdot (z'_{j(k)} - g'^k_{j(k)}) \ge \delta'$. Thus since we know $0 = g'^k \le z'[\cup_{t \ge k} N_t]$, then clearly $\delta' = \pi^k g'^k + \delta' \le \pi z'[\cup_{t \ge k} N_t] = \sum_{t \ge k} \beta_t$.

We know

$$0 \le \gamma_1^{-1} \sum_{t=1}^{l} \beta_t + (\gamma_2^{-1} - \gamma_1^{-1}) \sum_{t=2}^{l} \beta_t + \ldots + (\gamma_l^{-1} - \gamma_{l-1}^{-1})\beta_l = \hat{\pi} z'.$$

We know that the term $(\gamma_k^{-1} - \gamma_{k-1}^{-1})(\sum_{t \ge k} \beta_t) \ge (\gamma_k^{-1} - \gamma_{k-1}^{-1})\delta'$. Let $\delta = (\gamma_{k-1}^{-1} - \gamma_{k-1}^{-1})\delta'$. Since every term in the sum is nonnegative,

$$0 < 2^{-E} \le \delta \le \hat{\pi} z'$$

as long as we can get a proper lower bound on δ by demonstrating a lower bound for $\gamma_k^{-1} - \gamma_{k-1}^{-1}$. Note the equality

$$\gamma_k^{-1} - \gamma_{k-1}^{-1} = 1/\gamma_k - 1/\gamma_{k-1} = \frac{\gamma_{k-1} - \gamma_k}{\gamma_{k-1}\gamma_k}.$$

Since all prices are in Δ, $\gamma_{k-1}\gamma_k$ is sufficiently bounded, we need only lower bound $\gamma_{k-1} - \gamma_k$.

We know we have a proper lower bound on $\pi_{j'}^{k-1} - \pi_{j'}^{k}$ which we will call $\hat{\pi}_{j'}\sigma$. So,

$$\pi_{j'}^{k-1} - \pi_{j'}^{k} = \gamma_{k-1}\hat{\pi}_{j'} - \gamma_k\hat{\pi}_{j'} = \hat{\pi}_{j'}(\gamma_{k-1} - \gamma_k) \geq \hat{\pi}_{j'}\sigma$$

.

Thus, we have a good lower bound where

$$(\gamma_{k-1} - \gamma_k) \geq \sigma$$

and thus

$$0 < 2^{-E} \leq \delta \leq \hat{\pi}z'.$$

□

The separation lemma allows us to use the ellipsoid method to construct a polynomial time algorithm. As stated previously, this approach follows the work of [23] and utilizes the central-cut ellipsoid method.

The following theorem is the algorithmic result of the strong separation lemma for correspondences.

Theorem 2.3.6 *Let M be an exchange market where the excess demand is a GS correspondence. Assume that M is equipped with a demand oracle. A polynomial-time algorithm that given any $\pi \in \mathbf{R}_+^n$ and $\mu > 0$, asserts that π is a weak (1+ μ)-approximate equilibrium or that π is not a weak (1 + $\mu/2$)-approximate equilibrium* *is also assumed to exist. There then exists an algorithm that takes M, a rational* $\epsilon > 0$ *and returns a weak* $(1+\epsilon)$*-approximate equilibrium price vector in time that is polynomial in the input size of* M *and in* $log(\frac{1}{\epsilon})$*.*

For a more detailed look at how one uses a separation oracle and the central-cut ellipsoid method to derive an algorithmic result, see the end of Chapter 3.

2.3.4 The Spending Constraint Model

Nikhil Devanur and Vijay Vazirani have introduced a new market model which they call the "Spending Constraint Model" [33,35]. Their purpose in introducing their new model is to retain weak gross substitutability, but present an efficient algorithm for a wide class of concave utility functions. We present the spending constraint model for the Exchange or Arrow-Debreu Market and show that our techniques can compute equilibria for these markets in polynomial time. The Fisher case is largely similar.

There are n goods and n' traders. Each agent i has an endowment of $e_i \in [0,1]^n$. The income of the trader will be represented by $m_i = \sum_{1 \leq i \leq n} e_{ij}\pi_j$. There is one unit of each good in the market. For $i \in 1, 2, \ldots n$ and $j \in 1, 2, \ldots n'$, let $f^i_j : [0, m_i] \to \mathbf{R}^+$ be the *rate function* of trader i for good j; the rate at which i derives utility per unit of j received as a function of the amount of her budget spent on j. Define $g^i_j : [0, m_i] \to \mathbf{R}^+$ to be:

$$g^i_j = \int_o^x \frac{f^i_j(y)}{\pi_j} dy.$$

This function give the utility derived by trader i spending x dollars on good j at price π_j. We let $j = 0$ represent money, thus f_0^i and g_0^i will be used to determine the utility of unspent money. The price of money, π_o, is assumed to be 1. Devanur and Vazirani provide a further restriction that the f_j^i's be decreasing step functions. In this case, the g_j^i's will then be piecewise-linear concave functions.

Each step of f_j^i is called a *segment*. The set of segments defined by function f_j^i is denoted by seg(f_j^i). Suppose one of these segments, s has range $[a, b] \subseteq [0, m_i]$, and $f_j^i = c$, for $x \in [a, b]$. Then we define value(s) = $b - a$, rate(s) = c, and good(s) = j (good(0) = money.) Let segments(i) be the union of all the segments of buyer i.

Devanur and Vazirani also add the two following assumptions. For each good, there is a buyer who desires it. That is, For all $j \in 1, 2, \ldots n$, there is i ∈1, 2, ...n' such that there is $s \in seg(f_j^i) : rate(s) > 0$. Also, each buyer i wishes to use all of her money: $\sum_{s \in segments(i), rate(s)>0} value(s) \geq m_i$. These assumptions will ensure that an equilibrium exists and that all equilibrium prices are positive.

With all these assumptions in place, optimal baskets for traders are easily characterized. *Bang for the Buck* relative to prices π for segment $s \in seg(f_j^i)$, is defined as $rate(s)/\pi_j$ (or just $rate(s)$ if $j = 0$). Sort all segments $s \in segments(i)$ by decreasing bang per buck, and partition by equality into classes: $Q_1, Q_2, \ldots$. For a class Q_l, value(Q_l) is defined to be the sum of the values of segments in it. At prices p, goods corresponding to any segment in Q_l make i equally happy, and those in Q_l are desired strictly more by i than those in Q_{l+1}. There is k such that

$$\sum_{1 \le l \le k-1} value(Q_l) < e(i) \le \sum_{1 \le l \le k} value(Q_l).$$

Clearly, i's optimal allocation, that is i's demand, must contain all goods corresponding to segments in $Q_1, \ldots, Q_{k-1}$, and a bundle of goods worth $m_i - (\sum_{1 \le l \le k-1} value(Q_l))$ from segments in Q_k. It is said that for buyer i, at prices p, $Q_1, \ldots, Q_{k-1}$ are *forced partitions*, Q_k is the *flexible partition*, and $Q_{k+1}, \ldots$ are the undesirable partitions.

Note that the possibility of a flexible partition implies that the demand of this market need not be single-valued, it is a correspondence. It is assumed that prices are positive and the demand is homogenous of order zero. In particular for any price π (where π_0 represents the price of money):

$$\mathscr{Z}(\pi) = \mathscr{Z}(1, \frac{\pi_1}{\pi_0}, \frac{\pi_2}{\pi_0}, \ldots, \frac{\pi_n}{\pi_0}).$$

Thus, the standard spending constraint formulation provides us with a well defined demand correspondence. In the proofs of the lemmas below, one must use general prices, not simply prices where $\pi_0 = 1$, so a trader's income is taken to be $m_i * \pi_0$.

It is straightforward to show that this correspondence will sastify the mild conditions (homogeneity, Walras' Law, etc.) detailed earlier when correspondences were introduced. We must now show that this market's demand satisfies the Poltervitch-Spivak definition of gross substitutability. Then, lemma 2.3.5 can be applied and it will be shown that a polynomial time algorithm to compute an equilibrium for Spending Constraint Markets can be produced. If the equilibrium price produced by the algorithm has $\pi_0 \neq 1$, simply divide each price by the price of good 0 and we are provided with an equilibrium that makes sense in the spending constraint context.

It should be noted that in [35], Devanur and Vazirani show that the demand satisfies a different definition of Weak Gross Subsitutability. This is in the restricted case where the f_i are continuous and strictly decreasing and the demand is a differentiable function. This section is concerned with the case where the f_i are decreasing step functions (a case considered closely in [33,35]) and the demand need not be single valued. Therefore, the GS lemma in [35] does not apply since that definition of GS assumes the demand is a differentiable function. Vazirani does show that weak gross substituability holds for the Fisher case in [33]. The proof of the following lemma for the exchange market is similar to the Fisher market argument in [33].

Lemma 2.3.7 *A Spending constraint model exchange market M has a demand $\mathcal{Z}$ that is a GS correspondence.*

Proof:

Let positive price vectors π and π' with $\pi \leq \pi'$ and $\pi \neq \pi'$ be given. It should be noted that all traders will have at least as much income under π' as they do under π. Let $z \in \mathcal{Z}(\pi)$ and $z' \in \mathcal{Z}(\pi')$ be given. Suppose $\mathcal{Z}$ is not a GS correspondence. Then for all $i \in I(\pi, \pi')$, $z_i > z'_i$.

Consider a trader i that spends more on good $j \notin I(\pi, \pi')$ under π' than under π even though $\pi'_j > \pi_j$. Let k be such that a segment of good j is in class Q_k where this portion of j is purchased under π' but not π. Clearly, this segment could not have been in the forced partition under π or it would have been purchased at that price.

If it was in the flexible partition or undesirable partition under π then it has an even worse bang for buck relative to prices π'. This means that any good l in $I(\pi, \pi')$ that were in the forced or flexible partitions under π would have been purchased under π' and thus $z_l'^i \geq z_l^i$. All traders that do not spend more on a good $j \notin I(\pi, \pi')$ under π' than under π, then must spend at least as much on goods in $I(\pi, \pi')$ as they did under π since they have at least as much income under π' and all their nonnegative additional income goes to goods in $I(\pi, \pi')$. Thus, traders spend at least as much on goods in $I(\pi, \pi')$ under π' as under π. Therefore there is a good $i \in I(\pi, \pi')$, such that $z_i - z_i' \leq 0$ and we have a contradiction. Therefore, $\mathcal{Z}$ must be a GS correspondence.

□

2.3.4.1 Rationality of Prices in the Spending Constraint Model

This section demonstrates that when the Spending Constraint Model has rational input parameters, equilibirum prices will also be rational. The existence of a rational price equilibrium along with lemma 2.3.7, lemma 2.3.2 and an extension of the ellipsoid method due to Jain [58] will allow the computation of an exact equilibria in polynomial time.

Lemma 2.3.8 *Let M be an Spending Constraint Exchange Market with rational input parameters. There is a rational equilibrium price vector for M. The binary represenation of the numerator and denominator of this vector is bounded by a polynomial in the input size.*

Proof:

Suppose π is an equilibrium price.

Let $Q_1, \ldots, Q_k$ be the partition of segments into equivalence classes ordered so that they have decreasing "bang for the buck" for a given trader. Thus, we know that for all traders i, with all goods j and j', and corresponding good segments s and s' with $s \in Q_p$, $s' \in Q_q$, if $p \leq q$, we then have $\frac{rate(s)}{\pi_j} \geq \frac{rate(s')}{\pi_{j'}}$. This inequality constraint can be re written as $rate(s')\pi_j \leq rate(s)\pi_{j'}$.

At equilibrium with each trader maximizing utility subject to their budget constraints, we have each trader possessing a k such that

$$\sum_{1 \leq l \leq k-1} value(Q_l) < \sum_{1 \leq i \leq n} e_{ij}\pi_j \leq \sum_{1 \leq l \leq k} value(Q_l).$$

Trader i's demand must contain all goods corresponding to segments in $Q_1, \ldots, Q_{k-1}$, and a bundle of goods worth $m_i - (\sum_{1 \leq l \leq k-1} value(Q_l))$ from segments in Q_k.

The market clears at equilibrium. For each good j, $\sum_i x_{ij} \leq \sum_i e_{ij} = 1$.

If a price satisfies all these constraints, then it is an equilibrium. Note that all of the constraints are linear, thus a solution to the linear system, a price equilibrium, is rational.

□

A similar lemma has been shown for the Spending Constraint Fisher market in [33].

Theorem 2.3.9 *Exact equilibrium prices for Spending Constraint Markets can be computed in polynomial time.*

The theorem follows from lemma 2.3.8, lemma 2.3.7, lemma 2.3.2 and a straightforward application of Theorem 12 in [58].

CHAPTER 3
MARKET EQUILIBRIUM FOR CES EXCHANGE ECONOMIES: EXISTENCE AND COMPUTATION

3.1 Introduction

In this chapter[1], we consider exchange economies where the traders' preferences are expressed in terms of the extensively used *constant elasticity of substitution* (CES) utility functions. We show that for all these economies it is possible to say whether an equilibrium exists in polynomial time.

We then describe a convex formulation of the equilibrium conditions, which leads to polynomial time algorithms for a wide range of the parameter defining the CES utility functions. This range includes instances that do not satisfy weak gross substitutability. It is then demonstrated that the ellipsoid method can be shown to compute approximate equilibria in polynomial time.

An *exchange economy* consists of a collection of goods, initially distributed among a number of traders. The preferences of the traders for the bundles of goods are expressed by a utility function. Each trader wants to maximize her utility, subject to her budget constraint.

An equilibrium is a set of prices at which there are allocations of goods to traders such that two conditions are simultaneously satisfied: each trader's allocation maximizes her utility subject to the budget constraint, and the market clears.

---

[1]The results of this chapter based on joint work with Bruno Codenotti, Sriram Penumachta, and Kasturi Varadarajan and appeared in [19].

Existence. An early and fundamental triumph of Mathematical Economics was the 1954 result by Arrow and Debreu [6] that, even in a more general situation which includes the production of goods, subject to mild sufficient conditions, there is an equilibrium. However, given a set of traders, each endowed with a utility function and a nonnegative vector of initial endowments, an equilibrium does not need to exist.

Thus the problem arises of determining whether a given exchange economy has an equilibrium. In this paper, we show that this problem can be solved in polynomial time, whenever the utility functions are of the form $u(x_1, \ldots, x_n) = \left(\sum_{j=1}^{n} c_j x_j^{\rho}\right)^{\frac{1}{\rho}}$, with $\rho < 1$, $\rho \neq 0$, i.e., for constant elasticity of substitution (CES) utility functions [92].

This result generalizes methods of Eaves [38], who analyzed the existence of positive equilibrium prices for Cobb-Douglas utility functions, and Gale [48], who analyzed the existence of equilibria for linear utility functions. (See also Jain [58], who employs a sufficient condition for the existence of positive price equilibria for linear utility functions.) Our result is in contrast with the NP-hardness result of [27], which applies to Leontief utility functions. As described below, linear, Cobb-Douglas, and Leontief utility functions are limiting cases of CES utility functions.

Computation. The problem of computing equilibrium prices for exchange economies has attracted a lot of attention since the 1960s. In recent years, theoretical computer scientists have become interested in the polynomial-time solvability of the problem. Several results [75] seem to indicate that in order for the problem to admit polynomial time algorithms, certain restrictions should be satisfied by the market.

Two well studied restrictions are *gross substitutability* – GS (see [1], p. 611) and the *weak axiom of revealed preferences* – WARP (see [1], Section 2.F). Although restrictive, these conditions are useful and model some realistic scenarios.

It is well known that GS implies that the equilibrium prices are unique up to scaling ([95], p. 395), and that WGS and WARP both imply that the set of equilibrium prices is convex ([1], p. 608). When the set of equilibria is convex, it is enough to add a non-degeneracy assumption (which is almost always satisfied) to get the uniqueness of the equilibrium up to scaling [32].

All the polynomial-time algorithms developed so far apply to scenarios where either WGS or WARP hold. In this chapter we discuss a convex characterization of the equilibrium conditions which applies to exchange economies with CES functions such that $-1 \leq \rho < 0$. [2] Note that these economies do not fall into either WGS or WARP.

Related Work. In a series of papers which started with linear utility functions, polynomial time algorithms have been developed to compute equilibria for more and more general settings [14,20,28,49,58,60,69,80,100]. However, the corresponding market satisfies one of the two conditions discussed above (WGS or WARP) (see [21] for a review).

The technical tool used in some of these results is to reformulate the problem in terms of mathematical programming in a way that a polynomial time algorithm

[2] Similar convex formulations hold for $\rho > 0$, as well as for other functional forms. A comprehensive presentation of these results can be found in [28].

(or approximation scheme – in general the equilibrium point is not a vector of rationals) can be obtained by known optimization techniques. In particular, *convex programming* has been proven to be a particularly useful tool [23, 28, 58, 73, 100].

Summary of Our Contribution. We settle several issues concerning equilibria in exchange economies with CES utility functions. We first show that the existential problem can always be solved in polynomial time by checking the biconnectivity of a digraph associated with the input. We then discuss the range of these functions for which the economy's equilibria form a convex set and use this characterization to derive a polynomial-time algorithm. We leave open the important problem of whether or not it is possible to find polynomial-time algorithms for the range where the economies admit multiple disconnected equilibria.

Organization. In Section 3.2, we formally describe the model of an exchange economy, introduce CES functions, and hint at their economic relevance. Section 3.3 is devoted to a detailed discussion of the demand function of traders with CES utility functions.

In Section 3.4, we characterize the problem of existence of an equilibrium for CES exchange economies, in terms of a graph property that can be verified in polynomial time.

In Section 3.5 we present the demonstration that equilibrium prices and allocations for an exchange economy, where the traders are endowed with CES functions with $-1 \leq \rho < 0$, can be computed by solving a feasibility problem, defined in terms of explicitly given convex constraints. Here, it is then shown that this formulation can lead to a polynomial time algorithm for computing the equilibrium via the ellipsoid method.

3.2 Background

For definitions of the exchange model and equilibrium, see Chapter 1.

CES utility functions. The most popular family of utility functions is given by CES (constant elasticity of substitution) functions, which have been introduced in [92]. We refer the reader to the book by Shoven and Whalley [59] for a sense of their pervasiveness in applied general equilibrium models. A CES function ranks the trader's preferences over bundles of goods $(x_1, \ldots, x_n)$ according to the value of $u(x_1, \ldots, x_n) = \left(\sum_{j=1}^{n} c_j x_j^{\rho}\right)^{\frac{1}{\rho}}$. where $-\infty < \rho < 1$, but $\rho \neq 0$.

The success of CES functions is due to the useful combination of their mathematical tractability with their expressive power, which allows for a realistic modeling of a wide range of consumer preferences. Indeed, one can model markets with very different characteristics in terms of preference towards variety, substitutability versus complementarity, and multiplicity of price equilibria, by changing the values of ρ and of the utility parameters c_j.

CES functions have been thoroughly analyzed in [5], where it has also been shown how to derive, in the limit, their special cases, i.e., linear, Cobb-Douglas, and Leontief functions (see [5], p. 231). Let $\sigma = \frac{1}{1-\rho}$. The parameter σ is called the *elasticity of substitution*. For $\sigma \to \infty$ ($\rho \to 1$), CES take the linear form, and the goods are perfect substitutes, so that there is no preference for variety. For $\sigma > 1$

$(\rho > 0)$, the goods are partial substitutes, and different values of σ in this range allow us to express different levels of preference for variety. For $\sigma \to 1$ $(\rho \to 0)$, CES become Cobb-Douglas functions, and express a perfect balance between substitution and complementarity effects. Indeed it is not difficult to show that a trader with a Cobb-Douglas utility spends a fixed fraction of her income on each good.

For $\sigma < 1$ $(\rho < 0)$, CES functions model markets with significant complementarity effects between goods. This feature reaches its extreme (*perfect complementarity*) as $\sigma \to 0$ $(\rho \to -\infty)$, i.e., when CES takes the form of Leontief functions. In the latter case, the *shape* of the optimal bundle demanded by the consumer does not depend at all on the prices of the goods, but is fully determined by the parameters defining the utility function.

Whenever the relative incomes of the traders are independent of the prices, CES functions give rise to a market which satisfies WARP. This happens for instance in the Fisher model, a very special case of the exchange model. On the other hand, CES functions satisfy WGS if and only if $\rho \geq 0$, whereas, if $\rho < -1$, they allow for multiple disconnected equilibria [52].

