Computational Aspects of Game Theory and Microeconomics

A Thesis
Presented to
The Academic Faculty

by

Evangelos Markakis

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

College of Computing
Georgia Institute of Technology
August 2005

Computational Aspects of Game Theory and Microeconomics

Approved by:

Dr. Richard J. Lipton, Advisor
College of Computing
Georgia Institute of Technology

Dr. Vijay V. Vazirani
College of Computing
Georgia Institute of Technology

Dr. Dana Randall
College of Computing
Georgia Institute of Technology

Dr. Yan Z. Ding
College of Computing
Georgia Institute of Technology

Dr. Richard Duke
School of Mathematics
Georgia Institute of Technology

Date Approved: June 3, 2005

ACKNOWLEDGEMENTS

I would first like to thank my advisor, Dick Lipton, for his support throughout these years. I would also like to thank Milena Mihail and Vijay Vazirani for convincing me to join the Ph.D. program at Georgia Tech back in 2000. l certainly did not regret it! Special thanks to some of my friends and collaborators Amin Saberi, Aranyak Mehta, Parikshit Gopalan and Nayantara Bhatnagar for providing a great working environment. Finally I want to thank Evgenia Skoufa for bearing with me especially in my first two years in Atlanta.

TABLE OF CONTENTS

LIST OF TABLES

SUMMARY

The purpose of this thesis is to study algorithmic questions that arise in the context of game theory and microeconomics. In particular, we investigate the computational complexity of various economic solution concepts by using and advancing methodologies from the fields of combinatorial optimization and approximation algorithms.

We first study the problem of allocating a set of indivisible goods to a set of agents, who express preferences over combinations of items through their utility functions. Several objectives have been considered in the economic literature in different contexts. In fair division theory, a desirable outcome is to minimize the envy or the envy-ratio between any pair of players. We use tools from the theory of linear and integer programming as well as combinatorics to derive new approximation algorithms and hardness results for various types of utility functions. A different objective that has been considered in the context of auctions, is to find an allocation that maximizes the social welfare, i.e., the total utility derived by the agents. We construct PCP-based reductions from multi-prover proof systems to obtain hardness results, given standard assumptions for the utility functions of the agents.

We then consider equilibrium concepts in games. We derive the first subexponential algorithm for computing approximate Nash equilibria in 2-player noncooperative games. We extend our result to multi-player games and we further propose a second algorithm based on solving polynomial equations over the reals. Both algorithms improve the previously known upper bounds on the complexity of the problem.

Finally, we study game theoretic models that have been proposed recently to address incentive issues in Internet routing. We obtain a polynomial time algorithm for computing equilibria in such games, i.e., routing schemes and payoff allocations from which no subset of agents has an incentive to deviate. Our algorithm is based on linear programming duality theory. We also obtain generalizations when the agents have nonlinear utility functions.

CHAPTER I

INTRODUCTION

The interaction of economic agents has become a fundamental paradigm in computer science applications. The expansion of the Internet and the world wide web have led to numerous examples including electronic commerce (e.g. web auctions), Internet routing, network design and other related problems on multi-agent systems. It is therefore natural that the modern theory of algorithms should adopt game theoretic models to analyze applications as above.

Game theory and economics have been rich in providing models and solution concepts as well as in prescribing strategies for rational agents. However, the outcomes proposed by the economic theory often involve optimization problems with no known efficient solutions. Resolving such complexity questions is a challenge for the evolving field of algorithmic game theory and requires a combination of methodologies from computer science and economics.

The purpose of this thesis is to study algorithmic aspects of game theory and microeconomics. In particular, we investigate the computational complexity of various economic solution concepts. We have chosen representative problems from different areas of economics such as fair division, auction design and computation of equilibria in games and in many cases, we develop approximation algorithms or provide inapproximability results. Some of the main tools used include combinatorial methods, theory of linear and integer programming, and probabilistic techniques. Below we briefly describe the problems that we consider along with our contribution and the methodologies used.

1.1 Problems studied - Techniques and Contributions

In the second chapter, we study fair division questions from an algorithmic perspective. Fair division is a central topic in economic theory with numerous applications in everyday life. Ever since the first attempt for a mathematical treatment of the topic by Steinahus, Banach and Knaster [98], a vast literature on cake-cutting algorithms and allocations of divisible goods has emerged (see among others [15] and [87]). In the case of indivisible goods however, different techniques are required and very few algorithmic results are known. We (jointly with R. Lipton, E. Mossel and A. Saberi) study the problem of allocating a set of indivisible goods to a set of agents who express preferences over combinations of items through their utility functions. Our fairness criterion is to minimize the envy among the agents. We derive approximation algorithms and hardness results for various types of utility functions. In the case of identical and additive utilities, we use techniques from the theory of linear and integer programming as well as combinatorial tools to obtain a polynomial time approximation scheme. In the case of non-additive utilities, we use information theoretic arguments to show (unconditionally) that an exponential amount of communication is required to compute the optimal solution. We also show that our results hold for other allocation models in which we have a combination of divisible and indivisible goods (e.g. in partitioning measurable spaces with atoms [22]).

In the third chapter we study optimization problems that arise in the design of combinatorial auction protocols. Lately, a large volume of transactions is conducted through auctions including, among others, selling spectrum licenses, treasury bills and even flowers in Dutch auctions. In a combinatorial auction, a set of indivisible goods is to be sold to a set of agents. The most desirable outcome from an economic point of view is to pick the allocation that maximizes the *social welfare*, i.e., the total utility derived by the agents. We (jointly with S. Khot, R. Lipton and A. Mehta) derive inapproximability results when the agents have submodular utility functions. Submodularity can be seen as the discrete analog of concavity and is a natural property in many economic settings since it expresses the fact that an agent's utility gets saturated as we allocate more goods to him. We show that the problem is not approximable with a factor better than $1 - 1/e$, unless **P**= **NP**. Our result is based on the PCP theorem and on constructing reductions from multi-prover proof systems for MAX-3-COLORING.

In the fourth chapter we investigate the complexity of computing Nash equilibria in noncooperative games. Noncooperative game theory has been extensively used to analyze interactions among selfish players. The dominant concept of rationality in noncooperative games is the *Nash equilibrium*, a behavior from which no player has an incentive to deviate (unilaterally). Polynomial time algorithms for equilibria however are still elusive and there is a need to develop an algorithmic theory of equilibria. Rationality properties of a solution do not suffice if computing an equilibrium is an intractable problem. People in practice tend to follow strategies that are simple and easy to compute, therefore complexity issues should be taken into consideration. We obtain the currently best upper bounds for computing approximate Nash equilibria. In particular, we (jointly with R. Lipton and A. Mehta) derive the first subexponential algorithm for 2-player and constant-player games, by using probabilistic techniques. We also identify cases where exact equilibria are polynomial time computable such as games with low rank payoff matrices. In a subsequent work, we (jointly with R. Lipton) provide another algorithm for multi-player games, based on results for finding solutions of polynomial equations. The second algorithm improves the dependence on the degree of approximation and the number of players at the expense of raising the dependence on the number of strategies.

Finally, in the fifth chapter, we study a game theoretic model for addressing incentive issues in Internet routing. The Internet is composed of many administrative domains (Autonomous Systems, ASes) each of which can be seen as an agent trying to route its customers' traffic and maximize its own benefit. As an attempt to model the interaction of these entities, Papadimitriou [82] proposed the following cooperative game: Given a multicommodity flow satisfying capacity and demand constraints, the total payoff derived is the amount of flow routed (or more generally a function of the flow routed). An outcome of the game is a feasible multicommodity flow along with an allocation of the total payoff to the players. The dominant solution concept in cooperative games is the notion of the *core*, which consists of outcomes that are stable against deviations from any subset of agents. We (jointly with A. Saberi) show that under this model, the core is nonempty, i.e., there exists a routing scheme of the demand traffic and an allocation of the total payoff such that no subset of ASes has an incentive to secede from the scheme. We further obtain a polynomial time algorithm to compute such an outcome. Our algorithm is based on an economic interpretation of the dual variables of the multicommodity flow linear program and the payoff allocation is obtained by solving the dual program. We also extend our result to the case where the payoff is a concave function of the routed flow.

CHAPTER II

FAIR DIVISION

2.1 Introduction

Fair division is a central topic in economic theory with numerous applications in everyday life. The first attempt for a mathematical treatment of the problem was made by the Polish school of mathematicians (Steinhaus, Banach, and Knaster [98]) and was the source of many interesting questions. Over the past fifty years, a vast literature has emerged (see [15, 87] for a summary of related results) and several notions of fairness have been suggested.

The fairness criterion that we focus on is the amount of envy between any pair of players. An allocation is *envy-free* if and only if every player likes his own share at least as much as the share of any other player. The class of envy-free allocations as a solution concept was introduced by Foley [36] and Varian [101] and has been studied extensively since then in the economic literature [87, 15]. However, in most of the models considered so far, either all goods are divisible or there is at least one divisible good, e.g. money, so that players can compensate each other and achieve envy-freeness [1, 100]. We believe that indivisibility issues should be taken into consideration.

We study the problem of allocating m *indivisible* goods to a set of n players in a fair manner. When goods are indivisible, an envy-free allocation may not exist and we wish to find allocations with upper bounds on the envy. In our model, each player p has a certain utility value $v_p(S)$ for each subset S of the goods. Given an allocation of the goods to the players, a player p envies player q if her valuation for the bundle given to player q is more than her valuation for her own bundle. In that case, her *envy* is the difference.

2.1.1 Our Results

We first show that there always exists an allocation with maximum envy at most α, where α is the maximum marginal utility of the goods, i.e., the value by which the utility of a player is increased when one more good is added to her bundle. Assuming that we have oracle access to the players' utilities, we give an $O(mn^3)$ time algorithm for producing a desired allocation. The problem of finding allocations with bounded envy in the presence of indivisible goods was introduced in [22] and a bound of $O(\alpha n^{3/2})$ was obtained. Our bound is a substantial improvement and it is also tight.

We then look at the optimization problem of computing allocations with minimum possible envy. We show that in most cases the problem is hard. First, using a similar argument as in [77], we show that any algorithm needs exponential time to obtain enough information about the valuations of players even if the valuations are provided via an oracle. We then look at the special case of additive utilities, i.e., $v_p(S) = \sum_{i \in S} v_p(\{i\})$. Even in this case we prove that for any constant c, there can be no 2^{m^c}-approximation algorithm for the minimum envy problem, unless $\mathbf{P} = \mathbf{NP}$.

We believe that a more suitable objective function is the maximum envy-ratio. The envy-ratio of player p for player q is the utility of p for q's bundle over her utility for her own bundle. If all players have the same utility function, the problem is closely related to a class of scheduling problems on identical processors. If we think of the players as identical machines and the set of goods as a set of jobs, then our problem is equivalent to scheduling the jobs so as to minimize the ratio of the maximum completion time over the minimum completion time. In [17] it is shown that Graham's greedy algorithm [40] achieves an approximation factor of 1.4 for the envy-ratio problem. We improve this result and derive a polynomial time approximation scheme.

Finally the issue of incentive compatibility is addressed. We prove that any algorithm that produces an allocation with minimum envy cannot be truthful. We also show that randomly allocating the goods to the players results in an allocation with envy at most $O(\sqrt{\alpha}\, n^{1/2+\epsilon})$ with high probability.

2.2 Model, Definitions and Notation

Let $N = \{1, 2, ..., n\}$ be a set of players and $M = \{1, 2, ..., m\}$ be a set of indivisible goods. A utility function v_p is associated with each player p. For $S \subseteq M$, $v_p(S)$ is the happiness player p derives if she obtains the subset S. We assume that v_p is non-negative and monotone i.e. $v_p(S) \leq v_p(T)$ for every $S \subseteq T$ and every p.

An allocation A is a partition of the goods $A = (A_1, A_2, ..., A_n)$ where $\cup_{p=1}^{n} A_p = M$ and $A_p \cap A_q = \emptyset$ for all $p \neq q$. The subset A_p denotes the set of goods allocated to player p. Note that some of the sets A_p may be empty. A partial allocation will be a partition of some subset of M.

Given an allocation $A = (A_1, A_2, \ldots, A_n)$, we say that player p envies player q if she prefers the bundle allocated to q to her own i.e. $v_p(A_p) < v_p(A_q)$. We will denote by e_{pq} the envy of p for q:

$$e_{pq}(A) = \max\{0, v_p(A_q) - v_p(A_p)\}.$$

We define $e(A)$ to be the maximum envy between any pair of players.

$$e(A) = \max\{e_{pq}(A),\ p, q \in N\}.$$

We will often omit the parameter A in the notation.

2.3 Existence of Allocations with Bounded Maximum Envy

A natural question is whether there exist allocations with bounded envy. We obtain a bound on the envy in terms of the maximum marginal utility of the goods. The marginal utility of a good i with respect to a player p and a subset of goods S, is the amount by which it increases the utility of p, when added to S, i.e., $v_p(S \cup \{i\}) - v_p(S)$. The maximum marginal utility is:

$$\alpha = \max_{S,p,i}\ v_p(S \cup \{i\}) - v_p(S)$$

In addition to proving a bound on the envy, we present an efficient algorithm that computes a desired allocation. For that, we assume that the algorithm can ask an oracle for the utility of a player p for any subset S.

Theorem 1 *For any set of goods and any set of players, there exists an allocation A such that the maximum envy of A is bounded by the maximum marginal utility of the goods, α.*

Furthermore, given oracle access for the utility functions of the players, there is an $O(mn^3)$ time algorithm for finding such an allocation.

Given an allocation A, we define the *envy graph* of A as follows: every node of the graph represents a player and there is a directed edge from p to q if and only if p envies q. The proof of Theorem 1 is based on the following Lemma:

Lemma 2 *For any partial allocation A with envy graph G, we can find another partial allocation B with envy graph H such that:*

- $e(B) \leq e(A)$,
- *H is acyclic.*

Proof :

If G has no directed cycles, we are done. Suppose that $C = p_1 \to p_2 \to \cdots \to p_r \to p_1$ is a directed cycle in G. If $A = \{A_1, ..., A_n\}$, we can obtain $A' = (A'_1, ...A'_n)$ by re-allocating the goods as follows: $A'_p = A_p$ for all $p \notin \{p_1, \ldots, p_r\}$, and $A'_{p_1} = A_{p_2}, A'_{p_2} = A_{p_3}, \ldots, A'_{p_r} = A_{p_1}$.

Note that all players evaluate what they have in A' at least as much as what they have in A. Therefore it is easy to see that $e(A') \leq e(A)$.

We can also show that the number of edges in the envy graph G' corresponding to A' has decreased. To see this, first note that the set of the edges between pairs of vertices in $N \setminus C$ has not changed. Also every edge of the form $p \to p_j$ for $p \in N \setminus C$ and $p_j \in C$ has now become the edge $p \to p_{j-1}$ (or $p \to p_r$ if $j = 1$) in G' and no more edges of this form have been added. The number of edges of the form $p_j \to p$ has either decreased or remained the same since players in C are strictly happier. Finally for $p_i \in C$ the number of edges from p_i to vertices in C has decreased by at least 1.

Thus by repeatedly removing cycles using the above procedure, we will obtain an allocation B with corresponding envy graph H such that $e(B) \leq e(A)$ and H is acyclic. Since the number of edges decreases at every step, the process will terminate. □

Proof of Theorem 1 : We give an algorithm that produces the desired allocation. The algorithm proceeds in m rounds. At each round one more good is allocated to some player.

In the first round, we allocate good 1 to some player arbitrarily. Clearly the maximum envy is at most α. Suppose at the end of round i, the goods $\{1, ..., i\}$ have been allocated to the players and the maximum envy is at most α. At round $i+1$, we construct the envy graph corresponding to the current allocation. We use the procedure of Lemma 2 to obtain an allocation A in which the maximum envy is at most α and the new envy graph G is acyclic. Since G is acyclic, there is a player $p \in N$ with in-degree 0, which implies that nobody envies p. We then allocate good $i+1$ to p. Let $B = (B_1, ..., B_n)$ be the new allocation. For any 2 players q, r with $q, r \neq p$, $e_{qr}(B) = e_{qr}(A) \leq \alpha$. For $q \in N \setminus \{p\}$, since $e_{qp}(A) = 0$ we have:

$$\begin{aligned} e_{qp}(B) &= \max\{0, v_q(A_p \cup \{i\}) - v_q(A_q)\} \\ &\leq \max\{0, \alpha + v_q(A_p) - v_q(A_q)\} \leq \alpha \end{aligned}$$

The analysis for the running time of the algorithm is straightforward. In Lemma 2, we keep removing cycles until the envy-graph is acyclic. Finding a cycle and removing it takes at most $O(n^2)$ time and it decreases the number of edges by at least one. Initially the envy graph has no edges. Allocating a good at any round adds at most $n-1$ edges to the new envy graph. Since every cycle removal decreases the number of edges, the number of times we have to remove a cycle is at most $O(nm)$ and the total running time is $O(mn^3)$. $\square$

In [22], a similar model has been defined with the difference that the utility function of every player is additive and we have a combination of divisible and indivisible goods. More formally, in [22] the problem is to partition a measurable space $(\Omega, \mathcal{F})$. Each player has a utility function which is a probability measure v_p on $(\Omega, \mathcal{F})$ such that for each v_p the maximum value of an *atom* is α. A subset $S \subseteq \Omega$ is an atom for v_p if $v_p(S) > 0$ and $\forall E \subset S$, either $v_p(E) = 0$ or $v_p(E) = v_p(S)$. It is shown that there exist allocations with envy at most $O(\alpha n^{3/2})$.

We can prove that our result also holds for their model and hence it improves the bound of $O(\alpha n^{3/2})$ to α. The idea is that we can partition Ω into indivisible goods of value at most α and then apply Theorem 1. In particular, we use the following Lemma:

Theorem 3 *When the utilities of the players are probability measures on* $(\Omega, \mathcal{F}) = ([0,1]$, *Borel sets*) *with atoms of value at most* α, *there exists a partition* $A = (A_1, ..., A_n)$ *of* Ω *such that* $e(A) \leq \alpha$.

Proof : Since each measure v_p has atoms of value at most α, this means that for every $x \in [0,1]$, $v_p(\{x\}) \leq \alpha$. The case $\alpha = 0$ corresponds to an infinitely divisible cake and an envy-free allocation always exists [22]. For $\alpha > 0$ we can reduce the problem to allocating indivisible goods of value at most α and then use Theorem 1.

Lemma 4 *The interval* $[0,1]$ *can be partitioned in* m *disjoint sets* $S_1, ..., S_m$ *such that* $m = O(n/\alpha)$ *and* $v_p(S_j) \leq \alpha$ *for every* $p = 1, ..., n, j = 1, ..., m$

Proof : Find the minimum possible value for $x \in [0,1]$ such that $v_p([0,x]) \leq \alpha$ for every player p. Such an x exists since atoms have value at most α. Set $S_1 = [0,x]$ and consider the interval $(x,1]$. Again find the minimum value of $y \in (x,1]$ such that $v_p((x,y]) \leq \alpha$ for every p. Set $S_2 = (x,y]$. We can continue in the same manner until we partition the whole interval $[0,1]$. It is easy to check that the number of disjoint intervals $S_1, S_2, ..., S_m$ of the partition will be $O(n/\alpha)$. □

We can now treat the intervals $S_1, ..., S_m$ produced in the previous Lemma as indivisible goods and Theorem 1 will complete the proof. □

2.4 Minimizing Envy as an Optimization Problem

Although we can produce an allocation with bounded envy, in many instances the maximum envy can be smaller than α. Therefore we would like to look at the following two optimization problems:

Problem 1: Minimum envy

Compute an allocation A that minimizes the envy

$$\max_{p,q}\{0, v_p(A_q) - v_p(A_p)\}$$

Problem 2: Minimum envy-ratio

Compute an allocation A that minimizes the envy-ratio

$$\max_{p,q}\{1, \frac{v_p(A_q)}{v_p(A_p)}\}$$

As we will see it is not always possible to have a polynomial time algorithm for computing an optimal solution, hence we will also be interested in obtaining approximation algorithms. Given a minimization problem Π, we say that an algorithm has an approximation factor of ρ for Π, if for any instance I of Π, the algorithm outputs a solution which is guaranteed to be at most $\rho\, OPT(I)$, where $OPT(I)$ is the optimal solution. We will say that an algorithm is a *Polynomial Time Approximation Scheme* (PTAS) if for any instance I and any error parameter $\epsilon > 0$, the algorithm runs in time polynomial in the input size and outputs a solution which is at most $(1+\epsilon)OPT(I)$. If in addition the running time is polynomial in $1/\epsilon$ then we say that the algorithm is a *Fully Polynomial Time Approximation Scheme* (FPTAS).

