COMBINATORIAL AND EXCHANGE MARKETS: ALGORITHMS, COMPLEXITY, AND APPLICATIONS

A Dissertation
Presented to
The Academic Faculty

By

Sadra Yazdanbod

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
Algorithms, Combinatorics and Optimization

Georgia Institute of Technology

May 2018

COMBINATORIAL AND EXCHANGE MARKETS:
ALGORITHMS, COMPLEXITY, AND APPLICATIONS

Committee Members:

Dr. Vijay V. Vazirani, Advisor
Computer Science Department
University of California, Irvine

Dr. Ruta Mehta
Department of Computer Science
University of Illinois at Urbana-Champaign

Dr. Jamie Morgenstern
School of Computer Science
Georgia Institute of Technology

Dr. Prasad Tetali
School of Computer Science
Georgia Institute of Technology

Dr. Craig Tovey
School of Industrial and Systems Engineering
Georgia Institute of Technology

Expected Defense Date: May 10, 2018

ACKNOWLEDGEMENTS

I would like to thank my advisor Vijay Vazirani, for his support throughout my graduate studies and for being so patient with me. His guidance, patience and the fruitful discussions helped me to grow not only as a researcher, but also as a person.

I want to thank my collaborators for investing the time to work with me on research problems. I am especially indebted to Ruta Mehta whose valuable insights helped me move forward with my research many times.

Finally I want to thank my family who always supported me and helped me pursue my interests.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

In today's world, globalization and the Internet have resulted in the creation of enormously many different kinds of marketplaces. The marketplaces naturally tend to find an "equilibrium" in terms of prices and interaction of agents. Therefore, understanding the equilibria results in better understanding and prediction of the marketplaces. In this thesis, we study two broad classes of equilibria. The first one is called market price equilibria, which can explain and predict prices within a market. The second one is Nash equilibria (NE), which is arguably the most important and well-studied solution concept within game theory. NE helps us to explain and predict the interactions between agents within a market.

We can summarize the main contributions as follows:

- **Combinatorial markets with covering constraints.** We introduce a new class of combinatorial markets in which agents have covering constraints over resources required and are interested in delay minimization. Our market model is applicable to several settings including scheduling and communicating over a network. We give a proof of the existence of equilibria and a polynomial time algorithm for finding one, drawing heavily on techniques from LP duality and submodular minimization. Finally, we show that our model inherits many of the fairness properties of traditional equilibrium models as well as new models, such as Competitive Equilibrium with Equal Incomes (CEEI).

- **Settling the complexity of Leontief and PLC exchange, markets under exact and approximate equilibria.** We show FIXP-hardness of computing equilibria in Arrow-Debreu exchange markets under Leontief utility functions, and Arrow-Debreu markets under linear utility functions and Leontief production sets, thereby settling these open questions of Vazirani and Yannakakis (2011). As a consequence of the results stated above, and the fact that membership in FIXP has been established for PLC utilities, the entire computational

difficulty of Arrow-Debreu markets under PLC utility functions lies in the Leontief utility subcase. Finally, we give a polynomial time algorithm for finding an equilibrium in Arrow-Debreu exchange markets under Leontief utility functions provided the number of agents is a constant. This settles part of an open problem of Devanur and Kannan (2008).

- **$\exists\mathbb{R}$-Completeness for multi-player Nash equilibria.** As a result of a series of important works [22, 32, 33, 56, 89], the complexity of 2-player Nash equilibrium is by now well understood, even when equilibria with special properties are desired and when the game is symmetric. Our contribution is on settling the complexity of finding equilibria with special properties on multi-player games. We show that the following decision versions of 3-Nash are $\exists\mathbb{R}$-complete: checking whether (i) there are two or more equilibria, (ii) there exists an equilibrium in which each player gets at least h payoff, where h is a rational number, (iii) a given set of strategies are played with non-zero probability, and (iv) all the played strategies belong to a given set. $\exists\mathbb{R}$ is the class of decision problems which can be reduced in polynomial time to Existential Theory of the Reals. Next, we give a reduction from 3-Nash to symmetric 3-Nash, hence resolving an open problem of Papadimitriou.