3.3 Demand of CES Consumers

In this section, we characterize the demand function of traders with CES utility functions. Consider a setting where trader i has an initial endowment $w_i = (w_{i1}, \ldots, w_{in}) \in \mathbf{R}^n_+$ of goods, and the CES utility function $u_i(x_{i1}, \ldots, x_{in}) = \left(\sum_{j=1}^n \alpha_{ij} x_{ij}^{\rho_i}\right)^{\frac{1}{\rho_i}}$, where $\alpha_{ij} \geq 0$, and $-\infty < \rho_i < 1$, but $\rho_i \neq 0$.

We assume throughout that $w_{ij} > 0$ for some j, and also that $\alpha_{ij} > 0$ for some j. If $\alpha_{ij} > 0$, we say that trader i *wants* good j. If trader i does not want good j, it is easy to see that the utility of a bundle $x_i \in \mathbf{R}_+^n$ is independent of x_{ij}. We adopt the convention that $\alpha_{ij} x_{ij}^{\rho_i} = 0$ when $\alpha_{ij} = 0$ and $x_{ij} = 0$.

First consider the case where $\rho_i > 0$. Evidently, if we start with any bundle $x_i \in \mathbf{R}_+^n$ and add to it an arbitrarily small amount of a good that i wants, we get a bundle with more utility. From this, it follows that the demand of the trader is well-defined at a given price if and only if each of the goods that the trader wants has a strictly positive price.

Now consider the case where $\rho_i < 0$. It is easy to see that a bundle $x_i \in \mathbf{R}_+^n$ has a strictly positive utility if and only if it has a strictly positive amount of each of the goods that the trader wants. Evidently, if we start with any bundle $x_i \in \mathbf{R}_+^n$ that has strictly positive utility and add to it an arbitrarily small amount of a good that i wants, we get a bundle with more utility. Let π be a price at which the income $\pi \cdot w_i$ is positive. Since the trader can afford a bundle with positive utility, we conclude that the demand is well-defined if and only if each of the goods that the trader wants has a strictly positive price. Now let π be a price at which the income $\pi \cdot w_i$ is zero. We see that the demand is well-defined if and only if *at least one* of the goods that the trader wants is positively priced.

Irrespective of whether ρ_i is positive or negative, traders with positive income demand a positive amount of each good they want. Traders with CES utilities are also *non-satiable* on all goods they want which means that demand is not well-defined on the zero priced goods they want.

Also irrespective of whether ρ_i is positive or negative, the demand is well-defined at any strictly positive price vector $\pi \in \mathbf{R}^n_{++}$. It is in fact unique and is given by the expression

$$x_{ij}(\pi) = \frac{\alpha_{ij}^{1/1-\rho_i}}{\pi_j^{1/1-\rho_i}} \times \frac{\sum_k \pi_k w_{ik}}{\sum_k \alpha_k^{1/1-\rho_i} \pi_k^{-\rho_i/1-\rho_i}}. \tag{3.1}$$

The formula above is folklore and is derived using the Kuhn-Tucker conditions.

3.4 Existence of an Equilibrium

The celebrated paper of Arrow and Debreu [6] had a much weaker set of assumptions sufficient for the existence of equilibrium than earlier work. The assumptions were still somewhat restrictive though. Indeed, Arrow and Debreu themselves called the assumptions for their first existence theorem "clearly unrealistic" and immediately proceeded to weaken the sufficent conditions for their second theorem. See the introduction to Maxfield [66] for a discussion of the work on showing existence of equilibrium under progressively weaker assumptions. In general, it is NP-hard to determine whether a market possesses an equilibrium or not [27].

Gale [48] provided a very simple two trader example of a market that does not possess an equilibrium. Gale's example was for the linear exchange model, but it also serves as an example for the CES case with $\rho > 0$. Suppose trader one possesses both apples and oranges, but only wants apples. Trader two wants both apples and oranges, but owns only oranges. This simple market has no equilibrium. If oranges are priced at zero, then the demand of trader two is not well-defined. If oranges have a positive price, then trader one will want to sell all of her oranges to buy more apples even though she already owns all the apples present in the market. Gale's example will not work for the CES with $\rho < 0$ case though because that actually has an equilibrium with a positive price for apples and zero price for oranges.

In this section, we characterize the existence of equilibrium for an exchange economy where the traders have CES utility functions. The characterization immediately implies a polynomial time algorithm to decide whether the economy has an equilibrium. As before, we assume that each trader wants some good. That is for each trader i, there exists a j such that $\alpha_{ij} > 0$.

We assume in the remainder of this section that each trader has a positive amount of precisely one good. This assumption is without loss of generality: we may replace a trader with positive amounts of k different goods with k traders, each with the same utility function and a positive amount of one good. A straightforward argument that employs the homogeneity of the CES utility functions shows that this transformation preserves the equilibria.

It is easy to see, but nonetheless worth noting, that the traders with positive income will be precisely those traders whose single good is positively priced.

Definition 3.4.1 *There is a vertex v_i for each consumer i. We have an arc from v_i to v_k when trader i possesses a good which trader k wants. The resulting directed graph is called an economy graph.*

The following existence theorem is the main result we use from Maxfield [66].

Theorem 3.4.2 *If the economy graph is strongly connected, an equilibrium exists. Moreover, all goods are positively priced at any equilibrium in such a market.*

Proof: This follows from Theorem 2 of Maxfield [66] who obtains this result using strong connectivity and general results on the existence of a *quasi-equilibrium* ([1], Chapter 17). □

Definition 3.4.3 *We say that a strongly connected component in the economy graph is on if every trader within it has a positive income. If no trader in a strongly connected component has a positive income, then we say that that component is off.*

Lemma 3.4.4 *At equilibrium, every strongly connected component in an economy graph is either on or off.*

Proof: Suppose not. Suppose we are given an equilibrium price π, but there is a component that is neither on or off. In that case, there must be a trader with positive income that desires a good from a trader with no income. That means the zero income trader's good must have a price of zero. Since the trader with positive income is non-satiable on the zero priced good, demand is not well-defined for that good and therefore, π is not an equilibrium. This provides a contradiction. □

Consider a strongly connected component C of the economy graph that has no incoming arcs from traders outside C. We claim that a good held by any trader i in C is also desired by some trader i' in C. If C consists simply of the node v_i, then since there are no incoming arcs from outside, it must be that i desires his own good (because i must desire some good). If C consists of more than one node, the claim follows from strong connectivity.

Furthermore, it follows that a good held by a trader in C is not held by any trader outside C. Otherwise, C would have an incoming arc.

Lemma 3.4.5 *At equilibrium, a strongly connected component of an economy graph is on if and only if it has no incoming arcs.*

Proof: Suppose the economy has an equilibrium price π. Suppose a strongly connected component C_1 is on. We will show C_1 can have no incoming arcs. If C_1 has an incoming arc, that means some trader in C_1 wants some good in another component C_2. This trader will purchase a good that is present in C_2 since she has a positive income. If C_2 is off, then the trader in C_1 will demand an infinite amount of a good in C_2, thus contradicting the assumption of equilibrium. Thus, at equlibrium C_2 must be on. If C_2 has any incoming arcs, then we can make an identical argument to show that the components providing the incoming arcs must also be on. This implies that there is a chain of on components that only ends with an on component that has no incoming arcs. Therefore, without loss of generality, let us assume C_2 has no incoming arcs. A trader in C_1 will purchase a portion of some positively priced good in C_2. This is because there is no other place for the trader to acquire that good, otherwise there would be an arc incoming to C_2 from some external trader that owns that good. There is no way for traders in C_2 to get back the value from C_1. This is because traders in C_2 desire and thus purchase only goods that are in C_2, all of which are unique to C_2. This implies that traders in C_2 maximizing their utility cannot satisfy their budget constraint tightly even though they wish to consume more goods and have income to buy them. Thus, π is not an equilibrium which is a contradiction.

Suppose the economy has an equilibrium price π. Suppose further than a component, C, has no incoming arcs. We show that C must be on. Suppose C is off. Traders in C demand only goods in C. All goods in C are free so utility maximizing traders (who by assumption, desire at least one good) in C will have undefined demands without violating their budget constraints. Therefore, π is not an equilibrium. We have a contradiction and the lemma is proven.

□

There is an important distinction, which bears repeating, between CES utility functions with $\rho > 0$ and those with $\rho < 0$. Traders with $\rho > 0$ will have positive utility as long as they have a positive amount of some good that they desire. Traders with $\rho < 0$ will only have positive utility if they have a positive amount of all goods they desire. Moreover, traders with $\rho > 0$ with zero income have undefined demands if any of their desired goods are priced at zero. Zero income traders with $\rho < 0$ only have undefined demand if all of their desired goods are free.

The following theorem is the main result of this section.

Theorem 3.4.6 *An equilibrium exists if and only if for every vertex v in a strongly connected component with incoming arcs, either (a) v has a CES utility function with $\rho > 0$ and all its incoming arcs are from vertices in strongly connected components without incoming arcs, or (b) v has a CES utility function with $\rho < 0$ and has at least one incoming arc from a strongly connected component without incoming arcs.*

Proof: Suppose an equilibrium price π exists. Then by Lemma 3.4.5, the strongly connected components that are on are precisely those that have no incoming arcs. And it is precisely the goods that are held by traders in such components that have positive price. Let C_1 be a strongly connected component with incoming arcs (if none exist, then this direction of the theorem is trivially true). Suppose there is a vertex i with a CES utility function with $\rho > 0$, and it has an incoming arc from a vertex that is in a strongly connected component with incoming arcs. Then i wants a good with price zero and so her demand is not defined, contradicting the assumption that π is an equilibrium price. Now suppose that there is a vertex i with a CES utility function with $\rho < 0$, and none of its incoming arcs are from a vertex in a strongly connected component with no incoming arcs. This means that trader i desires only zero priced goods and thus has undefined demand contradicting the assumption that π is an equilibrium price.

For the other direction of the theorem, suppose that every vertex in a strongly connected component with incoming arcs has an incoming arc from a strongly connected component without incoming arcs. Each strongly connected component can be considered as an economy unto itself, and has an equilibrium with positive prices by Maxfield's theorem. For each good in a component with no incoming arcs we assign prices to goods identical to their equilibrium prices as subeconomies. As no good in one of these strongly connected components is owned outside the component, this assignment of prices is well-defined.

For each good held by a trader in a component with incoming arcs, we assign a price of zero. By the argument above, we know that none of these goods are the same as those that were priced positively so this price is well defined. We claim that this price π is an equilibrium price.

For a trader in a component without incoming arcs, we assign the bundle that is the same as the one she gets in the equilibrium for the corresponding subeconomy. Clearly, this is a valid demand.

Consider a trader in a component with incoming arcs. Her income is 0. We claim that her demand is well-defined and that the zero bundle is a valid demand vector. This is because she is either a CES trader with $\rho > 0$ and all the goods that she wants are in components with no incoming arcs and hence positively priced, or she is a CES trader with $\rho < 0$ and at least one of the goods that she wants is positively priced, and thus the best utility she can afford is 0.

We now verify that condition (2) in the definition of an equilibrium holds, that is, the demand is at most the supply. For a good held by a trader in a component with no incoming arc, this follows from the equilibrium conditions of the corresponding subeconomy, and the fact that any trader outside the component demands 0 units of the good. For a good held by a trader in a component with incoming arcs, the net demand is 0, so condition (2) trivially holds.

This completes the proof of the theorem. $\square$

It may be worth noting that this theorem easily extends to the Cobb-Douglas case. On the relevant properties, the Cobb-Douglas utility functions are identical to the $\rho < 0$ case. The linear case has been understood for some time [48]. As mentioned earlier, the final CES limit case, determing existence of equilibria when traders have Leontief utility functions is now known to be NP-hard [27].

We conclude by noting that besides yielding a polynomial time algorithm for checking the existence of equilibrium, the above characterization provides a polynomial-time reduction of the computation of an equilibrium for the original economy to the computation of positive price equilibria for sub-economies.

3.5 Efficient Computation by Convex Programming

In this section, we consider an economy in which each trader i has a CES utility function with $-1 \leq \rho_i < 0$. We present Codenotti and Varadarajan's demonstration that the positive price equilibria of such an economy can be characterized as the solutions of a convex feasibility problem. The results of the previous section show that the computation of an equilibrium for an economy can be reduced to the computation of a positive price equilibrium for a sub-economy that is represented by a strongly connected graph as discussed above. With this reduction, we can now show via the elliposoid method, that there is a polynomial time algorithm for computing an approximate equilibrium. This is significant because previous algorithms have been for markets that satisfy gross substituability.

Since the demand of every trader is well-defined and unique at any positive price, we may write the equilibria as the set of prices $\pi \in \mathbf{R}_{++}$ such that for each good j, we have $\sum_i x_{ij}(\pi) \leq \sum_i w_{ij}$. Let $\rho = -1$, and note that $\rho \leq \rho_i$, for each i. Let $f_{ij}(\pi) = \pi_j^{1/(1-\rho)} x_{ij}(\pi)$. Let $\sigma_j = \pi_j^{1/(1-\rho)}$. In terms of the σ_j's, the equilibria are the set of $\sigma = (\sigma_1, \ldots, \sigma_n) \in \mathbf{R}_{++}$ such that for each good j,

$$\sum_i f_{ij}(\sigma) \leq \sigma_j(\sum_i w_{ij}).$$

We argue that this is a convex feasibility program. Since the right hand side of each inequality is a linear function, it suffices to argue that the left hand side is a convex function. The latter is established via the following proposition.

Proposition 3.5.1 *The function $f_{ij}(\sigma)$ is a convex function over $\mathbf{R}_{++}$.*

Proof: If $\alpha_{ij} = 0$, f_{ij} is zero over the domain and the proposition follows. Otherwise, f_{ij} is positive at each point of the domian. It therefore suffices to show that the constraint $f_{ij} \leq t$ defines a convex set for positive t. Using formula 3.1 for demand, this constraint is

$$\frac{\alpha_{ij}^{\frac{1}{1-\rho_i}}}{\sigma_j^{\frac{\rho_i-\rho}{1-\rho_i}}} \times \frac{\sum_k \sigma_k^{1-\rho} w_{ik}}{\sum_k \alpha_{ik}^{\frac{1}{1-\rho_i}} \sigma_k^{\frac{-\rho_i(1-\rho)}{1-\rho_i}}} \leq t.$$

Rewriting, and raising both sides to the power $1/(1-\rho)$, we obtain

$$\alpha_{ij}^{\frac{1}{(1-\rho)(1-\rho_i)}} \times (\sum_k \sigma_k^{1-\rho} w_{ik})^{\frac{1}{1-\rho}} \leq t^{\frac{1}{1-\rho}} \sigma_j^{\frac{\rho_i-\rho}{(1-\rho_i)(1-\rho)}} v_i^{\frac{-\rho_i}{1-\rho_i}}, \tag{3.2}$$

where

$$v_i = \left(\sum_k \alpha_{ik}^{\frac{1}{1-\rho_i}} \sigma_k^{\frac{-\rho_i(1-\rho)}{1-\rho_i}}\right)^{\frac{1-\rho_i}{-\rho_i(1-\rho)}}. \tag{3.3}$$

The left hand side of inequality 3.2 is a convex function, and the right hand side is a concave function that is non-decreasing in each argument when viewed as a function of t, σ_j, and v_i, since the exponents are non-negative and add up to one. Since $0 < \frac{-\rho_i(1-\rho)}{1-\rho_i} < 1$, the right hand side of equality 3.3 is a concave function, in fact a CES function. It follows that the right hand side of inequality 3.2 remains a concave function when v_i is replaced by the right hand side of equality 3.3. This completes the proof. $\square$

Converting the Convex Program into an Algorithm We now show through a series of lemmas that this convex program produces a polynomial time algorithm to compute an approximate equilibrium via the ellipsoid method. If we let an $\epsilon > 0$ be given, then we wish to find a $(1+\epsilon)$-approximate equilibrium in polynomial time. Choose L to be an integer such that for all i, $-1 \leq \rho_i \leq -\frac{1}{L}$. Also, L must be chosen such that for all i and j, $\frac{1}{L} \leq w_{ij}$ (if $w_{ij} \neq 0$), $\sum_i w_{ij} \leq L$, $\frac{1}{L} \leq \alpha_{ij}$(if $\alpha_{ij} \neq 0$) and $\sum_i \alpha_{ij} \leq L$. L must also have the property that it bounds the number of goods and traders. Define $K = L^{8n}$.

The following lemma shows that at equilibrium, there is a bound on the ratio of the largest price to the smallest price.

Lemma 3.5.2 *At equilibrium,* $\frac{\sigma_{max}}{\sigma_{min}} \leq K$.

Proof: Suppose we have

$$\frac{\pi_{max}}{\pi_{min}} \leq K.$$

It would follow that,

$$\frac{\sigma_{max}}{\sigma_{min}} = (\frac{\pi_{max}}{\pi_{min}})^{\frac{1}{1-\rho}} \leq K^{\frac{1}{1-\rho}}.$$

Since $K \geq 1$, we have $K^{\frac{1}{1-\rho}} \leq K$. Therefore,

$$\frac{\sigma_{max}}{\sigma_{min}} \leq K.$$

Therefore, we must show that $\frac{\pi_{max}}{\pi_{min}} \leq K$.

Let π be a market equilibrium price. Scale the prices so the $\pi_{max} = 1$. Because this is a strongly connected component of an economy graph, we may construct a chain $(j_0, j_1, j_2, \ldots, j_m)$ where $m < n$, the number of goods, and for each j_k, there is a trader that posseses the j_k^{th} good and desires the j_{k+1}^{th} good.

Choose i and j' such that $j_0 = j'$ and $w_{ij'} \geq \frac{1}{L}$ and $\pi_{j'} = \pi_{max}$. Choose j_m such that $\pi_{j_m} = \pi_{min}$. Now we examine the demand for trader i for good j where $j = j_1 \neq j'$. We may do this because we are in strongly connected component with multiple traders and goods (equilibrium computations are trivial for a single good or single trader economy).

Recall the formula for demand from Formula 3.1.

$$x_{ij}(\pi) = \frac{\alpha_{ij}^{1/1-\rho_i}}{\pi_j^{1/1-\rho_i}} \times \frac{\sum_k \pi_k w_{ik}}{\sum_k \alpha_k^{1/1-\rho_i} \pi_k^{-\rho_i/1-\rho_i}}.$$

We can utilize the various bounds to see that

$$x_{ij}(\pi) \geq \frac{\frac{1}{L}}{\pi_j^{\frac{1}{2}}} \times \frac{\pi_{j'} w_{ij'}}{L}. \geq \frac{\frac{1}{L}}{\pi_j^{\frac{1}{2}}} \times \frac{\frac{1}{L}}{L}. = \frac{1}{L^3 \pi_j^{\frac{1}{2}}}$$

Since π is an equilibrium,

$$x_{ij}(\pi) \leq \sum_i w_{ij} \leq L.$$

Therefore,

$$L \geq \frac{1}{L^3 \pi_j^{\frac{1}{2}}}$$

.

Solving for π_j, we get

$$\pi_j \geq \frac{1}{L^8}.$$

Therefore, any trader that possesses good j has an income of at least $\frac{1}{L^9}$ and any prices for goods (such as j_2) that this trader desires can be bound below by $\frac{1}{L^{16}}$ simply by using the process in the proof above. If we iterate this process, we will reach j_m (a good with π_{min}) in fewer than n steps and can therefore bound the price of the last good in the chain by $\frac{1}{L^{8n}}$.

Therefore, for all j, $\pi_j \geq \frac{1}{K}$ and $\frac{\sigma_{max}}{\sigma_{min}} \leq K$ $\square$

We define

$$g_i(\sigma) = \sum_i f_{ij}(\sigma) - \sigma_j(\sum_i w_{ij}).$$

Note that if for all i, $g_i(\sigma) < 0$, then σ is an equilibrium. In general, for $\epsilon > 0$, we set $\epsilon' = \frac{\epsilon}{L^2}$. Define a box $B = \{\pi | \frac{1}{4K} \leq \pi_j \leq 1\}$. Define an interior box in B, $B_2 = \{\pi | \frac{1}{2K} \leq \pi_j \leq \frac{1}{2}\}$ Let $\hat{\sigma}$ be an equilibrium inside B_2. There must be an equilibrium inside B_2 by Lemma 3.5.2. Let D be the disk with radius $min(\frac{1}{4K}, \frac{\epsilon'}{K^{10}})$ centered around $\hat{\sigma}$.

The following lemma shows that if the $g_i(\sigma)$ are almost less than zero, then σ is an approximate equilibrium.