In the following theorem we show that any algorithm needs an exponential number of queries in the worst case to produce an optimal solution. Our construction is similar to Nisan and Segal [77].

Theorem 5 *Any (deterministic) algorithm that computes an allocation with minimum envy or minimum envy-ratio requires a number of queries which is exponential in the number of goods in the worst case.*

Proof : We give an outline of the proof. Suppose $m = 2k$. We consider the following class of utility functions $\mathcal{F}$. A function v is in $\mathcal{F}$ if:

$v(S) = 0$ for all S with $|S| < k$.

$v(S) = 1$ for all S with $|S| > k$.

$v(S) = 1 - v(\bar{S})$ for all $|S| = k$

Now, consider instances (v, v) in which there are two players with the same utility function v for some $v \in \mathcal{F}$. The number of such instances is doubly exponential in k. Since the algorithm asks only a polynomial number of queries, it can produce only an exponential number of different query sequences. Therefore, there exist two different functions $u, v \in \mathcal{F}$ such that the query sequences corresponding to the instances defined by u and v are the same.

Consider the instances (u, v) and (v, u). The algorithm will produce the same query sequences for both instances and therefore it will produce the same allocation for (u, v) and (v, u). It is easy to see that although for either case, there exists an allocation which is envy-free, there is no single allocation that is envy-free for both instances. □

We would like to note that an interesting fact about Theorem 5 is that it is unconditional, i.e., not dependent on any complexity theory assumption.

2.4.1 Additive Utilities

We consider a natural special case of the problem in which the utility functions of all players are additive i.e. for all $p \in N$, $v_p(S) = \sum_{i \in S} v_p(\{i\})$. In that case, an instance of the problem is specified by an $n \times m$ matrix $V = (v_{p,i})$.

2.4.1.1 The minimum envy problem

Still, the problem of finding a minimum-envy allocation is **NP**-hard, even when the number of players is two. This follows from the fact that for two players with the same utility functions, deciding whether an envy-free allocation exists is equivalent to deciding if there exists a partition of a set of positive integers in two subsets of equal sum, which is **NP**-complete [102].

Since the objective function of the minimum envy problem can be zero and since deciding whether the minimum envy is zero is **NP**-complete, we cannot have any polynomial time approximation algorithm, unless $\mathbf{P} = \mathbf{NP}$. One way to remedy this is to add 1 (or some positive constant) to our objective function. In that case, Theorem 1 guarantees a $(1+\alpha)$-approximation algorithm, where $\alpha = \max\ v_{p,i}$. Even in this case though, strong hardness results can be obtained. We can show that for any constant c, there is no 2^{m^c}-approximation algorithm, unless $\mathbf{P} = \mathbf{NP}$. The proof is along the same lines as the inapproximability result for the problem Subset-Sums Difference in [10] and we omit it.

2.4.1.2 The minimum envy-ratio problem

We believe that a more suitable objective function is the envy-ratio. In the rest of this section, we study the envy-ratio problem in the case where the players have the same utility function. We will denote the utility that players derive from having good i by $v(i)$.

This special case is closely related to a class of scheduling problems on identical processors. We can think of the set of players as a set of identical machines and the set of goods as a set of m jobs to be scheduled on the machines. Every job has a positive processing time and the load of every processor is the sum of the processing times of the jobs assigned to it. Several objective functions have been considered such as minimizing the maximum completion time (makespan) [40, 44] or maximizing the minimum completion time [27, 106, 2]. Our problem is equivalent to minimizing the ratio of the maximum completion time over the minimum completion time.

The following greedy algorithm was proposed by Graham for the minimum makespan problem [40]: Sort the goods in decreasing order of their values and allocate them one by one in that order. At every step, allocate the next good to the player whose current bundle has the least value. In [17] it was proved (in the context of scheduling) that the approximation factor of Graham's algorithm is 1.4 for the ratio problem.

Theorem 6 *[17] Graham's algorithm achieves an approximation factor of* 1.4 *for the envy-ratio problem.*

In the next Theorem, we improve this result and show that we can achieve any constant factor arbitrarily close to 1 for the envy-ratio problem.

Theorem 7 *There is a PTAS for the envy-ratio problem when all players have the same utility for each good.*

Proof : Before going into the details of the proof we give a brief outline of the technique. Our algorithm is similar to [2] and [106]. However our objective function does not fit in their framework. The algorithm is as follows: Given our original instance, we round the utility of each good to obtain a coarser instance in which there is a constant number of distinct utilities, i.e., a constant number of different types of goods. We then show that in the new instance, we can find an optimal solution by searching, for every player, among a constant number of distinct assignments of goods. The constant will be exponential in the approximation parameter $1/\epsilon$. This observation enables us to compute the optimal solution in the rounded instance by solving a series of integer programs with a constant number of variables using Lenstra's algorithm [64]. After finding an optimal allocation in the rounded instance, we will convert it into an allocation for the original instance. In the whole process, there are 2 sources of error: computing the rounded instance from the original one and transforming the optimal allocation of the rounded instance to an allocation of the original instance. We are able to bound the error by $1 + \epsilon$.

Let I be an instance of the problem, with n players, m goods and utility $v(i)$ for good i. If $m < n$ then the optimal envy-ratio is ∞ and any allocation is optimal. Hence we can assume without loss of generality that $m \geq n$. We start with some basic facts about the optimal solution.

Let L be the average utility of the players,

$$L = \frac{1}{n} \sum_{i \in M} v(i)$$

If all the goods were divisible, we could allocate a fraction of $1/n$ from each good to a player and everybody would receive a utility of exactly L.

We briefly sketch how to handle goods with utility greater than L. Suppose there exists a good i with $v(i) \geq L$. If i is assigned to a player p in an optimal allocation, then there is an allocation with the same or less envy-ratio in which i is the only good allocated to p. To see this, suppose that player p receives good i and some other good, say j in an optimal solution. Let q be the person with minimum utility and bundle S_{min}. Then $v(S_{min}) < L$ and by giving good j to q, it is easy to see that the ratio does not increase, and hence the new solution is also optimal. Therefore goods with high utility can be assigned to players until we are left with goods that satisfy $v(i) < L$. This does not mean that if we have a PTAS for the case when $v(i) < L$ for all i, we can derive a PTAS for the general problem, as is the case in [2]. Instead we will have to round "big" goods appropriately so that in the rounded instance their utility is also higher than the corresponding average utility, L. We will then output an optimal solution for the rounded instance in which such goods are assigned to players with no other good in their bundle. We omit the details for handling goods with $v(i) \geq L$ and from now on we will assume that $v(i) < L$ for every i. We have the following fact:

Claim 8 *If $v(i) < L$ for every good i, then there exists an optimal allocation $A = (A_1, ..., A_n)$ such that $\frac{1}{2}L < v(A_i) < 2L$.*

The proof is by showing that in a given optimal solution, it is possible to reallocate the goods so that the envy-ratio does not increase and the conditions of the claim are satisfied.

We will now describe how to round the values of the goods and obtain an instance in which there is only a constant number of different types of goods (i.e. a constant number of distinct values for the goods). The construction is the same as in [2].

We will denote the rounded instance by $I^R(\lambda)$, where λ is a positive constant and will be determined later (λ will be $O(1/\epsilon)$). We will often omit λ in the notation.

We first round the value of every good with relatively high value. In particular, for every good i with $v(i) > L/\lambda$, we round $v(i)$ to the next integer multiple of L/λ^2. Roughly this means that we round up the first least significant digits of $v(i)$. We cannot afford to do the same for goods with small value since the error introduced by this process might be very big. Instead, let S be the sum of the values of the goods with value less than L/λ. We round S to the next integer multiple of L/λ, say S^R. Instance $I^R(\lambda)$ will have $S^R\lambda/L$ new goods with value L/λ. This completes the construction. Note that in $I^R(\lambda)$ all values are of the form kL/λ^2, where $\lambda \leq k \leq \lambda^2$. Hence we have only a constant number of distinct values, since λ is a constant.

Let M^R be the set of goods in the new instance and $v^R(i)$ be the value of each good. Let $L^R = \frac{1}{n}\sum_{j \in M^R} v^R(i)$. It is easy to see that $L \leq L^R$ and that all values in $I^R(\lambda)$ are at most L^R. Hence by Claim 8 there is an optimal solution $A^R = (A_1^R, ..., A_n^R)$ such that $\frac{1}{2}L^R < v(A_p^R) < 2L^R$ for every p. In the algorithm below we will search for such a solution.

We represent a player's bundle by a vector $t = (t_\lambda, ..., t_{\lambda^2})$, where t_k is the number of goods with value kL/λ^2 assigned to her. We will then say that the player is of type t. The utility derived from t is $v(t) = \sum_k t_k kL/\lambda^2$. Let U be the set of all possible types t, with $\frac{1}{2}L^R < v(t) < 2L^R$. It is easy to see that $|U|$ is bounded by a constant which is exponential in λ. Hence for a player of type $t \in U$, there is only a constant number of distinct values for her utility. Let $V(U)$ be the set of these values, i.e. $V(U) = \{u \,:\, v(t) = u \text{ for some } t \in U\}$.

We can now show how to find the optimal envy-ratio in $I^R(\lambda)$. For all pairs of values $u_1, u_2 \in V(U)$, we will solve the following decision problem: Is there an allocation in which the utility of every player is between u_1 and u_2? Since $|V(U)|$ is constant, after solving the above problem for all u_1, u_2 we can output the allocation corresponding to u_1^*, u_2^* for which the minimum ratio u_2^*/u_1^* is attained.

To solve the decision problem, we will write an integer program (IP) with a constant number of variables and use Lenstra's algorithm [64]. In the following IP, for each $t \in U$ we have an integer variable X_t indicating how many players are of type t. The first constraint implies that all players will obtain an allocation of type $t \in U$ and the second that all goods are assigned. It is obvious that the decision problem with inputs u_1, u_2 has a solution iff the corresponding integer program is feasible. Therefore we can find the optimal solution of $I^R(\lambda)$ in polynomial time. Note that the actual running time is exponential in λ which is the reason why we will finally obtain a PTAS and not an FPTAS.

In the following IP, $U_{u_1}^{u_2}$ is the set of all types $t \in U$ such that $u_1 \le v(t) \le u_2$ and n_k is the number of goods in $I^R(\lambda)$ of value kL/λ^2.

$$
\begin{aligned}
&\sum_{t \in U} X_t = n \\
&\sum_{t \in U} X_t t_k = n_k && \forall k \\
&X_t \ge 0 && \forall X_t \text{ with } t \in U_{u_1}^{u_2} \\
&X_t = 0 && \forall X_t \text{ with } t \in U \setminus U_{u_1}^{u_2}
\end{aligned}
$$

We need to see how the original instance is related to the rounded instance. The following Lemma has been proved in [2]:

Lemma 9 *Let $A = (A_1, ..., A_n)$ be an allocation in I, where $\frac{1}{2}L < v(A_i) < 2L$. Then there exists an allocation $B = (B_1, ..., B_n)$ in the rounded instance, I^R, such that:*

$$v(A_i) - \frac{1}{\lambda}L \leq v(B_i) \leq \frac{\lambda+1}{\lambda}v(A_i) + \frac{1}{\lambda}L$$

Also if $B = (B_1, ..., B_n)$ is an allocation in I^R such that $\frac{1}{2}L^R < v(B_i) < 2L^R$, then there exists an allocation $A = (A_1, ..., A_n)$, in I such that:

$$\frac{\lambda}{\lambda+1}v(B_i) - \frac{2}{\lambda}L \leq v(A_i) \leq v(B_i) + \frac{1}{\lambda}L$$

We are now ready to prove our Theorem. Our algorithm will be: Given instance I, compute the instance I^R, find an optimal allocation $A^R = (A^R_1, ..., A^R_n)$ for I^R, and then convert A^R to an allocation $A = (A_1, ..., A_n)$ for I using Lemma 9. Output A.

Suppose without loss of generality that $v(A^R_1) \leq ... \leq v(A^R_n)$ and $v(A_1) \leq ... \leq v(A_n)$. Let $A^* = (A^*_1, ..., A^*_n)$ be an optimal solution to I satisfying the conditions of Claim 8 and assume $v(A^*_1) \leq ... \leq v(A^*_n)$. We want to show:

$$\frac{v(A_n)}{v(A_1)} \leq (1+\epsilon)\frac{v(A^*_n)}{v(A^*_1})$$

By Lemma 9 we know that:

$$v(A_n) \leq v(A^R_n) + \frac{1}{\lambda}L \leq v(A^R_n) = \frac{1}{\lambda}L^R \leq v(A^R_n)(1 + \frac{2}{\lambda})$$

Similar calculations yield: $v(A_1) \geq v(A^R_1)(\frac{\lambda}{\lambda+1} - \frac{4}{\lambda})$. Therefore:

$$\frac{v(A_n)}{v(A_1)} \leq \frac{v(A^R_n)}{v(A^R_1)}(\frac{1+\frac{2}{\lambda}}{\frac{\lambda}{\lambda+1} - \frac{4}{\lambda}})$$

We need to relate the optimal solution in I^R with the optimal solution in I. By using the first part of Lemma 9 and by performing similar calculations we have that there exists an allocation $A' = (A'_1, ..., A'_n)$ in I^R such that:

$$\frac{v(A'_n)}{v(A'_1)} \leq \frac{v(A^*_n)}{v(A^*_1)}(\frac{\frac{\lambda+1}{\lambda} + \frac{2}{\lambda}}{1 - \frac{2}{\lambda}})$$

Since A^R is an optimal solution in I^R the ratio in A' will be at least as big as in A^R. Hence by combining the above equations we finally have:

$$\begin{aligned}
\frac{v(A_n)}{v(A_1)} &\leq \frac{v(A^*_n)}{v(A^*_1)}(\frac{1+\frac{2}{\lambda}}{\frac{\lambda}{\lambda+1} - \frac{4}{\lambda}})(\frac{\frac{\lambda+1}{\lambda} + \frac{2}{\lambda}}{1 - \frac{2}{\lambda}}) \\
&\leq \frac{(\lambda+1)(\lambda+2)(\lambda+3)}{(\lambda-2)(\lambda^2-4\lambda-4)}OPT
\end{aligned}$$

Thus, if we set $\lambda = 56/\epsilon$, it is easy to see that the factor will be at most $1 + \epsilon$.

$\square$

2.5 Truthfulness

So far we have assumed that we can obtain the actual utilities of the players for the goods. However, in many situations this is private information and the players may have incentives to lie about their valuations in order to obtain a better bundle. We would like to investigate the question of whether we can have mechanisms that elicit the true valuations from the players and also produce allocations with minimum or bounded envy. A mechanism is truthful if for every player, her profit is maximized by declaring her true utility, i.e., being truthful is a dominant strategy.

Truthful mechanisms have been the subject of research especially in the context of auctions. However, unlike our problem, auction mechanisms are allowed to compensate the players with money to ensure truthfulness.

We present a simple argument to prove that any mechanism that computes a minimum-envy allocation cannot be truthful even in the special case where the utility functions are additive.

Theorem 10 *Any mechanism that returns an allocation with minimum possible envy is not truthful.*

Proof : Let $\mathcal{M}$ be a mechanism that outputs an allocation with minimum envy. Consider the following instance: We have two players, 1 and 2 and $k+2$ goods. In particular we have good a, good b and k eggs. Let's say $k = 100$. The k eggs are playing the role of an almost divisible good of value 0.2. Suppose the players have the following utilities for the goods:

$$v_1(a) = 0.45, v_1(b) = 0.35, v_1(\text{egg}) = 0.2/k,$$

$$v_2(a) = 0.35, v_2(b) = 0.45, v_2(\text{egg}) = 0.2/k$$

The specific instance admits an envy-free allocation: give to player 1 good a and 25 eggs and give the rest to player 2. Therefore in the allocation that $\mathcal{M}$ will output there will be no envy. Let A be the partition that $\mathcal{M}$ outputs for this instance. Note that in A each player receives exactly one of the goods a, b because if one player received both then the other player would envy her. Also we note that it is Player 1 who receives a. To see this, suppose on the contrary that player 1 receives b. Then in order for A to be envy-free player 1 should receive at least 75 eggs (otherwise the bundle S of player 1 is worth less than 1/2 and she will be envious). But then player 2 will receive a and at most 25 eggs so she will be envious, a contradiction. Therefore in A player 1 receives a and T eggs and player 2 receives b and $k - T$ eggs. It is also easy to see that $25 \leq T \leq 75$.

Case 1: $T < 74$

In this case player 1 can increase her utility by lying and declaring that good a has less value for her. It is possible for her to lie in such a way to force the mechanism to give her the good a and at least $T + 1$ eggs (assuming that 2 does not change her declaration). She can declare that her valuation function is: $v_1(a) = 0.45 - \delta, v_1(b) = 0.35 + \delta, v_1(egg) = 0.2/k$ where δ is such that:

$$0.45 - \delta + (T + 1)0.2/k = 1/2$$

Notice that under this new declaration, there still exists an envy-free outcome. Let A' be the new output of $\mathcal{M}$. Again player 1 will receive a. This is true because if player 1 receives b then she has to receive at least $k - T - 1$ eggs so that she is not jealous. But then player 2 will receive good a and at most $T + 1 \leq 74$ eggs which in total is worth less than 1/2. Hence player 1 will get a and at least $T + 1$ eggs (otherwise her bundle is worth less than 1/2) which is more than what she gets if she is honest.

Case 2: $T \geq 74$

Now it is player 2 who can try to cheat. By misreporting her utilities in a similar manner as in Case 1, she can obtain a higher utility than before. □

In the rest of the section, we present a simple truthful mechanism which allocates the goods to the players uniformly at random. We assume that the sum of the utilities of each player over all goods is one. We will show that with high probability the maximum envy of the resulting allocation is no more than $O(\sqrt{\alpha} n^{1/2+\epsilon})$.

Theorem 11 *Suppose that $v_{p,j} \leq \alpha \quad \forall p \in N, j \in M$. Then for every $\epsilon > 0$, and for large enough n, there exists a truthful algorithm such that with high probability the allocation output by the algorithm has maximum envy at most $O(\sqrt{\alpha}\ n^{1/2+\epsilon})$.*

Proof : The proof is based on the probabilistic method. Allocate each good independently to player p with probability $1/n$. Clearly this is a truthful mechanism. We will show that with high probability, the allocation produced satisfies the desired bound. Fix two players p, q. Given p and q we define a random variable Y_j indicating the contribution of good j to the envy of player p for q. The variable Y_j is equal to 1, if good j is allocated to player q, -1, if it is allocated to player p, and 0 otherwise. Hence: $Y_j = 1$ with probability $1/n$, -1 w.p. $1/n$ and 0 w.p. $(n-2)/n$. We now define the random variable: $f_{pq} = \sum_j v_{p,j} Y_j$. Clearly the envy of p for q is $e_{pq} = max\{0, f_{pq}\}$. We will show that with high probability, for every p, q, $f_{pq} \leq O(\sqrt{\alpha}\ n^{1/2+\epsilon})$ and this will complete the proof.

The expectation of f_{pq} is:

$$E[f_{pq}] = \sum_j E[Y_j] v_{p,j} = 0$$

To compute the variance, note that the variables $\{Y_j\}$ are independent. Thus:

$$Var[f_{pq}] = \sum_j v_{p,j}^2 Var[Y_j] = \frac{2}{n} \sum_j v_{p,j}^2 \leq \frac{2\alpha}{n} \sum_j v_{p,j} = \frac{2\alpha}{n}$$

By using Chebyshev's inequality, we have that for any ordered pair of players p, q such that $p \neq q$ and for $t > 0$:

$$Pr[|f_{pq}| \geq t] \leq \frac{2\alpha}{nt^2}$$

Hence we have:

$$\begin{aligned} Pr[\max_{p,q} f_{pq} < t] &= Pr[\bigcap_{(p,q)} f_{pq} < t] = 1 - Pr[\bigcup_{(p,q)} f_{pq} \geq t] \\ &\geq 1 - \sum_{(p,q)} \frac{2\alpha}{nt^2} \geq 1 - \frac{2\alpha n}{t^2} \end{aligned}$$

If we set $t = 2\sqrt{\alpha}\ n^{1/2+\epsilon}$ we have that:

$$Pr[\text{max. envy } < 2\sqrt{\alpha} n^{1/2+\epsilon}] \geq 1 - n^{-2\epsilon} \tag{1}$$

$\square$

2.6 Discussion

Our algorithm for minimizing the envy-ratio works only if the utility functions of the players are additive and identical. It would be very interesting to find an approximation algorithm for the general case. One approach is to use a linear programming relaxation similar to Lenstra et al. [65].