CHAPTER 1

INTRODUCTION

In a free market economy, prices naturally tend to find an "equilibrium" under which there is parity between supply and demand. The power of this pricing mechanism is well explored and understood in economics: It allocates resources efficiently since prices send strong signals about what is wanted and what is not, and it prevents artificial scarcity of goods while at the same time ensuring that goods that are truly scarce are conserved [76]. Furthermore, equilibrium-based mechanisms have been designed even for certain applications which do not involve any exchange of money but require fairness properties such as envy-freeness and the sharing incentive property; a popular one being CEEI[1] [81]. The surge of markets on the Internet, in which pricing and allocation are done in a centralized manner via computation, has led to a long line of work in the Theoretical Computer Science community on the computation of economic equilibria.

In this thesis, we study the computational complexity of price equilibria in the classical Arrow-Debreu exchange market model (see Section 3), and we define a new class of combinatorial markets with covering constraint to appropriately model several new markets on the internet, including scheduling jobs and bandwidth allocation in networks (see Section 1.1).

Next, we study the complexity of another broad class of equilibria, namely, Nash Equilibria (see Section 1.3). Nash equilibrium (NE) is arguably the most important and well-studied solution concept within game theory and understanding its complexity has led to an impressive theory which was discovered largely over the last decade.

[1] Competitive Equilibrium with Equal Incomes

1.1 Combinatorial Markets with Covering Constraints

A quickly emerging market today is cloud computing and scheduling market and most projections predict that this market will dwarf even the adwords market which is one of the most important market on the internet. In order to appropriately model it, we define a broad class of market models that we call combinatorial markets with covering constraints. A common feature of our markets is that these are resource allocation markets in which each agent desires a *specific amount of resources* to complete a task, i.e., each agent has a *covering constraint*. If the agent does not get all the resources requested, then she will not be able to complete the task and hence has no value for this partial allocation. With several agents vying for the same set of resources, a new parameter that becomes crucially important is the *delay* experienced by agents. This naturally leads to a definition of supply and demand, as well as pricing and allocation, based on *temporal considerations*.

We define an equilibrium-based model for pricing and allocation in these markets. Our model is fundamentally different from traditional market models: Each agent needs only a bounded amount of resources to finish her tasks and has no use for more, and her utility, which corresponds to the delay she experiences, also has a finite maximum value, i.e., her "utility function" satiates. On the other hand, traditional models satisfy non-satiation, i.e., no matter what bundle of goods an agent gets, there is a way of giving her additional goods so her utility strictly increases. Non-satiation turns out to be a key assumption in the Arrow-Debreu Theorem, which established the existence of equilibrium in traditional markets. Despite this, we manage to give an existence proof for our model. Additionally, we prove that all the above-stated benefits of equilibria, including the fairness properties of CEEI, continue to hold for our model.

We next address the issue of computing equilibria in our model. Rubinstein [93] recently showed that computing an equilibrium in our general model is PPAD-hard via a reduction from Fisher markets with separable piecewise linear concave (SPLC) utilities. Since the former prob-

lem is known to be PPAD-hard [23], the market problem with strong feasibility is PPAD-hard. For this reason, we define a sub-model which is still sufficiently rich to capture the applications mentioned above, and present a polynomial time algorithm to compute an equilibria in this sub-model. In general, markets and games exhibit a kind of dichotomy: typically the structure of equilibria is complicated in which case they are computationally intractable [25, 33, 34, 52, 68], or in very special cases equilibria have a nice structure (usually, forming a convex set) and polynomial time algorithms follow using standard methods, such as convex programming and the primal-dual method [27, 40, 41, 62, 88, 106]. However, the sub-model we define breaks this dichotomy. The equilibria have a different structure than those of models for which polynomial time algorithms have been designed. In particular, we give examples in which the set of equilibrium prices is non-convex. Hence techniques used for designing polynomial time algorithms for traditional models, such as the primal-dual method and convex programming, are not applicable. Our algorithms are based on new ideas: we make heavy use of LP duality and the way optimal solutions to LPs change with changes in certain parameters. Submodular minimization, combined with binary search, is used as a subroutine in this process.

1.2 Leontief and PLC Exchange Markets

A decade and a half of work in TCS has led to a deep understanding of computability of market equilibria for classic market models under fundamental utility functions. At this point, perhaps the most basic utility functions whose complexity remains unresolved are Leontief[2] and piecewise-linear concave (PLC). For both exact and approximate computation of equilibria, only partial results are know as detailed below. In this thesis, we resolve the remaining open questions, thereby pinning down the classes which characterize their complexity.