Lemma 3.5.3 *If $\sigma \in B$ and for all i, $g_i(\sigma) \leq \epsilon^{'}$, then π is a $(1+\epsilon)$-approximate equilibrium.*

Proof:

Let $\epsilon > 0$ be given. Recall that $\epsilon' = \frac{\epsilon}{L^2}$. Suppose $g_i(\sigma) \leq \epsilon'$. Utilizing the definitions of f and g,

$$g_i(\sigma) = \sum_i \sigma_j x_{ij}(\pi) - \sigma_j \sum_i w_{ij} \leq \epsilon'$$

Divide by σ_j to get

$$\sum_i x_{ij}(\pi) - \sum_i w_{ij} \leq \frac{\epsilon'}{\sigma_j} \leq \frac{\epsilon'}{\sigma_{min}} \leq \frac{\epsilon'}{\frac{1}{L}} \leq L\epsilon'$$

Thus,

$$\sum_i x_{ij}(\pi) \leq \sum_i w_{ij} + L\epsilon'$$

Since $L\sum_i w_{ij} \geq 1$, we have

$$\sum_i w_{ij} + L\epsilon' \leq \sum_i w_{ij} + L\epsilon'(L\sum_i w_{ij}) = \sum_i w_{ij}(1 + L^2\epsilon') = (1+\epsilon)\sum_i w_{ij}.$$

Thus,

$$\sum_i x_{ij}(\pi) \leq (1+\epsilon)\sum_i w_{ij}$$

and π is a $(1+\epsilon)$-approximate equilibirum. $\square$

The following lemma bounds the degree to which g_i can change in the box B.

Lemma 3.5.4 *The magnitude of the derivative of g_i over B is bounded by K^9.*

Proof:

First, we examine the partial derivative.

$$\frac{\partial g_i}{\partial \sigma_j} = \sum_i \left(\frac{\partial f_{ij}(\sigma)}{\partial \sigma_j} - w_{ij}\right)$$

Define $S = \sum_k \alpha_k^{1/(1-\rho_i)} \sigma_k \frac{-\rho_i(1-\rho)}{1-\rho_i}$

$$\frac{\partial f}{\partial \sigma_j} = \frac{\alpha_{ij}^{1/(1-\rho_i)}}{\sigma_j^{(\rho_i-\rho)/(1-\rho_i)}} \left(\frac{(\sigma_j^{1-\rho} w_{ij})S - (\sum_k \sigma_k^{1-\rho} w_{ik})(\alpha_{ij}^{1/(1-\rho_i)}(\frac{-\rho_i(1-\rho)}{1-\rho_i})\sigma_j^{\frac{\rho-1}{1-\rho_i}}}{S^2} \right)$$

$$+ \frac{\sum_k \sigma_k^{1-\rho} w_{ik}}{S} \left(-\alpha_{ij}^{1/(1-\rho_i)} (\frac{\rho_i - \rho}{1-\rho_i}) \sigma^{\frac{\rho-1}{1-\rho_i}} \right)$$

If we examine each term, we see that $-K^7 \leq \frac{\partial f}{\partial \sigma_j} \leq K^6$. This quickly yields $|\frac{\partial g_i}{\partial \sigma_j}| \leq K^8$. The gradient has n components so its magnitude is bounded above by

$$(n * (K^8)^2)^{\frac{1}{2}} \leq (K^{17})^{\frac{1}{2}} \leq K^9.$$

$\square$

Lemma 3.5.5 *Let $\sigma_0 \in B$. We can compute an approximation $\bar{\alpha}$ of $\nabla g_i(\sigma_0)$ so that $|\nabla g_i(\sigma_0) \cdot (\sigma - \sigma_0) - \bar{\alpha} \cdot (\sigma - \sigma_0)| \leq \frac{\epsilon'}{4}$ for each $\sigma \in B$. This computation can be done in time polynomial in n, and the encoding length of K, σ_0 and ϵ.*

Proof: Both σ and σ_0 are in B, so it must be the case that for each j, $|\sigma_j - \sigma_{0j}| < 1$. Therefore, we only need to bound each component of $\nabla g(\sigma_0) - \bar{\alpha}$ by $\frac{\epsilon'}{4n}$. One can see the formula for the partial derivatives of g_i in the proof of Lemma 3.5.4. All of the terms can be approximated closely enough to bound $|g(\sigma_0) - \bar{\alpha}|$ in polynomial time.

□

This is the separating oracle lemma that will be used in the ellipsoid method.

Lemma 3.5.6 *Given any $\sigma_0 \in \mathbb{Q}^n$, we can either assert that σ_0 is an ϵ-approximate equilibrium, or we can return a vector $c \in \mathbb{Q}^n$ such that $c \cdot \sigma \leq c \cdot \sigma_0$ for any $\sigma \in D$. This compuation can be done in time polynomial in n and the encoding length of K, σ_0, and ϵ.*

Proof:

Note that due to the radius of the disk being no larger than $\frac{1}{4K}$ and $\hat{\sigma} \in B_2$, that $D \subset B$. Let $\sigma \in D$ be given.

There are 3 cases to consider.

Case 1: $\sigma_0 \notin B$.

Case 2: $\sigma_0 \in B$ and $g_i(\sigma_0) \leq \frac{\epsilon'}{2}$ for each i.

Case 3: $\sigma_0 \in B$ and $g_i(\sigma_0) > \frac{\epsilon'}{2}$ for each i.

Case 1: We can check that $\sigma_0 \notin B$ by scanning the components of σ_0. There is some i where either (a) $\sigma_{0i} > 1$ or (b) $\sigma_{0i} < \frac{1}{4K}$. If (a), choose $c = e_i$ and $c \cdot \sigma = \sigma_i \leq 1 < \sigma_{0i} = c \cdot \sigma_0$. If (b), choose $c = -e_i$ and $c \cdot \sigma = -\sigma_i \leq \frac{-1}{4K} < -\sigma_{0i} = c \cdot \sigma_0$

Case 2: We can check that $g_i(\sigma_0) \leq \frac{\epsilon'}{2}$ for each i and then assert that σ_0 is an ϵ-approximate equilibrium by Lemma 3.5.3.

Case 3: First, we argue that $g_i(\sigma) \leq \frac{\epsilon'}{4}$ (for all i). Because g_i is convex and $\hat{\sigma}$ is an equilibrium, we know that $0 \geq g_i(\hat{\sigma}) \geq g_i(\sigma) + \nabla g_i(\sigma) \cdot (\hat{\sigma} - \sigma)$. From Lemma 3.5.4 and the size of D, we know that $\nabla g_i(\sigma) \cdot (\hat{\sigma} - \sigma) \geq -(K^9)(\frac{\epsilon'}{K^{10}}) = -\frac{\epsilon'}{K}$. Therefore $0 \geq g_i(\sigma) - \frac{\epsilon'}{K}$, so $g_i(\sigma) \leq \frac{\epsilon'}{K} \leq \frac{\epsilon'}{4}$.

By convexity, we know that the tangent $g_i(\sigma_o) + \nabla g_i(\sigma_0)(\sigma - \sigma_0) \leq g_i(\sigma)$. Because $g_i(\sigma_o) \geq \frac{\epsilon'}{2}$ and $g_i(\sigma) \leq \frac{\epsilon'}{4}$, we may conclude that $g_i(\sigma) \leq g(\sigma_0) - \frac{\epsilon'}{4}$. Thus, $g_i(\sigma_o) + \nabla g_i(\sigma_0)(\sigma - \sigma_0) \leq g_i(\sigma_0) - \frac{\epsilon'}{4}$ which implies that that $\nabla g_i(\sigma_0)(\sigma - \sigma_0) \leq -\frac{\epsilon'}{4}$. By Lemma 3.5.5, we may compute $\bar{\alpha}$ in polynomial time such that $|\nabla g_i(\sigma_0) \cdot (\sigma - \sigma_0) - \bar{\alpha} \cdot (\sigma - \sigma_0)| \leq \frac{\epsilon'}{4}$. This implies that $\bar{\alpha} \cdot (\sigma - \sigma_0) \leq 0$. If we choose $c = \bar{\alpha}$, we have $c \cdot (\sigma - \sigma_0) = c \cdot \sigma - c \cdot \sigma_0 \leq 0$. Therefore $c \cdot \sigma \leq c \cdot \sigma_0$.

□

Applying the Ellipsoid Method This separating oracle lemma will allow us to directly apply (with some slight modifications) the central-cut ellipsoid method, Theorem 3.21 from [53]. Here is the modified central-cut ellipsoid theorem:

Theorem 3.5.7 *There is an algorithm , called the* **central-cut ellipsoid method***, that solves the following problem:*

Input: *A rational number $\mu > 0$ and a closed convex set $C \subset \mathbb{R}^n$ contained in a ball of radius R. There is an oracle that for any $y \in \mathbb{Q}^n$ either accepts y or finds a vector $c \in \mathbb{Q}^n$ such that $c \cdot x \leq c \cdot y$ for any $x \in C$.*

Output:

Either

(i) a vector $a \in \mathbb{Q}^n$ that the oracle accepts, or

(ii) an ellipsoid E such that $C \subseteq E$ and $vol(E) \leq \mu$.

The number of calls that the algorithm makes to the oracle is polynomial in n and the encoding length of its input parameters R and μ,. The number of bits used to represent the rational numbers in the vectors given to the oracle is also bounded by such a polynomial.

Proof: For the proof of the original version of the central-cut ellipsoid method theorem, see Theorem 3.21 in [53]. $\square$

Now, we show that the central-cut ellipsoid method theorem can utilize our separation oracle to find an approximate equilibrium.

First, we set $\mu = (\frac{\epsilon'}{nK^{10}})^n$. This disk D is shall be our closed, convex set, and we shall set $R = n$ in order to easily bound it. If we input σ_0 and if the oracle from Lemma 3.5.6 accepts, then we already have an approximate equilibrium otherwise the oracle gives a separating hyperplane just like it needs to for the theorem. Thus, the ellipsoid method will either produce a $1+\epsilon$-approximate equilibrium or it will produce an ellipsoid E such that $D \subseteq E(A, a)$ and $vol(E(A, a)) \leq \mu$

This condition can't be reached though, because the volume of an m-ball of radius r is bounded below by $(\frac{r}{m})^m$. This means that $vol(D) > \left(\frac{\frac{\epsilon'}{K^{10}}}{n}\right)^n \geq (\frac{\epsilon'}{nK^{10}})^n = \mu$. Thus it can never be the case that $D \subseteq E(A, a)$ if $vol(E(A, a)) \leq \mu$. Therefore, the method always finds a $(1+\epsilon)$-approximate equilibrium and the entire algorithm runs in time polynomial in the input size of the market and the encoding length of ϵ.

CHAPTER 4
AN EXPERIMENTAL STUDY OF DIFFERENT APPROACHES TO COMPUTING MARKET EQUILIBRIA

4.1 Introduction

Over the last few years, the problem of computing market equilibrium prices for exchange economies has received a good deal of attention in the theoretical computer science community. Such activity led to a flurry of polynomial time algorithms for various restricted, yet significant, settings. The most important restrictions arise either when the traders' utility functions satisfy a property known as *gross substitutability* or when the initial endowments are proportional (the Fisher model).

In this chapter[1] we experimentally compare the performance of some of these recent algorithms against that of the most used software packages. In particular, we evaluate the following approaches: (i) the solver PATH, available under GAMS/MPSGE, a popular tool for computing market equilibrium prices; (ii) a discrete version of a simple iterative price update scheme called tâtonnement; (iii) a discrete version of the welfare adjustment process; (iv) convex feasibility programs that characterize the equilibrium in some special cases.

We analyze the performance of these approaches on models of exchange economies where the consumers are equipped with utility functions which are widely used in real world applications.

---

[1]This chapter is based on joint work with Bruno Codenotti, Sriram Pemmaraju, Rajiv Raman, and Kasturi Varadarajan. This work appeared in [25, 26], with the final journal version in [24].

The outcomes of our experiments consistently show that many market settings allow for an efficient computation of the equilibrium, well beyond the restrictions under which the theory provides polynomial time guarantees. For some of the approaches, we also identify models where they are are prone to failure.

In its exchange version, the market equilibrium problem consists of finding prices and allocations of goods to traders such that each trader maximizes her utility function and the market clears (see below for precise definitions). A fundamental result in economic theory states that, under mild assumptions, market clearing prices exist [6].

Soon after this existential result was shown, researchers started analyzing economic processes leading to the equilibrium. The most popular of these is *tâtonnement*, which, starting from an arbitrary price vector, updates it according to the market excess demand generated by such prices [3, 4]. In its continuous version, the tâtonnement process is known to converge [4] whenever the market satisfies *weak gross substitutability* (see below for the definition). However, it need not converge if the market does not satisfy this property (see [1], Chapter 17).

The failure of tâtonnement to provide global convergence stimulated a substantial amount of work on the computation of the equilibrium. Scarf and some coauthors developed pivoting algorithms which search for an equilibrium within the simplex of prices [39, 54, 85, 86]. Unlike tâtonnement, these algorithms always reach a solution, but they lack a clear economic interpretation and they require exponential time, even on certain simple instances.

Motivated by the lack of global convergence of tâtonnement, and by the lack of a clear economic interpretation for Scarf's methods, Smale developed a global Newton's method for the computation of equilibrium prices [90]. His approach provides a price adjustment mechanism which takes into account all the components of the Jacobian of the excess demand functions. However, Smale's technique does not come with polynomial time guarantees, and its behavior, when the current price is far from equilibrium, seems complicated. For this reason, most solvers based on Newton's method, including PATH, the solver used in Section 4.3 and which is available under the popular GAMS framework, do a line search within each Newton's iteration, in order to guarantee that some progress is made even far from equilibrium (see [45, 46, 63]).

A different line of work has attempted to take advantage of the convexity of the set of equilibrium prices in certain exchange markets (see chapter one). For example, in [4] it is shown that when the market satisfies weak gross substitutability, a fundamental inequality holds which defines an infinite collection of hyperplanes that separates equilibrium prices from the rest. A stream of work has extended this characterization to handle settings where the demand need not be a single-valued function of the prices. These settings include in particular the case of linear utility functions (see [74, 78, 79] and the references therein). Some of these papers build upon the characterization above to propose Ellipsoid and cutting-plane algorithms to compute the equilibrium.

There has also been some work on writing the equilibria for certain special exchange economies as solutions to explicit convex programs – Nenakov and Primak

[73] for linear and Cobb-Douglas utilities, and Eaves [38] for Cobb-Douglas utilities.

Another family of computational techniques follow from Negishi's characterization of the market equilibrium as the solution to a welfare maximization problem, where the *welfare function* that is maximized is a linear combination of individual utility functions obtained by using certain positive weights [72]. This characterization transforms the problem of computing equilibrium prices into the problem of computing the appropriate weights of the linear combination mentioned above. For this computation, there is a natural *welfare adjustment* process or *joint maximization* procedure that works in the space of the weights in a manner that is analogous to how the tâtonnement process works, in the space of prices. As a result, this process is convergent under conditions similar to those implying the convergence of the tâtonnement process [65].

Recently, the question of when the market equilibrium problem can be solved in polynomial time has received considerable attention, starting with the work of Deng et al. [99]. The focus has been on isolating restrictive yet important families of markets for which the problem can be solved in polynomial time [20,23,29,49,58,60,69,80,100]. Several techniques have emerged in the process – primal-dual methods, auction-based algorithms, variants of welfare adjustment, tâtonnement, and convex programming formulations. See [22] for a review of this body of work.

This paper aims to complement the flurry of recent theoretical advances in the design of polynomial time algorithms for the market equilibrium problem with an experimental investigation. The specific goal of this paper is to comparatively study four approaches to the problem.

1. The popular software tool based on the modeling language GAMS (short for "General Algebraic Modeling System") and specifically its subsystem MPSGE (short for "Mathematical Programming System for General Equilibrium Analysis"). GAMS/MPSGE is the most commonly used tool for practical applications involving the solution of market equilibrium problems. The solver we used for the market equilibrium problem within the GAMS/MPSGE framework is the Newton-based solver PATH [63].

2. A version of the tâtonnement process. The continuous tâtonnement process converges for markets satisfying weak gross substitutability, and is particularly attractive due to its simplicity. The main question is whether the theoretically well understood continuous tâtonnement process can be turned into a simple discrete algorithm that has good convergence properties.

3. The sequential joint maximization algorithm of [81]. Such an algorithm roughly corresponds to "Algorithm 2" in [60] that computes an approximate equilibrium in an exchange market by iteratively solving a special case of exchange which arises when the initial endowments are collinear (a.k.a. Fisher's model). Algorithm 2 in [60] does not fit perfectly into the framework of sequential joint maximization because it uses an extra fictitious trader.

4. Solving convex programming formulations for the market equilibrium problem for some speical cases [29, 41]. In the experiments reported in this paper, we

use the "convex" option in the general purpose non-linear solver LOQO, in combination with AMPL, its modeling language.

In our experiments, the utility functions of the traders are derived from the family of CES and nested CES functions. (See below for the definitions.) Our main motivation for studying these functions is that they are widely used to model production and consumption [59, 93]. Our main observations are:

1. The PATH solver (used within GAMS) typically converges when applied to CES functions. For certain choices of market types with nested CES functions, however, the PATH solver exhibits a large variance in performance and often fails to converge when applied to exchange economies with nested CES functions.

2. The convex programming approach, which is applicable to a subclass of CES functions [29], compares favorably against PATH and seems to be competitive in terms of scalability.

3. With an appropriately chosen price update rule, the discrete version of tâtonnement performed remarkably well on CES functions and in particular scaled well compared to PATH. On nested CES functions, however, the tâtonnement algorithm also often fails to converge for certain market types, although it generally converges for most market types. The market types on which tâtonnement performs well complement those on which PATH performs well.

4. The welfare adjustment process is almost always convergent on CES exchange economies, where the initial endowment vectors are almost proportional. Even

when the initial endowments are farthest from being proportional, the process converges, although with a significantly larger number of iterations, for all but a few market types.

Theoretical studies of the market equilibrium problem (for the exchange model) have revealed that polynomial time solvability depends very much on the nature of the utility functions and the nature of the initial endowments. In the case of CES functions for example, gross substitutability implies polynomial time solvability whenever the *elasticity of substitution* (see below for the definition) is greater than or equal to one; the existence of convex feasibility formulations implies polynomial time solvability when the elasticity is greater than or equal to $\frac{1}{2}$ [29]; when the elasticity is smaller than $\frac{1}{2}$ we do not know much about polynomial time solvability but we do know that convex feasibility formulations are ruled out because multiple disconnected equilibria can occur. For the extreme case of Leontief utility functions (zero elasticity), we know that polynomial time solvability is unlikely because the problem turns out to be $PPAD$-complete (see [75] for the definition of the complexity class $PPAD$).

In the special case where the initial endowments of the traders happen to be proportional, we have polynomial time solvability for all CES functions due to Eisenberg's convex program [41].

This background led us to the formulation of different input market types, and motivated us to study how the experimental performance of each algorithm varies with the market type.

Prior to this paper, there has been some work analyzing the practical performance of different algorithms for the market equilibrium problem. In [17] the performance of a distributed implementation of tâtonnement is discussed; in [89] several complementarity solvers are implemented and their relative merits analyzed, while in [54] the efficiency of Newton's method is investigated. In [7] an approach based on global minimization is illustrated, and the outcomes of some numerical experiments are reported. More recently, in [43] the performance of interior point methods has been analyzed, and computational data have been obtained for some small scale benchmarks. A common feature of the experiments reported in these works is that the sizes of the problems considered were quite small. To the best of our knowledge, this paper provides the first attempt at an experimental evaluation of different algorithms for large-scale problems.

The rest of this paper is organized as follows. In Section 4.2 we provide the basic definitions, introduce the market models, describe the different types of input data, and the computational settings used for the experiments.

In Section 4.3 we present the results of the experimental work performed using the PATH solver, available under the GAMS/MPSGE system. We identify settings where PATH consistently performs well, and, by contrast, scenarios where the running-time of PATH is less predictable.

In Section 4.4 we analyze some simple price update schemes, which are discrete versions of the tâtonnement process, and show that, for certain market types, they rapidly converge well beyond what the theory predicts.

In Section 4.5 we describe the outcomes of the experiments done using the *sequential joint maximization* algorithm, which is based on Negishi's approach for establishing the existence of the equilibrium [72]. This algorithm seems to converge in a few iterations, even for settings where the theory does not guarantee convergence. However the experiments have been limited to medium size instances, due to the fact that each iteration requires solving a convex program with a large number of variables.

Finally, in Section 4.6 we report on an experimental study of some of the convex-programming based approaches for computing equilibria in various special cases. Here too the experiments on large scale problems have not been possible, for the same reason as above.