There are many related notions of fairness such as max-min fairness or proportional fairness and we would like to know the complexity of these solution concepts as well. Some progress along these lines has already been made in [11].

Another question concerns the tradeoff between fairness and optimality of a solution. An allocation is optimal if it maximizes the sum of the utilities of the players. Such a tradeoff can be seen as the social cost of fairness or the "price of socialism".

CHAPTER III

COMBINATORIAL AUCTIONS

3.1 Introduction

A large volume of transactions is nowadays conducted via auctions, including auction services on the internet (e.g., eBay) as well as FCC auctions of spectrum licences. Recently, there has been a lot of interest in auctions with complex bidding and allocation possibilities that can capture various dependencies between a large number of items being sold. A very general model which can express such complex scenarios is that of combinatorial auctions.

In a combinatorial auction, a set of goods is to be allocated to a set of players. A utility function is associated with each player specifying the happiness of the player for each subset of the goods. One natural objective for the auctioneer is to maximize the economic efficiency of the auction, which is the sum of the utilities of all the players. Formally, the *allocation problem* is defined as follows: We have a set M of m indivisible goods and n players. Player i has a monotone utility function $v_i : 2^M \to \mathrm{R}$. We wish to find a partition $(S_1, \ldots, S_n)$ of the set of goods among the n players that maximizes the total utility or *social welfare*, $\sum_i v_i(S_i)$. Such an allocation is called an optimal allocation.

We are interested in the computational complexity of the allocation problem, and we would like an algorithm which runs in time polynomial in n and m. However, one can see that the input representation is itself exponential in m for general utility functions. Even if the utility functions have a succinct representation (polynomial in n and m), the allocation problem may be **NP**-hard [60, 4]. In the absence of a succinct representation, it is typically assumed that the auctioneer has oracle access to the players' utilities and that he can ask queries to the players. There are 2 types of queries that have been considered. In a *value query* the auctioneer specifies a subset $S \subseteq M$ and asks player i for the value $v_i(S)$. In a *demand query*, the auctioneer presents a set of prices for the goods and asks a player for the set S of goods that maximizes his profit (which is his utility for S minus the sum of the prices of the goods in S). Note that if we have a succinct representation of the utility functions then we can always simulate value queries. The problem remains hard in the query model and we are therefore interested in approximation algorithms and inapproximability results.

A natural class of utility functions that has been studied extensively in the literature is the class of submodular functions. A function v is submodular if for any 2 sets of goods $S \subseteq T$, the marginal contribution of a good $x \notin T$, is bigger when added to S than when added to T, i.e., $v(S \cup x) - v(S) \geq v(T \cup x) - v(T)$. Submodularity can be seen as the discrete analog of concavity and arises naturally in economic settings since it captures the property that marginal utilities are decreasing as we allocate more goods to a player. It is known that the class of submodular utility functions contains the functions with the Gross Substitutes property [42], and also that submodular functions are complement-free.

3.1.1 Previous Work

For general utility functions, the allocation problem is **NP**-hard. Approximation algorithms have been obtained that achieve factors of $O(\frac{1}{\sqrt{m}})$ ([61, 12], using demand queries) and $O(\frac{\sqrt{\log m}}{m})$ ([46], using value queries). It has also been shown that we cannot have polynomial time algorithms with a factor better than $O(\frac{\log m}{m})$ ([12], using value queries) or better than $O(\frac{1}{m^{1/2-\epsilon}})$ ([61, 91], even for single minded bidders). If we allow demand queries, exponential communication is still required to achieve any approximation guarantee better than $O(\frac{1}{m^{1/2-\epsilon}})$ [77]. For single-minded bidders, as well as for other classes of utility functions, approximation algorithms have been obtained, among others, in [5, 7, 61]. For more results on the allocation problem with general utilities, see [20].

For the class of submodular utility functions, the allocation problem is still **NP**-hard. The following positive results are known: In [60] it was shown that a simple greedy algorithm using value queries achieves an approximation ratio of $1/2$. An improved ratio of $1-1/e$ was obtained in [4] for a special case of submodular functions, the class of additive valuations with budget constraints. Very recently, approximation algorithms with ratio $1 - 1/e$ were obtained in [30, 31] using demand queries. As for negative results, it was shown in [77] that an exponential amount of communication is needed to achieve an approximation ratio better than $1 - O(\frac{1}{m})$. In [30] it was shown that there cannot be any polynomial time algorithm in the succinct representation or the value query model with a ratio better than 50/51, unless **P**= **NP**.

3.1.2 Our Result

We show that there is no polynomial time approximation algorithm for the allocation problem with monotone submodular utility functions achieving a ratio better than $1-1/e$, unless **P**= **NP**. Our result is true in the succinct representation model, and hence also in the value query model. The result does not hold if the algorithm is allowed to use demand queries.

A hardness result of $1 - 1/e$ for the class XOS (which strictly contains the class of submodular functions) is obtained in [30] by a gadget reduction from the maximum k-coverage problem. For a definition of the class XOS, see [60]. Similar reductions do not seem to work for submodular functions. Instead we provide a reduction from multi-prover proof systems for MAX-3-COLORING. Our result is based on the reduction of Feige [33] for the hardness of set-cover and maximum k-coverage. The results of [33] use a reduction from a multi-prover proof system for MAX-3-SAT. This is not sufficient to give a hardness result for the allocation problem, as explained in Section 3.3. Instead, we use a proof system for MAX-3-COLORING. We then define an instance of the allocation problem and show that the new proof system enables all players to achieve maximum possible utility in the yes case, and ensure that in the no case, players achieve only a $(1 - 1/e)$-fraction of the maximum utility on the average. The crucial property of the new proof system is that when a graph is 3-colorable, there are in fact many different proofs, i.e., colorings, that make the verifier accept. This would not be true if we start with a proof system for MAX-3-SAT. By introducing a correspondence between colorings and players of the allocation instance, we obtain the desired result.

The current state of the art for the allocation problem with submodular utilities, including our result, is summarized in Table 1. We note that we do not address the question of obtaining truthful mechanisms for the allocation problem. For some classes of functions,

Table 1: Approximability results for submodular utilities

	Algorithms	Hardness
Value Queries	1/2 [60]	$1-1/e$
Demand Queries	$1-1/e$ [31]	$1-O(1/m)$ [77]

incentive compatible mechanisms have been obtained that also achieve reasonable approximations to the allocation problem (e.g. [61, 5, 7]). For submodular utilities, the only truthful mechanism known is obtained in [30]. This achieves an $O(\frac{1}{\sqrt{m}})$-approximation. Obtaining a truthful mechanism with a better approximation guarantee seems to be a challenging open problem.

In the next section we define the model formally and introduce some notation. In Section 3.3 we present a weaker hardness result of 3/4 using a 2-prover proof system to illustrate the ideas in our proof. In Section 3.4 we present the hardness of $1-1/e$ based on the k-prover proof system of [33].

3.2 The Model

We assume we have a set of players $N = \{1, ..., n\}$ and a set of goods $M = \{1, ..., m\}$ to be allocated to the players. Each player has a utility function v_i, where for a set $S \subseteq M$, $v_i(S)$ is the utility that player i derives if he obtains the set S. We make the standard assumptions that v_i is monotone and that $v_i(\emptyset) = 0$.

Definition 1 *A function $v : 2^M \rightarrow R$ is submodular if for any sets $S \subset T$ and any $x \in M \backslash T$:*

$$v(S \cup \{x\}) - v(S) \geq v(T \cup \{x\}) - v(T)$$

An equivalent definition of submodular functions is that for any sets S, T: $v(S \cup T) + v(S \cap T) \leq v(S) + v(T)$.

An allocation of M is a partition of the goods $(S_1, ..., S_n)$ such that $\bigcup_i S_i = M$ and $S_i \cap S_j = \emptyset$. The allocation problem we will consider is:

> **The allocation problem with submodular utilities:** Given a monotone, submodular utility function v_i for every player i, find an allocation of the goods

$(S_1, ..., S_n)$ that maximizes $\sum_i v_i(S_i)$.

To clarify how the input is accessed, we assume that either the utility functions have a succinct representation[1], or that the auctioneer can ask value queries to the players. In a value query, the auctioneer specifies a subset S to a player i and the player responds with $v_i(S)$. In this case the auctioneer is allowed to ask at most a polynomial number of value queries.

3.3 A Hardness of 3/4

We first present a hardness result of 3/4. The reduction of this section is based on a 2-prover proof system for MAX-3-COLORING, which is analogous to the proof system of [69] for MAX-3-SAT. We remark that this proof is provided here only to illustrate the main ideas of our result and to give some intuition. In the next Section we will see that by moving to a k-prover proof system we can obtain a hardness of $1 - 1/e$.

In the MAX-3-COLORING problem, we are given a graph G and we are asked to color the vertices of G with 3 different colors so as to maximize the number of properly colored edges, where an edge is properly colored if its vertices receive different colors. Given a graph G, let $OPT(G)$ denote the maximum fraction of edges that can be properly colored by any 3-coloring of the vertices. We will start with a *gap* version of MAX-3-COLORING: Given a constant c, we denote by GAP-MAX-3-COLORING(c) the promise problem in which the yes instances are the graphs with $OPT(G) = 1$ and the no instances are graphs with $OPT(G) \leq c$. By the PCP theorem [6], and by [80], we know:

Proposition 12 *There is a constant $c < 1$ such that GAP-MAX-3-COLORING(c) is* **NP***-hard, i.e., it is* **NP***-hard to distinguish whether*
YES Case: $OPT(G) = 1$, and
NO Case: $OPT(G) \leq c$.

Let G be an instance of GAP-MAX-3-COLORING(c). The first step in our proof is a

[1] By this we mean a representation of size polynomial in n and m, such that given S and i, the auctioneer can compute $v_i(S)$ in time polynomial in the size of the representation. For example, additive valuations with budget limits [60] have a succinct representation.

reduction to the Label Cover problem. A label cover instance L consists of a graph G', a set of labels Λ and a binary relation $\pi_e \subseteq \Lambda \times \Lambda$ for every edge e. The relation π_e can be seen as a constraint on the labels of the vertices of e. An assignment of one label to each vertex is called a *labeling*. Given a labeling, we will say that the constraint of an edge $e = (u, v)$ is satisfied if $(l(u), l(v)) \in \pi_e$, where $l(u), l(v)$ are the labels of u, v respectively. The goal is to find a labeling of the vertices that satisfies the maximum fraction of edges of G', denoted by $OPT(L)$.

The instance L produced from G is the following: G' has one vertex for every edge (u, v) of G. The vertices (u_1, v_1) and (u_2, v_2) of G' are adjacent if and only if the edges (u_1, v_1) and (u_2, v_2) have one common vertex in G. Each vertex (u, v) of G' has 6 labels corresponding to the 6 different proper colorings of (u, v) using 3 colors. For an edge between (u_1, v_1) and (u_2, v_2) in G', the corresponding constraint is satisfied if the labels of (u_1, v_1) and (u_2, v_2) assign the same color to their common vertex. From Proposition 12 it follows easily that:

Proposition 13 *It is* **NP**-*hard to distinguish between:*
YES Case: $OPT(L) = 1$, *and*
NO Case: $OPT(L) \leq c'$, *for some constant* $c' < 1$

We will say that 2 labelings L_1, L_2 are *disjoint* if every vertex of G' receives a different label in L_1 and L_2. Note that in the YES case, there are in fact 6 disjoint labelings that satisfy all the constraints.

The Label Cover instance L is essentially a description of a 2-prover 1-round proof system for MAX-3-COLORING with completeness parameter equal to 1 and soundness parameter equal to c' (see [33, 69] for more on these proof systems).

Given L, we will now define a new label cover instance L', the hardness of which will imply hardness of the allocation problem. Going from instance L to L' is equivalent to applying the parallel repetition theorem of Raz [85] to the 2-prover proof system for MAX-3-COLORING.

We will denote by H the graph in the new label cover instance L'. A vertex of H is now an ordered tuple of s vertices of G', i.e., it is an ordered tuple of s edges of G, where s is a constant to be determined later . We will refer to the vertices of H as nodes to distinguish them from the vertices of G. For 2 nodes of H, $u = (e_1, ..., e_s)$ and $v = (e'_1, ..., e'_s)$, there is an edge between u and v if and only if for every $i \in [s]$, the edges e_i and e'_i have exactly one common vertex (where $[s] = \{1, ..., s\}$). We will refer to these s common vertices as the *shared* vertices of u and v. The set of labels of a node $v = (e_1, ..., e_s)$ is the set of 6^s proper colorings of its edges ($\Lambda = [6^s]$). The constraints can be defined analogously to the constraints in L. In particular, for an edge $e = (u, v)$ of H, a labeling satisfies the constraint of edge e if the labels of u and v induce the same coloring of their shared vertices.

By Proposition 13 and Raz's parallel repetition theorem [85], we can show that:

Proposition 14 *It is* **NP***-hard to distinguish between:*

YES Case: $OPT(L') = 1$, *and*

NO Case: $OPT(L') \leq 2^{-\gamma s}$, *for some constant* $\gamma > 0$.

Remark 1 *In fact, in the YES case, there are* 6^s *disjoint labelings that satisfy all the constraints.*

This property will be used crucially in the remaining section. The known reductions from GAP-MAX-3-SAT to label cover, implicit in [33, 69], are not sufficient to guarantee that there is more than one labeling satisfying all the edges. This was our motivation for using GAP-MAX-3-COLORING instead.

The final step of the proof is to define an instance of the allocation problem from L'. For that we need the following definition:

Definition 2 *A Partition System* $P(U, r, h, t)$ *consists of a universe* U *of* r *elements, and* t *pairs of sets* $(A_1, \bar{A}_1), ...(A_t, \bar{A}_t)$, $(A_i \subset U)$ *with the property that any collection of* $h' \leq h$ *sets without a complementary pair* $A_i, \bar{A}_i$ *covers at most* $(1 - 1/2^{h'})r$ *elements.*

If $U = \{0, 1\}^t$, we can construct a partition system $P(U, r, h, t)$ with $r = 2^h$ and $h = t$. For $i = 1, ..., t$ the pair $(A_i, \bar{A}_i)$ will be the partition of U according to the value of each element in the i-th coordinate. In this case the sets $A_i, \bar{A}_i$ have cardinality $r/2$.

For every edge e in the label cover instance L', we construct a partition system $P^e(U^e, r, h, t = h = 3^s)$ on a separate subuniverse U^e as described above. Call the partitions $(A_1^e, \bar{A}_1^e)$,....,$(A_t^e, \bar{A}_t^e)$.

Recall that for every edge $e = (u, v)$, there are 3^s different colorings of the s shared vertices of u and v. Assuming some arbitrary ordering of these colorings, we will say that the pair $(A_i^e, \bar{A}_i^e)$ of P^e corresponds to the ith coloring of the shared vertices.

We define a set system on the whole universe $\bigcup U^e$. For every node v and every label i, we have a set $S_{v,i}$. For every edge e incident on v, $S_{v,i}$ contains one set from every partition system P^e, as follows. Consider an edge $e = (v, w)$. Then A_j^e contributes to all the sets $S_{v,i}$ such that label i in node v induces the jth coloring of the shared vertices between v and w. Similarly $\bar{A}_j^e$ contributes to all the $S_{w,i}$ such that label i in node w gives the jth coloring to the shared vertices (the choice of assigning A_j^e to the $S_{v,i}$'s and $\bar{A}_j^e$ to the $S_{w,i}$'s is made arbitrarily for each edge (v, w)). Thus

$$S_{v,i} = \bigcup_{(v,w)\in E} B_j^{(v,w)}$$

where E is the set of edges of H, $B_j^{(v,w)}$ is one of $A_j^{(v,w)}$ or $\bar{A}_j^{(v,w)}$, and j is the coloring that label i gives to the shared vertices of (v, w).

We are now ready to define our instance I of the allocation problem. There are $n = 6^s$ players, all having the same utility function. The goods are the sets $S_{v,i}$ for each node v and label i. If a player is allocated a collection of goods $S_{v_1,i_1}...S_{v_l,i_l}$, then his utility is

$$|\bigcup_{j=1}^{l} S_{v_j,i_j}|$$

It is easy to verify that this is a monotone and submodular utility function. Let $OPT(I)$ be the optimal solution to the instance I.

Lemma 15 *If $OPT(L') = 1$, then $OPT(I) = nr|E|$.*

Proof : From Remark 1, there are $n = 6^s$ disjoint labelings that satisfy all the constraints of L'. Consider the ith such labeling. This defines an allocation to the ith player as follows: we allocate the goods $S_{v,l(v)}$ for each node v, to player i, where $l(v)$ is the label of v in this

ith labeling. Since the labeling satisfies all the edges, the corresponding sets $S_{v,l(v)}$ cover all the subuniverses. To see this, fix an edge $e = (v, w)$. The labeling satisfies e, hence the labels of v and w induce the same coloring of the shared vertices between v and w, and therefore they both correspond to the same partition of P^e, say $(A^e_j, \bar{A}^e_j)$. Thus U^e is covered by the sets $S_{v,l(v)}$ and $S_{w,l(w)}$ and the utility of player i is $r|E|$. We can find such an allocation for every player, since the labelings are disjoint. $\square$

For the No case, consider the following simple allocation: each player gets exactly one set from every node. Hence each player i defines a labeling, which cannot satisfy more than $2^{-\gamma s}$ fraction of the edges. For the rest of the edges, the 2 sets of player i come from different partitions and hence can cover at most 3/4 of the subuniverse. This shows that the total utility of this simple allocation is at most 3/4 of that in the Yes case. In fact, we will show that this is true for any allocation.

Lemma 16 *If $OPT(L') \leq 2^{-\gamma s}$, then $OPT(I) \leq (3/4 + \epsilon)nr|E|$, for some small constant $\epsilon > 0$ that depends on s.*

Proof : Consider an allocation of goods to the players. If player i receives sets $S_1, ..., S_l$, then his utility p_i can be decomposed as $p_i = \sum_e p_{i,e}$, where

$$p_{i,e} = |(\cup_j S_j) \cap U^e|$$

Fix an edge (u, v). Let m_i be the number of goods of the type $S_{u,j}$ for some j. Let m'_i be the number of goods of the type $S_{v,j}$ for some j, and let $x_i = m_i + m'_i$. Let N be the set of players. For the edge $e = (u, v)$, let N^e_1 be the set of players whose sets cover the subuniverse U^e and $N^e_2 = N \backslash N^e_1$. Let $n^e_1 = |N^e_1|$ and $n^e_2 = |N^e_2|$. Note that for $i \in N^e_1$, the contribution of the x_i sets to $p_{i,e}$ is r. For $i \in N^e_2$, it follows that the contribution of the x_i sets to $p_{i,e}$ is at most $(1 - \frac{1}{2^{x_i}})r$ by the properties of the partition system of this edge[2].

[2] To use the property of P^e, we need to ensure that $x_i \leq 3^s$. However since $i \in N^e_2$, even if $x_i > 3^s$, the distinct sets A^e_j or $\bar{A}^e_j$ that he has received through his x_i goods are all from different partitions of U_e and hence they can be at most 3^s.

Hence the total utility derived by the players from the subuniverse U^e is

$$\sum_i p_{i,e} \leq \sum_{i \in N_1^e} r + \sum_{i \in N_2^e} (1 - \tfrac{1}{2^{x_i}})r$$

In the YES case, the total utility derived from the subuniverse U^e was nr. Hence the loss in the total utility derived from U^e is

$$\Delta_e \geq nr - \sum_{i \in N_1^e} r - \sum_{i \in N_2^e} (1 - \tfrac{1}{2^{x_i}})r = r \sum_{i \in N_2^e} \frac{1}{2^{x_i}}$$

By the convexity of the function 2^{-x}, we have that

$$\Delta_e \geq r \; n_2^e \; 2^{-\frac{\sum_{i \in N_2^e} x_i}{n_2^e}}$$

But note that $\sum_{i \in N_1^e} x_i \geq 2n_1^e$, since players in N_1^e cover U^e and they need at least 2 sets to do this. Therefore $\sum_{i \in N_2^e} x_i \leq 2n_2^e$ and $\Delta_e \geq r \; n_2^e/4$. The total loss is

$$\sum_e \Delta_e \geq r/4 \sum_e n_2^e$$

Hence it suffices to prove $\sum_e n_2^e \geq (1-\epsilon)n|E|$, or that $\sum_e n_1^e \leq \epsilon n|E|$.