In economics, concave utilities occupy a special place because of their generality and because

[2]Leontief utility function for a bundle $\boldsymbol{x}$ of goods is given by $U(\boldsymbol{x}) = \min_j x_j/A_j$, where A_j's are non-negative constants. It captures the situation when goods are complements and required in a fixed proportion.

they capture the natural condition of decreasing marginal utilities. Since computer science assumes a finite precision model of computation, one is forced to restrict attention to PLC[3] utility functions. Price equilibria are clearly quintessential to economics, and therefore it is important to obtain a precise understanding of the complexity of computing Arrow-Debreu equilibria under PLC utility functions Leontief utilities form a subcase of PLC utilities and are very widely used in economic modeling [75].

For both these utility functions, computation of approximate equilibria has been known to be PPAD-hard for more than a decade [29, 36, 60]; however, membership in PPAD has not been established yet. In fact, [103] goes further to say that these problems may not even be in PPAD Furthermore, certain consequences stated in the literature are not true without establishing this result (see Section 3.1). Our first result shows membership of these problems in PPAD. The only fixed point formulation known for these problems was obtained in the context of proving membership in FIXP [48]. This formulation is our starting point; however, working with it is not straightforward. The main technical challenge lies in showing that an approximate fixed point captures an approximate market equilibrium. This turns out to be quite involved and technical, and requires new ideas as elaborated in Section 3.1. On the other hand, for both these utility functions, exact computation of equilibria is known to be in FIXP [48, 109]; however, FIXP-hardness was not established before and was stated as an open problem in [103]. This is our second result.

Proofs of membership in FIXP for Leontief and PLC utility functions were given by Yannakakis [109] and Garg et. al. [48], respectively. In Chapter 3, we prove FIXP-hardness for Arrow-Debreu exchange markets under Leontief utility functions, and Arrow-Debreu markets under linear utility functions and Leontief production sets. As corollaries, we obtain FIXP-hardness for PLC utilities and for Arrow-Debreu markets under linear utility functions and polyhedral production sets (membership in FIXP for production was also shown in [48]). In all cases, as required under FIXP, the set of instances mapped onto will admit equilibria, i.e., will be "yes" instances. If

[3]Clearly, by making the pieces fine enough, we can obtain a good approximation to the original utility functions.

all instances are under consideration, then we prove that the problem of deciding if a given instance admits an equilibrium is ETR-complete, where ETR is the class Existential Theory of Reals.

As a consequence of the results stated above, the entire computational difficulty of Arrow-Debreu markets under PLC utility functions lies in the Leontief utility subcase. This is perhaps the most unexpected aspect of our result, since Leontief utilities are meant only for the case that goods are perfect complements, whereas PLC utilities are very general, capturing not only the cases when goods are complements and substitutes, but also arbitrary combinations of these and much more.[4]

The class PPAD was defined by Papadimitriou [89]; he also proved PPAD-completeness of computing an approximate equilibrium of Arrow-Debreu exchange markets given by aggregate excess demand functions. The class FIXP was defined by Etessami and Yannakakis [68] and they proved FIXP-completeness of Arrow-Debreu exchange markets whose aggregate excess demand functions are algebraic. However, these results do not establish PPAD or FIXP-completeness of Arrow-Debreu markets under any specific class of utility functions[5]. We note that there has been no progress on giving proofs of FIXP-hardness for market equilibria under any specific utility functions.

Perhaps the most elementary way of stating the main technical part of our second result is the following reduction, which we will denote by $\mathcal{R}$: Given a set $\mathcal{S}$ of simultaneous multivariate polynomial equations in which the variables are constrained to be in a closed bounded region in the positive orthant, we construct an Arrow-Debreu market with Leontief utilities, say $\mathcal{M}$, which has one good corresponding to each variable in $\mathcal{S}$. We prove that the equilibria of $\mathcal{M}$, when projected onto prices of these latter goods, are in one-to-one correspondence with the set of solutions of the polynomials. This reduction, together with the fact that the 3-player Nash equilibrium problem (3-Nash) is FIXP-complete [68] and that 3-Nash can be reduced to such a system $\mathcal{S}$, yield FIXP-

[4]We had expected the precise complexity of computing an equilibrium in Arrow-Debreu exchange markets to be easier in case of Leontief utilities than in case of PLC utilities.