4.2 Definitions and Market Models

For the definitions of an exchange economy, market equilibria, and related properties such as gross subsitutability, see the introductory chapter. The CES utility function is central to the experiment and a nested version of CES is also introduced, so the definition is repeated below.

In this chapter, we address the computational problem of finding the equilibrium price vector for an exchange economy given the initial endowment and an appropriate representation of the utility function for each trader.

4.2.1 Utility Functions

An important aspect of our experiments is the generation of markets with enough variety so as to represent a wide range of phenomenon. The utility function of every agent is a generalization of the *constant elasticity of substitution* (CES) functional form. A CES function is a concave function defined as

$$u(x_1, x_2, \ldots, x_n) = \Big(\sum_{j=1}^{n} \alpha_j^{\frac{1}{\sigma}} x_j^{\frac{\sigma-1}{\sigma}} \Big)^{\frac{\sigma}{\sigma-1}},$$

where $\alpha_j \geq 0$ for each j and $\sigma > 0, \sigma \neq 1$. The α_j's and σ are parameters that can be assigned values to obtain different utility functions. The parameter σ represents the *elasticity of substitution*, a natural measure of the curvature of the indifference curves of the utility function. We call an *elastic market* a market where consumers are highly sensitive to price changes. In the case of CES functions, this happens when all the consumers have a utility function with elasticity of substitution at least one. The CES functions range from *linear utility functions* (when $\sigma \to \infty$) that are fully elastic to *Leontief functions* (when $\sigma \to 0$) that are completely inelastic. When the utility function is linear, goods are perfect substitutes and when the utility function is Leontief, goods are perfect complements. In between, when $\sigma \to 1$, CES functions become the *Cobb-Douglas functions* that express a balance between substitution and complementarity effects. While σ models the elasticity of substitution, the α_j's capture how much an agent desires good j. CES functions are ubiquitous in economics literature because of their power to express a wide variety of substitution and complementarity effects as well as their mathematical tractability which allows for explicit computation of the associated demand function.

In our experiments, the traders' utility functions $u(x)$ are chosen from a simple family of nested CES functions that generalize CES functions. Let $u^1(x)$ be a CES function with elasticity σ_b that depends only on the quantity of the first $\lfloor n/3 \rfloor$ goods.

That is,

$$u^1(x_1, x_2, \ldots, x_n) = \Big(\sum_{j=1}^{\lfloor n/3 \rfloor} \alpha_j^{\frac{1}{\sigma_b}} x_j^{\frac{\sigma_b - 1}{\sigma_b}} \Big)^{\frac{\sigma_b}{\sigma_b - 1}},$$

where each $\alpha_j \geq 0$. Similarly, let

$$u^2(x_1, x_2, \ldots, x_n) = \Big(\sum_{j=\lfloor n/3 \rfloor + 1}^{\lfloor 2n/3 \rfloor} \alpha_j^{\frac{1}{\sigma_b}} x_j^{\frac{\sigma_b - 1}{\sigma_b}} \Big)^{\frac{\sigma_b}{\sigma_b - 1}},$$

and

$$u^3(x_1, x_2, \ldots, x_n) = \Big(\sum_{j=\lfloor 2n/3 \rfloor + 1}^{n} \alpha_j^{\frac{1}{\sigma_b}} x_j^{\frac{\sigma_b - 1}{\sigma_b}} \Big)^{\frac{\sigma_b}{\sigma_b - 1}}.$$

Then

$$u(x) = \Big(u^1(x)^{\frac{\sigma_t - 1}{\sigma_t}} + u^2(x)^{\frac{\sigma_t - 1}{\sigma_t}} + u^3(x)^{\frac{\sigma_t - 1}{\sigma_t}} \Big)^{\frac{\sigma_t}{\sigma_t - 1}},$$

for some elasticity σ_t. We refer to $u()$ as a 2-level nested CES function with 3 nests, the *bottom elasticities* being all equal to σ_b, and the *top elasticity* being σ_t. Note that if the bottom and top elasticities are equal, then u becomes a CES function with elasticity $\sigma_b = \sigma_t$.

Nested CES functions are used extensively to model both production and consumption in applied general equilibrium: We refer the reader to the book by Shoven and Whalley [59] for a sense of their pervasiveness. The popular modeling language MPSGE [83] uses nested CES functions to model production and consumption.

4.2.2 Input Generators

Assuming that we use 2-level nested CES functional forms as described above to represent agent's preferences, generating a market corresponds to generating α_j's, σ_t, σ_b, and the endowments for each agent. Let m be the number of agents and n

the number of goods. For $1 \leq i \leq m$, $1 \leq j \leq n$, let α_{ij} denote the coefficient $\alpha_j^{\frac{1}{\sigma_b}}$ of the term $x_j^{\frac{\sigma_b}{\sigma_b - 1}}$ in agent i's utility function. For notational convenience let A denote the $m \times n$ matrix of the α_{ij}'s. We will call this the *desirability matrix* since it represents the distribution of agents extent of desire for different goods. Without loss of generality, we can assume that the α_{ij}'s are normalized so that the entries in each row in A sum to 1. Let W denote the $m \times n$ matrix of endowments. Without loss of generality, we assume that endowments are normalized so that entries in W are are in the range $[0, 1]$ and all column sums are 1. This implies that the total quantity of each good is one. While the σ_t (and σ_b) values for different agents can be different in general, for our experiments we typically assume that these are all identical. Thus generating a market corresponds to generating $m \times n$ matrices A and W and the two values σ_t and σ_b. We generate matrices A and W independently, using several generators we have implemented.

4.2.2.1 Generators for the desirability matrices

We have implemented several generators for the desirability matrix A.

Uniform Generator This constructs matrices A such that each agent's desire is uniformly distributed among the n goods. Specifically, each row in A is chosen independently by first picking uniformly a random vector from $[0, 1]^n$ and then normalizing so that the sum of the numbers in the vector is 1.

Concentrated Generator This constructs matrices in which for $1 \leq i \leq m$, agent i desires a fraction .8 of good $n - i$. That is, $\alpha_{i,n-i} = .8$. Two goods j_1 and j_2

are chosen at random from among the other goods, and their desires are set to 0.1 each. The goods j_1 and j_2 are chosen at random with replacement and so they may be identical in which case the agent's desire for this good is just 0.2.[2] This generator assumes that[3] $m \leq n$.

Sharply Concentrated Generator The desire of agent i is 1 for good i and zero for the remaining goods. This generator assumes that $m \leq n$.

Subset Generator For each agent i, a random subset J_i of the goods with expected size $n/4$ is chosen. Agent i's desire for each good outside J_i is set to 0 and the rest of the mass is distributed uniformly among the goods in J_i. Rows in the matrix are generated independently of each other.

4.2.2.2 Introducing correlation in desires

In the description of the uniform generator and the subset generator above, we mentioned that rows of the desirability matrix are independently generated. We get matrices with richer structure and asymmetry between goods by introducing some dependence. We explored the following type of dependence.

Replicated desires In this case, after the first row of A is generated the remaining rows are generated by simply copying the first row.

---

[2] To ensure the existence of an equilibrium, we sometimes perturb the desirability matrix so that each entry is at least some very small positive number ε.

[3] In most of our experiments, we have $m = n$.

By combining desirability matrices of two types, we can get a new type of desirability matrix. For example, we can generate a desirability matrix A_1 using the sharply concentrated generator, a desirability matrix A_2 using the subset generator, pick a parameter β in the range $[0,1]$ and output $\beta A_1 + (1-\beta)A_2$. We do in fact use such combinations in some our experiments.

4.2.2.3 Generators for the endowment matrices

We have implemented several generators for the endowment matrix that are very similar to the generators for the desirability matrix. We have a uniform generator and the subset generator for endowment matrices that are identical to the corresponding generators for the desirability matrices, except that now columns are independently generated, instead of rows. (Recall that the entries in each column add up to 1.) The sharply concentrated generator for the endowment matrix assumes that $m \geq n$ and gives the entire 1 unit of the j-th good to the j-th trader. The concentrated generator for the endowment matrix also assumes that $m \geq n$. It takes the output of the concentrated generator and perturbs it so that the off-diagonal entries are small positive numbers.

Some dependence can also be introduced among the columns of the endowment matrix with the uniform and subset generators. The main kind of dependence we use is that of replicated columns – having generated the first column, the remaining columns are simply copies of it. Note that this leads to the situation where the endowments of the agents are proportional – the Fisher model.

Finally we may combine the output of two generators, as indicated for the desirability matrix, to get a new type of generator.

4.2.3 Computational Environment

All of our experiments were performed on a machine with an AMD Athlon, 64 bit, 2202.865 Mhz processor, 2GB of RAM, and running Red Hat Linux Release 3, Kernel Version - 2.4.21-15.0.4

- The experiments in Section 4.3 use the mixed-complementarity solver PATH that is available under GAMS, which is a modeling system for mathematical programming problems. GAMS consists of a language compiler integrated with high-performance solvers, and it is tailored for complex, large scale modeling applications (see [10]). GAMS has been extended to GAMS/MPSGE by Rutherford [82] to easily handle economic equilibrium problems. The web site `http://www.gams.com/solvers/mpsge/pubs.htm` lists a number of scientific papers which have been using MPSGE.

- The experiments in Sections 4.5 and 4.6 involve solving convex programs. For this task, we used the general nonlinear solver LOQO [94], which is based on an infeasible primal-dual interior point method, and runs faster on convex programs (for which it can be run with the "convex option" on) than on non-convex ones.

4.3 The Performance of an Efficient General Purpose Solver

In this section, we present the outcomes of the experiments we have carried out with the PATH solver available under GAMS/MPSGE.

PATH is a sophisticated solver, based on Newton's method, which is the most used technique to solve systems of nonlinear equations [36, 46]. Newton's method constructs successive approximations to the solution, and works very well in the proximity of the solution. However, there are no guarantees of progress far from the solution. For this reason, PATH combines Newton's iterations with a line search which makes sure that at each iteration the error bound decreases, by enforcing the decrease of a suitably chosen *merit function*. This line search corresponds to a linear complementarity problem, which PATH solves by using a pivotal method [45].

Our experiments with GAMS/PATH are of two kinds. The first kind aims at understanding the sensitivity of PATH to specific market types and parameter ranges of interest. The second kind studies how the running time scales with input size.

4.3.0.1 Sensitivity to Market Type

Figures 4.1, 4.2, and 4.3 summarize the performance of PATH on economies with 50 traders and goods for three different choices of generators for the desirability and endowment matrices. One phenomenon that clearly stands out is that the performance is quite good when the top elasticity σ_t is less than or equal to the bottom elasticity σ_b. This range includes the special case of CES functions. In contrast, the performance can be significantly worse for some instances where $\sigma_t > \sigma_b$. This phenomenon can be seen both in the running time and in the number of runs in which PATH declares a failure in computing the equilibrium.

The degradation in performance is more striking for some configurations of

	0.1	0.3	0.5	0.9	1.3	1.7
0.1	1.96	1.11	1.11	1.00	1.00	1.03
0.3	52.31	1.67	1.68	1.64	1.33	1.37
0.5	82.08	86.06	1.85	1.75	1.76	1.68
0.9	40.87	99.08	2.15	1.72	1.76	1.76
1.3	85.29	122.69	1.95	1.44	1.40	1.50
1.7	88.73	58.34	61.11	1.36	1.35	1.39

(a)

	0.1	0.3	0.5	0.9, 1.3, 1.7
0.1	0	0	0	0
0.3	5	0	0	0
0.5	3	2	0	0
0.9	5	5	0	0
1.3	5	3	0	0
1.7	4	1	1	0

(b)

Figure 4.1: PATH on markets with 50 traders and goods. The desirability matrix is obtained by adding β times the output of a sharply concentrated generator and $(1-\beta)$ times the output of a subset generator, with $\beta = 0.95$. The endowment matrix is from the sharply concentrated generator. Rows of the table correspond to top elasticity σ_t, and columns to bottom elasticity σ_b. Six values – 0.1, 0.3, 0.5, 0.9, 1.3, and 1.7 – were chosen for these elasticities. (a) Each entry of this table corresponds to a choice of σ_t and σ_b, and the number shown is the average running time in seconds over five inputs. (b) Each entry shows the number of failures out of the five runs.

	0. 1	0.3	0.5	0.9	1.3	1.7
0.1	1.77	1.23	1.23	1.19	1.15	1.07
0.3	110.01	1.84	1.88	1.97	1.90	1.86
0.5	64.38	20.00	2.18	2.01	1.97	2.06
0.9	76.65	43.25	2.29	2.04	1.98	1.99
1.3	2.42	1.82	1.68	1.61	1.68	1.72
1.7	1.62	1.58	1.67	1.64	1.56	1.54

(a)

	0. 1	0.3	0.5, 0.9, 1.3, 1.7
0.1	0	0	0
0.3	5	0	0
0.5	3	1	0
0.9	3	0	0
1.3	0	0	0
1.7	0	0	0

(b)

Figure 4.2: PATH on markets with 50 traders and goods. The desirability and endowment matrices are generated using the concentrated generators.

	0.1	0.3	0.5	0.9	1.3	1.7
0.1	1.25	1.04	1.03	1.04	1.04	1.04
0.3	2.60	1.37	1.24	1.25	1.16	1.08
0.5	2.04	1.58	1.45	1.24	1.25	1.25
0.9	2.85	1.74	1.66	1.58	1.41	1.33
1.3	3.13	1.77	1.91	1.78	1.54	1.49
1.7	3.12	1.84	1.90	1.98	1.66	1.70

(a)

	0.1, 0.3, 0.5, 0.9, 1.3, 1.7
0.1	0
0.3	0
0.5	0
0.9	0
1.3	0
1.7	0

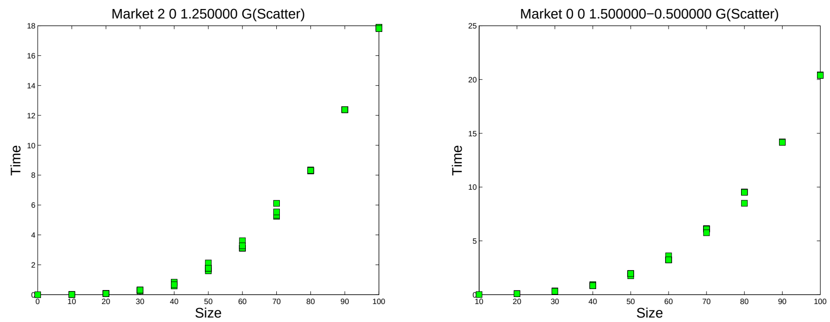
(b)

Figure 4.3: PATH on markets with 50 traders and goods. The desirability and endowment matrices are generated using the uniform generators.

generators than others. For instance, PATH performs much better on the configuration corresponding to Figure 4.3 than on the configuration corresponding to Figure 4.1.

4.3.0.2 Running Time as a Function of Input Size

To get an estimate of how the running time of PATH varies with input size, we experimented with some generator configurations and choices of σ_t and σ_b where PATH does not report failure. Figure 4.4 illustrates how the running time grows fairly



(a) **(b)**

Figure 4.4: The running time of PATH, in seconds, as a function of the input size ($m = n$). (a) The concentrated generator is used for the desirability matrix and the uniform generator for the endowment matrix; $\sigma_t = \sigma_b = 1.25$. (b) The uniform generator is used for both the desirability and endowment matrix; $\sigma_t = 1.5$ and $\sigma_b = 0.5$. There were five runs for each input size.

rapidly with size for two such parameter choices. The runs on other benchmarks (in PATH's good range) yield very similar figures.

4.4 An Algorithm Derived from the Tâtonnement process

In 1874 Léon Walras proposed that an equilibrium price vector could be reached via a discrete price-adjustment process that he called *tâtonnement*. In Samuelson's (1947) now-standard version of tâtonnement, competitive agents receive a price signal, and report their excess demands at these prices to the central auctioneer. The auctioneer then computes aggregate excess demands, adjusts the prices incrementally in proportion to the magnitude of excess demands, and announces the new incrementally adjusted price level. In each round, agents recalculate their excess demands upon receiving the newly adjusted price signal and report these to the auctioneer. The process continues until prices converge to an equilibrium.

In its continuous version, the tâtonnement process is governed by the differential equation system: $\frac{d\pi_i}{dt} = G_i(Z_i(\pi))$ for each $i = 1, 2, \ldots, n$ where $G_i()$ is some continuous and differentiable, sign-preserving function and the derivative of π_i is with respect to time. The continuous version of tâtonnement is more amenable to analysis of convergence, and it is this process that is shown to be convergent by Arrow, Block, and Hurwicz [4] for markets satisfying weak GS.

In our implementation of tâtonnement, the starting price vector is $(1, 1, \ldots, 1)$. Let π^k be the price vector after k iterations (price updates). In iteration $(k+1)$, the algorithm computes the excess demand vector $Z(\pi^k)$ and then updates each price using the rule $\pi_i^{k+1} \leftarrow \pi_i^k + c_{i,k} \cdot Z_i(\pi^k)$. One specific choice of $c_{i,k}$ that we have used in many of our experiments is

$$c_{i,k} = \frac{\pi_i^k}{i \cdot \max_j |Z_j(\pi^k)|}.$$

This choice of $c_{i,k}$ ensures that $|c_{i,k} \cdot Z_i(\pi)| \leq \pi_i^k$ and therefore π continues to remain non-negative. Also noteworthy is the role of i that ensures that the "step size" diminishes as the process (hopefully) approaches the equilibrium.

Our experiments with tâtonnement were of three kinds. In the first, we attempt to understand the sensitivity of its performance, measured in terms of the number of iterations, to the market type. In the second, we study how the performance scales with size. For these two kinds of experiments, we terminated tâtonnement when $\max_i |Z_i(\pi^k)|$ fell below a threshold value of $\epsilon = 10^{-4}$ (success) or when the

	0. 1	0.3	0.5	0.9	1.3	1.7
0.1	0.21	100.00	100.00	0.07	3.96	5.66
0.3	0.14	0.55	75.48	0.06	6.79	2.04
0.5	0.35	0.85	0.45	0.04	5.78	2.68
0.9	0.83	0.12	0.52	0.22	4.10	4.19
1.3	0.59	0.27	1.24	0.54	0.03	0.18
1.7	0.10	0.78	0.67	1.67	0.02	0.01

(a)

	0.1	0.3	0.5	0.9, 1.3, 1.7
0.1	0	5	5	0
0.3	0	0	3	0
0.5	0	0	0	0
0.9	0	0	0	0
1.3	0	0	0	0
1.7	0	0	0	0

(b)

Figure 4.5: Performance of tâtonnement on markets with 50 traders and goods. σ_t varies with the rows and σ_b with the columns. The desirability matrix is obtained by adding β times the output of a sharply concentrated generator, and $(1-\beta)$ times the output of a subset generator, with $\beta = 0.95$. The endowment matrix is from the sharply concentrated generator. (a) The number of iterations, in thousands, averaged over 5 runs. (b) The number of failures out of 5 runs.

number of iterations exceeded $100,000$ (failure), whichever happened first. In the third kind of experiment, where we study how the performance depends on ε, we increased the limit on the number of iterations to 10 million.

	0. 1	0.3	0.5	0.9	1.3	1.7
0.1	0.29	1.27	2.90	1.07	2.64	0.63
0.3	0.10	1.10	0.16	0.60	1.73	1.29
0.5	0.12	1.20	1.85	7.69	0.14	0.48
0.9	0.90	0.10	2.51	2.91	2.70	3.54
1.3	0.50	0.13	0.33	0.21	0.29	0.32
1.7	0.75	0.15	0.21	0.48	1.21	3.41

(a)

	0.1, 0.3, 0.5, 0.9, 1.3, 1.7
0.1	0
0.3	0
0.5	0
0.9	0
1.3	0
1.7	0

(b)

Figure 4.6: Performance of tâtonnement on markets with 50 traders and goods. The desirability and endowment matrices are generated using the concentrated generators.