For an edge (u,v), let $N_1^{e,\leq s}$ be the set of players from N_1^e who have at most s goods of the type $S_{u,j}$ or $S_{v,j}$. Let $N_1^{e,>s} = N_1^e \backslash N_1^{e,\leq s}$.

$$\sum_e n_1^e = \sum_e |N_1^{e,>s}| + |N_1^{e,\leq s}| \leq \frac{2n|E|}{s} + \sum_e |N_1^{e,\leq s}|$$

where the inequality follows from the fact that for the edge e we cannot have more than $2n/s$ players receiving more than s goods from u and v.

Claim 17 $\sum_e |N_1^{e,\leq s}| < \delta n|E|$, *where* $\delta \leq c' s 2^{-\gamma s}$, *for some constant* c'.

Proof : Suppose that the sum is $\delta n|E|$ for some $\delta \leq 1$. Then it can be easily seen that for at least $\delta|E|/2$ edges, $|N_1^{e,\leq s}| \geq \delta n/2$. Call these edges *good* edges.

Pick a player i at random. For every node, consider the set of goods assigned to player i from this node, and pick one at random. Assign the corresponding label to this node. If player i has not been assigned any good from some node, then assign an arbitrary label to this node. This defines a labeling. We look at the expected number of satisfied edges.

For every good edge $e = (u, v)$, the probability that e is satisfied is at least $\delta/2s^2$, since e has at least $\delta n/2$ players that have covered U^e with at most s goods. Since there are at least $\delta|E|/2$ good edges, the expected number of satisfied edges is at least $\delta^2|E|/4s^2$. This means that there exists a labeling that satisfies at least $\delta^2|E|/4s^2$ edges. But, since $OPT(L') \leq 2^{-\gamma s}$, we get $\delta \leq c' s 2^{-\gamma s}$, for some constant c'. □

We finally have

$$\sum_e n_1^e \leq \frac{2n|E|}{s} + \delta n|E| \leq \epsilon n|E|$$

where ϵ is some small constant depending on s. Therefore the total loss is

$$\sum_e \Delta_e \geq \frac{1}{4}(1-\epsilon)nr|E|$$

which implies that $OPT(I) \leq (3/4 + \epsilon)nr|E|$. □

Given any $\epsilon > 0$, we can choose s large enough so that Lemma 16 holds. From Lemmas 15 and 16, we have:

Theorem 18 *For any $\epsilon > 0$, there is no polynomial time $(3/4+\epsilon)$-approximation algorithm for the allocation problem with monotone submodular utilities, unless* $\mathbf{P} = \mathbf{NP}$.

3.4 A Hardness of 1-1/e

In this section we obtain a stronger result by using a k-prover proof system (i.e., a label cover problem on hypergraphs) for MAX-3-COLORING. The new proof system is obtained in a similar manner as the proof system for MAX-3-SAT by Feige [33].

Let G be an instance of GAP-MAX-3-COLORING(c). From the graph G, we will define a new label cover instance. The label cover instance is now defined on a hypergraph H instead of a graph. Let s and k be constants to be determined later. The hypergraph H consists of k layers of vertices, $V_1, ..., V_k$. To every layer V_i, we associate a binary string $C_i \in \{0,1\}^s$ of weight $s/2$, with the property that the Hamming distance between any 2 strings is at least $s/3$. One can obtain such a collection of strings by using the codewords of an appropriate binary code (see [33] for more details). Notice that each C_i defines a partition of the indices $\{1, ..., s\}$ into 2 sets A_i, B_i, each of cardinality $s/2$, where an index l belongs to A_i (resp. B_i) if the l-th bit of C_i is 1 (resp. 0).

We will refer to the vertices of H as nodes. One difference from Section 3.3 is that now a node of H will contain both edges and vertices of G. To be more specific, a node v in V_i is an ordered s-tuple $v = (v^1, ..., v^s)$, where for $l \in \{1, ..., s\}$, if $l \in A_i$, then v_l is an edge of G, otherwise it is a vertex of G. Clearly there are at most $n^{O(s)}$ nodes in each layer V_i and since k and s are constants, the size of H is polynomial in the size of G.

A label of a node v in V_i will be a proper coloring of the $s/2$ edges corresponding to v and a coloring of the $s/2$ vertices corresponding to v. Hence there are $6^{s/2}3^{s/2}$ distinct labels. For technical reasons we make $6^{s/2}3^{s/2}$ copies of each label so that in total we have 6^s labels in every node.

Edges of the hypergraph H have cardinality k and contain one node from each layer. For every ordered tuple of s edges $(e_1, ..., e_s)$, of G and a choice of s vertices $(u_1, ..., u_s)$, one from each e_i, we will have a hyperedge $(v_1, ..., v_k)$ in H, with $v_i \in V_i$ if and only if the nodes $v_1, ..., v_k$ satisfy the following:

1. $v_i^l = e_l$ if $l \in A_i$.

2. $v_i^l = u_l$ if $l \in B_i$.

We will call the vertices $u_1, ..., u_s$ the *shared* vertices of the hyperedge $(v_1, ..., v_k)$. Given a labeling to the nodes of H, let $(l(v_1), ..., l(v_k))$ be the labels of the hyperedge $e = (v_1, ..., v_k)$. We will say that e is *weakly* satisfied if there exists a pair of nodes v_i, v_j that agree on the coloring of the shared vertices as induced by their labeling. We will call the pair of labels $(l(v_i), l(v_j))$ a consistent pair with respect to the hyperedge e and the labeling. We will say that a hyperedge is *strongly* satisfied if for every pair v_i, v_j, $(l(v_i), l(v_j))$ is consistent. This completes the description of the label cover instance L. Let $OPT^{weak}(L)$ (resp. $OPT^{strong}(L)$) be the maximum fraction of hyperedges that can be weakly (resp. strongly) satisfied by any labeling. The following lemma is essentially Lemma 5 in [33].

Lemma 19 *It is* **NP***-hard to distinguish between:*

YES Case: $OPT^{strong}(L) = 1$

NO Case: $OPT^{weak}(L) \leq k^2 2^{-\gamma s}$, *for some constant* $\gamma > 0$.

Remark 2 *In the YES Case of Lemma 19, there are 6^s disjoint labelings that strongly satisfy all the hyperedges.*

This follows from a similar argument as for Remark 1.

To define the instance of the allocation problem, we will first construct a set system as in Section 3.3. For this we will need a more general notion of a partition system:

Lemma 20 ([33]) *Let $U = [k]^n$. We can construct a partition system on U of the form $P = \{(A_1^1, ..., A_k^1), (A_1^2, ..., A_k^2), ..., (A_1^n, ..., A_k^n)\}$, with the properties that*

1. *For $i = 1, ..., n$, $\cup A_j^i = U$.*
2. *Any collection of $l \leq n$ sets, one from each partition, covers exactly $(1-(1-1/k)^l)|U|$ elements.*

For every hyperedge e, we will have a separate subuniverse U^e. Let $n = 6^s$ be the number of labels of each node. For each hyperedge e we construct a partition system P^e on the subuniverse U^e as in Lemma 20. Let $P^e = \{(A_{1,1}^e, ..., A_{1,k}^e), (A_{2,1}^e, ..., A_{2,k}^e), ..., (A_{n,1}^e, ..., A_{n,k}^e)\}$. Notice that for a hyperedge $e = (v_1, ..., v_k)$, we can always find n disjoint labelings of the nodes $v_1, ..., v_k$ that strongly satisfy the hyperedge e. This follows from the fact that there are 6^s proper colorings of an s-tuple of edges of G and for each such coloring we can pick a label from each node v_i that respects this coloring. Due to the multiple copies of each distinct label, we in fact have more than n labelings that strongly satisfy e. We arbitrarily pick n of these disjoint labelings (note that any other labeling that strongly satisfies e is "isomorphic" to one of the n labelings picked). Assuming some arbitrary ordering among the n labelings, we associate the jth partition of P^e with the jth labeling of e, for every e. If $(l_1^j, ..., l_k^j)$ is the jth labeling of e and $(A_{j,1}^e, ..., A_{j,k}^e)$ is the jth partition of P^e we will also say that the set $A_{j,i}^e$ corresponds to the label l_i^j of v_i.

We can now define our set system. We will have one set $S_{v,i}$ for every node v and label i. Let $v \in V_l$ for some $l \in [k]$. For an edge e that contains node v, suppose label i is in the jth labeling of e. We will then include the set $A_{j,l}^e$ from the jth partition in $S_{v,i}$. Hence

$S_{v,i}$ is the following union of sets:

$$S_{v,i} = \bigcup_{e:v\sim e} A^e_{j_e(i),l}$$

where $j_e(i)$ is the labeling of edge e that contains i.

As in Section 3.3, the instance of the allocation problem contains $n = 6^s$ players with the same submodular utility function. The goods are the sets $S_{v,i}$ and the utility of a player for a collection of sets is the cardinality of their union. Let I denote the instance of the allocation problem and let $OPT(I)$ be the optimal solution of I. Let $r = |U^e|$ and let E be the set of the hyperedges of H. Our hardness result is established by the following two lemmas.

Lemma 21 *If $OPT^{strong}(L) = 1$, then $OPT(I) = nr|E|$.*

Proof : Since $OPT^{strong}(L) = 1$, consider a labeling that strongly satisfies all the hyperedges. By the discussion above, we can always pick a labeling such that when restricted to the nodes of an edge, it corresponds to one of the n disjoint labelings of that edge. Let $l(v)$ be the label of each node. Pick a player and allocate to him all the sets $\{S_{v,l(v)}\}$. We claim that the sets cover the subuniverse U^e for every edge e and the utility of the player is therefore $r|E|$. To see this, fix an edge $e = (v_1, ..., v_k)$. Since the labeling strongly satifies the edge, it corresponds to some partition of the partition system P^e, say the jth partition. Hence for $i = 1, ..., k$, the set $A^e_{j,i}$ which corresponds to label $l(v_i)$ is contained in $S_{v_i,l(v_i)}$. Thus the player covers the entire subuniverse U^e with the sets $S_{v_i,l(v_i)}$. Since this is true for every edge, his utility is exactly $r|E|$. By Remark 2 we can repeat the above for all the 6^s players. $\square$

Lemma 22 *If $OPT^{weak}(L) \leq k^2 2^{-\gamma s}$, then $OPT(I) \leq (1 - 1/e + \epsilon)nr|E|$, where $\epsilon > 0$ is some small constant depending on s and k.*

Proof : Consider an allocation of the goods to the players, i.e., an allocation of the labels of each node. We decompose the utility p_i of player i as: $p_i = \sum p_{i,e}$, where $p_{i,e}$ is as in

Section 3.3. For a node v and a player i, let m_i^v be the number of sets of the type $S_{v,j}$ that player i has received. Fix an edge $e = (v_1, ..., v_k)$. Let $x_i^e = \sum_{l=1}^{k} m_i^{v_l}$. Define the set of players:

$$N_1^e = \{i : \exists v_j, v_l \text{ such that } i \text{ has a pair of consistent labels for these 2 nodes}\}$$

Let $N_2^e = N \backslash N_1^e$, and let $n_1^e = |N_1^e|, n_2^e = |N_2^e|$. Trivially, for $i \in N_1^e$, the contribution of the x_i^e sets to $p_{i,e}$ is at most r. For $i \in N_2^e$, the x_i^e sets of the type $S_{v_l,j}$ do not contain even one pair of labels which are consistent for some pair of nodes in e. For each set $S_{v_l,j}$ that player i has received, let $A_{t,l}^e$ be the set from the partition system P^e contained in $S_{v_l,j}$. It follows that the sets $A_{t,l}^e$ corresponding to the labels of player i come from different partitions of U^e. Therefore, by Lemma 20, we get that the sets $S_{v_l,j}$ cover exactly $1-(1-\frac{1}{k})^{x_i^e}$ fraction of the subuniverse U^e. Hence the total utility derived by the players from the subuniverse U^e is

$$\sum_i p_{i,e} \leq \sum_{i \in N_1^e} r + \sum_{i \in N_2^e} \left(1-(1-\tfrac{1}{k})^{x_i^e}\right) r$$

The loss in the total utility compared to the YES case is:

$$\Delta_e \geq nr - \sum_{i \in N_1^e} r - \sum_{i \in N_2^e} (1-(1-\tfrac{1}{k})^{x_i^e}) r = r \sum_{i \in N_2^e} (1-\frac{1}{k})^{x_i^e}$$

By the convexity of the function $(1-\frac{1}{k})^x$, we have that

$$\Delta_e \geq r n_2^e (1-\frac{1}{k})^{\frac{\sum_{i \in N_2^e} x_i^e}{n_2^e}} \tag{2}$$

Let $N_1^{e,\leq k^2}$ be the set of players from N_1^e who have at most k^2 goods of the type $S_{v_l,j}$. Let $N_1^{e,>k^2} = N_1^e \backslash N_1^{e,\leq k^2}$.

$$\sum_e n_1^e = \sum_e |N_1^{e,>k^2}| + |N_1^{e,\leq k^2}| \leq \frac{kn|E|}{k^2} + \sum_e |N_1^{e,\leq k^2}|$$

where the inequality follows from the fact that for the edge e we cannot have more than n/k players receiving more than k^2 goods from the nodes $v_1, v_2, ..., v_k$.

Claim 23 $\sum_e |N_1^{e,\leq k^2}| < \delta n|E|$, *for* $\delta \leq ck^3 2^{-\gamma s}$, *for some constant* c.

Proof : The proof is similar to that of Claim 6. If we assume the contrary to the statement then we can find a labeling which weakly satisfies more than $k^2 2^{-\gamma s}$ fraction of the edges, a contradiction. $\Box$

Hence $\sum_e n_1^e \leq \frac{n|E|}{k} + \delta n|E|$, which implies $\sum_e n_2^e \geq (1-\beta)n|E|$, for some small constant $\beta > 0$. In Section 3.3, this sufficed to obtain the hardness result of 3/4, because $\sum_{i \in N_2^e} x_i^e \leq 2n_2^e$. Here a similar argument would need that $\sum_{i \in N_2^e} x_i^e \leq kn_2^e$, which may not be true for every edge because players in N_1^e are only weakly satisfying e. However, we will see that for most edges, $\sum_{i \in N_2^e} x_i^e$ is still small.

Since $\sum_e n_1^e \leq \beta n|E|$, it follows that for at least a $1 - \sqrt{\beta}$ fraction of the edges, $n_2^e \geq (1 - \sqrt{\beta})n$. Call these edges good. For each good edge e:

$$\frac{\sum_{i \in N_2^e} x_i}{n_2^e} \leq \frac{kn}{(1 - \sqrt{\beta})n} \leq k(1 + \beta')$$

for some small constant $\beta' > 0$. From (2), we get that for every good edge the loss $\Delta_e \geq rn_2^e(1 - \frac{1}{k})^{k(1+\beta')} \geq rn_2^e(1 - \beta'')\frac{1}{e}$, for some small constant $\beta'' > 0$. Summing the loss over all the good edges, we get that the total loss in utility is at least

$$r \sum_{e: e \text{is good}} (1 - \sqrt{\beta})n(1 - \beta'')\tfrac{1}{e} \geq \frac{n}{e} r|E|(1 - \sqrt{\beta})^2(1 - \beta'') \geq \frac{1}{e} nr|E|(1 - \epsilon)$$

where $\epsilon > 0$ is some small constant. Hence the total utility is at most $(1 - \frac{1}{e} + \epsilon)nr|E|$ $\Box$

Given any $\epsilon > 0$, we can choose large enough constants s, k so that Lemma 22 holds. Hence we get:

Theorem 24 *For any $\epsilon > 0$, there is no polynomial time $(1-\frac{1}{e}+\epsilon)$-approximation algorithm for the allocation problem with monotone submodular utilities, unless* **P**=**NP**.

3.5 Conclusion

We have provided a $(1 - 1/e \simeq 0.632)$-hardness of approximation in the value query model. There is a gap between the upper and lower bounds in both the value query and demand query model. It would be interesting to narrow these gaps. It will also be interesting to obtain truthful mechanisms with good approximation guarantees.

CHAPTER IV

EQUILIBRIUM CONCEPTS IN GAMES

4.1 Introduction

Noncooperative game theory has been extensively used for modeling and analyzing situations of strategic interactions. One of the dominant solution concepts in noncooperative games is that of a Nash equilibrium [73]. Briefly, a Nash equilibrium of a game is a situation in which no agent has an incentive to deviate from her current strategy. A nice property of this concept is the well known fact that every game has at least one such equilibrium [73].

We consider the problem of computing a Nash equilibrium in finite games. The proof given by Nash for the existence of equilibria is based on Brouwer's fixed point theorem and is nonconstructive. Even for 2-player games there is still no polynomial time algorithm. The running time of all known algorithms (see among others [55, 57, 59, 62, 63]) is either exponential (in the number of available pure strategies) or has not been determined yet (and is believed to be exponential). For m-person games, $m > 2$, the problem seems to be even more difficult. While for 2-player games it can be formalized as a Linear Complementarity Problem (and hence some of the algorithms above) the problem for 3-player games is a Nonlinear Complementarity Problem. Algorithms for equilibria in multi-player games (among others, [88, 105]) are also believed to be exponential. Recently it has been shown that finding equilibria with certain natural properties (e.g. maximizing payoff) is **NP**-hard [19, 38]. The complexity of finding a single equilibrium has been of considerable interest in the computer science community and has been addressed as one of the current challenges in computational complexity [82]. In fact it is known that the problem for 2-person games lies in some class between **FP** and **FNP** [81] (the search versions of **P** and **NP**). For a summary of results on algorithms for Nash equilibria see the surveys [104, 71].

An issue related to the complexity of the problem is that even for 3-player games, there exist examples [73] in which the payoff data are rational numbers but all the Nash equilibria have irrational entries. Hence it is not even clear whether an equilibrium can be finitely represented on a Turing machine. This problem does not exist in 2-player games, in which it is known that there is always an equilibrium that can be described by polynomially sized rational numbers.

A second and related issue is the need to play simple strategies. Even if Nash strategies can be computed efficiently, they may be too complicated to implement. This has been pointed out, among others, by Simon [97] and later by Rubinstein [90] in the context of bounded rationality. Players tend to prefer a sub-optimal strategy (with respect to rationality) instead of following a complex plan of action. In this chapter, we consider normal form games and we will call a strategy *simple* if it is a uniform distribution on a small support (multi)set. The importance of small support strategies becomes clear if we consider the pure strategies to be resources. In this case an equilibrium is almost impractical if a player has to use a mixed strategy which randomizes over a large set of resources. However, there exist games whose Nash equilibria are completely mixed, i.e., a player has to randomize over all his available pure strategies.

4.1.1 Our Results

We address the above issues (namely, the need for efficient algorithms and the need for simple strategies), by using the weaker concept of ϵ-equilibrium (strategies from which each player has only an ϵ incentive to defect). In particular, we propose two algorithms for computing approximate equilibria both of which improve the previously known upper bounds on the complexity of the problem.

In Section 4.2.1 we show that for any 2-person game, there exists an ϵ-equilibrium with only logarithmic support (in the number of available pure strategies). Moreover the strategy of each player in such an equilibrium is uniform on its support set and can be expressed in polylogarithmically many bits. In our opinion, this is an interesting observation on the structure of competitive behavior in various scenarios - namely, extremely simple approximate solutions exist. Our proof is based on the probabilistic method and it directly yields a quasi-polynomial ($n^{O(\ln n)}$, where n is the number of available pure strategies) algorithm for computing such an approximate equilibrium. To our knowledge this is the first subexponential algorithm for ϵ-equilibria. In addition to having small support, our approximate equilibria provide both players with a good payoff too: the payoff that each player receives using these strategies is almost the same as that in some exact Nash equilibrium. Finally, our result can be easily generalized to multi-player games and our algorithm remains subexponential as long as the number of players is constant. It is interesting to note that although the problem of finding exact equilibria is believed to become more difficult in the "transition" from 2-player to 3-player games (due to the nonlinearity of the complementarity problem), this is not the case for approximate equilibria.

A second result (Section 4.2.2) is that if the players are allowed to communicate and "sign treaties" then there are *constant* support strategies which approximate the payoffs that each player gets in an equilibrium (in fact there are constant support strategies that approximate the payoffs of *any* pair of strategies). In real life, such treaties are not unknown (though often tacit) - this result can be considered as an explanation of why certain small strategies behave well and are used in real games, as opposed to a large and complicated Nash equilibrium.

In Section 4.2.3 we investigate the question: when does a game have small support exact Nash equilibria? We give a sufficient condition for 2-person games: if the payoff matrices of the players have low rank then there exists a Nash equilibrium with small support. Our proof is based on Caratheodory's theorem and the result has some interesting corollaries regarding the computation of Nash equilibria. In particular, we show that if the matrices can be well approximated by low rank matrices, then there exists an approximate equilibrium with small support. It also follows that if the payoff matrices have constant rank, we can compute an exact Nash equilibrium in polynomial time.