[5]In the economics literature, there are two parallel streams of results on market equilibria, one assumes being given an excess demand function and the other a specific class of utility functions.

hardness for the Leontief case.

On positive results for PLC utilities, Devanur and Kannan had given a polynomial time algorithm for finding an equilibrium in Arrow-Debreu markets under these utility functions provided the number of goods is a constant, using algebraic cell decomposition [38]. They had stated the open problem of handling the case of constant number of agents. Our third result settles a part of this open problem by obtaining a polynomial time algorithm for the subcase of Leontief utilities.

1.3 Multi-Player (Symmetric) Nash Equilibria

Nash equilibrium (NE) is arguably the most important and well-studied solution concept within game theory and understanding its complexity has led to an impressive theory which was discovered largely over the last decade. We denote by k-Nash the problem of computing a NE in a k-player game for a constant k. For the case of 2-Nash, the seminal results of Daskalakis, Goldberg and Papadimitriou [33], and Chen, Deng and Teng [22] exactly characterized the complexity of this problem, namely it is PPAD-complete. This leads us to another basic question: of finding a k-Nash solution that satisfies special properties, e.g., has a payoff of at least h for each player. For the case of 2-players, these questions were first studied by Gilboa and Zemel for non-symmetric games [56] and later by Conitzer and Sandholm for symmetric games [32]. Both papers considered 2-Nash under numerous special properties and showed them all to be NP-complete. More recently, Bilò and Mavronicolas [12] extended the results of Gilboa and Zemel to win-lose games, in which all payoffs are either 0 or 1. Thus the complexity of the 2-player case is very well understood.

Although the 2-player case is the most classical and well studied case, it is also important to study the complexity of the multi-player case, especially in the context of new applications arising on the Internet and other large networks where multiple players are locked in strategic situations. Indeed there has been much activity on this front, e.g., see [3, 66, 92], but the picture is not as clear as the 2-player case. A fundamental difference between 2-Nash and k-Nash, for $k \geq 3$, is that whereas the former always admits an equilibrium that can be written using rational numbers [73],

the latter require irrational numbers in general, as shown by Nash himself [84] (we will assume that all numbers in the given instance are rational). It is easy to see that in the latter case, equilibria are algebraic numbers. This difference makes the multi-player case much harder.

Daskalakis, Goldberg and Papadimitriou [33] showed that for k-player games, $k \geq 3$, finding an ϵ-approximate Nash equilibrium is PPAD-complete. The complexity of exact equilibrium was resolved by Etessami and Yannakakis [68], who showed this case to be complete for their class FIXP. How about the complexity of finding a k-Nash solution that satisfies special properties? Due to the inherent difficulty of dealing with irrational numbers, this problem remained open until 2011, when Schaefer and Štefankovič [97] formally defined class $\exists\mathbb{R}$, and showed that checking if a 3-player game has a NE in which every strategy is played with probability at most 0.5 (**InBox**) is $\exists\mathbb{R}$-complete. $\exists\mathbb{R}$ is the class of "yes" instances of existentially quantified formulas with bases $\{+, -, *, \wedge, \vee, =, <, >\}$ on real numbers; we note that this class was informally known and used earlier than [97], e.g., see [19]. In [35], Datta showed that an arbitrary semi-algebraic set can be encoded as *totally mixed* NE of a 3-player game. However, the reduction is not polynomial time and therefore is not applicable to show $\exists\mathbb{R}$-completeness of the decision problems in 3-Nash. Recently, in [74] Levy gave another construction to *precisely* capture any compact semi-algebraic set of mixed-strategies of a game as a projection of Nash equilibrium strategies of another game with additional binary players, however, no bound is provided on the number of additional players.

Our first set of results extends $\exists\mathbb{R}$-completeness to NE computation with a number of special properties in ≥ 3 player games: (i) checking if a game has more than one NE (**NonUnique**). NE where, (ii) each player gets at least h payoff (**MaxPayoff**), (iii) a given set of strategies are played with positive probability (**Subset**), or (iv) all the played strategies belong to a given set (**Superset**).