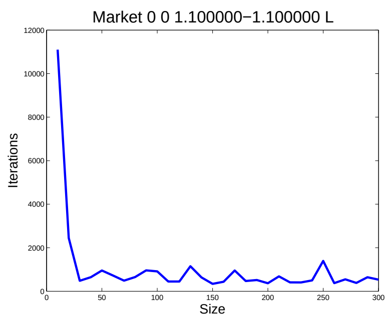
4.4.0.3 Sensitivity to Market Type

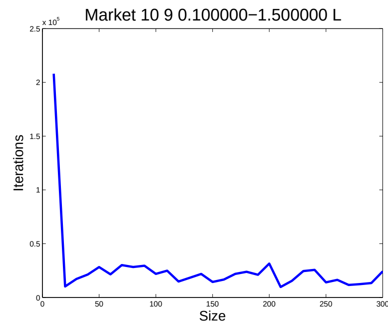
Figures 4.5 and 4.6 illustrate the performance of tâtonnement on economies with 50 traders and goods for two different choices of generators for the desirability and endowment matrices. As in the case of PATH, the choice of generators significantly impacts the performance of tâtonnement. In contrast with what we observed for PATH, tâtonnement performs better on markets with $\sigma_t \geq \sigma_b$ than on markets with $\sigma_t < \sigma_b$. This can be seen in Figure 4.5 in the number of failures as well as the number of iterations used. We observed this phenomenon consistently in generator configurations for which there was a degradation in the performance of tâtonnement.

4.4.0.4 Performance as a function of input size

We measured how the number of iterations taken by the tâtonnement algorithm changes with size for various configurations of the desirability and endowment generators, σ_t and σ_b. As with PATH, we focussed on the parts of the configuration space where tâtonnement does not tend to fail. In these experiments, the input size is the number of traders which equals the number of goods. Quite remarkably, we find that the number of iterations does not grow significantly with input size, but stays nearly flat once the input size is beyond a certain threshold. This phenomenon happens consistently across the input configurations, and is illustrated by the plots shown in Figures 4.7 (a) and (b). The running time, on the other hand, does grow with the input size.

At each iteration of the tâtonnement algorithm, the essential and most expen-





(a) **(b)**

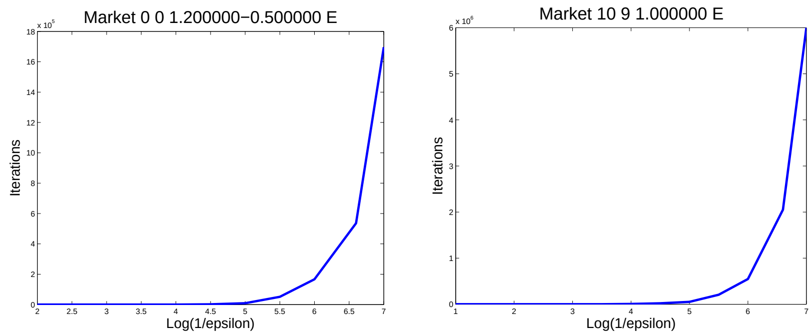
Figure 4.7: The number of iterations of tâtonnement, as a function of the input size , with $m = n$. (a) The uniform generator is used for both the desirability matrix and the endowment matrix; $\sigma_t = \sigma_b = 1.1$. (b) The desirability matrix is obtained by adding β times the output of a sharply concentrated generator and $(1-\beta)$ times the output of a subset generator, with $\beta = 0.95$. The endowment matrix is from the sharply concentrated generator; $\sigma_t = 0.1$ and $\sigma_b = 1.5$. The number of iterations for each input size is averaged over five runs.

sive task is the computation of the demand at the new price. The demand function for traders with CES utilities has an explicit formulation which can be easily derived. If M_i denotes the income of the i-th trader, then her demand for good j is

$$(\frac{\alpha_{ij}}{\pi_j})^\sigma \frac{M_i}{\sum_{1\le j\le n} \alpha_{ij}^\sigma \pi_j^{1-\sigma}}.$$

Using this formula, the aggregate excess demand can be computed in $O(mn)$ time.

The same asymptotic bound applies to the computation of the demand for nested CES functions, when the number of nests is a constant, as was the case in the experiments we conducted.



(a) **(b)**

Figure 4.8: The number of iterations of tâtonnement, as a function of $\log_{10}(1/\varepsilon)$, with $m = n = 50$. (a) The uniform generator is used for both the desirability matrix and the endowment matrix; $\sigma_t = 1.2$ and $\sigma_b = 0.5$. (b) The desirability matrix is obtained by adding β times the output of a sharply concentrated generator and $(1-\beta)$ times the output of a subset generator, with $\beta = 0.95$. The endowment matrix is from the sharply concentrated generator. ; $\sigma_t = \sigma_b = 1.0$. The number of iterations for each input size is averaged over five runs.

4.4.0.5 Performance as a function of ε

The number of iterations of tâtonnement seems to grow rapidly with respect to $\log(\frac{1}{\epsilon})$, quite independently of the market type. Figures 4.8(a) and (b) show typical plots.

4.5 Welfare Adjustment Schemes

In this section we report on some experimental work for computing equilibria in the exchange model using the *sequential joint maximization* algorithm, which is based on Negishi's approach for establishing the existence of the equilibrium [72]. Let R^n_{++} denote the subset of $\mathbf{R}^n$ with all positive coordinates. Let $\alpha = (\alpha_1, \ldots, \alpha_m) \in R^m_{++}$, be any vector. Consider the allocations that solve the following optimization problem over $x_i \in \mathbf{R}^n_+$:

$$\begin{array}{ll} \text{Maximize} & \sum_{i=1}^{m} \alpha_i u_i(x_i) \\ \text{Subject to} & \sum_i x_{ij} \leq \sum_i w_{ij} \text{ for each good } j. \end{array}$$

The optimal allocations $\bar{x}_i$ are called the *Negishi welfare optimum* at the *welfare weights* α_i. Let $\pi = (\pi_1, \ldots, \pi_n) \in \mathbf{R}^n_+$, where the "dual price" π_j is the Lagrangian multiplier associated with the constraint in the program corresponding to the j-th good. Define $B_i(\alpha) = \pi \cdot w_i - \pi \cdot \bar{x}_i$, the budget surplus of the i-th trader at prices π and with allocation $\bar{x}_i$. Define $f_i(\alpha) = B_i(\alpha)/\alpha_i$, and $f(\alpha) = (f_1(\alpha), \ldots, f_m(\alpha))$.

Under some standard assumptions on the utility functions, the following properties hold for the map $f : R^m_{++} \to \mathbf{R}^m$ (see Chapter 7 of the book by Ginsburgh and Waelbroeck [51] for a systematic exposition.)

1. $f(\alpha)$ is single valued, continuous, and differentiable at each $\alpha \in \Re^m_{++}$.

2. $\sum_i \alpha_i f_i(\alpha) = 0$, which corresponds to Walras' Law.

3. For any real $\lambda > 0$, $f(\lambda\alpha) = f(\alpha)$, which is positive homogeneity.

4. There exists an $\alpha^* \in R^m_{++}$ such that $f(\alpha^*) = 0$. The corresponding dual prices constitute an equilibrium for the economy.

Properties (1)-(3) follow from definitions and basic optimization theory; property (4) follows from Negishi's theorem [72]. This characterization suggests an approach for finding an equilibrium by a search in the space of *Negishi weights*. This approach, which is complementary to the traditional price space search, is elaborated in [51, 65, 81]. In particular, Mantel shows [65] that if the utility functions are strictly concave and log-homogeneous, and generate an excess demand that satisfies gross substitutability, then we have $\frac{\partial f_i(\alpha)}{\partial \alpha_i} < 0$ and $\frac{\partial f_j(\alpha)}{\partial \alpha_i} > 0$ for $j \neq i$. This is the analogue of gross substitutability in the "Negishi space." He also shows that a differential welfare-weight adjustment process, which is the equivalent of tâtonnement, converges to the equilibrium in these situations. The related computational methods that work in the space of welfare weights to find an α^* such that $f(\alpha^*) = 0$ are usually called *welfare adjustment* or *joint maximization* methods.

If each u_i is the logarithm of a function that is homogeneous of degree one, which is the case in our experiments[4], then a result of Eisenberg [41] implies that the dual prices corresponding to welfare weights $\alpha = (\alpha_1, \ldots, \alpha_m)$ are precisely the Fisher equilibrium prices for the model in which the traders have incomes $\alpha_1, \ldots, \alpha_m$. The welfare adjustment methods can then be seen as attempting to compute an equilibrium for the exchange economy by iteratively solving Fisher instances. This idea has been explored by Ye [100] for the case of linear utility functions. The second algorithm of Jain, Mahdian, and Saberi [60] may also be seen in this light, although it uses an extra trader and thus does not fit directly into this framework.

We implemented an algorithm for computing the equilibrium for an exchange

[4]The nested CES functions we use are homogeneous of degree one; however, it is easy to verify that replacing each trader's utility function by its logarithm does not change the equilibria.

market that uses an algorithm for the Fisher setting as a black box. The algorithm starts off from an arbitrary initial price π^0, and computes a sequence of prices as follows. Given π^k, the algorithm sets up a Fisher instance by setting the money of each trader to be $e_i^k = \pi^k \cdot w_i$, where w_i is the i-th trader's initial endowment. Let π^{k+1} be the price vector that is the solution of the Fisher instance with incomes $e_1^k, \ldots, e_m^k$. The goods in the Fisher instance are obtained by aggregating the initial endowment w_i of each trader. If π^{k+1} is within a specified tolerance of π^k, we stop and return π^{k+1}; one can show that π^{k+1} must be an approximate equilibrium. Otherwise, we compute π^{k+2} and proceed.

This may be seen as a version of tâtonnement in the Negishi space. We are simply performing the step $e_i^{k+1} \leftarrow e_i^k + e_i^k f_i(e^k)$.

In our implementation of the iterative Fisher algorithm, we stop when the Euclidean distance between the successive prices falls below 0.001 (success) or when the number of iterations exceeds 100 (failure). In our experiments, we studied how the convergence of the iterative Fisher algorithm varied with elasticity of the CES utility functions (we set $\sigma_t = \sigma_b$) and the initial endowments. If the elasticity is greater than 1, then gross substitutability holds, and Mantel's results [65] show that a differential version of welfare adjustment converges to the equilibrium welfare weights. On the other hand, if the initial endowments of the traders are proportional, then Eisenberg's result [41] implies that our iterative Fisher algorithm should terminate in two iterations. Setting $m = n$, we generate the endowment matrix by taking β times the output of the sharply concentrated generator plus $(1 - \beta)$ times the output of a uniform generator with replicated columns (recall that this yields proportional endowments). We varied the parameter β from 0 to 1. Note that when $\beta = 0$, the initial endowments are proportional, whereas when $\beta = 1$, the initial endowments are orthogonal. We varied the elasticity of the utility functions of the traders from 0.1 to 1.3.

Figure 4.9 shows the result of such an experiment when the uniform generator is used for the desirability matrix. The algorithm converges in a very small number of iterations for all elasticity and β values, though the number of iterations tends to be somewhat higher when β equals 1.

	0.0	0.2	0.4	0.6	0.8	1.0
0 .1	2	2	2	2	2	5.2
0.3	2	2	2	2	2	5
0.5	2	2	2	2	2	5
0.7	2	2	2	2	2	4.8
0.9	2	2	2	2	2	4.8
1.1	2	2	2	2	2	4.4
1.3	2	2	2	2	2	4

Figure 4.9: Number of iterations of the iterative Fisher algorithm. The elasticity of the CES functions of the traders varies with the rows; β varies with the columns; each entry is the average number of iterations over 5 runs. We have $m = n = 25$ and the desirability matrix is computed using the uniform generator

Figure 4.10 (a) tabulates the number of iterations where we use the concentrated generator for the desirability matrix. Note that the number of iterations is quite small when $\beta \leq 0.8$, indicating that the iterative Fisher algorithm tends to converge quickly when the initial endowments are even reasonably close to being proportional. The number of iterations is significantly larger when the initial endowments are orthogonal ($\beta = 1$). Even in this case, failure to converge tends to happen only for small elasticity values, as can be seen from Figure 4.10 (b).

These results are quite representative of the behavior we observed over different choices of generators for the desirability matrix. The explanation for these good convergence results should be read in the light of the discussion above, which shows that we are actually performing a welfare adjustment process in the Negishi space.

Ye [100] gives an example with two traders and two goods and linear utilities for which the simple iterative Fisher algorithm described above cycles between two prices (and therefore does not converge). As explained, the iterative update in the above algorithm corresponds to making the update $\alpha_i^{k+1} \leftarrow \alpha_i^k + \alpha_i^k f_i(\alpha^k)$ in the Negishi space. It is conceivable (and probably even provable using the technology in [4]) that a differential version of the above converges to an equilibrium when the utility functions satisfy GS; note that Mantel [65] has shown that a differential version of $\alpha_i^{k+1} \leftarrow \alpha_i^k + f_i(\alpha^k)$ converges to an equilibrium in this situation. In our experiments, we worked with the simple iterative Fisher and not a differential version, motivated by the fact that this is done in practice [81].

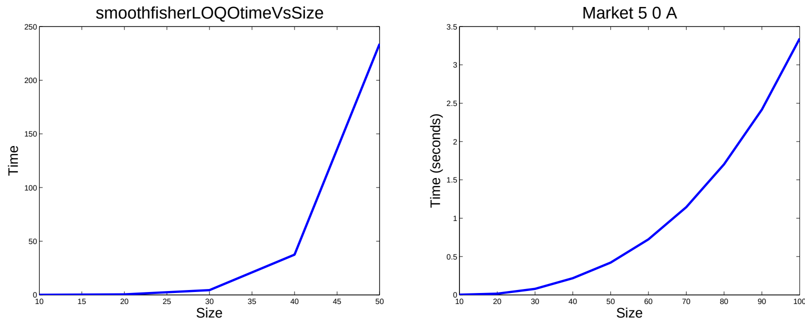
	0.0	0.2	0.4	0.6	0.8	1.0
0 .1	2	2	2.4	3	3	75.2
0.3	2	2	3	3	3	87
0.5	2	2	3	3	3	47.6
0.7	2	2	2.8	3	3	48.6
0.9	2	2	2.4	3	3	46.2
1.1	2	2	2.8	3	3	56.4
1.3	2	2	2.6	3	3	62.4

(a)

	0.0	0.2	0.4	0.6	0.8	1.0
0 .1	0	0	0	0	0	3
0.3	0	0	0	0	0	2
0.5	0	0	0	0	0	0
0.7	0	0	0	0	0	0
0.9	0	0	0	0	0	0
1.1	0	0	0	0	0	0
1.3	0	0	0	0	0	0

(b)

Figure 4.10: The iterative Fisher algorithm when the concentrated generator is used for the desirability matrix; $m = n = 25$. (a) The average number of iterations over 5 runs. (b) The number of failures out of 5 runs.



(a) **(b)**

Figure 4.11: Running time as a function of size for **(a)** the convex program for Fisher instances with $\sigma = 0.25$, and **(b)** for the convex program for exchange instances with $\sigma = 1.25$.

4.6 Explicit Convex Programs

In this section, we report on an experimental study of some of the convex-programming based approaches for computing equilibria in various special cases.

The Fisher Setting. We implemented the convex program of Eisenberg [41] for computing the equilibrium in the Fisher setting when the traders have homogeneous utility functions. The program has $n * m$ variables for a market with m traders and n goods, has n linear constraints, and a concave objective function. Our interest is in measuring how well the running time scales with size. In our experiments, we set $n = m$. Figure 4.11(a) depicts how the average running time varies with n for a a typical run where traders having CES utilities with $\sigma = 0.25$. We did not run the experiments beyond $n = 50$ since the solver LOQO took too long to complete. We suspect that this happens because the number of variables in the program is $n * m$.

Exchange Economies with CES utilities. Codenotti et al. [29] present convex programs that characterize the equilibria in exchange economies where traders have CES functions with elasticity that lies in the range (a) $[1/2, 1)$, and (b) $(1, \infty)$. We implemented a version of their convex program when the elasticities lie in the latter range. This program has $n + m$ variables for a market with m traders and n goods and $n + m$ constraints. In our experiments, all the traders have the same elasticity and $n = m$. We measured how well the running time scales with problem size. Figure 4.11(b), which is quite typical, depicts how the average running time varies with n for $\sigma = 1.25$. Note that the running time compares favorably with that of PATH for the same input configuration (see Figure 4.4 (a)).

We also measured how the running time varies with elasticity. For a market with $n = 25$, we varied the elasticity of substitution from 1 to 20 and found that, beyond a certain point, the running time is stable and increases only very mildly with the elasticity. We experimented with different kinds of markets and found this behavior to be fairly typical.

CHAPTER 5
AGENT HEURISTICS AND PATHS TO NASH EQUILIBIRUM

5.1 Introduction

There has recently been a great deal work investigating the complexity of computing Nash equilibria. Christos Papadimitriou [68] articulates the view that a solution concept must not only be intuitively compelling, but also tractable computationally. Even though the concept of Nash equilibrium is not inherently a computational one, if rational utility maximizing agents are expected to arrive at the equilibrium, one might expect a computer could compute the equilibrium efficiently. In the words of Papadimitriou, "Efficient computability is an important modeling prerequisite for solution concepts."

One could go even further than Papadimitriou since a solution could potentially be computed efficiently, but only with information unavailable to agents within the game. If a solution cannot be computed based on the information available to agents, it might be thought unlikely to be a valid prediction of behavior. In general, the problem with the concept of Nash equilibrium is that it is not clear how agents will arrive at it.

In 2001, Papadimitriou placed the computation of Nash Equilibrium as one of the central problems in theoretical computer science. It has been shown by Chen and Deng [16] that even in the two player case, the problem of computing a Nash equilibrium is PPAD-complete. Some problems [61] in PPAD("Polynomial Parity

Arguments on Directed Graphs") [75], such as computing Brouwer fixed points and finding an n-player Nash equilibrium are widely thought to be intractable. Therefore, it is strongly suspected that $PPAD \neq P$. For a precise definition of PPAD, see [31,75].

This chapter uses simulations of a restricted form of Two Player Texas Hold'em to show that in this setting, the Nash equilibrium is a reasonable long run expectation for player behavior. If players update their strategies based on simple adaptations to their performance with hands up until that point, they find themselves near the Nash equilibrium after they play many hands.

5.2 Two Player Push-Fold No Limit Holdem

Models of poker have been studied in conjuction with game theory from the very beginning [9,44,96]. These models have tended to be very simple approximations of actual poker games. More recently, game theoretic ideas have been applied to improve artificial intelligence for actual poker games, specifically two player Texas Hold'em (e.g. [50]). Chen and Ankenman analyze several simplified models of poker [13].

The model we wish to examine in detail is two player push-fold No Limit Holdem. This game is quite close to the actual game of Texas Holdem late in tournaments where the blinds are very large relative to the size of the players' chip stacks. With deeper stacks, players would want to take advantage of later betting rounds.

I follow Chen and Ankemenan's description of the game [13]. Each player has a stack of size S. The first player to act, the *pusher*, or the *attacker* pays a small blind (forced bet) of 0.5 units. The second player to act, the *caller*, or the *defender*, pays a big blind of 1 unit. Both players receive two private "hole" cards. At this point, the attacker can "push", that is, bet the remainder of her chips or fold. If the attacker folds then the defender wins the 1.5 unit pot. If the attacker pushes, then the defender can call or fold. If the defender folds then the attacker wins the pot. If the defender decides to call the attacker's push then five community cards are dealt out and we have a showdown where both players display their cards. Both the attacker and the defender can use their hole cards and the community cards to construct the best possible five card poker hand. The winning hand receives the entire pot of $2S$ units. In case of a tie, the pot is split evenly. The Push-Fold game can be seen graphically in Figure 5.1 with an example showdown in Figure 5.2.

Suppose the attacker pushes all in. If the defender knew her opponent's hole cards, the decision to call or fold would be a straightforward calculation. The defender would want to call if calling had a greater expected value in chips gained than folding. The defender can calculate her probability p of winning with her hole cards versus the attacker's hole cards. She will want to call if $(S+1)p-(S-1)(1-p)>0$. A simple manipulation shows that she will call if $p>\frac{S-1}{2S}$. One can see that p approaches $1/2$ as S grows.