In Section 4.3, we take an algebraic approach by using the observation that Nash equilibria are essentially the roots of a single polynomial equation. Based on this, we first show that every game has at least one Nash equilibrium for which all the entries in the probability distributions are algebraic numbers and hence can be finitely represented. A finite representation has been known for 2-player games but nothing was known for games with three or more players. The current bounds for the size of the representation are exponential. We also use results from the existential theory of reals and propose a second algorithm for computing an approximate equilibrium, which runs in time $poly(\log 1/\epsilon, L, m^n)$. Here m is the number of players, n is the total number of available strategies, ϵ is the degree of approximation and L is the maximum bit size of the payoff data. For the case of two players our algorithm can be modified to compute an exact Nash equilibrium in time $2^{O(n)}$. This is yet another exponential algorithm for computing an exact equilibrium in 2-person games.

Finally we address the question of whether the existence of Nash equilibria can have any additional algebraic implications. We observe that since the set of Nash equilibria is a nonempty semi-algebraic set, this implies the existence of nontrivial solutions in certain systems of polynomial inequalities. We believe that this can be a new approach for providing simple proofs for the existence of solutions in such systems.

4.1.2 Related Work

The problem of looking for small support equilibria has been studied earlier. Koller and Megiddo [55] prove that for two person games in *extensive form* there exist equilibrium strategies whose support is at most the number of leaves of the game tree. However, not all games can be represented in the extensive form with a small number of leaves (where by small we mean logarithmic in the number of pure strategies). Our result guarantees the existence of equilibria with logarithmic support for any two person normal form game (and also for multiple players as stated above) but the equilibria are only approximate.

For the class of 2-person zero-sum games, results for approximate minmax strategies have been proved independently by Althöfer [3], Lipton and Young [69], and Newman [75]. In fact the proofs of Section 4.2.1 use the same technique (sampling). Recent algorithms for exact or approximate equilibria but only for special classes of games have been obtained among others in [66, 50, 51].

The fact that Nash equilibria are fixed points of a certain map [73] gives rise to many algorithmic approaches that are based on Scarf's algorithm [92], which is a general algorithm for approximating fixed points of continuous mappings. The worst case complexity of this algorithm and its variants is exponential in all the parameters, namely the total number of strategies, the number of players and the digits of accuracy [43]. Our first algorithm (Section 4.2.1) is subexponential in the number of strategies and exponential in the accuracy parameter and the number of players. Our second algorithm is polynomial in the digits of accuracy but exponential in the number of strategies and players. Our second algorithm is also stronger in the sense that not only players have very small incentive to deviate from the approximate equilibrium, but also the set of strategies which are output are exponentially close to some exact Nash equilibrium. This is not ensured neither by our first algorithm nor by Scarf's algorithm. More information on algorithmic approaches can be found in the surveys [71, 104].

The algebraic characterization of Nash equilibria as the set of solutions to a system of polynomial inequalities has been used before. In [99], algebraic techniques are presented for counting the number of completely mixed equilibria. In [23] it is shown that every real algebraic variety is isomorphic to the set of completely mixed Nash equilibria of some three-person game. However representation and complexity issues are not addressed there. A similar approach to ours was developed independently in [83], yielding a polynomial time algorithm for symmetric games with relatively small number of strategies.

4.2 Small Support ϵ-equilibria

We start with some definitions and notation for 2-player games. As we will see the results of Section 4.2.1 generalize to multiple player games. Consider a 2-player game where for simplicity the number of available (pure) strategies for each player is n. We will refer to the two players as the row and the column player and we will denote their payoff matrices by R, C respectively. The meaning of the payoff matrices is that if the row player chooses his ith pure strategy and the column player chooses his jth pure strategy, then they receive a payoff of R_{ij}, C_{ij} respectively.

A *mixed strategy* (or a randomized strategy) for a player is a probability distribution over the set of his pure strategies and will be represented by a vector $x = (x_1, x_2, ..., x_n)$, where $x_i \geq 0$ and $\sum x_i = 1$. Here x_i is the probability that the player will choose his

ith pure strategy. If $x_i > 0$ we say that the mixed strategy x *uses* the ith pure strategy. The *support* of x ($Supp(x)$) is the set of pure strategies that it uses. A mixed strategy is called k-*uniform* if it is the uniform distribution on a multiset S of pure strategies, with $|S| = k$. For a mixed strategy pair x, y, the payoff to the row player is the expected value of a random variable which is equal to R_{ij} with probability $x_i y_j$. Therefore the payoff to the row player is (x, Ry), where $(.\, ,.)$ denotes the inner product of two n-dimensional vectors. Similarly the payoff to the column player is (x, Cy).

The notion of a Nash equilibrium [73] is formulated as follows:

Definition 3 *A pair of strategies x^*, y^* form a Nash equilibrium if and only if:*

(i) for every (mixed) strategy $\bar{x}$ of the row player, $(\bar{x}, Ry^*) \leq (x^*, Ry^*)$,

(ii) for every (mixed) strategy $\bar{y}$ of the column player, $(x^*, C\bar{y}) \leq (x^*, Cy^*)$.

In other words, no player has an incentive to (unilaterally) deviate from his strategy. In the same manner we can define Nash equilibria for multi-player games, requiring that no player has an incentive to deviate, given that the other players do not change their strategy. Similarly we can define ϵ-equilibria:

Definition 4 *For any $\epsilon > 0$ a pair of mixed strategies x', y' is called an ϵ-Nash equilibrium if and only if:*

(i) for every (mixed) strategy $\bar{x}$ of the row player, $(\bar{x}, Ry') \leq (x', Ry') + \epsilon$,

(ii) for every (mixed) strategy $\bar{y}$ of the column player, $(x', C\bar{y}) \leq (x', Cy') + \epsilon$.

4.2.1 A Subexponential Algorithm for 2-person Games and Generalizations

For the present we assume that all entries of R and C are between 0 and 1. Our main result is:

Theorem 25 *For any Nash equilibrium x^*, y^* and for any $\epsilon > 0$, there exists, for every $k \geq \frac{12 \ln n}{\epsilon^2}$, a pair of k-uniform strategies x', y', such that:*

1. *x', y' is an ϵ-equilibrium,*

2. $|(x', Ry') - (x^*, Ry^*)| < \epsilon$ *(row player gets almost the same payoff as in the Nash equilibrium),*

3. $|(x', Cy') - (x^*, Cy^*)| < \epsilon$ *(column player gets almost the same payoff as in the Nash equilibrium).*

Proof :

The proof is based on the probabilistic method. For the given $\epsilon > 0$, fix $k \geq 12 \ln n/\epsilon^2$. Form a multiset A by sampling k times from the set of pure strategies of the row player, independently at random according to the distribution x^*. Similarly form a multiset B by sampling k times from the pure strategies of the column player, independently at random according to the distribution y^*.

Let x' be the mixed strategy for the row player which assigns probability $1/k$ to each member of A and 0 to other pure strategies. Let y' be the mixed strategy for the column player which assigns probability $1/k$ to each member of B and 0 to other pure strategies. Clearly, if a pure strategy occurs α times in the multiset, then it is assigned probability α/k.

Denote by x^i the ith pure strategy of the row player, and by y^j the jth pure strategy of the column player. In order to analyze the probability that x', y' is an ϵ-Nash equilibrium it suffices to consider only deviations to pure strategies.

We define the following events:

$$\begin{aligned}
\phi_1 &= \{|\,(x', Ry') - (x^*, Ry^*)\,| < \epsilon/2\} \\
\pi_{1,i} &= \{(x^i, Ry') < (x', Ry') + \epsilon\}, \qquad (i = 1, ..., n) \\
\phi_2 &= \{|\,(x', Cy') - (x^*, Cy^*)\,| < \epsilon/2\} \\
\pi_{2,j} &= \{(x', Cy^j) < (x', Cy') + \epsilon\}, \qquad (j = 1, ..., n) \\
GOOD &= \phi_1 \cap \phi_2 \bigcap_{i=1}^{n} \pi_{1,i} \bigcap_{j=1}^{n} \pi_{2,j}
\end{aligned}$$

We wish to show that $Pr[GOOD] > 0$. This would mean that there exists a choice of A and B such that the corresponding strategies x' and y' satisfy all three conditions in the statement of the theorem.

In order to bound the probabilities of the events ϕ_1^c and ϕ_2^c we introduce the following events:

$$\begin{aligned}
\phi_{1a} &= \{\mid (x', Ry^*) - (x^*, Ry^*) \mid < \epsilon/4\} \\
\phi_{1b} &= \{\mid (x', Ry') - (x', Ry^*) \mid < \epsilon/4\} \\
\phi_{2a} &= \{\mid (x^*, Cy') - (x^*, Cy^*) \mid < \epsilon/4\} \\
\phi_{2b} &= \{\mid (x', Cy') - (x^*, Cy') \mid < \epsilon/4\}
\end{aligned}$$

Note that $\phi_{1a} \cap \phi_{1b} \subseteq \phi_1$. The expression (x', Ry^*) is essentially a sum of k independent random variables each of expected value (x^*, Ry^*). Each such random variable takes value between 0 and 1. Therefore we can apply a standard tail inequality [45] and get:

$$Pr[\phi_{1a}^c] \leq 2e^{-k\epsilon^2/8}$$

Using a similar argument we have:

$$Pr[\phi_{1b}^c] \leq 2e^{-k\epsilon^2/8}$$

Therefore $Pr[\phi_1^c] \leq 4e^{-k\epsilon^2/8}$ and the same holds for the event ϕ_2^c.

In order to bound the probabilities of the events $\pi_{1,i}$'s and $\pi_{2,j}$'s we define the following auxilliary events:

$$\psi_{1,i} = \{(x^i, Ry') < (x^i, Ry^*) + \epsilon/2\}, \qquad (i = 1, ..., n)$$

$$\psi_{2,j} = \{(x', Ry^j) < (x^*, Ry^j) + \epsilon/2\}, \qquad (j = 1, ..., n)$$

We can easily see that

$$\psi_{1,i} \cap \phi_1 \subseteq \pi_{1,i}, \qquad (i = 1, ..., n)$$

$$\psi_{2,j} \cap \phi_2 \subseteq \pi_{2,j}, \qquad (j = 1, ..., n)$$

Using the Hoeffding bound again we get:

$$Pr[\psi_{1,i}^c] \leq e^{-k\epsilon^2/2}$$

$$Pr[\psi_{2,j}^c] \leq e^{-k\epsilon^2/2}$$

Now by combining the above equations we see that:

$$Pr[GOOD^c] \leq Pr[\phi_1^c] + Pr[\phi_2^c] + \sum_{i=1}^{n} Pr[\pi_{1,i}^c] + \sum_{j=1}^{n} Pr[\pi_{2,j}^c]$$

$$\leq 8e^{-k\epsilon^2/8} + 2n[e^{-k\epsilon^2/2} + 4e^{-k\epsilon^2/8}] < 1$$

Thus $Pr[GOOD] > 0$. □

Note that not only do the strategies x', y' form an ϵ-equilibrium, but they also provide both players with a payoff ϵ-close to the payoffs they would get in some Nash equilibrium. In fact, the payoffs of every Nash equilibrium can be thus approximated by a small strategy ϵ-equilibrium. Furthermore x', y' are k-uniform, which implies the following corollary:

Corollary 26 *For a 2-person game, there exists a quasi-polynomial algorithm for computing all k-uniform ϵ-equilibria.*

Proof : Given an $\epsilon > 0$, fix $k = \frac{12 \ln n}{\epsilon^2}$. By an exhaustive search, we can compute all k-uniform ϵ-equilibria (by Theorem 25 at least one such equilibrium exists; verifying ϵ-equilibrium condition is easy as we need to check only for deviations to pure strategies). The running time of the algorithm is quasi-polynomial since there are $\binom{n+k-1}{k}^2$ possible pairs of multisets to look at. □

To our knowledge this is the first subexponential algorithm for finding an approximate equilibrium. Furthermore, given the payoffs of any Nash equilibrium the algorithm can find an ϵ-Nash equilibrium in which both players receive payoffs ϵ-close to the given values.

When the entries of R and C are not between 0 and 1 the ϵ-incentive to defect and the ϵ-change in payoff both get magnified by $R_{max} - R_{min}$ for the row player and by $C_{max} - C_{min}$ for the column player. Here R_{max} and R_{min} denote the maximum and minimum entry of R, and similarly for C. Additionally if the players do not have the same number of pure strategies (say n_1, n_2) then the same result holds with $k \geq \frac{12 \ln max\{n_1, n_2\}}{\epsilon^2}$.

Our results can also be generalized to games with more than two players. In particular for an m-person game:

Theorem 27 *Let $x_1^*, ..., x_m^*$ be a Nash equilibrium in an m-person game. Let $p_1^*, ..., p_m^*$ be the payoffs to the players in the Nash equilibrium. Then for any $\epsilon > 0$, there exists, for every $k \geq \frac{3m^2 \ln m^2 n}{\epsilon^2}$, a set of k-uniform strategies $x_1', ..., x_m'$, such that:*

1. *$x_1', ..., x_m'$ is an ϵ-equilibrium,*

2. *$|p_i' - p_i^*| < \epsilon$ for $i = 1, ..., m$*

where $p_1', ..., p_m'$ are the payoffs to the players if they play strategies x_i'.

The proof of Theorem 27 is completely analogous to that of Theorem 25 and we omit it. As we see, we can guarantee an ϵ-equilibrium with logarithmic support only when the number of players, m, is constant. It seems to us that the technique of sampling cannot help us prove a more general theorem than that. It is an interesting question to see whether this can be done using a different technique. However, it is still interesting that we can prove the existence of simple approximate equilibria even for 3-player games. This is so because the problem of finding exact equilibria for 3-player games seems to be more difficult than for 2-player games due to the existence of irrational equilibria and the non-linearity of the Complementarity Problem.

Corollary 26 also generalizes to games with a constant number of players since in this case the number of combinations of multisets that the algorithm has to look at is still quasi-polynomial.

4.2.2 Approximating Payoffs of Nash equilibria with Constant Support

In terms of the size of the support we can do much better, if we have weaker requirements. There may be applications in which we would not even insist on an approximate equilibrium. All we would care for is to approximate the payoffs in an actual Nash equilibrium. The next result is in that direction:

Theorem 28 *For any Nash equilibrium x^*, y^* and any $\epsilon > 0$, there exists, for every $k \geq 5/\epsilon^2$, a pair of k-uniform strategies (x, y), such that*

1. $|(x, Ry) - (x^*, Ry^*)| < \epsilon$,

2. $|(x, Cy) - (x^*, Cy^*)| < \epsilon$.

Again this result can be generalized to multiple player games. For an m-person game the support of the k-uniform strategies will be $O(m^2 \ln m/\epsilon^2)$.

Theorem 28 establishes the existence of *constant* support strategies which approximate the payoffs that both players get in a Nash equilibrium. The techniques used to prove this are the same as those used to prove Theorem 25, and the proof is omitted. Again, we assume that the entries of R and C are between 0 and 1 (in the general case we get a magnification by $R_{max} - R_{min}$ and $C_{max} - C_{min}$ as before). Note that Theorem 28 is true for any pair of strategies x^*, y^*, not necessarily for Nash equilibria.

A situation in which this result could be applicable is the following: Consider a game between two players both having a very large number of pure strategies at their disposal. Let v_1, v_2 be the payoffs in a Nash equilibrium to the row and column player respectively. If the support of the equilibrium strategies is very big, then it would be preferable for both players to sign a "bilateral treaty" and use only a small number of strategies, as provided by the result. In that case, both players would still receive a payoff close to v_1 and v_2 respectively, while using a small number of strategies. Furthermore, each player will be able to check, during the game, if the other player has violated the treaty, in which case he can switch to any other strategy.

4.2.3 A Sufficient Condition for Small Support Exact Equilibria

In this section we investigate the question: when does a 2-person game have small support exact Nash equilibria? We show that if the payoff matrices have low rank then the game has a small support Nash equilibrium. Furthermore we show that if the payoff matrices can be approximated by low rank matrices then the game has a small support approximate equilibrium (where the approximation factor depends on how well the matrices can be approximated).

Suppose that the two players have n_1 and n_2 available pure strategies respectively. Then the payoff matrices, R, C, are of dimension $n_1 \times n_2$.

Theorem 29 *Let x^*, y^* be a Nash equilibrium. If $rank(C) \leq k$, then there exists a mixed strategy x for the row player with $|Supp(x)| \leq k+1$ such that x, y^* is a Nash equilibrium. Similarly, if $rank(R) \leq k$, then there exists a mixed strategy y for the column player with $|Supp(y)| \leq k+1$ such that x^*, y is a Nash equilibrium. Furthermore the payoff that both players receive in the equilibria x, y^* and x^*, y is equal to the payoff in the initial equilibrium x^*, y^*.*

Our original proof of Theorem 29 was a generalization of a result due to Raghavan [84], which deals with "completely mixed equilibria", i.e., equilibria which use all the pure strategies. The generalization was based on a careful Gaussian elimination type step. However, we now suspect that this theorem may not be unknown to the game theory community as we recently realized that a simple proof follows from the polyhedral structure of the set of Nash equilibria (see [104, 48]). We would still like to bring the theorem to the attention of the broader computer science and economics community as it has some interesting corollaries regarding the computation of Nash equilibria. We present below another simple proof, based on Caratheodory's theorem, suggested to us by N. Vishnoi and N. Devanur [29]:

Proof of Theorem 29 : Let S be the k-dimensional space spanned by the columns of R. Since Ry^* is a convex combination of the columns of R, it can be written as a convex combination of at most $k+1$ columns of R (by Caratheodory's Theorem). Let this new convex combination be Ry. Note that $Supp(y) \subseteq Supp(y^*)$. This implies that y is a best response to x^*. Since $Ry^* = Ry$, x^* is also a best reponse to y. Hence x^*, y is a Nash equilibrium. Since $Ry^* = Ry$ the first player receives the same value in x^*, y as in x^*, y^*. The second player will also receive the same value as in the initial equilibrium because $Supp(y) \subseteq Supp(y^*)$. $\square$

Definition 5 *For two matrices C, D, D is an ϵ-approximation of C if $C = D + E$, where $|E_{ij}| \leq \epsilon$ for every i, j.*

Lemma 30 *Let D be an ϵ-approximation of C. Let x^*, y^* be a Nash equilibrium for the game with payoff matrices R, D. Then x^*, y^* is a 2ϵ-Nash equilibrium for the game with payoff matrices R, C.*

Proof : Clearly $(x^*, Ry^*) \geq (\bar{x}, Ry^*)$, $\forall\ \bar{x}$. For any strategy $\bar{y}$:

$$(x^*, Cy^*) = (x^*, Dy^*) + (x^*, Ey^*) \geq (x^*, D\bar{y}) + (x^*, Ey^*)$$

Since $|E_{ij}| \leq \epsilon,\ \forall\ i, j$,

$$(x^*, E\bar{y}) - (x^*, Ey^*) \leq 2\epsilon$$

Hence,

$$(x^*, Cy^*) \geq (x^*, D\bar{y}) + (x^*, E\bar{y}) - 2\epsilon = (x^*, C\bar{y}) - 2\epsilon$$

□

Corollary 31 *For any game R, C, and for any $k < \min\{n_1, n_2\}$, if C can be ϵ-approximated by a rank k matrix, then there exists a 2ϵ-equilibrium x, y with $|Supp(x)| \leq k+1$. Similarly for R.*

In particular, we can use the Singular Value Decomposition to approximate the payoff matrices R, C by rank k matrices for any k. The approximation factor ϵ of Corollary 31 is then a function of the singular values of the matrices.

A useful corollary arises from the observation that for 2-person games, if we know the support of a Nash equilibrium, then we can compute the exact equilibrium strategies in polynomial time. This is because an equilibrium strategy y for the column player equalizes the payoff that the row player gets for every pure strategy in his support and vice versa. Hence we can write a linear program and compute the Nash equilibrium with the given support. The following is a direct consequence of this observation and Theorem 29.