Our second set of results deals with symmetric games. Symmetry arises naturally in numerous strategic situations and with the growth of the Internet, on which typically users are indistinguishable, such situations are only becoming more ubiquitous. In a *symmetric game* all players participate under identical circumstances, i.e., strategy sets and payoffs. Thus the payoff of player

i depends only on the strategy, s, played by her and the multiset of strategies, S, played by the others, without reference to their identities. Furthermore, if any other player j were to play s and the remaining players S, the payoff to j would be identical to that of i. A *symmetric Nash equilibrium* (SNE) is a NE in which all players play the same strategy. Nash [84], while providing game theory with its central solution concept, also defined the notion of a symmetric game and proved, in a separate theorem, that such games always admit a symmetric equilibrium.

A simple reduction is known from 2-Nash to symmetric 2-Nash, and it shows that the latter is also PPAD-complete. The questions studied by Gilboa and Zemel [56] for 2-player games were studied by Conitzer and Sandholm [32] for symmetric games and were shown to be NP-complete. On the other hand, no reduction is known from 3-Nash to symmetric 3-Nash. Indeed, after giving the reduction from 2-Nash to symmetric 2-Nash, Papadimitriou [90] states, "Amazingly, it is not clear how to generalize this proof for three player games!".

To obtain our results on symmetric k-player games, for $k \geq 3$, we first give a reduction from 3-Nash to symmetric 3-Nash, hence settling the open problem of [90]. This also enables us to show that symmetric 3-Nash is complete for the class FIXP$_a$, Strong Approximation FIXP, which is a variant of FIXP that is restricted to working with rational numbers only. It also yields $\exists\mathbb{R}$-completeness for **Superset** and **Subset** in such games. Once the 3-player case is settled, we prove analogous results for symmetric k-player games, for $k > 3$.

[48] gave a dichotomy for NE, showing a qualitative difference between 2-Nash and k-Nash along three different criteria, see Table 1.1. The results of this thesis add a fourth criterion to this dichotomy, namely complexity of decision problems. Additionally, we get an analogous dichotomy for symmetric NE, see Table 1.2. Results of Chapter 4 are indicated by $\mathcal{CP}$ in the tables.

We note that the results were first presented in [51]. In that paper, we had left the open problem of extending our $\exists\mathbb{R}$-completeness results to decision versions of other 3-Nash and symmetric 3-Nash problems. Since then, there has been much progress on this open problem. First, Bilò and Mavronicolas [10] showed that the 3-player versions of all problems studied by Gilboa and Zemel

Table 1.1: Dichotomy for Nash equilibria

	2-Nash	**k-Nash, $k \geq 3$**
Nature of solution	Rational [73]	Algebraic; irrational example [84]
Complexity	PPAD-complete [22, 33, 89]	FIXP-complete [68]
Practical algorithms	Lemke-Howson [73]	?
Decision problems	NP-complete [32, 56]	$\exists\mathbb{R}$-complete: [97] $\mathcal{CP}$ (Theorems 32, 34)

Table 1.2: Dichotomy for symmetric Nash equilibria

	Symmetric 2-Nash	**Symmetric k-Nash, $k \geq 3$**
Nature of solution	Rational [73]	Algebraic; irrational example $\mathcal{CP}$ together with [84]
Complexity	PPAD-complete [22, 33, 89]	FIXP$_a$-complete: $\mathcal{CP}$ (Theorem 42)
Practical algorithms	Lemke-Howson [73, 95]	?
Decision problems	NP-complete [32]	$\exists\mathbb{R}$-complete: $\mathcal{CP}$ (Theorem 41)

[56] are $\exists\mathbb{R}$-complete; moreover, they do so via a unified reduction from **InBox**. Next, the same authors [11] showed $\exists\mathbb{R}$-completeness for several decision versions of symmetric 3-Nash, this time via reductions from **Subset**.

CHAPTER 2

COMBINATORIAL MARKETS WITH COVERING CONSTRAINTS: ALGORITHMS AND APPLICATIONS

In this chapter, we introduce a new class of combinatorial markets in which agents have covering constraints over resources required and are interested in delay minimization. Our market model is applicable to several settings including scheduling and communicating over a network.

This model is quite different from the traditional models, to the extent that neither do the classical equilibrium existence results seem to apply to it nor do any of the efficient algorithmic techniques developed to compute equilibria. In particular, our model does not satisfy the condition of non-satiation, which is used critically to show the existence of equilibria in traditional market models and we observe that our set of equilibrium prices could be a connected, non-convex set.

We give a proof of the existence of equilibria and a polynomial time algorithm for finding one, drawing heavily on techniques from LP duality and submodular minimization. Finally, we show that our model inherits many of the fairness properties of traditional equilibrium models as well as new models, such as CEEI.