In a real game, the defender will not know the attacker's cards and thus not know the probability p of winning the hand. If instead, she knew the attacker's pushing strategy then she could calculate her chance of winning against the attacker's range of hands. There are 1326 possible hands. If one played each hand the same way

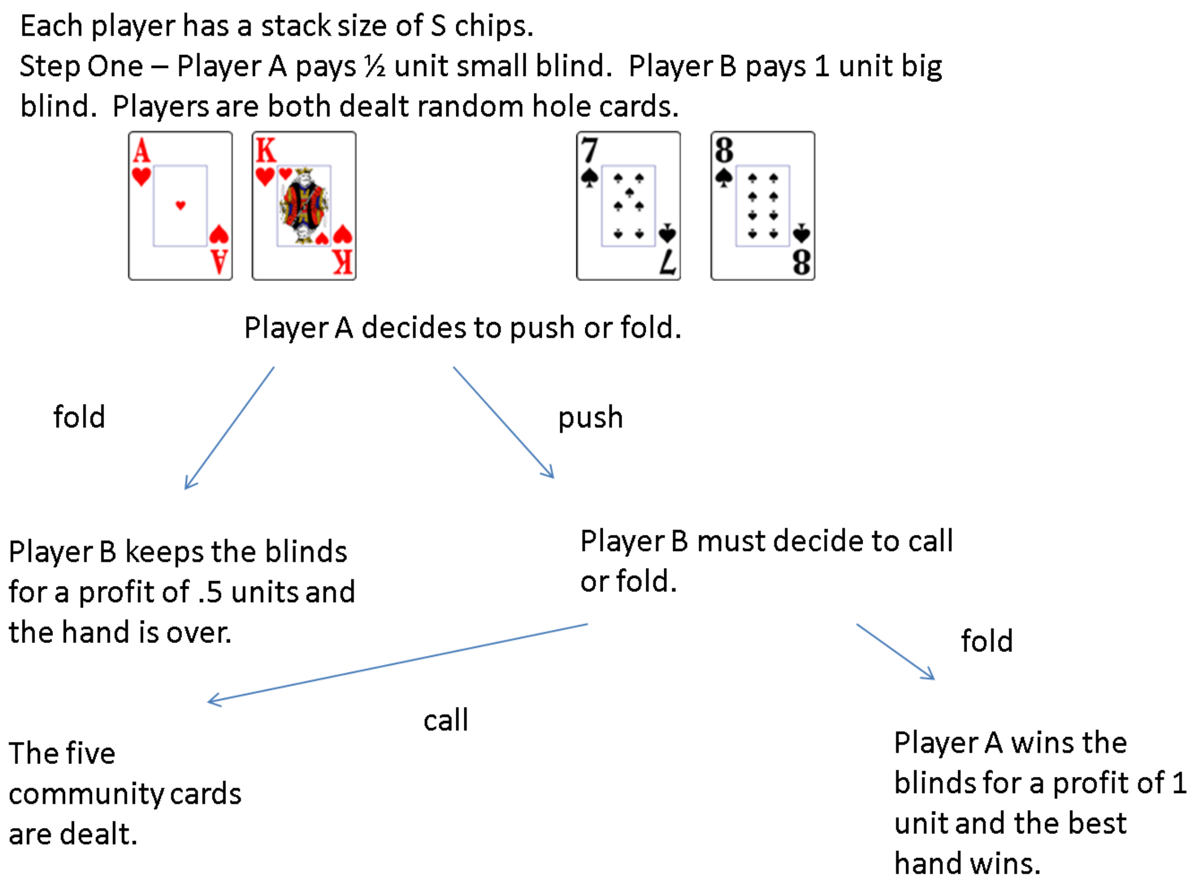


Figure 5.1: Diagram of the Push Fold Game

every time, then a strategy could be represented as a vctor in $\{0,1\}^{1326}$. If a player wished to randomize their behavior (e.g. push (K hearts, 2 clubs) 85% of the time and fold it 15% of the time) and use a mixed strategy, then their strategy would be represented by a vector in $[0,1]^{1326}$.

Some hands are strategically equivalent though - there is no reason to treat (7 clubs, 8 hearts) any differently than (7 spades, 8 diamonds). One would treat suited hands like (7 spades, 8 spades) differently since they have a greater chance of winning

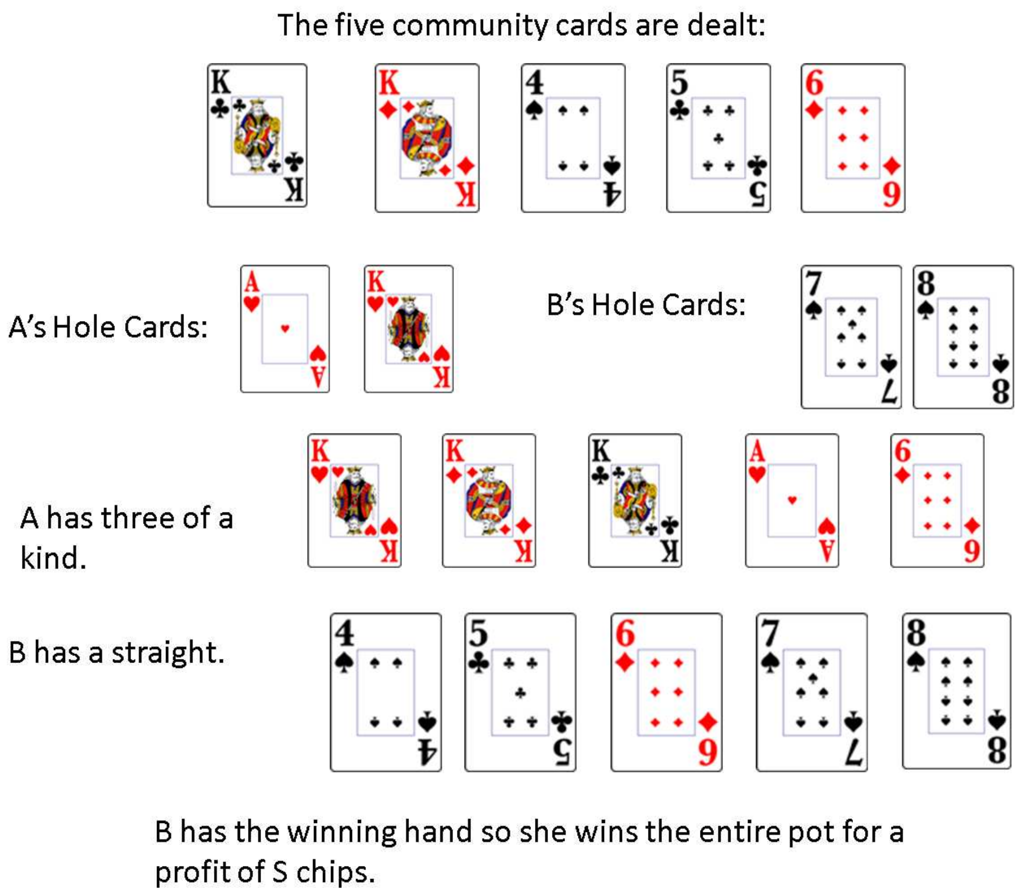


Figure 5.2: Showdown when Player A pushes and Player B calls

by getting a flush. There are only 169 strategically different hands. This includes 13 pairs, $13 * 12/2$ suited hands, and $13 * 12/2$ non suited hands. A pushing (or calling) strategy can thus be represented by vector $x \in [0, 1]^{169}$ where x_i represents the probability of pushing (or calling) with the i^{th} hand. Typically, a strategy will dictate that a player always play or always fold a given hand so most recommended strategies have the overwhelming majority of components either 0 or 1.

Some hands are clearly superior to other hands. For example, pocket aces are superior to pocket kings since AA has a higher probability of winning against every hand than KK does. Thus there will never be any rational strategy where one will call with KK, but fold AA. One can not put a complete ordering of hands in this manner though since the relation is not transitive. 22 is a favorite against Ace-King offsuit and Ace-King offsuit is a strong favorite against Jack-Ten suited, but Jack-Ten suited is a favorite against 22.

5.2.1 Equilibrium Strategy

If x is a strategy for player 1, and y is the strategy for player 2 then we can calculate $E(x, y)$, the expected gain in chips for player 1. There is randomness in both what hole cards the players will receive and in who the winner of the hand is, but the probability distributions are known. We know the probability of receiving a certain hand and can calculate the probability that one specific set of hole cards will defeat another specific set of hole cards. Therefore, once the strategies are given, the expected gain in chips for each player can be calculated. In some cases, one might want to calculate the conditional expected value of playing a particular hand (e.g. AK offsuit) against an opponent's strategy.

Chen and Ankeman were the first to publish the Nash Equilibrium or "optimal" strategies for the two player push-fold game [13]. They published a table that gives the equilibrium strategy for a game with stack sizes of less than 50 big blinds. This is a strategy where each player is playing a best response to the other player and neither has any reason to deviate. If x is a strategy for player 1, and y is the strategy for player 2, we represent the expected gain in chips for player 1 by $E(x, y)$. The expected gain for player 2 is necessarily $-E(x, y)$.

Definition 5.2.1 *Strategies x^* and y^* are equilibrium strategies if and only if $E(x^*, y^*) \geq E(x, y^*)$ for all $x \in [0, 1]^{169}$ and $-E(x^*, y^*) \geq -E(x^*, y)$ for all $y \in [0, 1]^{169}$*

As mentioned earlier, many Texas Holdem tournaments essentially become the 2 player push-fold game at the end due to the stack sizes becoming small relative to the blinds thus precluding multiple betting rounds. Empirically, it is quite clear that most participants do not employ the equilibrium strategy. The following strategies are attempts to model sensible strategies. In fact, these strategies though in some sense simple, are likely superior to many players because they will have access to perfect memory of previous hands and can perform fast and accurate calculations of probablities. Because this is a zero-sum game, a more tractable problem than the general case, there is some hope that some of these strategies might lead to a Nash equilibrium.

5.2.2 Results Oriented Strategy

The first strategy is what I will interchangably call *basic adapative* or the *results oriented strategy* (ROS). In this strategy, the player is wholly ignorant of poker strategy and the value of poker hands. The player only sees the results, the change in the size of her chipstack after playing or folding a hand. The player begins by pushing (or calling) every hand with probability .5. The player keeps track of how successful the strategy is with each hand. The vector $z \in [0,1]^{169}$ stores this information with z_i being equal to the total profitability of hand i when it was pushed (or called) with compared to what would've happened if the hand had not been played. For example, suppose the pusher has pushed J8 offsuit ten times and lost a total of 3 chips. If the pusher had folded each time, he would have lost $10 \times .5 = 5$ chips, thus $z_i = -3 - (-5) = 2$ for i corresponding to J8 offsuit. There will be a separate vector for calling and folding. The strategy vector x will be determined by z. Specifically, we set $x_i(z_i)$ to be the sigmoid function, a special case of the logistic function:

$$x_i(z_i) = \frac{1}{(1 + e^{-\alpha z_i})}$$

with α being a sensitivity paramter. As the z_i diverge from 0, the x_i will tend towards 0 or 1. This is desirable since the equilibrium strategy for all stack sizes is a pure strategy with the exception of one or two hands. Larger values of α would cause faster convergence towards a pure strategy but could be less likely to converge to a best response against their opponent. This is because a fluke early result (e.g. losing twice with pocket aces) could have the effect of a player never playing a hand that has a very high expected value.

The player has no knowledge of poker hands, but has perfect memory on the success of hands that have been played so far. There will be separate x vectors for guiding pushing and calling decisions, x^p and x^c.

5.2.3 Odds Aware Strategy

The *odds aware strategy* (OAS) is similar to the ROS. The player begins by playing every hand with .5 probability. The difference is that when there is a push and a call, she sees the other players cards' and can determine the probability of winning the hand. She can thus determine the probablity that she will win that hand. This probability can be used to determine the expected gain in chips and this expected gain in chips is used in the average for z_i rather than the actual profit from the given hand.

This strategy is best illustrated with an example. Each Player has a stack size of ten units. Suppose that on the first hand, the first player pushes with (2h, 2c) and player two calls with (Ah, Kh). The community cards are then dealt out and they are (2s, 2d, Th, Jh, Qh) so Player 1 has her four of a kind 2's (2h, 2c, 2s, 2d, Qh) defeated by Player 2's Royal Flush (Th, Jh, Qh, Kh, Ah). Under ROS, Player 1 has a profit of -10 which will result in a $z_i = -9.5$. If player 1 had been using the OAS, they would realize that the community cards could have been different and that over all possible community cards, (2h, 2c) will defeat (Ah, Kh) 855,521 times, lose 845,329 times, and tie 11,454 times for an overall pot equity of 50.3%. In OAS, the pushing player would have a $z_i = (.503 \times 20) - 10 - (-.5) = .56$. The OAS player would be expected to move more smoothly in the equilibrium direction since they would be less prone to being influenced by uncommon random events.

5.3 Simulations

The simulations consist of repeated plays of the push-fold game with the stack sizes reset to a constant at the beginning of each hand. The majority of experiments were conducted with a stack size of 10. This is due to the fact that the equilibrium strategies for stack size 10 are more interesting. If stack sizes are very large (say 1000 units), then the equilibrium strategy would simply have the pusher and caller playing only when dealt two pocket aces. If the stack size was extremely small like 2 units, then the equilibrium play would be for the pusher to push almost every hand and the caller call every hand. With a stack size of ten, the equilibrium pusher will be pushing roughly half the time.

5.3.1 Measures of Distance

There are multiple ways one can measure the effectiveness of an adaptive strategy. One can look at the Euclidean distance of the strategy from the equilibrium strategy and how that changes over time. One problem with that method is that not all hands and their associated push-fold decisions carry equal monetary weight for the player. For some hands, there is very little difference in expected gain in chips between pushing and folding. If a strategy differs from the equilibrium strategy only on these marginal hands, this strategy will be very close to a best response despite being some distance away in 169 dimension Euclidean space.

I present graphs where we see both the Euclidean distance from equilibrium and the expected gain in chips for a strategy compared to the expected gain in chips for a strategy that is a best response to the opponent's strategy. Because the results for the pusher and caller are largely similar, I present graphs only for the pushing player.

The first simulation has two basic adapative or ROS players playing against one another. Initially, the pusher pushes every hand with probability .5 and the caller calls every hand with probability .5. They then adjust based on their results according the ROS update rule described above with each player utilizing an $\alpha = .01$.

You can see in figure 5.3 that the player moves in the direction of the equilibrium strategy, but that progress slows down rather quickly. After 2 million hands, the distance to the equilibrium vector has been cut in half, but there is still a rather large gap. It turns out that most of the gap is indeed produced by hands where the player is nearly indifferent between pushing or folding. The hands where the player disagrees with the equlibrium play tend to have an expected value of near -.5 big blinds, the exact expected value one would get for folding. The strategy for really obvious hands with expected values far from -.5 tended to be close to the equilibrium strategy.

In figure 5.4, you can see the evolution of the expected value of the pusher's strategy against the caller's actual strategy. This is compared with the expected value of a perfectly exploitative strategy against the caller's strategy. The gap between the two starts out extremely large, but eventually converges to the point where the expected value of the pusher's strategy is within 1 big blind per 100 hands of a best response to his opponent's actual strategy. The caller's strategy is also close to a best

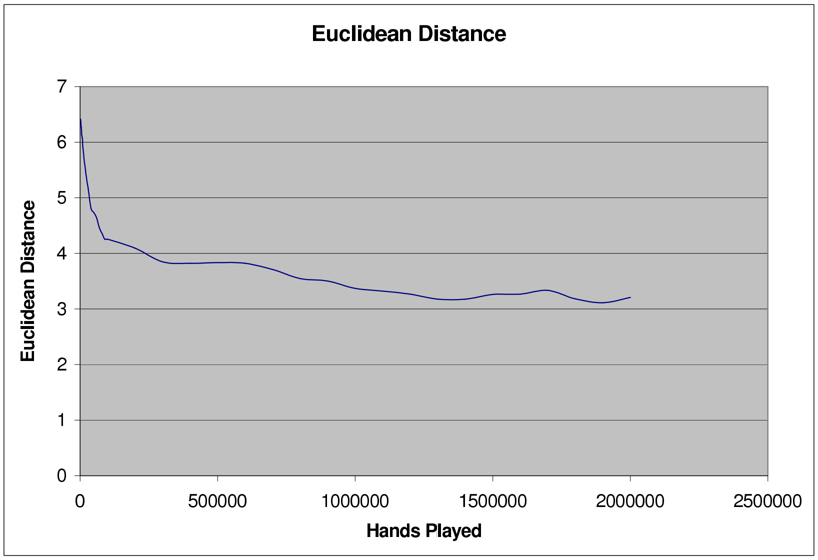


Figure 5.3: Euclidean Distance of Results Oriented Pusher from Nash Equilibrium pushing strategy while playing a Results Oriented Caller for 2 million hands. Both players had a stack size of 10 big blinds.

response. This is the case despite never knowing the opponent's actual strategy and only adapting based on the results of previous hands. Convergence is not quick, but this gives some support to the notion of Nash Equilibrium as a prediction of behavior in the context of the push-fold game.

5.3.2 Sensitivity Parameter

For the basic adaptive or the results oriented strategy, the sensitivity parameter α for the update function is important. I found that in practice, .01 was an effective value. It could be argued that convergence is too slow and it takes an unreasonable number of hands to converge, but the results in the previous section show that we do indeed see convergence to an effective strategy, a near equilibrium. If we

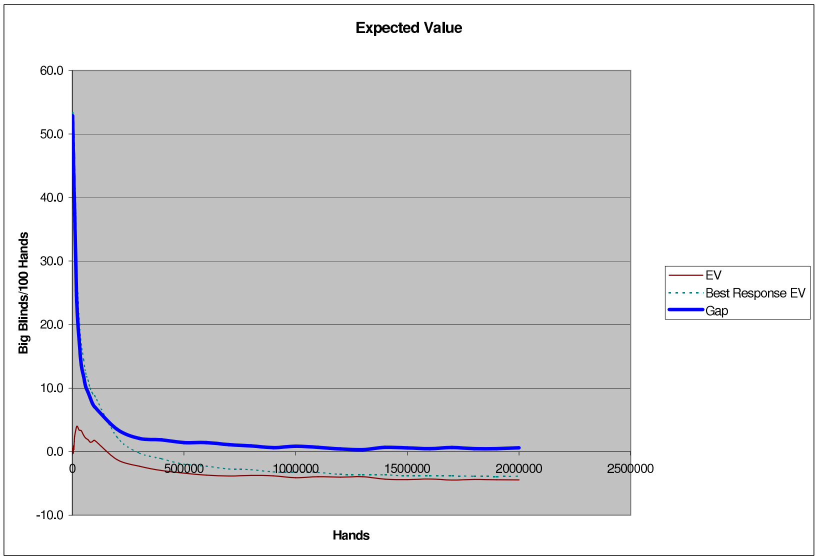


Figure 5.4: Expected Value of ROS Pusher against ROS Caller compared to the Expected Value of a best response to that same caller.

up the sensitivity parameter to .1 then we do see an initial improvement in convergence speed, but the strategy never gets very close to being within 1 big blind/100 hands of a best response even after two million hands. If we up α all the way to 1, we don't go anywhere near equilibrium. In fact, in this simulation, the Euclidean distance actually *increases* from 6.5 to 8.1 and the gap in expected value stays above 20 big blinds per 100 hands. This can be seen in Figure 5.5.

5.3.3 Different Update Rules

In this section, I compare and contrast the performance of the basic adaptive or results-oriented strategy (ROS)to the odds-aware strategy (OAS). We see, as one would expect, somewhat faster convergence to near equilibrium. We run both ROS and OAS pushers against an equilibrium caller. Both players converge towards equi-

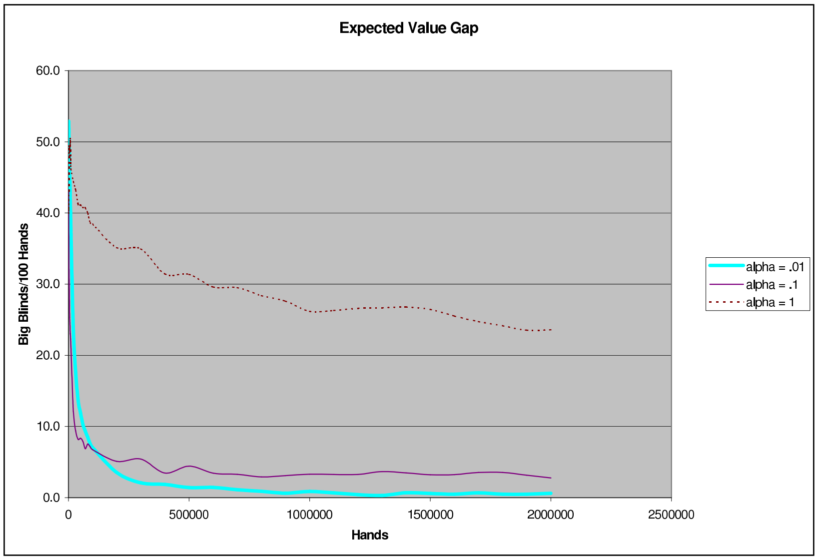


Figure 5.5: Three Simulations with ROS pushers and callers. Graph shows the gap in expected value between a best response and the actual strategy for three different values of the sensitivity parameter.

librium, but the OAS is somewhat faster. The OAS is within 1 big blind per 100 hands after 500,000 hands. The ROS takes 1 million hands to get to that level Figure 5.6 shows the ratio of the OAS gap (difference in strategy expected value to best response expected value) to the ROS gap. As you can see, the OAS is superior, but the difference is not overwhelming.