Corollary 32 *If the payoff matrices R, C have constant rank, then we can compute an exact Nash equilibrium in polynomial time. In particular if one of the players has a constant number of pure strategies, we can compute a Nash equilibrium in polynomial time.*

4.3 An Algebraic Approach

In this Section, we will mainly be interested in multi-player games. We start with the observation that the set of Nash equilibria of an m-player game is the set of solutions to a system of polynomial inequalities. For this we need to introduce some additional notation. Consider a game with m players and suppose that the number of available pure strategies for player i is n_i. Let $n_0 = \max n_i$. We will denote the m-dimensional payoff matrix of player i by A^i. If players $1, \cdots, m$ play the pure strategies $j_1, \cdots, j_m$ respectively, player i receives a payoff equal to $A^i(j_1, \cdots, j_m)$. We assume that the entries of the matrices are integers, at most L bits long and $H = 2^L$ is their maximum absolute value.

A *mixed strategy* for player i will be represented by a vector $x_i = (x_{i1}, x_{i2}, \cdots, x_{i,n_i})$, where $x_{ij} \geq 0$ and $\sum x_{ij} = 1$. We denote by $\mathcal{S}_i$ the strategy space of player i, i.e., the $(n_i - 1)$-dimensional unit simplex. For an m-tuple of mixed strategies $x = (x_1, \cdots, x_m) \in \mathcal{S}_1 \times \cdots \times \mathcal{S}_m$, the expected payoff to the ith player is:

$$P^i(x) = \sum_{j_1=1}^{n_1} \cdots \sum_{j_m=1}^{n_m} A^i(j_1, \cdots, j_m) x_{1,j_1} \cdots x_{m,j_m} \tag{3}$$

For a tuple of mixed strategies $x = (x_1, \cdots, x_m)$, we will denote by x^{-i} the set of strategies: $\{x_j : j \neq i\}$. We will also denote by (x^{-i}, x_i') the tuple $(x_1, \cdots, x_{i-1}, x_i', x_{i+1}, \cdots, x_m)$, i.e., the ith player switches to the strategy x_i' while all other players keep playing the same strategy as in x. Under this notation, a Nash equilibrium is formulated as follows:

Definition 6 *A tuple of strategies $x = (x_1, \cdots, x_m) \in \mathcal{S}_1 \times \cdots \times \mathcal{S}_m$ is a Nash equilibrium if for every player i and for every mixed strategy $x_i' \in \mathcal{S}_i$, $P^i(x^{-i}, x_i') \leq P^i(x)$.*

As we have already seen in the proof of Theorem 25, it is enough to consider only deviations to pure strategies. For a player i, let s_i^j denote her jth pure strategy. Then an equivalent definition is the following: x is a Nash equilibrium if for any player i and any pure strategy of player i, s_i^j: $P^i(x^{-i}, s_i^j) \leq P^i(x)$.

Similarly, a tuple of strategies $x = (x_1, \cdots, x_m)$ is an ϵ-equilibrium if for every player i and for every pure strategy s_i^j, $P^i(x^{-i}, s_i^j) \leq P^i(x) + \epsilon$.

Another notion of approximation that we will use is that of ϵ-closeness:

Definition 7 *A point $x = (x_1, \cdots, x_m) \in \mathcal{S}_1 \times \cdots \times \mathcal{S}_m$ is ϵ-close to a point $y \in \mathcal{S}_1 \times \cdots \times \mathcal{S}_m$ if and only if $||x_i - y_i||_\infty \leq \epsilon$ for all $i = 1, ..., m$*

Note that an ϵ-equilibrium is not necessarily close to a real Nash equilibrium.

4.3.1 Nash Equilibria as Roots of a Polynomial

From the above definitions, a Nash equilibrium is a solution of the following system of polynomial inequalities and equalities:

$$
\begin{array}{ll}
x_{ij} \geq 0 & i = 1, ..., m, j = 1, ..., n_i \\
\sum_{j=1}^{n_i} x_{ij} = 1 & i = 1, ..., m \\
P^i(x^{-i}, s_i^j) \leq P^i(x) & i = 1, ..., m, j = 1, ..., n_i
\end{array}
\tag{4}
$$

Let $n = \sum n_i$. The system has n variables and $2n + m = O(n)$ multilinear constraints. By adding slack variables we can convert every constraint to an equation, where each polynomial is of degree at most m (the degree of a polynomial is the maximum total degree of its monomials). Note that the slack variables are squared so that we do not have to add any more constraints for their nonnegativity:

$$
\begin{array}{ll}
B_{ij} = x_{ij} - \beta_{ij}^2 = 0 & i = 1, ..., m, j = 1, ..., n_i \\
\Gamma_i = \sum_{j=1}^{n_i} x_{ij} - 1 = 0 & i = 1, ..., m \\
\Delta_{ij} = P^i(x) - P^i(x^{-i}, s_i^j) - \delta_{ij}^2 = 0 & i = 1, ..., m, j = 1, ..., n_i
\end{array}
\tag{5}
$$

We can now combine all the polynomial equations into one by taking the sum of squares ($P_1 = 0$ and $P_2 = 0$ is equivalent to $P_1^2 + P_2^2 = 0$, when looking for solutions over the real numbers). Therefore we have the following polynomial which we will refer to as the polynomial of the game $(A^1, ..., A^m)$:

$$
\Phi(A^1, ..., A^m) = \sum_{i=1}^{m} \sum_{j=1}^{n_i} B_{ij}^2 + \sum_{i=1}^{m} \Gamma_i^2 + \sum_{i=1}^{m} \sum_{j=1}^{n_i} \Delta_{ij}^2 \tag{6}
$$

Claim 33 $\Phi(A^1, ..., A^m)$ *has degree* $2m$*,* $O(n)$ *variables,* $n_0^{O(m)}$ *monomials and maximum absolute value of its coefficients* $O(nH^2)$*.*

4.3.2 Finite Representation of Nash Equilibria

The fact that there exist games whose equilibria have irrational entries in their probability distributions is not necessarily an obstacle towards obtaining a finite representation for a Nash equilibrium. For example, a real algebraic number α can be uniquely described by the irreducible polynomial with integer coefficients, P, for which $P(\alpha) = 0$ and an interval which isolates the root α from the other roots of P. In the next Theorem we show that every game has a Nash equilibrium that can be finitely represented. The proof is based on a deep result from the theory of real closed fields known as the transfer principle [9]. We also need to use the fact that equilibria always exist. We are not aware if there is an alternative way of proving Theorem 34. The original topological proof of existence by Nash via Brouwer's fixed point theorem, though powerful enough to guarantee an equilibrium, does not seem to give any further information on the algebraic properties of the equilibria.

Theorem 34 *For every finite game there exists a Nash equilibrium $x = (x_1, ..., x_m)$ such that every entry in the probability distributions $x_1, \cdots, x_m$ is an algebraic number.*

To prove this theorem, we first need the following definition:

Definition 8 *An ordered field R is a real closed field if*

1. *every positive element $x \in R$ is a square (i.e., $x = y^2$ for some $y \in R$).*
2. *every univariate polynomial of odd degree with coefficients in R has a root in R.*

Obviously the real numbers are an example of a real closed field. The following theorem on the existence of roots over real closed fields is known as the transfer principle and is due to Tarski and Seidenberg:

Theorem [Tarski-Seidenberg] : Let R be a real closed field and P be a polynomial with coefficients from R. Let $R' \supseteq R$ be another real closed field that contains R. Then P has a root in R if and only if it has a root in R'.

Proof of Theorem 34 : Given a game $(A^1, ..., A^m)$, the set of its Nash equilibria is the set of roots of the corresponding polynomial Φ (excluding the slack variables). By

Nash's proof [73] we know that the equation $\Phi(A^1, ..., A^m) = 0$ has a solution over the reals. Consider the field of the real algebraic numbers R_{alg}. It is known that R_{alg} is a real closed field [9]. The real numbers form a real closed field which contains R_{alg}. Since the coefficients of Φ are integers, it follows immediately from the Tarski-Seidenberg Theorem that there exists a Nash equilibrium in R_{alg}. □

A natural question is whether there are reasonable upper bounds for the degree and the coefficient size of the polynomials that represent the entries of an equilibrium. The known upper bounds are exponential. In particular, it follows by [9][Chapter 13] and by Claim 33 that the degrees of the polynomials will be $m^{O(n)}$ and the coefficient size will be $O(L + \log n)m^{O(n)}$.

4.3.3 Algorithmic Implications

We will use the formulation of Nash equilibria as roots of the polynomial Φ to propose an algorithm for computing approximate equilibria. For this we will use as a subroutine a decision algorithm for the existential theory of reals.

A special case of the decision problem for the existential theory of reals is to decide whether the equation $P(x_1, ..., x_k) = 0$ has a solution over the reals. Here P is a polynomial in k variables of degree d and with integer coefficients. The best upper bound for the complexity of this problem is $d^{O(k)}$, as provided by the algorithms of Basu et al. [8] and Renegar [86].

Theorem 35 *For an m-person game, $m \geq 2$, and for $0 < \epsilon < 1$, there is an algorithm which runs in time poly$(\log 1/\epsilon, L, m^n)$ and computes an m-tuple of strategies $x \in \mathcal{S}_1 \times \cdots \times \mathcal{S}_m$ such that:*

1. *x is ϵ/d-close to some Nash equilibrium y, where $d = 2^{m+1}n_0^m H$.*
2. *$|P^i(x) - P^i(y)| < \epsilon/2$ for all $i = 1, ..., m$.*
3. *x is an ϵ-Nash equilibrium.*

To prove Theorem 35, we need the following Lemma:

Lemma 36 *Let $y = (y_1, ..., y_m)$ be a Nash equilibrium. Let $x = (x_1, ..., x_m)$ be Δ-close to y, where $\Delta < 1$. Then:*

1. *x is an ϵ-Nash equilibrium for $\epsilon = 2^{m+1} n_0^m H\Delta$.*

2. *$|P^i(x) - P^i(y)| < \epsilon/2$ for all $i = 1, ..., m$.*

Proof :

We give a sketch of the proof. Let $\epsilon = 2^{m+1} n_0^m H\Delta$. Since x is Δ-close to y, each x_i can be written in the form $x_i = y_i + e_i$, where $e_i = (e_{i1}, ..., e_{i,n_i})$ and $|e_{ij}| \le \Delta$. For the first claim, we need to prove that for every player i, $P^i(x) \ge P^i(x^{-i}, s_i^j) - \epsilon$, for every pure strategy s_i^j. Fix a pure strategy s_i^j. Then:

$$\begin{aligned} P^i(x) &= \sum_{j_1} \cdots \sum_{j_m} A^i(j_1, ..., j_m)(y_{1,j_1} + e_{1,j_1}) \cdots (y_{m,j_m} + e_{m,j_m}) \\ &= P^i(y) + E_1 + \cdots + E_{2^m - 1} \end{aligned}$$

where each term E_i is an m-fold sum. Since y is a Nash equilibrium we have:

$$P^i(x) \ge P^i(y^{-i}, s_i^j) + \sum E_i = P^i(x^{-i}, s_i^j) + \sum F_i + \sum E_i$$

where each F_i is an $(m-1)$-fold sum similar to the E_i terms. By performing some simple calculations we can actually show that: $|\sum E_i + \sum F_i| \le \epsilon$. Hence $\sum E_i + \sum F_i \ge -\epsilon$, which proves the first claim. The second claim can also be verified along the same lines. $\square$

From now on, let $\mathcal{A}$ be an algorithm that decides whether $P(x_1, ..., x_k) = 0$ has a solution over the reals in time $d^{O(k)}$, for a degree d polynomial P (either the algorithm of [8] or [86] will do).

Proof of Theorem 35 : By Lemma 36, we only need to find an m-tuple x such that x is ϵ/d-close to some Nash equilibrium y. Let $\Phi(A^1, ..., A^m)$ be the corresponding polynomial of the game. By Claim 33, the time to compute the coefficients of all the monomials of Φ, given the payoff matrices, is $n_0^{O(m)}$ which is *poly*(m^n). We can now use $\mathcal{A}$ combined with binary search to compute a rational approximation of some root. Suppose we start with the variable x_{11}. We can add two more constraints to Φ expressing the fact that $x_{11} \in [0, 1/2]$ (by adding slack variables to the inequalties $x_{11} \geq 0$ and $x_{11} - 1/2 \leq 0$, then squaring the equations and adding them to Φ). We then run $\mathcal{A}$ for the new polynomial and if the answer is yes we know that there exists an equilibrium with $x_{11} \in [0, 1/2]$. We can replace the constraints that we added with the ones corresponding to $x_{11} \in [0, 1/4]$. If the answer is no then there exists an equilibrium with $x_{11} \in [1/4, 1/2]$, hence we can continue our binary search in that interval. Proceeding in this manner we will find an interval I_{11} with length at most $\epsilon/(n_1 d)$. For this we need to run $O(\log n_1 d/\epsilon) = O(\log 1/\epsilon + m + m \log n + L) = poly(\log 1/\epsilon, L, m, n)$ times the algorithm $\mathcal{A}$. We will then add to Φ the constraints corresponding to $x_{11} \in I_{11}$ and we will go on to the next variable. When we are done with the variable x_{1,n_i-1}, the interval I_{1,n_i} for x_{1,n_i} is also determined. This is because x_{1,n_i} should be equal to $1 - \sum_{j \neq n_i} x_{1j}$, so that x_1 is a probability distribution. Therefore the length of I_{1,n_i} will be at most ϵ/d. Hence by the end of this step we know that we can select a probability distribution x_1 for the first player such that $|x_1 - y_1|_\infty \leq \epsilon/d$ for some Nash equilibrium y. We continue the procedure to determine an interval for every variable x_{ij}. We can then output a rational number in I_{ij} for each variable so as to ensure that $x_1, ..., x_m$ are probability distributions. Note that by the end we have only added $O(n)$ additional slack variables and constraints. Therefore the total running time will be $poly(\log 1/\epsilon, L, m^n)$.

□

We can also show that for 2-person games we can compute an exact Nash equilibrium using algorithm $\mathcal{A}$ as a subroutine. The crucial observation is that if we know the support of the Nash equilibrium strategies for 2-person games, the exact strategies can be computed by solving a linear program, as explained in Section 4.2.3. By adding constraints of the form $x_{ij} = 0$ and by running $\mathcal{A}$ a linear number of times, we can identify the support of some Nash equilibrium.

Theorem 37 *There exists an algorithm that runs in time $2^{O(n)}$ and computes an exact Nash equilibrium.*

Proof : Let (A^1, A^2) be a 2-person game and let $\Phi(A^1, A^2)$ be the corresponding polynomial. We can add the constraint $x_{11} = 0$ and decide if $\Phi + x_{11}^2 = 0$ has a solution. If yes then there exists a Nash equilibrium in which player 1 does not use strategy 1 and we can go on to the second variable x_{12} while keeping the constraint $x_{11} = 0$. If on the other hand the answer was no, then we know that all Nash equilibria use strategy 1. We can proceed in the same manner: suppose that at step i, we have already included the constraints $x_{1,j_1} = 0, ..., x_{1,j_k} = 0$. We now add $x_{1i} = 0$. If the answer of $\mathcal{A}$ is yes we go to the next step and include all the constraints we have already added in our polynomial otherwise we know that all equilibria in which $x_{1,j_1} = 0, ..., x_{1,j_k} = 0$, use strategy i. After $2n$ steps we will know exactly the support of an equilibrium and then we can solve the corresponding linear program to compute it. □

This is yet another exponential algorithm for computing an exact equilibrium in 2-person games. An upper bound on the compexity of the problem can be obtained by the naive algorithm that tries all possible pairs of supports for the two players, which is $O(2^n LP_n^n) = 2^{O(n)}$, where LP_n^n is the time to solve a linear program with $O(n)$ variables and $O(n)$ constraints. Our algorithm achieves the same asymptotic bound but is in fact worse since the constant in the exponent is bigger than two. However we would still like to bring it to the attention of the community firstly because it is a different approach that has not been addressed before to the best of our knowledge and secondly because a future improvement in decision algorithms for low degree polynomial equations would directly imply an improvement in our algorithm too.

4.3.4 An application: systems of polynomial inequalities

Much of the research on equilibria in economic models has focused on the algorithmic problem of computing an equilibrium. A common approach has been to reduce the question to an already known and studied problem (e.g. fixed point approximations, linear and nonlinear complementarity problems, systems of polynomial equations and many others). In this section we would like to propose an alternative viewpoint and take advantage of the fact that Nash or market equilibria always exist. In particular, if a problem can be reduced to the existence of an equilibrium in a game or market, then we are guaranteed that a solution exists. As an example, we give the following theorem:

Theorem 38 *Let A be a $n \times n$ matrix and a_i be the i-th row of A. Let $S \subseteq \{1, ..., n\}$. Then the following system of inequalities in n variables $x = (x_1, ..., x_n)$*

$$x^T Ax - a_i x \geq 0, \quad i \in S$$

has a nonzero solution. In fact it has a probability distribution as a solution.

Proof :

Consider the *symmetric* game (A, A^T). It is known that every symmetric game has an equilibrium in which both players play the same strategy. The inequalities of the system correspond to the constraints that if both players play strategy x, a deviation to a pure strategy i, for $i \in S$ does not make a player better off. □

Deciding whether a set of polynomial equations and inequalities has a solution (or a non-trivial solution) has been an active research topic. Similar theorems can be obtained for any system that corresponds to partial constraints for the existence of Nash equilibria or market equilibria. We do not know if an algebraic proof of Theorem 38 is already known. We believe that the existence of equilibria in games and markets can yield a way of providing simple proofs for the existence of solutions in certain systems of polynomial inequalities.

4.4 Discussion

Another attempt to prove the results of Section 4.2.1 would be to approximate the vectors of a Nash equilibrium by vectors of small support. It is not difficult to see that we can approximate any probability distribution vector by a vector of logarithmic support in the l_∞ norm with error at most $1/\log n$. However, approximating an equilibrium x^*, y^* in this manner does not imply that the approximating vectors will form an ϵ-equilibrium, for any given fixed ϵ. On the other hand it can be shown that an ϵ-approximation in the l_1 norm does yield an ϵ-equilibrium, but such an approximation is not always possible (e.g. if the Nash strategies are the uniform distributions).

An interesting open question is whether we can generalize the results of Section 4.2.1 to games where the number of players is not constant. Another question would be to generalize the result so that the incentive to defect does not depend on the range of the payoff matrices (which can be much higher than the expected payoff in any equilibrium).

CHAPTER V

GAME-THEORETIC MODELS FOR INTERDOMAIN ROUTING

5.1 Introduction

The Internet, which has become a common playground for a large number of entities with selfish motives and varying degrees of collaboration, naturally gives rise to new game theoretic issues [82]. Problems that stem from Internet applications are very different from traditional algorithmic problems as the behavior of the participants is determined by their own goals and not by the instructions of the designer. It seems that such problems would require techniques and ideas from both computer science and game theory. In this chapter we focus on one specific problem, namely incentive issues in Internet routing.

The Internet is composed of many administrative domains or *Autonomous Systems* (ASes). Each AS is usually administered by a single entity. For example, a corporation or a university campus often defines an autonomous system. The connectivity of the Internet is determined by agreements between ASes for routing each other's traffic. The current protocol for routing between ASes is the Border Gateway Protocol (BGP). BGP works without a centralized authority by allowing ASes to constantly announce and exchange routing paths.

ASes can be considered independent self-interested agents, following routing policies that serve their own interests. In particular, ASes would like to satisfy their own and their customers' traffic demands and at the same time they would prefer to avoid carrying transit traffic, i.e., traffic that is neither originated nor destined to them or their customers. Avoiding transit traffic though, may result in sub-optimal efficiency and instability in the network and may even affect the network connectivity.

A question that arises naturally is whether it is possible to have a routing scheme which maximizes network efficiency and is stable in the sense that no AS or subset of ASes has an incentive to secede. In [82], Papadimitriou proposed a game theoretic formulation of this question by defining the following coalitional game: Given a network with a multicommodity flow, satisfying node capacity and demand constraints, the payoff of a node is the total flow originated or terminated at it (flow passing through a node is not included in its payoff). One of the open problems in [82] was to find sufficient conditions under which the *core* of the game is non-empty. An outcome of a game is in the core if no subset of players can collude and obtain a better payoff for its members, either viewed as a set (transferable payoff), or for each player in the coalition individually (non-transferable payoff).

We show that the core of this game is always non-empty. In the transferable case, an allocation in the core can be computed in polynomial time by solving the dual program of the multicommodity flow problem. For the same game with non-transferable payoff our proof of non-emptiness of the core is non-constructive. It is still an open question whether a core allocation can be computed efficiently for this case.

We also generalize this result to the case where a strictly concave utility function is associated with each commodity. In [52], Kelly proposed such a model for analyzing charging, rate control and routing in communication networks. An optimal outcome in his model is expressed as the solution of a non-linear program. The dual variables of that program (shadow prices) can be interpreted as actual payments of the nodes for their traffic. Using a similar argument as before, we show that if ASes compensate each other according to these shadow prices, the resulting payoff allocation is in the core.