2.1 Model and Main Results

We introduce a combinatorial version of the well studied Fisher market model [16, 41]. In market $\mathcal{M}$, let A be a set of n agents, indexed by i, and G be a set of m divisible goods, indexed by j. We represent an allocation of goods to agents using the variables $x_{ij} \in \mathbb{R}_+, i \in A, j \in G$. Each agent $i \in A$ wants to procure goods that satisfy a set of *covering constraints*, C, where C is a set indexing the constraints (C is the same for all agents for ease of notation).

$$\forall\, k \in C, \textstyle\sum_{j \in G} a_{ijk} x_{ij} \geq r_{ik}, \quad \text{and} \quad \forall\, j \in G, x_{ij} \geq 0. \tag{CC(i)}$$

The objective of each agent is to minimize the "delay" she experiences, while meeting these constraints. We refer to the term d_{ij} as the delay faced by agent i on using good j, and the terms r_{ik}s as the "requirements"; d_{ij}s and r_{ik}s are assumed to be non-negative. Agent i wants an allocation that optimizes the following LP.

$$\min \textstyle\sum_{j \in G} d_{ij} x_{ij} \text{ s.t.} \qquad \text{(Delay LP}(i)\text{)}$$

$$\forall\, k \in C, \textstyle\sum_{j \in G} a_{ijk} x_{ij} \geq r_{ik}.$$

$$\forall\, j \in G, x_{ij} \geq 0.$$

We use the notation $\boldsymbol{d}_i := (d_{ij})_{j \in G}, \boldsymbol{r}_i := (r_{ik})_{k \in C}$, $A_i := (a_{ijk})_{j \in G, k \in C}$, $\boldsymbol{x}_i := (x_{ij})_{j \in G}$, and $X := (\boldsymbol{x}_i)_{i \in A}$. Although our results hold for any LP, the most interesting cases are when the constraints are covering constraints, i.e., the matrix A_i has only non-negative entries.

We will use a market mechanism to allocate resources. Let $p_j \in \mathbb{R}_+$ denote the price per unit amount of good j, and assume agent i has a total budget of $m_i \in \mathbb{R}_+$. Then, as is standard in the Fisher markets, the bundle $\boldsymbol{x}_i$ that the agent may purchase is restricted by,

$$\textstyle\sum_{j \in G} p_j x_{ij} \leq m_i. \qquad \text{(Budget constraint}(i)\text{)}$$

Allocation $\boldsymbol{x}_i$ is an *optimal allocation (bundle)* for agent i relative to prices $\boldsymbol{p} := (p_j)_{j \in G}$, if it optimizes LP (Delay LP(i)) with an additional budget constraint (Budget constraint(i)). We denote the set of these optimal allocations by $OPT_i(\boldsymbol{p})$. Each good has a given supply which, after normalization, may be assumed to be equal to 1. The allocation needs to be *supply respecting*, that is, it has to satisfy the supply constraints:

$$\forall\, j \in G, \textstyle\sum_{i \in A} x_{ij} \leq 1. \qquad \text{(Supply constraints)}$$

Finally, a supply respecting allocation X and prices $\boldsymbol{p}$ are a *market equilibrium* of $\mathcal{M}$ iff

1. Each agent gets an optimal allocation relative to prices $\boldsymbol{p}$.

2. If some good $j \in G$ is not fully allocated, i.e., $\sum_{i \in A} x_{ij} < 1$, then $p_j = 0$.

The equilibrium condition requires that each agent does the best for herself, regardless of what the other agents do or even what the supply constraints are. From the perspective of the goods, the aim is market clearing (rather than, say, profit maximization). Some goods may not have sufficient demand and therefore we may not be able to clear them. This is handled by requiring these goods to be priced at zero.

In Theorem 3 we obtain a characterization of the equilibria in this general model in terms of solutions of a parameterized linear program that has one parameter per agent.

2.1.1 Existence of equilibria

We show how the above model is a special case of the classic Arrow-Debreu market model with quasi-concave utility functions in Section 2.6.1. Unfortunately, these utility functions do not satisfy the "non-satiation" condition required by the Arrow-Debreu theorem for the existence of an equilibrium: utility does not increase beyond a point even if additional goods are allocated. In fact, an equilibrium doesn't always exist for all covering LPs, as shown via a simple example in Section 2.4, Figure 2.2. And therefore we next identify conditions under which it does exist; the example in Figure 2.2 shows why this condition is necessary.