5.3.4 Different Stack Sizes

All prevous experiments were done with a stack size of 10. As argued previously, this stack size makes for a more interesting equilibrium strategy for the push fold game. This section examines simulations with stack sizes of 5 and 40. At equilibrium, players with stack size of 5 will play most hands equilibrium players with

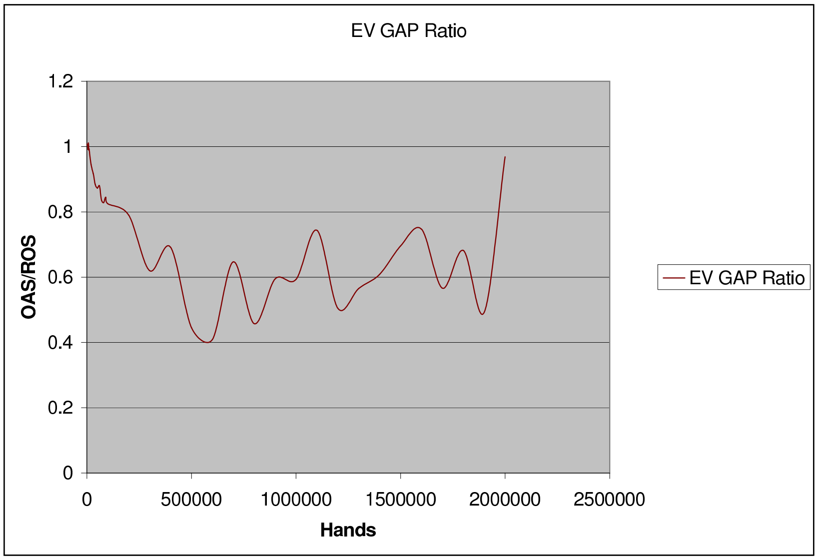


Figure 5.6: Graph shows the ratio of an Odds Aware player's expected value gap to the Results Oriented player's expected value gap.

stack size 40 will play very few hands. Also, the equilibrium play of Push-Fold players with a stack size of 40 will bear less resemblance to optimal play in full blown Texas Hold'em because those players would have more room for betting in later betting rounds. Interestingly, with a stack size of 5, the pusher has a slight advantage in expected value at the Nash equilibrium. At size 10, the pusher has a small disadvantage. At size 40, the pusher has an extreme disadvantage with the Nash equilibrium resulting in the pusher losing an expected 30 big blinds per 100 hands.

The results of the experiments largely confirm the convergence results with stack size ten. The expected value gap for stack size 40 naturally starts out much larger since there are more big blinds to win, but in the end there is convergence so that the gap between the resulting strategy and the best response is close to 1 big blind per 100 hands. We find slightly better convergence in the case of stack size equal to 5. This can be seen in Figure 5.7. Note that the graph begins after 50,000 hands so some of the differences can be seen. In the full graph, the initial gap for stack size equal to 40 was over 150 big blinds per 100 hands and it was difficult to distinguish the smaller differences at the end of the figure.

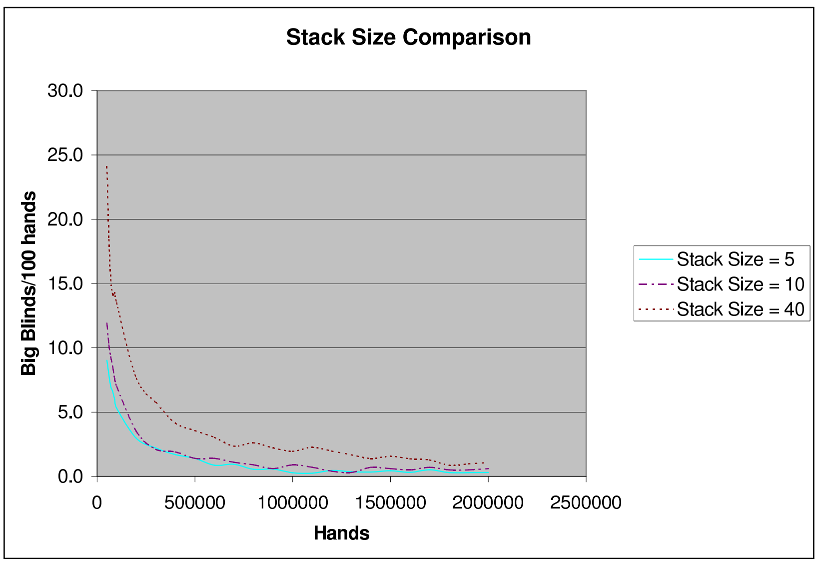


Figure 5.7: Simulations for three ROS players with differing stack sizes. The graph shows the gap in expected value between the actual strategy and a best response.

5.4 Discussion

The simulations in this chapter lend some credence to the concept of Nash equilibrium as a prediction of the behavior of actors with imperfect information. Despite a lack of theoretical knowledge of the game and no knowledge of the strategies of their opponents, players adjusting their play through a fairly simple heuristic based on past performance were able to eventually converge to nearly a best response versus their opponent's actual strategy.

CHAPTER 6
SKILL VS. CHANCE IN THE POKER TOURNAMENT ECONOMY - A MONTE CARLO SIMULATION

6.1 Introduction

Poker has long been studied in game theoretic and artificial intelligence circles [9, 30, 44, 96]. Poker has also been examined in some depth by legal scholars [12]. A central question in legal circles is whether poker should be considered a game of skill or a game of chance as American law treats such games differently. The question has also been examined scientifically. Noga Alon analyzes some toy models of Texas Hold'em and applies the Central Limit Theorem to a long sequence of hands to conclude that "poker is predominantly a game of skill." [2] Psychologists have shown that even a modest amount of instruction can improve the results of inexperienced players [64]. The role of skill is assumed to be quite central within the artificial intelligence and poker instructional literature.

One area of poker in which luck is thought by many to be most prominent is tournament poker. Most poker is played in cash or "ring" games with chips that directly represent money. A player can leave the game at any time and take her chips with her. In a tournament, the players continue to play until they have all the chips or no chips. Players are rewarded based on where they finished in the tournament, typically with only the top ten percent receiving any money at all. Most tournament payouts are very top heavy with the top few places receiving much more than anybody else.

In cash games, it is fairly straightforward to determine if one was is a statistically significant winner. One first calculates their historical win rate in big blinds per hand. Then, one can calculate the standard deviation and find a confidence interval for a player's win rate. This can be done easily because the distribution of hand outcomes is not that far from normal [13] and one can accumulate a large sample size relatively quickly and the central limit theorem applies.

We can illustrate this with an example. Suppose a player plays 2,000 hands of poker and wins 50 big blinds during that time. Thus, we have a mean of $\mu_{2000} = 2.5$ big blinds per 100 hands. Suppose the standard deviation is $\sigma = 15$ per 100 hands. This gives a standard error of $SE = \frac{15}{\sqrt{20}} = 3.35$. This gives a 95 percent confidence interval on the player's true win rate of roughly $[2.5 - 1.96 * 3.35, 2.5 + 1.96] = [-4.04, 9.04]$. If the player had instead played 20,000 hands with the same observed win rate and standard deviation, the confidence interval would instead be $[.42, 4.58]$ and they could say they were a statistically signicant winning player.

For tournaments, the main problem in applying this approach is that in addition to the very high variance, the distribution of pay outcomes is highly skewed. Also, the distribution can vary from tournament to tournament and an entire tournament takes a much longer time than a single hand of poker so it takes a long time to get a significant sample size. If one plays very large live tournaments, then a lifetime is not long enough for this statistical approach to have much merit.

The most prominent poker tournament in the world is the annual World Series of Poker (WSOP) Main Event in Las Vegas. This event has a ten thousand dollar buyin and routinely draws thousands of players. From 2002 to 2007 in the largest fields of players seen, this tournament has been won by an amateur hitherto unkown to the poker world. This has added to the popular notion that poker tournament results are mostly a result of chance.

This chapter models poker tournaments and the resulting poker economy using a monte carlo simulation. I am therefore able to offer quantitative estimates on the contribution of luck versus skill over time. I will also examine the role of differing poker bankroll strategies and how they can effect long term player profit. Another question of interest is whether it is possible for high stakes tournaments to arise if there is no skill and players are limited in the size of their initial poker bankrolls.

6.2 Tournament Model

In an actual poker tournament, players are randomly assigned tables (typically with nine players). Players will play poker hands and whenever a player loses all of their chips, they are forced to exit the tournament. The blinds (forced bets) rise as time goes on forcing more and more players to risk all of their chips. As players are eliminated, the remaining players are consolidated into fewer tables. Eventually, one player will have all of the chips and is declared the winner. The other players are paid out according to what place they finished in with most player receiving nothing.

To model poker tournaments, I make no attempt to model the play of individual hands. Instead, I make the assumption that when one holds the skill of opponents constant, a player's probability of doubling the size of their chip stack is constant.

This is a simplifiying assumption that is not exactly correct since generally more (and different) skill can be applied in the early stages of a tournament when stack sizes are large relative to the blinds, but this effect is likely quite small in practice. In fact, this assumption is weaker than in [13] where in their "Theory of Doubling Up" they assume that the probability of doubling up is constant throughout the tournament regardless of a change in the skill level of opponents during the course of the tournament. The key to this approach is that the details of the mechanism for doubling one's chip stack are not important - it is only relevant how likely one is to double their chip stack.

A player's skill will be represented by an integer $P \in [40, 60]$. Player skill will be assumed to be binomial distribution centered around 50 so the probability that a player has skill equal to P is simply $\binom{20}{P-40}(\frac{1}{2})^{20}$. When two players are in an all-in confontation, the percentage chance of player i defeating player j is $50 + P_i - P_j$. The value of P is therefore the probability (in percentage terms) that the player defeats an average player. Given the nature of the most prominent form of poker, Texas Holdem, and the probabilities of the hands going up against each other, 40 and 60 seem to be at the boundary of the skill edge that is possible. Even simple strategies such as going all in every hand could not be at a much more extreme disadvantage against expert play than what is assumed here.

In a modelled tournament, two players are chosen at random and forced into an all-in confrontation. This process is iterated until one player has all of the chips and is declared the winner. Players will then be paid out cash based on the order of where they finish with at least ten percent of players receiving money and the majority of the money going to the top few finishers. Each player will have to pay a fixed amount of money, or a *buyin* (BI) in order to play the tournament. The total prizepool consists of the sum of all player buyins. This study uses the published tournament payout structures from PokerStars.com, the largest online poker site [56].

6.3 Poker Economy Model

I propose that an initial pool of thousands players each deposit one hundred dollars into their poker bankroll B. Players can have different tolerance for risk of ruin ($B = 0$) and will choose a tournament with a buy in appropriate to that level of risk tolerance R. Even skilled players with a positive expectation in tournaments will have some significant risk of ruin if they are too agressive with their bankroll management. Players will play the largest buyin (BI) tournament available such that $\frac{B}{BI} > R$. For example, suppose a player has a bankroll of $B = 150$ dollars, an $R = 50$, and available tournaments with buyins of 1, 2, and 10 dollars. Since $\frac{150}{10} = 15 < 50 = R$, the player will not play a ten dollar tourney. But $\frac{150}{2} = 75 > 50 = R$, so the player can play in the 2 dollar tournament. I will assume players will always be willing to play in 1 dollar tournaments if they have a bankroll of at least one dollar.

At each point in time, every player will play in a tournament by this buyin rule. When players win, they place the winnings in their bankroll so players with growing bankrolls will increase the stakes of the tournaments that they choose to play. At each point in time, there will a record of how profitable each player has been. When players run out of money, they will redeposit one hundred dollars into their bankroll. In this model, the casino does not take any money out of the prize pool.

There are other variations one might explore on the basic model that go beyond the scope of this chapter. Profitable players might also withdraw winnings from their bankroll at different rates. Losing players might have different chances of quitting or redepositing when they go broke and this could have an effect on the overall average skill levels even without any player learning and improvement. One could also look at the effect of new players joining at varying rates.

6.4 Experiments

We wish to determine how much of poker success is determined by skill. There have been a wide variety of arguments that skill is a portion of poker success - many players would argue that that point hardly needs an argument. The degree to which success is determined by skill or chance has not been explored and is a significant open question. This chapter assumes the existence of skill and addresses the degree to which skill determines player outcomes in poker tournaments.

The fundamental approach of this chapter is to have each player play thousands of tournaments so that we have a relatively large sample. Then we can look at the correlation between player skill and player profit. One would expect the correlation to increase with the number of tournaments played. We then do a plot with the number of tournaments played on the x-axis and the variation in in profitability explained by skill (the r^2) y axis.

The first experiment consists of a poker economy of 5000 players who all deposit 100 dollars into their poker accounts. The available buyins for tournaments are 1, 2, 5, 10, 20, 30, 50, 100, 200, 500, and 1000. All players have the same risk tolerance or bankroll strategy ($R = 50$) and will play the highest buyin tournament for which they can afford more than 50 buyins. A player will always be willing to play the 1 dollar buyin tournament. If they cannot afford the 1 dollar tournament, they will deposit 50 more dollars into their poker account. The players each play a total of 5000 tournaments.

At first, all 5000 players play in a 1 dollar tournament. As time progresses, losing players run out of money and redeposit, thus growing the poker economy. After 5000 tournaments, there is a total of 4,896,300 dollars in the economy for an average deposit of nearly 1,000 dollars. Winning players move up to larger stakes. After 5000 tournaments, there are players playing at every stake from 1 to 1000, with fewer and more skilled players as you move up the stakes. You can see the effict in table 6.1. There are 2970 players with an average skill of 49.1 playing 1 dollar tournaments. There are only 12 players with an average skill level of 56.6 playing the 1000 dollar tournaments.

I then examined how much of the variation in profitability of the players is explained by their skill level P. I used an ordinary least squares regression on P, P^2, P^3, P^4, P^5, and 10^P. When only a few tournaments have been played, the model is almost useless and very little of the variation in player profit is explained. After a

Buyin	Players	Average Skill
1	2970	49.1
2	783	50.4
5	462	51.0
10	297	51.8
20	121	52.4
30	116	53.1
50	103	53.4
100	65	54
200	55	54.9
500	16	55.6
1000	12	56.6

Table 6.1: Buyin Table

few hundred tournaments, a significant amount of the variation is explained by the model. After a few thousand, the majority of the profit variance is explained and in many cases, all of the variables are highly significant. This experiment is best summarized by figure 6.1. As you can see, the trend is clear, but the graph is not monotonic. This was typical - there were often stretches of a few hundred tournaments where odd results from skilled players getting unlucky slightly decreased the explained variation.

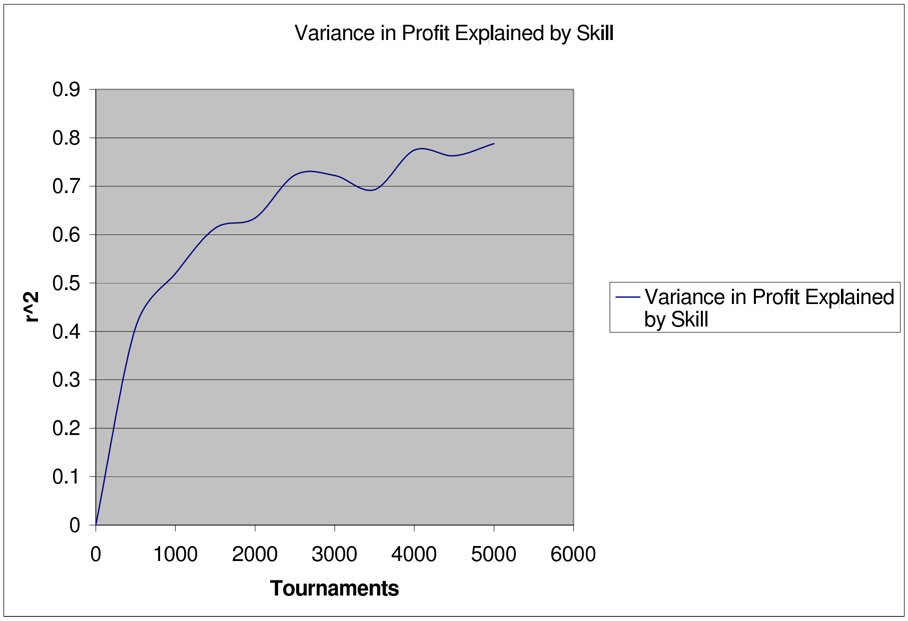


Figure 6.1: The variance explained by skill as a function of the number of tournaments played by 5000 players.

6.4.1 A No Skill Poker Tournament Economy

I ran another simulation that was identical except that there was no skill. All 5000 players were given a skill of 50. The size of the economy was similar to the scenario with skilled players with the average deposit being 957 dollars as compared to 979 in the skilled case. Perhaps suprisingly, the distribution of players into tournaments of different buyins after 5000 tournaments was strikingly similar as one can see examining table 6.2. Even without skill, *somebody* has to win the tournaments and when those winning players move up in stakes, somebody has to get lucky and win at those stakes and move up. It turns out that the distribution of players into tournament buyins does not tell us anything about the nature of skill in poker tournaments.

Buyin	Skilled Players	Unskilled Players
1	2970	2973
2	783	761
5	462	463
10	297	317
20	121	114
30	116	111
50	103	100
100	65	70
200	55	63
500	16	15
1000	12	13

Table 6.2: Buyin Table

There was however a difference between the skilled and unskilled poker economies regarding the distrubution of player profits after 5,000 tournaments. A few of the most skilled players had greater profit than the most profitable player in the unskilled economy. This is because even in the highest stakes games, these players have an edge whereas, by definition, nobody in the unskilled economy has an edge. On the bottom end, there are 320 players in the skilled economy that lost more money than the losingest player in the unskilled economy. These are players that are at a disadvantage even at the easiest games at the lowest stakes. These players had an average skill of 45.9 as compared with 49.1 for the average player in the 1 dollar tourney. These aspects can be summarized in the difference of standard deviation in player profit for the two types of economies after 5,000 tournaments. In the poker economy with skill, the standard deviation in profit is 7597 whereas in the unskilled poker economy, it is only 5954.

6.4.2 Varying Bankroll Strategies

Playing skill and chance are not the only determinants of poker success. There is also the factor of bankroll strategy. If a very skilled player is too agressive with his bankroll strategy, she could lose a great deal of her winnings with a modest stretch of bad luck. If a player is too cautious, she could be missing out on a great deal of potential profit in higher stakes games.

I ran a simulation similar to the first one, but with varying bankroll strategies. There were 5000 players with the same skill distribution and 5000 tournaments played, but with R uniformly at random taken to be a multiple of 10 with $10 \leq R \leq 100$. The differing bankroll strategies, particularly the aggressive strategies with low R caused more players to go bust and forced them to redeposit more often. Thus, the size of the overall poker economy was larger with an average deposit of \$1127 as compared to \$979 with R fixed at 50. The variance was also larger, $10,873$ compared to 7597, as bad players with aggressive bankroll management racked up big losses. Nonetheless, the players final distribution in tournaments was similar as can be seen in Table 6.3. We also see the same pattern with higher buyins associated with higher average skill.

Buyin	Fixed R	Varied R
1	2970	2852
2	783	892
5	462	469
10	297	283
20	121	128
30	116	113
50	103	107
100	65	75
200	55	59
500	16	10
1000	12	12

Table 6.3: Buyin Table

Very aggressive bankroll management appeared to be counterproductive when seeking long term profit. One player with a skill of 57 and an R of 10 lost 161 dollars over the 5000 tournament span. In the first simulation with everybody's $R = 50$, the least profitable player with a skill of 57 won over $19,000$ dollars. The most skilled player in that scenario to lose money had a skill of 54. If we group players by their R value, we find that the $R = 10$ people lost the most on average while an $R \geq 50$ was effective. If we regress average profitability on R and $\sqrt{R}$, we explain 89% of the variation (adjusted $r^2 = .86$) in average profit. This is best seen in Figure 6.2.