The use of dual variables for producing an allocation in the core is not new. In [16, 24, 39, 41, 47, 49, 79, 95] classes of games are defined in which a core allocation is obtained as a function of the dual variables. In fact if the demand constraints are dropped and all the nodes have unit capacity then the non-emptiness of the core in the multicommodity flow game with transferable payoff follows from Theorem 1 in [24]. For facility-location games [16, 39] show that the dual of the facility location problem is equivalent to the problem of finding core allocations if there is no integrality gap. In some games, e.g. [95] every allocation in the core is obtained via a dual solution. However this is not the case in our game. Several complexity results have also been obtained (e.g. for testing membership or non-emptiness of the core) among others by [24, 32, 25, 18].

Incentive issues in routing have also been addressed in [76] and [34] from a mechanism design point of view. In their models each link [76] or node [34] incurs a cost for routing a packet. VCG-type payment mechanisms are obtained to make the links or the nodes behave truthfully regarding the cost of routing.

In the next section we give some definitions and results from coalitional game theory which will be used later on. In Section 5.3 we focus on the linear multicommodity flow game and prove that the core is always non-empty in both the transferable and the non-transferable case. In Section 5.4 we give a game-theoretic formulation of Kelly's non-linear model [52] and prove that again the core is non-empty. We conclude in Section 5.5 with open problems and directions for further research.

5.2 Definitions and Notation

A coalitional (or cooperative) game is determined by a set of players $N = \{1, \cdots, n\}$, a set of possible outcomes $O(S)$ for every coalition $S \subseteq N$ and a set of payoff vectors $V(S)$ corresponding to the outcomes. A payoff vector $x \in V(S)$ corresponding to the outcome $o \in O(S)$ determines the payoff of each player if the outcome o is realized. The set N is sometimes referred to as the *grand* coalition.

A solution concept in coalitional game theory is usually defined as a set of payoff allocations that are *stable* in some certain sense. Among all the solution concepts that have been proposed over the years, the *core* is probably the most intuitive one. The core consists of all payoff allocations for which no subset of players (coalition) can improve upon by cooperating only among themselves. This means that once an agreement in the core has been reached, no coalition has an incentive to secede.

We will define the core for two scenarios of coalitional games. In games with *transferable* payoff, players can compensate each other with side payments. In such games a coalition S can be completely characterized by the maximum total payoff that it can achieve in $O(S)$. We will denote this number by $v(S)$. The coalition is allowed to split the payoff $v(S)$ among its members in any possible way. The *core* of the game will be the set of payoff allocations for which no coalition can gain more.

More formally, with each such game we associate a *characteristic function* $v : P(N) \rightarrow R^+$, where $P(N)$ is the powerset of N. Following standard assumptions in the literature we require that:

(i) $v(\emptyset) = 0$.

(ii) $v(S \cup T) \geq v(S) + v(T)$, if $S \cap T = \emptyset$.

We will denote a payoff allocation by a vector $x = (x_1, \cdots, x_n),\ x_i \geq 0$, where x_i is the payoff allocated to player i. Given an allocation x, we will denote by $x(S)$ the payoff that is allocated to a coalition S, i.e., $x(S) = \sum_{i \in S} x_i$. A payoff allocation is an *imputation* if $x(N) = v(N)$. The core is the set of stable imputations:

$$\textit{core} = \{x : x(N) = v(N) \text{ and } x(S) \geq v(S) \ \ \forall S \subset N\}.$$

In games with non-transferable payoff, compensations among different players are not possible. In this case a coalition S is characterized by the set $V(S)$ of payoff vectors. The interpretation of $V(S)$ is that it contains all the possible payoff allocations that can be obtained by S. A coalition S can improve upon a payoff vector x if there exists an allocation $y \in V(S)$ such that $x_i < y_i$ for all $i \in S$. Hence the core will be:

$$\textit{core} = \{x \in V(N) : \forall S \ \nexists y \in V(S) \text{ s.t. } y_i > x_i \ \forall i \in S\}.$$

Necessary and sufficient conditions for the non-emptiness of the core in games with transferable payoff were given by Bondareva and Shapley [13, 94]. In [93], Scarf generalized their result and provided a sufficient condition in games with non-transferable payoff.

Definition 9 *Let T be a collection of coalitions. T is said to be a balanced collection if and only if we can find nonnegative weights δ_S for all $S \in T$ such that for every $i \in N$, $\sum_{S \in T : i \in S} \delta_S = 1$.*

Given a coalition S, we will call a vector u *attainable* by S if $u \in V(S)$. We will also denote by u_S the vector whose entries are the entries of u that correspond to the players of S (i.e., the projection of u to S).

Definition 10 *A game is balanced if and only if for every balanced collection T, if u is such that u_S is attainable by S, for all $S \in T$, then u is attainable by N.*

Theorem [Scarf] : Every balanced game has a non-empty core.

5.3 The Multicommodity Flow Game with Linear Utility Functions

As an attempt to address incentive issues in the Internet, Papadimitriou [82] defined the following coalitional game: let G be an undirected graph on a set of nodes N with a capacity c_i on each node and a symmetric demand matrix D (where d_{ij} is the demand between nodes i and j). Each node represents an AS and the capacity of node i is a simplification attempting to capture the capability of the corresponding subnetwork. An outcome of the game is a feasible multicommodity flow subject to demand and capacity constraints, i.e., a vector $\{f_p\}$ where for a path p from i to j, f_p is the flow exchanged between these nodes along path p. The total flow exchanged between i and j will then be equal to $f_{ij} = f_{ji} = \sum f_p$, where the sum is taken over all paths connecting i and j. Therefore the matrix $F = (f_{ij})$ will satisfy $F \leq D$. In the game with transferable payoff the value $v(S)$ for a coalition $S \subseteq N$ is the maximum flow subject to demand and capacity constraints in the graph induced by S. In other words $v(S) = \max \sum_i \sum_j f_{ij}/2$, where the maximum is taken over feasible flows. In the non-transferable case the set $V(S)$ consists of the vectors $u = (u_1, \cdots, u_{|S|})$ such that there exists a feasible flow F in the graph induced by S for which $u_i = \sum_j f_{ij}/2$. Note that for a vector $u \in V(S)$, it is not necessarily true that the sum $\sum u_i$ is equal to the maximum flow in the graph induced by S.

Finding sufficient conditions for the non-emptiness of the core was posed as an open problem in [82]. In the following subsections we will show that the core is always non-empty in both cases.

5.3.1 The Coalitional Game with Transferable Payoff

For each $i, j \in N$, let P_{ij} denote the set of all paths between i and j and let $P = \bigcup P_{ij}$. A maximum flow satisfying as much of the demands as possible is the solution of the following linear program:

$$
\begin{array}{lll}
\text{maximize} & \displaystyle\sum_{p \in P} f_p & \\
\text{subject to} & \displaystyle\sum_{p: i \in p} f_p \leq c_i & \forall i \in N \\
& \displaystyle\sum_{p \in P_{ij}} f_p \leq d_{ij} & \forall i, j \in N \\
& f_p \geq 0 & \forall p \in P
\end{array}
\tag{7}
$$

The dual program is:

$$
\begin{array}{lll}
\text{minimize} & \displaystyle\sum_{i \in N} c_i x_i + \sum_{i,j \in N} d_{ij} y_{ij} & \\
\text{subject to} & y_{ij} + \displaystyle\sum_{i \in p} x_i \geq 1 & \forall p \in P_{ij} \\
& x_i \geq 0 & \forall i \in N \\
& y_{ij} \geq 0 & \forall i, j \in N
\end{array}
\tag{8}
$$

Here the dual variable x_i corresponds to node i in the graph and the variable y_{ij} corresponds to the unordered pair of nodes (i, j).

The first part of the following theorem can also be proved by directly applying the Bondareva-Shapley theorem. However that would be a nonconstructive proof. Instead, our proof yields a polynomial time algorithm for computing an allocation in the core.

Theorem 39 *The core of the multicommodity flow game with transferable payoff is non-empty. Furthermore, a payoff allocation in the core can be computed in polynomial time.*

Proof : Consider an optimal dual solution $\{x_i\}, \{y_{ij}\}$. For each node i define its payoff to be:

$$
p_i = c_i x_i + \frac{\sum_j d_{ij} y_{ij}}{2}
$$

To show that the payoff vector $\{p_i\}$ belongs to the core we need to show that:

(i) $\sum_{i \in N} p_i = OPT(N)$.

(ii) For every subset S, $\sum_{i \in S} p_i \geq OPT(S)$.

where for $S \subseteq N$, $OPT(S)$ is the optimal value of (7) when restricted to the subgraph induced by S.

For the first part note that:

$$\sum_{i \in N} p_i = \sum_{i \in N} c_i x_i + \sum_{i,j \in N} d_{ij} y_{ij} = OPT(N)$$

by the strong duality theorem.

For the second part, consider a coalition S and the network that is induced by S. Let $i \in S, j \in S$ and $p \in P_{ij}$ such that p is entirely in the induced graph. Since $\{x_i : i \in N\}, \{y_{ij} : i, j \in N\}$ is a dual optimal (and hence feasible) solution to the original problem it holds that :

$$y_{ij} + \sum_{i \in p} x_i \geq 1$$

Therefore, $(\{x_i : i \in S\}, \{y_{ij} : i, j \in S\})$ is a dual feasible solution for the induced linear program on S. Thus:

$$\sum_{i \in S} c_i x_i + \sum_{i,j \in S} d_{ij} y_{ij} \geq OPT(S)$$

But now the following holds:

$$\begin{aligned}
\sum_{i \in S} p_i &= \sum_{i \in S} c_i x_i + \sum_{i \in S} \left(\frac{\sum_{j \in N} d_{ij} y_{ij}}{2}\right) \\
&\geq \sum_{i \in S} c_i x_i + \frac{1}{2} \sum_{i \in S} \sum_{j \in S} d_{ij} y_{ij} \\
&= \sum_{i \in S} c_i x_i + \sum_{i,j \in S} d_{ij} y_{ij} \\
&\geq OPT(S)
\end{aligned}$$

Hence $\{p_i\}$ is in the core.

The above argument directly yields a polynomial time algorithm for computing an allocation that lies in the core by solving the dual program. It should be noted here that even though the dual program in general has an exponential number of constraints, it is known that it can be solved in polynomial time [102].

□

In the payoff allocation that we constructed, each of the nodes i, j receives exactly half of the payoff term $y_{ij}d_{ij}$. It is easily seen that if we arbitrarily allocate $\alpha_{ij}y_{ij}d_{ij}$ to node i and $(1-\alpha_{ij})y_{ij}d_{ij}$ to node j for $0 \leq \alpha_{ij} \leq 1$ the resulting allocation is also in the core. We should note however that these are not the only core allocations of the game.

5.3.2 The Coalitional Game with Non-Transferable Payoff

In a coalitional game with transferable payoff, we assume that players can compensate each other with a side payment. This assumption is not justified in many cases [78].

We will show that the core of the multicommodity flow game without transferable payoff is not empty using Scarf's Theorem (Section 5.2). Thus, we only need to show that the game is balanced.

Theorem 40 *The multicommodity flow game with non-transferable payoff is balanced and hence has a non-empty core.*

Proof : Consider a balanced collection of coalitions T. Let δ_S be the corresponding weight to each coalition such that for every $i \in N$, $\sum_{S\in T, i\in S} \delta_S = 1$. Consider a payoff vector u which is attainable by every coalition $S \in T$. We need to show that u is attainable by N. For a coalition $S \in T$, since u is attainable by S, there exists a feasible flow f^S subject to demand and capacity constraints such that for every player i: $\sum_j f^S_{ij}/2 = u_i$ (f^S_{ij} is the flow routed for the commodity (i, j) in the subgraph induced by S). We construct the flow $f = \sum_{S\in T} \delta_S f^S$. For a node i the total flow that we route for the commodities containing i (divided by 2) is:

$$\frac{1}{2} \sum_{S\in T, i\in S} \delta_S \sum_j f^s_{ij} = \sum_{S\in T, i\in S} \delta_S u_i = u_i$$

It is also easy to see that this flow satisfies capacity and demand constraints. Hence u is attainable by N and the game is balanced, which implies that the core is non-empty. □

5.4 The Game with Concave Utility Functions

In [52], Kelly defines a mathematical model for analyzing issues of pricing, rate control and routing in communication networks. Similar models have also been used among others by [53, 67, 68]. The model consists of a network with a set of nodes N, a capacity for each node c_i and a set of commodities K. We will denote by P_s the set of paths that commodity s is using to send flow from its source to its sink and $P = \cup P_s$. If a commodity s is sending flow at a rate of x_s then its source and sink derive a utility of $U_s(x_s)$ where U_s is an increasing, strictly concave and continuously differentiable function (according to Shenker [96] traffic that leads to such utility functions is called *elastic* traffic). We further assume that the aggregate utility of the network for flow rates $\{x_s\}$ is $\sum_s U_s(x_s)$.

In this setting, if flow f_p is sent along each path $p \in P$ then the total flow rate for commodity s is $\sum_{p \in P_s} f_p$. To find the system's optimal rates we need to solve the following non-linear optimization problem:

$$
\begin{array}{lll}
\text{maximize} & \displaystyle\sum_{s \in K} U_s(x_s) & \\
\text{subject to} & \displaystyle\sum_{i \in p} f_p \le c_i & \forall i \in N \\
 & \displaystyle\sum_{p \in P_s} f_p = x_s & \forall s \in K \\
 & f_p \ge 0 & \forall p \in P \\
 & x_s \ge 0 & \forall s \in K
\end{array} \qquad (9)
$$

Note that unlike Section 5.3 we do not have any demand constraints. This is purely for ease of exposition and our results hold even when a demand matrix is specified.

We construct the dual of (9). Consider the Lagrangian form:

$$
\begin{aligned}
L(x, f, \lambda, \mu) &= \sum_{s \in K} U_s(x_s) + \sum_i \mu_i (c_i - \sum_{i \in p} f_p) - \sum_{s \in K} \lambda_s (x_s - \sum_{p \in P_s} f_p) \\
&= \sum_{s \in K} (U_s(x_s) - \lambda_s x_s) + \sum_{p \in P} f_p (\lambda_{s(p)} - \sum_{i \in p} \mu_i) + \sum_{i \in N} c_i \mu_i
\end{aligned}
$$

where $\lambda = \{\lambda_s : s \in K\}$, $\mu = \{\mu_i : i \in N, \mu_i \ge 0\}$ are vectors of Lagrange multipliers and for a path p, $s(p)$ denotes the commodity that the path serves. Define the function

$$D(\lambda, \mu) = \max_{x \ge 0, f \ge 0} L(x, f, \lambda, \mu)$$

We can simplify the function $D(\lambda, \mu)$ by noting that:

$$\frac{\partial L}{\partial f_p} = \lambda_{s(p)} - \sum_{i \in p} \mu_i$$

This means that at a maximum of L over the orthant $x \geq 0, f \geq 0$ the following should be true:

$$\text{If } f_p > 0 \text{ then } \lambda_{s(p)} = \sum_{i \in p} \mu_i.$$

Thus

$$\begin{aligned} D(\lambda, \mu) &= \max_{x \geq 0} \sum_{s \in K} (U_s(x_s) - \lambda_s x_s) + \sum_{i \in N} c_i \mu_i \\ &= \sum_{s \in K} \max_{x_s} (U_s(x_s) - \lambda_s x_s) + \sum_{i \in N} c_i \mu_i \end{aligned}$$

The dual program of (9) is:

$$\begin{aligned} &\text{minimize} && D(\lambda, \mu) \\ &\text{subject to} && \mu_i \geq 0 \quad \forall i \in N \end{aligned} \tag{10}$$

The objective function of (9) is differentiable and strictly concave and the feasible region is compact. Hence (9) has an optimal solution. By the duality theorem [74], there exists a dual optimal solution for (10).

As in [52, 67] the dual variables of an optimal solution (shadow prices) can be interpreted as congestion control signals. Furthermore they can also indicate actual payments to the nodes for routing traffic. In this case we show that payments defined by an optimal dual solution result in a payoff allocation which lies in the core.

As in Section 5.3 we can view the nodes of the network as players in a coalitional game with transferable payoff. The outcome of the game is again a multicommodity flow satisfying the constraints in (9) and for a coalition $S \subset N$ we define its payoff $v(S)$ to be: $v(S) = 2OPT(S)$ where $OPT(S)$ is the optimal value of (9) when restricted to the subgraph induced by the nodes in S. This is a natural generalization of the game that we studied in Section 5.3 where now the total payoff of a coalition is not the maximum flow it can send but a concave function of the maximum flow.

The question that arises of course is whether this game has a non-empty core. We will answer this question in the affirmative.

Theorem 41 *Any optimal solution (λ, μ) to the dual program (10) gives rise to a payoff allocation which is in the core.*

Proof : The argument is essentially the same as in the proof of Theorem 39. For a node $i \in N$, let $K(i)$ be the set of commodities in which i is either a source or a sink. We can define the following payoff allocation to the nodes:

$$p_i = \sum_{s \in K(i)} \max_{x_s}(U_s(x_s) - \lambda_s x_s) + 2c_i\mu_i$$

To show that $p = \{p_i\}$ is in the core note first that:

$$\sum_{i \in N} p_i = \sum_{i \in N}\Big(\sum_{s \in K(i)} \max_{x_s}(U_s(x_s) - \lambda_s x_s) + 2c_i\mu_i\Big)$$
$$= 2\sum_{s \in K} \max_{x_s}(U_s(x_s) - \lambda_s x_s) + 2\sum_{i \in N} c_i\mu_i = v(N)$$

Therefore it remains to show that for every coalition S, $\sum_{i \in S} p_i \geq 2OPT(S)$. Consider a coalition S and the dual variables that correspond to commodities and nodes in the subgraph induced by S. These variables form a feasible solution to the dual of (9) when restricted to this subnetwork. Therefore we have:

$$\sum_{i \in S} p_i = 2D(\lambda, \mu; S) \geq 2OPT(S) = v(S)$$

where by $D(\lambda, \mu; S)$ we denote the dual objective function restricted to the subnetwork of S. Hence the allocation $\{p_i\}$ lies in the core.

□

We should also note that our proof for the non-transferable case in Section 5.3.2 also holds when the utilities are concave functions of the flow, which is the case here.

5.5 Discussion

In [54], Kelly and Vazirani showed that the problem of charging and rate control as defined in Kelly [52] can be seen as a generalization of Fisher's market equilibrium problem [14, 28].

The optimum dual variables in that model correspond to market clearing prices. Hence, Theorem 41 on core allocations in section 5.4 is along the same lines of the classic result in coalitional game theory that allocations corresponding to an equilibrium in the market lies in the core [78].

The core of a game is a useful concept in a cooperative setting where all the information regarding preferences, capacities and demands is known to all agents. Clearly this is not the case in the Internet. Moreover, in the core allocation that we constructed in sections 5.3.1 and 5.4, the payoff that a node receives depends on its capacity. It can be seen by using complementary slackness conditions that a node might receive a bigger payoff if it announces a smaller capacity. It is an interesting problem to design a distributed strategy-proof mechanism such that no node has an incentive to lie about its capacity. For related results on algorithmic mechanism design see [76, 34, 35].

The proof of non-emptiness of the core in Section 5.3.2 is based on Scarf's Theorem which is non-constructive. An open problem is to find an algorithm for computing a solution in the core efficiently.

Finally, in the non-linear model, if each commodity uses only one path, it is shown in [37, 67] that shadow prices can be computed by a distributed algorithm where the local computation is done on each link (on each AS in our case). We are not aware of any result for the general case.

REFERENCES

[1] Alkan, A., Demange, G., and Gale, D., "Fair allocation of indivisible goods and criteria of justice," *Econometrica*, vol. 59, pp. 1023–1039, 1991.

[2] Alon, N., Azar, Y., Woeginger, G., and Yadid, T., "Approximation schemes for scheduling on parallel machines," *Journal of Scheduling*, vol. 98, pp. 55–66, 1998.

[3] Althöfer, I., "On sparse approximations to randomized strategies and convex combinations," *Linear Algebra and Applications*, vol. 199, pp. 339–355, 1994.

[4] Andelman, N. and Mansour, Y., "Auctions with budget constraints," in *SWAT*, 2004.

[5] Archer, A., Papadimitriou, C., Talwar, K., and Tardos, E., "An approximate truthful mechanism for combinatorial auctions with single parameter agents," in *SODA*, pp. 205–214, 2003.