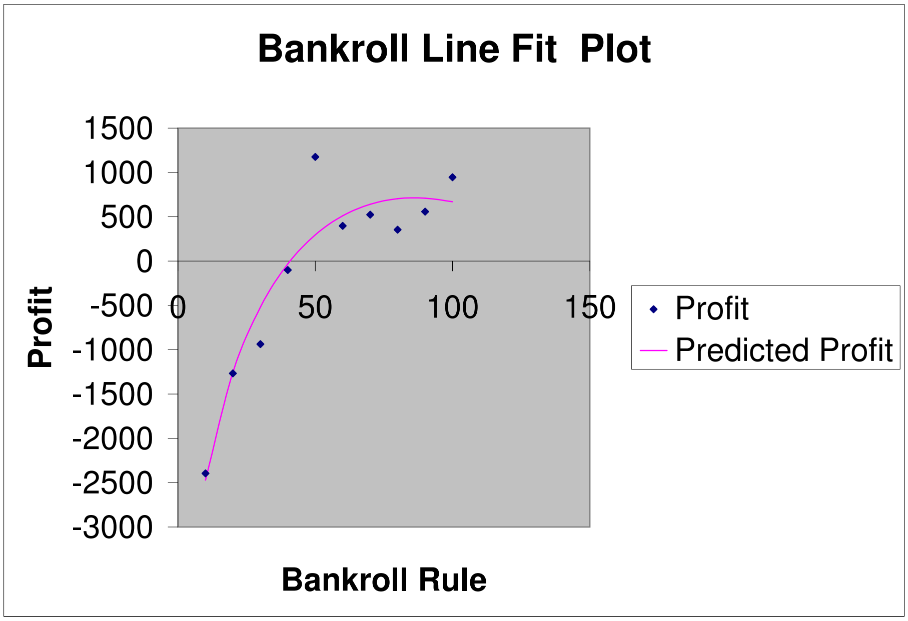


Figure 6.2: Average profit as a function of the Bankroll Management Strategy.

6.5 Discussion

The simulations in this chapter provide a quantitative estimate of the role that skill plays in tournament poker. In accord with popular perception, the role of skill in determing the outcome of a single tournament is rather small. However, over time, the role of skill explains a large majority of the variation in player profit in the poker tournament economy. Skilled players can threaten their potential profit with overly aggressive bankroll management, but barring that, the most skilled players will eventually be profitable and the unskilled players will lose money.

CHAPTER 7
OPEN PROBLEMS

7.1 Introduction

The open problems in algorithmic game theory are legion. There are two important problems that flow directly out of the work in this thesis. There are some remaining CES markets for which it is unknown whether a polynomial time equlibrium computation algorithm exists, but there are also no hardness results. The first problem is either finding a polynomial time algorithm for these markets or alternatively demonstrating PPAD-completeness. The other problem is proving that players in Two Player Push-Fold Texas Holdem employing heuristics without access to information about their opponent's past strategies will converge to an approximate equilibrium.

7.2 Completing the Complexity Classification of CES Economies

Recall that CES utility functions can be defined in terms of a paramter ρ such that $\rho < 1$ and $\rho \neq 0$. Utility functions with $\rho > 0$ satisfy gross substitutability. The limit cases of $\rho \to 1$ (linear utilities) and $\rho \to 0$ (Cobb-Douglas utilities) also satisfy gross substitutability. Approximate equilibria in these markets can all be computed in polynomial time by the algorithm introduced in Chapter 2. For $-1 \leq \rho < 0$, a polynomial time algorithm is provided in Chapter 3.

For $\rho \to -\infty$, the Leontief utilities, it is NP-hard just to determine if a market has an equilibrium or not [27]. When the equilibrium is known to exist, it is PPAD-complete to compute it [16, 27]. There are no results for $\rho < -1$. Settling the question of whether one can compute approximate equilibria for this range of ρ in polynomial time or whether it is perhaps complete for a difficult complexity class is completely open. Attempting to show PPAD-completeness would probably be the more promising route. This is because Gjerstad [52] has shown that the set of price equilibria for markets where traders have ρ in this range can be disconnected. This introduces difficulties not present in other regions of ρ where the set of price equilibria is convex.

The solution of this problem would be signicant for two reasons. One is that we would now be able to completely classify markets with CES utility functions. This would be remarkable because CES utility functions encompass such a wide range of behavior. The other is that regardless of what is shown for $\rho < -1$, it would be a novel finding. If there is as polynomial time algorithm, it would be quite striking to have one when the equilibrium structure can be so complicated. The other is that the first and most prominent hardness results for markets, the Leontief utility case, is a somewhat pathalogical case where you have no possibility of substituting one good for another. CES utility functions with $\rho < -1$ are smoother and more well behaved so it would be remarkable to have a hardness result for these markets.

7.3 Theoretical Derivation of Experimental Findings

In chapter 5, it is shown experimentally that players employing relatively simple heuristics converge to an approximate equilibrium. When equilibria for Push-Fold

Holdem were first computed [13], a technique called *fictitious play* [8, 11] was used. With fictitious play, an iterative procedure to compute equilibria, players (at least in part) play a best response to the strategy their oppenent has been playing. This cannot be employed by actual players in the Push-Fold game because they do not know what strategies their opponents have been employing even after the hands have been played.

It would be an advance if it could be proven that players employing the heuristics from chapter 5 which rely only on public information will converge upon an approximate equilibrium. The simulations conducted in this thesis provide some reason to think that such a demonstration is possible. This would give further credence to the Nash equlibrium as a solution concept in at least some settings.

REFERENCES

[1] J. R. Green A. Mas-Colell, M. D. Whinston. *Microeconomic Theory*. Oxford University Press, 1995.

[2] N. Alon. Poker, chance and skill. http://www.math.tau.ac.il/ nogaa/PDFS/skill4.pdf, 2007.

[3] K.. Arrow and L.Hurwicz. Competitive stability under weak gross substitutability: The "euclidean distance" approach. *International Economic Review*, 1:38–49, 1960.

[4] K.J. Arrow, H.D. Block, and L.Hurwicz. On the stability of the competitive equilibrium, ii. *Econometrica*, 27:82–109, 1959.

[5] K.J. Arrow, H.B. Chenery, B.S. Minhas, and R.M. Solow. Capital-labor substitution and economic efficiency. *The Review of Economics and Statistics*, 43(3):225–250, 1961.

[6] K.J. Arrow and G. Debreu. Existence of an equilibrium for a competitive economy. *Econometrica*, 22(3):265–290, 1954.

[7] A.M. Bagirov and A.M. Rubinov. Global optimization of marginal functions with applications to economic equilibrium. *Journal of Global Optimization*, 20:215–237, 2001.

[8] Ulrich Berger. Brown's original fictitious play. Game Theory and Information 0503008, EconWPA, March 2005.

[9] E. Borel. *Traite du Calcul des Probabilites et de ses Applications*. Gauthier-Villars, Paris, 1938.

[10] A. Brooke, D. Kendrick, and A. Meeraus. *GAMS: A user's guide*. The Scientific Press, South San Francisco, 1988.

[11] G.W. Brown. Iterative solutions of games by fictitious play. In T.C. Koopmans, editor, *Activity Analysis of Production and Allocation*, pages 374–376. Wiley, 1951.

[12] A. Cabot and R. Hannum. Public policy, law, mathematics and the future of an american tradition. *Cooley Law Review*, 2006.

[13] B. Chen and J. Ankenman. *The Mathematics of Poker*. ConJelCo, Pittsburgh, PA, 2006.

[14] Ning Chen, Xiaotie Deng, Xiaoming Sun, and Andrew Chi-Chih Yao. Fisher equilibrium price with a class of concave utility functions. In *ESA*, pages 169–179, 2004.

[15] Xi Chen, Decheng Dai, Ye du, and Shang-Hua Teng. Settling the complexity of arrow-debreu equilibria in markets with additively separable utilities. *CoRR*, 2009.

[16] Xi Chen and Xiaotie Deng. Settling the complexity of two-player nash equilibrium. In *FOCS '06: Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 261–272, Washington, DC, USA, 2006. IEEE Computer Society.

[17] J. Q. Cheng and M. P. Wellman. The walras algorithm: A convergent distributed implementation of general equilibrium outcomes. *Computational Economics*, 12(1):1–24, 1998.

[18] P.A. Chiappori and I. Ekeland. Individual excess demands. *Journal of Mathematical Economics*, 40:41–57, 2004.

[19] B. Codenotti, B. McCune, S. Penumatcha, and K. Varadarajan. Market equilibrium for ces exchange economies: Existence, multiplicity, and computation. In *FSTTCS 2005: Foundations of Software Technology and Theoretical Computer Science*, pages 505–516. Springer Verlag, 2005.

[20] B. Codenotti, B. McCune, and K. Varadarajan. Market equilibrium via the excess demand function. In *Proceedings of the 37th annual ACM symposium on Theory of computing*, pages 74 – 83, New York, NY, USA, 2005. ACM Press.

[21] B. Codenotti, S. Pemmaraju, and K. Varadarajan. Algorithms column: The computation of market equilibria. *SIGACT News*, 35, December 2004.

[22] B. Codenotti, S. Pemmaraju, and K. Varadarajan. The computation of market equilibria. *ACM SIGACT News*, 35(4):23–37, 2004.

[23] B. Codenotti, S. Pemmaraju, and K. Varadarajan. On the polynomial time computation of equilibria for certain exchange economies. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 72–81, Philadelphia, PA, USA, 2005. Society for Industrial and Applied Mathematics.

[24] Bruno Codenotti, Benton Mccune, Sriram Pemmaraju, Rajiv Raman, and Kasturi Varadarajan. An experimental study of different approaches to solve the market equilibrium problem. *J. Exp. Algorithmics*, 12:1–21, 2008.

[25] Bruno Codenotti, Benton McCune, Sriram V. Pemmaraju, Rajiv Raman, and Kasturi Varadarajan. An experimental study of different approaches to solve the market equilibrium problem. In *ALENEX/ANALCO*, pages 167–179, 2005.

[26] Bruno Codenotti, Benton McCune, Rajiv Raman, and Kasturi Varadarajan. Computing equilibrium prices : Does theory meet practice? In *European Symposium on Algorithms*, pages 83–94, 2005.

[27] Bruno Codenotti, Amin Saberi, Kasturi R. Varadarajan, and Yinyu Ye. Leontief economies encode nonzero sum two-player games. In *SODA*, pages 659–667, 2006.

[28] Bruno Codenotti and Kasturi Varadarajan. Market equilibrium in exchange economies with some families of concave utility functions. Computational Economics 0503001, EconWPA, March 2005.

[29] Bruno Codenotti and Kasturi R. Varadarajan. Efficient computation of equilibrium prices for markets with leontief utilities. In *ICALP*, pages 371–382, 2004.

[30] J. Schaeffer D. Billings, A. Davidson and D. Szafron. The challenge of poker. *Artificial Intelligence Journal*, 134, 2002.

[31] Constantinos Daskalakis, Paul W. Goldberg, and Christos H. Papadimitriou. The complexity of computing a nash equilibrium. In *STOC '06: Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 71–78, New York, NY, USA, 2006. ACM.

[32] G. Debreu. Economies with a finite set of equilibria. *EConometrica*, 38:387–392, 1970.

[33] N. Devanur and V.V. Vazirani. Market equilibrium when buyers have spending constraints. Webpage: `http://www.cc.gatech.edu/fac/Vijay.Vazirani/`, 2004.

[34] Nikhil R. Devanur and Ravi Kannan. Market equilibria in polynomial time for fixed number of goods or agents. In *FOCS*, pages 45–53, 2008.

[35] Nikhil R. Devanur and V.V. Vazirani. The spending constraint model for market

equilibrium: algorithmic, existence and uniqueness results. In *STOC*, pages 519–528, 2004.

[36] S.P. Dirkse and M.C. Ferris. A pathsearch damped newton method for computing general equilibria. *Annals of Operations Research*, pages 211–232, 1996.

[37] B. C. Eaves. A finite algorithm for the linear exchange model. *Journal of Mathematical Economics*, 3:197–203, 1976.

[38] B. C. Eaves. Finite solution of pure trade markets with cobb-douglas utilities. *Mathematical Programming Study*, 23:226–239, 1985.

[39] B. C. Eaves and H. Scarf. The solution of systems of piecewise linear equations. *Mathematics of Operations Research*, 1(1):1–27, 1976.

[40] B.C. Eaves. Homotopies for computation of fixed points. *Mathematical Programming*, 3:1–22, 1972.

[41] E. Eisenberg. Aggregation of utility functions. *Management Sciences*, 7(4):337–350, 1961.

[42] E. Eisenberg and D. Gale. Consensus of subjective probabilities: The pari-mutuel method. *Annals of Mathematical Statistics*, 30:165–168, 1959.

[43] M. Esteban-Bravo. Computing equilibria in general equilibrium models via interior-point methods. *Computational Economic*, 23:147–171, 2004.

[44] C. Ferguson and T. Ferguson. On the borel and von neumann poker models. *Game Theory and Applications*, 9:17–32, 2003.

[45] M. C. Ferris and T. S. Munson. Path 4.6. *http://www.gams.com/solvers/path.pdf*.

[46] M. C. Ferris and T. S. Munson. Complementarity problems in gams and the path solver. *Journal of Economic Dynamics and Control*, 24:165–188, 2000.

[47] D. Gale. *The Theory of Linear Economic Models*. McGraw Hill, New York, 1960.

[48] D. Gale. The linear exchange model. *Journal of Mathematical Economics*, 3:205–209, 1976.

[49] R. Garg and S. Kapoor. Auction algorithms for market equilibrium. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, pages 511–518. ACM, 2004.

[50] Andrew Gilpin and Tuomas Sandholm. A texas hold'em poker player based on automated abstraction and real-time equilibrium computation. In *AAMAS '06: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 1453–1454, New York, NY, USA, 2006. ACM.

[51] V. A. Ginsburgh and J. L. Waelbroeck. *Activity Analysis and General Equilibrium Modelling*. North Holland, 1981.

[52] S. Gjerstad. Multiple equilibria in exchange economies with homothetic, nearly identical preference. *University of Minnesota, Center for Economic Research, Discussion Paper*, 288, 1996.

[53] Martin Grotschel, Lazlo Lovasz, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag, 1991.

[54] P.T. Harker and B. Xiao. Newton's method for the nonlinear complementarity problem: a b-differential equation approach. *Mathematical Programming*, 48:339–358, 1990.

[55] http://cepa.newschool.edu/het/home.htm. The history of economic thought websites. Web Page.

[56] http://www.pokerstars.com/poker/tournaments/rules/prize structure/20/. Poker stars multi table tournament prize structures. Web Page, 2007.

[57] Li-Sha Huang and Shang-Hua Teng. On the approximation and smoothed complexity of leontief market equilibria. In *FAW*, pages 96–107, 2007.

[58] Kamal Jain. A polynomial time algorithm for computing the arrow-debreu market equilibrium for linear utilities. In *FOCS*, pages 286–294, 2004.

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

[60] M. Mahdian K. Jain and A. Saberi. Approximating market equilibria. In *Approximation, Randomization, and Combinatorial Optimization: Algorithms and Techniques, 6th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, APPROX 2003 and 7th International Workshop on Randomization and Approximation Techniques in Computer Science, RANDOM 2003, Princeton, NY, USA, August 24-26, 2003, Proceedings*, pages 98–108. Springer, 2003.

[61] S. Kintali. A compendium of ppad-complete problems. Webpage: `http://www.cc.gatech.edu/ kintali/ppad.html`, 2009.

[62] H.W. Kuhn. Simplicial approximation of fixed points. In *Proc. National Academy of Sciences of the United States of America*, volume 61, pages 1238–1242, 1968.

[63] T. S. Munson M. C. Ferris and D. Ralph. A homotopy method for mixed complementarity problems based on the path solver. pages 143–167, London, 2000. Chapman and Hall.

[64] D. Detterman M. DeDonno. Poker is a skill. 12:31–36, 2008.

[65] R. R. Mantel. The welfare adjustment process: its stability properties. *International Economic Review*, 12:415–430, 1971.

[66] R. R. Maxfield. General equilibrium and the theory of directed graphs. *Journal of Mathematical Economics*, 27:23–51, 1997.

[67] Benton McCune. Extending polynomial time computability to markets with demand correspondences. In *WINE*, pages 347–355, 2007.

[68] E. Tardos N. Nisan, T. Roughgarden and V. Vazirani. *Algorithmic Game Theory*. Cambridge University Press, New York, NY, 2007.

[69] A. Saberi V. V. Vazirani N. R. Devanur, C. H. Papadimitriou. Market equilibrium via a primal-dual-type algorithm. In *43rd Symposium on Foundations of Computer Science*, pages 389–395. IEEE Computer Science, 2002.

[70] J.F. Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences of the United States of America*, 36:48–49, 1950.

[71] J.F. Nash. Non-cooperative games. *Annals of Mathematics*, 54:286–295, 1951.

[72] T. Negishi. Welfare economics and existence of an equilibrium for a competitive economy. *Metroeconomica*, 12:92–97, 1960.

[73] E. I. Nenakov and M. E. Primak. One algorithm for finding solutions of the arrow-debreu model. *Kibernetica*, 3:127–128, 1983.

[74] D. J. Newman and M. E. Primak. Complexity of circumscribed and inscribed ellipsoid methods for solving equilibrium economical models. *Applied Mathematics and Computation*, 52:223–231, 1992.

[75] C.H. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *Journal of Computer and System Sciences*, 48.

[76] Christos H. Papadimitriou. Algorithms, games, and the internet. In *In STOC*, pages 749–753. ACM Press, 2001.

[77] V.M. Polterovich and V.A. Spivak. Gross substitutability of point to set correspondences. *Journal of Mathematical Economics*, 11:113–140, 1983.

[78] M. E. Primak. An algorithm for finding a solution of the linear exchange model and the linear arrow-debreu model. *Kibernetika*, 5:76–81, 1984.

[79] M. E. Primak. A converging algorithm for a linear exchange model. *Journal of Mathematical Economics*, 22:181–187, 1993.

[80] S. Kapoor R. Garg and V. Vazirani. An auction-based market equilibrium algorithm for the separable gross substitutability case. In *Approximation, Randomization, and Combinatorial Optimization, Algorithms and Techniques, 7th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, APPROX 2004, and 8th International Workshop on Randomization and Computation, RANDOM 2004, Cambridge, MA, USA, August 22-24, 2004, Proceedings*, pages 128–138. Springer, 2004.

[81] T. Rutherford. Sequential joint maximization. 18, 1999.

[82] T.F. Rutherford. Extensions of gams for complementarity problems arising in applied economic analysis. *Journal of Economic Dynamics and Control*, 19:1299–1324, 1995.

[83] T.F. Rutherford. Applied general equilibrium modeling with mpsge as a gams sybsystem: An overview of the modeling framework and syntax. *Computational Economics*, 14:1–46, 1999.

[84] Paul Samuelson. *Foundations of economic analysis*. Harward University Press, 1947.

[85] H. Scarf. The approximation of fixed points of a continuous mapping. *SIAM J. Applied Math*, 15:1328–1343, 1967.

[86] H. Scarf. The computation of equilibrium prices: An exposition. In *Handbook of Mathematical Economics*, volume 2, pages 1008–1061, 1982.

[87] H. Scarf. *Applied General Equilibrium Analysis*, chapter The Computation of Equilibrium Prices. 1984.

[88] H.E. Scarf. Some examples of global instability of the competitive equilibrium. *International Economic Review*, 1:157–172, 1960.

[89] S.C.Billups, S.P.Dirkse, and M.C. Ferris. A comparison of large scale mixed complemetarity problem solvers. *Comput. Optim. Appl.*, 7:3–25, 1997.

[90] S. Smale. A convergent process of price adjustment. *Journal of Mathematical Economics*, 3:107–120, 1976.

[91] S. Smale. Exchange processes of price adjustment. *Journal of Mathematical Economics*, 3:211–226, 1976.

[92] R. Solov. A contribution to the theory of economic growth. *Journal of Mathematical Economics*, 70:65–94, 1956.

[93] T.N. Srinivasan T.J. Kehoe and John Whalley. *Frontiers in Applied General Equilibrium Modelling*. Cambridge University Press, 2005.

[94] R. J. Vanderbei. *LOQO users' manual: version 4.05, Technical Report*. Operations Research and Financial Engineering, Princeton University, ., 2000.

[95] H. Varian. *Microeconomic Analysis*. W.W. Norton, 1992.

[96] J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. John Wiley, 1944.

[97] A. Wald. On some systems of equations of mathematical economics. *Econometrica*, 19:368–403, 1951.

[98] Leon Walras. *Elements of Pure Economics, or The Theory of Social Wealth*. 1874.

[99] M. Safra X. Deng, C. H. Papadimitriou. On the complexity of equilibria. In *Proceedings of the 34th annual ACM symposium on Theory of computing*, pages 67–71, New York, NY, USA, 2002. ACM Press.

[100] Yinyu Ye. A path to the arrow-debreu competitive market equilibrium. *Math. Program.*, 111(1-2):315–348, 2008.