[6] Arora, S., Lund, C., Motwani, R., Sudan, M., and Szegedy, M., "Proof verification and hardness of approximation problems," in *FOCS*, pp. 14–23, 1992.

[7] Bartal, Y., Gonen, R., and Nisan, N., "Incentive compatible multi unit combinatorial auctions," in *TARK*, pp. 72–87, 2003.

[8] Basu, S., Pollack, R., and Roy, M. F., "On the combinatorial and algebraic complexity of quantifier elimination," *Journal of the ACM*, vol. 43, no. 6, pp. 1002–1045, 1996.

[9] Basu, S., Pollack, R., and Roy, M. F., *Algorithms in Real Algebraic Geometry.* Springer-Verlag, 2003.

[10] Bazkan, C., Santha, M., and Tuza, Z., "Efficient approximation algorithms for the subset-sum problem," in *ICALP*, pp. 387–396, 1991.

[11] Bezakova, I. and Dani, V., "Allocating indivisible goods." Unpublished manuscript, 2005.

[12] Blumrosen, L. and Nisan, N., "On the computational power of ascending auctions i: Demand queries." ACM Conference on Electronic Commerce, 2005.

[13] Bondareva, O., "The core of an n person game," *Vestnik Leningrad University*, vol. 13, pp. 141–142, 1962.

[14] Brainard, W. C. and Scarf, H. E., "How to compute equilibrium prices in 1891," *Cowles Foundation Discussion Paper*, vol. 1270, 2000.

[15] Brams, S. J. and Taylor, A. D., *Fair Division: from Cake Cutting to Dispute Resolution.* Cambrige University press, 1986.

[16] Chardaire, P., *Facility Location Optimization and Cooperative Games*. PhD Thesis, 1998.

[17] Coffman, E. G. and Langston, M. A., "A performance guarantee for the greedy set-partitioning algorithm," *Acta Informatica*, vol. 21, pp. 409–415, 1984.

[18] Conitzer, V. and Sandholm, T., "Complexity of determining non-emptiness of the core," Tech. Rep. CS-02-137, CMU, 2002.

[19] Conitzer, V. and Sandholm, T., "Complexity results about nash equilibria," in *International Joint Conference on Artificial Intelligence*, 2003.

[20] Cramton, P., Shoham, Y., and Steinberg(editors), R., *Combinatorial Auctions*. MIT Press, Forthcoming (2005).

[21] Czumaj, A. and Voecking, B., "Tight bounds for worst-case equilibria," in *Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 413–420, 2002.

[22] Dall'Aglio, M. and Hill, T., "Maximin-share and minimax-envy in fair division problems," *Journal of Mathematical Analysis and Applications*, 2003.

[23] Datta, R., "Universality of nash equilibria," *Math. Op. Res.*, vol. 28, no. 3, pp. 424–432, 2003.

[24] Deng, X., Ibaraki, T., and Nagamochi, H., "Algorithms and complexity in combinatorial optimization games," in *Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 720–729, 1997.

[25] Deng, X. and Papadimitriou, C., "On the complexity of cooperative game solution concepts," *Mathematics of Operations Research*, vol. 19, no. 2, pp. 257–266, 1994.

[26] Deng, X., Papadimitriou, C. H., and Safra, M., "On the complexity of equilibria," in *ACM Symposium on Theory of Computing*, pp. 67–71, 2002.

[27] Deuermeyer, B. L., Friesen, D. K., and Langston, M. A., "Scheduling to maximize the minimum procesor finish time in a multiprocessor system," *SIAM Journal of Algebraic and Discrete Methods*, vol. 3, pp. 190–196, 1982.

[28] Devanur, N., Saberi, A., Papadimitriou, C., and Vazirani, V. V., "Market equilibrium via a primal-dual-type algorithm," in *Annual IEEE Symposium on the Foundations of Computer Science*, pp. 389–395, 2002.

[29] Devanur, N. and Vishnoi, N., "Private communication," 2003.

[30] Dobzinski, S., Nisan, N., and Schapira, M., "Approximation algorithms for combinatorial auctions with complement-free bidders," in *STOC*, 2005.

[31] Dobzinski, S. and Schapira, M., "An improved approximation algorithm for combinatorial auctions with submodular bidders." Working paper, 2005.

[32] Faigle, U., Fekete, S., Hochstattler, W., and Kern, W., "On the complexity of testing membership in the core of min-cost spanning tree games," *International Journal of Game Theory*, vol. 26, pp. 361–366, 1997.

[33] Feige, U., "A threshold of lnn for approximating set cover," *Journal of the ACM*, vol. 45, no. 4, pp. 634–652, 1998.

[34] Feigenbaum, J., Papadimitriou, C., Sami, R., and Shenker, S., "A bgp-based mechanism for lowest-cost routing," in *Symposium on Principles of Distributed Computing*, pp. 173–182, 2002.

[35] Feigenbaum, J. and Shenker, S., "Distributed algorithmic mechanism design: Recent results and future directions," in *International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, pp. 1–13, ACM Press, 2002.

[36] Foley, D., "Resource allocation and the public sector," *Yale Economics Essays*, vol. 7, pp. 45–98, 1967.

[37] Garg, N. and Young, N., "On-line end-to-end congestion control," in *Annual IEEE Symposium on the Foundations of Computer Science*, pp. 303–312, 2002.

[38] Gilboa, I. and Zemel, E., "Nash and correlated equilibria: Some complexity considerations," *Games and Economic Behavior*, vol. 15, no. 5, pp. 745–770, 1989.

[39] Goemans, M. and Skutella, M., "Cooperative facility location games," in *Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 76–85, 2000.

[40] Graham, R. L., "Bounds on multiprocessing timing anomalies," *SIAM Journal of Applied Mathematics*, vol. 17, pp. 416–429, 1969.

[41] Granot, D., "A generalized linear production model: A unified model," *Mathematical Programming*, vol. 34, pp. 212–222, 1986.

[42] Gul, F. and Stacchetti, E., "Walrasian equilibrium with gross substitutes," *Journal of Economic Theory*, vol. 87, pp. 66–95, 2000.

[43] Hirsch, M. D., Papadimitriou, C. H., and Vavasis, S. A., "Exponential lower bounds for finding brouwer fixed points," *Journal of Complexity*, vol. 5, pp. 379–416, 1989.

[44] Hochbaum, D. S. and Shmoys, D. B., "Using dual approximation algorithms for scheduling problems," *Journal of the ACM*, vol. 34, pp. 144–162, 1987.

[45] Hoeffding, W., "Probability inequalities for sums of bounded random variables," *American Statistical Journal*, pp. 13–30, March 1963.

[46] Holzman, R., Kfir-Dahav, N., Monderer, D., and Tennenholtz, M., "Bundling equilibrium in combinatorial auctions," *Games and Economic Behavior*, vol. 47, pp. 104–123, 2004.

[47] Jain, K. and Vazirani, V. V., "Application of approximation algorithms to cooperative games," in *Annual Symposium on Theory of Computing*, pp. 364–372, 2001.

[48] Jurg, A. P., "Some topics in the theory of bimatrix games." www.ub.rug.nl/eldoc/dis/non-rug/a.p.jurg/.

[49] Kalai, E. and Zemel, E., "Generalized network problems yielding totally balanced games," *Operations Research*, vol. 30, pp. 998–1008, 1982.

[50] Kearns, M. J., Littman, M. L., and Singh, S. P., "Graphical models for game theory," in *UAI*, pp. 253–260, 2001.

[51] Kearns, M. J. and Mansour, Y., "Efficient nash computation in large population games with bounded influence," in *UAI*, 2002.

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

[53] Kelly, F. P., Maulloo, A., and Tan, D., "Rate control in communication networks: Shadow prices, proportional fairness and stability," *Journal of the Operational Research Society*, vol. 49, pp. 237–252, 1998.

[54] Kelly, F. P. and Vazirani, V. V., "Rate control as a market equilibrium." In preparation, 2003.

[55] Koller, D. and Megiddo, N., "Finding mixed strategies with small support in extensive form games," *International Journal of Game Theory*, vol. 25, pp. 73–92, 1996.

[56] Koller, D., Megiddo, N., and von Stengel, B., "Fast algorithms for finding randomized strategies in game trees," in *Annual ACM Symposium on the Theory of Computing*, pp. 750–759, 1994.

[57] Koller, D., Megiddo, N., and von Stengel, B., "Efficient computation of equilibria for extensive two-person games," *Games and Economic Behavior*, vol. 14, no. 2, pp. 247–259, 1996.

[58] Koutsoupias, E. and Papadimitriou, C. H., "Worst case equilibria," in *Annual IEEE Symposium on Theoretical Aspects of Computer Science*, pp. 404–413, 1999.

[59] Kuhn, H. W., "An algorithm for equilibrium points in bimatrix games.," in *Proceedings of the National Academy of Sciences*, pp. 1657–1662, 1961.

[60] Lehmann, B., Lehmann, D., and Nisan, N., "Combinatorial auctions with decreasing marginal utilities," in *ACM Conference on Electronic Commerce*, 2001.

[61] Lehmann, D., O'Callaghan, L., and Shoham, Y., "Truth revelation in approximately efficient combinatorial auctions," in *ACM Conference on Electronic Commerce*, 1999.

[62] Lemke, C. E., "Bimatrix equilibrium points and mathematical programming," *Management Science*, vol. 11, pp. 681–689, 1965.

[63] Lemke, C. E. and Howson, J. T., "Equilibrium points of bimatrix games," *Journal of the Society for Industrial and Applied Mathematics*, vol. 12, pp. 413–423, 1964.

[64] Lenstra, H. W., "Integer programming with a fixed number of variables," *Math. Oper. Res.*, vol. 8, pp. 538–548, 1983.

[65] Lenstra, J. K., Shmoys, D. B., and Tardos, E., "Approximation algorithms for scheduling unrelated parallel machines," *Mathematical Programming*, vol. 46, pp. 259–271, 1990.

[66] Littman, M. L., Kearns, M. J., and Singh, S. P., "An efficient, exact algorithm for solving tree-structured graphical games," in *NIPS*, pp. 817–823, 2001.

[67] Low, S. and Lapsley, D., "Optimization, flow control 1: Basic algorithm and convergence," *IEEE Transactions on Networking*, vol. 7, no. 6, pp. 861–874, 1999.

[68] Low, S., Paganini, F., and Doyle, J., "Internet congestion control," *IEEE Control Systems Magazine*, vol. 22, pp. 28–43, Feb. 2002.

[69] Lund, C. and Yannakakis, M., "On the hardness of approximating minimization problems," *Journal of the ACM*, vol. 41, no. 5, pp. 960–981, 1994.

[70] Mangasarian, O. L., "Equilibrium points of bimatrix games," *Journal of the Society for Industrial and Applied Mathematics*, vol. 12, no. 4, pp. 778–780, 1964.

[71] McKelvey, R. and McLennan, A., "Computation of equilibria in finite games," *Amman, H., Kendrick, D., Rust, J. eds, Handbook of Computational Economics*, vol. 1, 1996.

[72] Moulin, H. and Vial, L., "Strategically zero-sum games: The class of games whose completely mixed equilibria cannot be improved upon," *International Journal of Game Theory*, vol. 7, no. 3, pp. 201–221, 1978.

[73] Nash, J. F., "Non-cooperative games," *Annals of Mathematics*, vol. 54, pp. 286–295, 1951.

[74] Nemirovskii, A. and Ben-Tal, A., "Lectures on modern convex optimization: Analysis, algorithms, and engineering applications." MPS-SIAM Series on Optimization, 2001.

[75] Newman, I., "Private vs. common random bits in communication complexity," *Information Processing Letters*, vol. 39, pp. 67–71, 1991.

[76] Nisan, N. and Ronen, A., "Algorithmic mechanism design," in *Annual ACM Symposium on the Theory of Computing*, pp. 129–140, 1999.

[77] Nisan, N. and Segal, I., "The comminication requirements of efficient allocations and supporting lindahl prices." To appear in Journal of Economic Theory, preliminary version available at http://www.cs.huji.ac.il/ noam/mkts.html, 2004.

[78] Osborne, M. and Rubinstein, A., *A Course in Game Theory.* Cambridge, Massachusetts: MIT Press, 1994.

[79] Owen, G., "On the core of linear production games," *Mathematical Programming*, vol. 9, pp. 358–370, 1975.

[80] Papadimitriou, C. and Yannakakis, "Optimization, approximation and complexity classes," *Journal of Computer and System Sciences*, vol. 43, pp. 425–440, 1991.

[81] Papadimitriou, C. H., "On the complexity of the parity argument and other inefficient proofs of existence," *Journal of Computer and System Sciences*, vol. 48, no. 3, 1994.

[82] Papadimitriou, C. H., "Algorithms, games, and the internet," in *Annual ACM Symposium on the Theory of Computing*, pp. 749–753, 2001.

[83] Papadimitriou, C. H. and Roughgarden, T., "Computing equilibria in multi-player games," in *Annual ACM-SIAM Symposium on Discrete Algorithms*, 2005.

[84] Raghavan, T. E. S., "Completely mixed strategies in bimatrix games," *Journal of London Math Society*, vol. 2, no. 2, pp. 709–712, 1970.

[85] Raz, R., "A parallel repetition theorem," *SIAM Journal of Computing*, vol. 27, no. 3, pp. 763–803, 1998.

[86] Renegar, J., "On the computational complexity and geometry of the first order theory of reals," *Journal of Symbolic Computation*, vol. 13, no. 3, pp. 255–352, 1992.

[87] Robertson, J. M. and Webb, W. A., *Cake Cutting Algorithms: be fair if you can.* AK Peters, 1998.

[88] Rosenmuller, J., "On a generalization of the lemke-howson algorithm to non-cooperative games," *SIAM Journal of Applied Mathematics*, vol. 21, pp. 73–79, 1971.

[89] Roughgarden, T. and Tardos, E., "How bad is selfish routing," in *Annual IEEE Symposium on Foundations of Computer Science*, pp. 93–102, 2000.

[90] Rubinstein, A., *Modeling Bounded Rationality.* Cambridge, Massachusetts: MIT Press, 1998.

[91] Sandholm, T., "An algorithm for optimal winner determination in combinatorial auctions," in *IJCAI*, 1999.

[92] Scarf, H., "The approximation of fixed points of a continuous mapping," *SIAM Journal of Applied Mathematics*, vol. 15, pp. 1328–1343, 1967.

[93] Scarf, H., "The core of an n person game," *Econometrica*, vol. 35, no. 1, pp. 50–69, 1967.

[94] Shapley, L., "On balanced sets and cores," *Naval Research Logistics Quarterly*, vol. 14, pp. 453–460, 1967.

[95] Shapley, L. and Shubik, M., "The assignment game 1: The core," *International Journal of Game Theory*, vol. 1, pp. 111–130, 1972.

[96] Shenker, S., "Fundamental design issues for the future internet," *IEEE Journal Selected Areas Communication*, vol. 13, pp. 1176–1188, 1995.

[97] Simon, H., *Models of Bounded Rationality, Volume 2.* Cambridge, Massachusetts: MIT Press, 1982.

[98] Steinhaus, H., "The problem of fair division," *Econometrica*, vol. 16, pp. 101–104, 1948.

[99] Sturmfels, B., "Solving systems of polynomial equations." Regional Conference Series in Mathematics, 97, 2002.

[100] Tadenuma, K. and Thomson, W., "The fair allocation of an indivisible good when monetary compensations are possible," *Mathematical Social Sciences*, vol. 25, no. 2, pp. 117–132, 1993.

[101] Varian, H., "Equity, envy and efficiency," *Journal of Economic Theory*, vol. 9, pp. 63–91, 1974.

[102] Vazirani, V. V., *Approximation Algorithms*. Springer-Verlag, 2001.

[103] Vetta, A., "Nash equilibria in competitive societies with applications to facility location, traffic routing and auctions," in *Annual IEEE Symposium on Foundations of Computer Science*, pp. 416–425, 2002.

[104] von Stengel, B., "Computing equilibria for two-person games," *Aumann, R. and Hart, S. eds, Handbook of Game Theory*, vol. 3, 2002.

[105] Wilson, I., "Computing equilibria of n-person games," *SIAM Journal of Applied Mathematics*, vol. 21, pp. 80–87, 1971.

[106] Woeginger, G., "A polynomial time approximation scheme for maximizing the minimum machine completion time," *Operations Research Letters*, vol. 20, pp. 149–154, 1997.

VITA

Personal Data

Birth: September 28, 1977, Athens, Greece

Address: 337410 Georgia Tech Station, Atlanta, GA 30332, USA

E-mail: vangelis@cc.gatech.edu

Homepage: `http://www.cc.gatech.edu/~vangelis`

Education

Georgia Institute of Technology, Atlanta, Georgia, USA

Fall 2000 - Summer 2005:

M.Sc. in Computer Science

Ph.D. in Computer Science (Algorithms, Combinatorics, and Optimization Program)

Thesis: Computational Aspects of Game Theory and Microeconomics

Advisor: Richard Lipton

National Technical University of Athens, Athens, Greece

Fall 1995 - Summer 2000:

Diploma (5-year B.Sc. degree with thesis) in Electrical and Computer Engineering

Thesis: Logics of Knowledge and Belief with Applications to Artificial Intelligence

Advisor: Efstathios Zachos

Research Interests

Approximation Algorithms, Combinatorial Optimization,

Computational issues in Game Theory and Mathematical Economics, Auction Design,

Machine Learning, Robotics (multi-robot routing applications).

Honors and Awards

Best student paper award (ACM Conference on Electronic Commerce, EC 2003).

Dean's fellowship, College of Computing, Georgia Institute of Technology, 2000-2001.

Annual scholarship from the State Scholarships Foundation of Greece, 1995-2000.

Ranked 2nd in nationwide exams for college admission, Greece, 1995.

Award from the Greek Mathematical Society, 1994.

Professional Experience

Georgia Institute of Technology, Atlanta, Georgia, USA

Research Assistant: Fall 2000 - Fall 2001 and Summer 2002 - present

Teaching Assistant for the course 'Introduction to the Theory of Computation', Spring 2002.

National Technical University of Athens, Athens, Greece

Teaching Assistant for the course 'Compilers', Spring 2000.

Educational Software: design and implementation of an operating systems simulator, now used in the high school 'Operating Systems' course in Greece, 1999.

Publications

R. Lipton, E. Markakis, A. Mehta, N. Vishnoi. On the Fourier Spectrum of Symmetric Boolean Functions with Applications to Learning Symmetric Juntas, *IEEE Conference on Computational Complexity* (CCC '05), 2005.

D. Kempe, P. Keskinocak, A. Kleywegt, S. Koenig, M. Lagoudakis, E. Markakis, A. Meyerson, C. Tovey, S. Jain. Auction-based Multi-robot Routing, *Robotics: Science and Systems*, 2005.

R. Lipton, E. Markakis, E. Mossel, and A. Saberi, On Approximately Fair Allocations of Indivisible Goods, *5th ACM Conference on Electronic Comerce* (EC '04), pp. 125–131, 2004.

R. Lipton, E. Markakis. Nash Equilibria via Polynomial Equations, *Latin American Theoretical Informatics* (LATIN '04), pp. 413–422, 2004.

K. Jain, M. Mahdian, E. Markakis, A. Saberi, V. V. Vazirani. Greedy Facility Location Algorithms Analyzed using Dual Fitting with Factor-Revealing LP, *Journal of the ACM*, 50 (6), pp. 795–824, November 2003.

E. Markakis, A. Saberi. On the Core of the Multicommodity Flow Game, *4th ACM Conference on Electronic Commerce* (EC '03), pp. 93-97, 2003. Recipient of the best student paper award. Invited for publication in special issue of *Decision Support Systems*.

R. Lipton, E. Markakis, A. Mehta. Playing Large Games using Simple Strategies, *4th ACM Conference on Electronic Commerce* (EC '03), pp. 36-41, 2003.

M. Mahdian, E. Markakis, A. Saberi, V. V. Vazirani. A Greedy Facility Location Algorithm Analyzed using Dual Fitting, *Proceedings of 5th International Workshop on Randomization and Approximation Techniques in Computer Science* (APPROX-RANDOM), pp. 127–137, 2001.

I. Kerenidis, E. Markakis, K. Potika, C. Siaterlis. MEDUSA: An Operating Systems Simulator, *Internet- Education-Science*, pp. 52-54, 2000.

Papers under review or in preparation

S. Khot, R. Lipton, E. Markakis, A. Mehta. Inapproximability Results for Combinatorial Auctions with Submodular Utility Functions, submitted, 2005.

M. Kolountzakis, E. Markakis, A. Mehta. Learning Symmetric Juntas in time $n^{o(k)}$, submitted, 2005.