The equilibrium condition requires at a minimum that there exists a supply respecting allocation that also satisfies CC(i) for all the agents. In fact, it is easy to see that a somewhat stronger feasibility condition is necessary: suppose that a subset of agents all have high budgets while the remaining agents have budgets that are close to 0. Then at an equilibrium, agents in the former set get their "best" goods, which means that whatever supply remains must be sufficient to allocate a feasible bundle to the remaining agents.

We require a similar condition for all minimally feasible allocations, *i.e.,* an allocation $\boldsymbol{x}_i$ such that reducing the amount of any good would make CC(i) infeasible. We call this condition strong feasibility.

Definition 1 (Strong feasibility)**.** Market $\mathcal{M}$ satisfies *strong feasibility* if any minimally feasible and supply respecting solution for a subset of agents can be extended to a feasible and supply respecting allocation for the entire set. Formally, $\forall\ S \subset A$, and $\forall\ (\boldsymbol{x}_i)_{i\in S}$ that are *minimally* feasible for $(\text{CC}(i))_{i\in S}$ and are supply respecting (with $x_{ij} = 0\ \forall\ i \in S^c$), $\exists$ solutions $(\boldsymbol{x}_i)_{i\in S^c}$ that are feasible for $(\text{CC}(i))_{i\in S^c}$ and $(\boldsymbol{x}_i)_{i\in(S\cup S^c)}$ is supply respecting.

Theorem 1. *[Strong feasibility implies the existence of an equilibrium] If* $(\text{CC}(i))_{i\in A}$ *of market* $\mathcal{M}$ *satisfies strong feasibility, then* $\exists$ *an allocation* X *and prices* $\boldsymbol{p}$ *that constitute a market equilibrium of* $\mathcal{M}$*.*

The proof of this theorem is in Section 2.6. Strong feasibility is quite general in the following sense: it is satisfied if there is a "default" good that has a large enough capacity and may have a large delay but occurs in every constraint with a positive coefficient. In other words, any agent's covering constraints may all be met by allocating a sufficient quantity of the default good.

2.1.2 Efficient computation

Ideally we would want to design an efficient algorithm for markets with *Strong feasibility* condition, however this problem turns out to be PPAD-hard [93]. In order to circumvent this hardness, we define a stronger condition called *extensibility*, and design a polynomial time algorithm to compute a market equilibrium under it. Extensibility requires that any "optimal allocation" to a subset of agents can be "extended" to an "optimal allocation" for a set that includes one extra agent. Hence this is a matroid-like condition. For this we first formally define the notion of an "optimal allocation" for a subset of agents.

Definition 2. For any subset of agents $S \subseteq A$, we say that an allocation X is *jointly optimal for* S if (i) it satisfies $(\text{CC}(i))_{i \in S}$, (ii) it is supply respecting, and (iii) it minimizes $\sum_{i \in S} \boldsymbol{d}_i \cdot \boldsymbol{x}_i$. (Observe that X may not be optimal for individual agents in S.)

Definition 3 (Extensibility)**.** Market $\mathcal{M}$ satisfies *extensibility* if $\forall\ S \subset A$, given an allocation X that is jointly optimal for S, the following holds: for any $i \in S^c$, $\exists$ an allocation X' that is jointly optimal for $S' = S \cup \{i\}$, while not changing the delay of the agents in S, i.e., $\boldsymbol{d}_i \cdot \boldsymbol{x}'_i = \boldsymbol{d}_i \cdot \boldsymbol{x}_i,\ \forall i \in S$. In other words, total delay cost of agents in S' can be minimized without changing the delay cost of agents in S.

Extensibility seems somewhat stronger than strong feasibility, but the two conditions are formally incomparable; see example in Section 2.4, Figure 2.2. In Section 2.1.3 we show that extensibility condition captures many interesting problems as special cases. Even for very simple markets that satisfy extensibility, e.g., Tables 2.1 and 2.2 in Section 2.4, the set of equilibria may turn out to be highly non-convex. Therefore the techniques used to obtain polynomial time algorithms for traditional models are not applicable. In Section 2.3 we design a polynomial time algorithm by making a heavy use of parameterized LP, duality and submodular minimization, and obtain the following result.

Theorem 2. *[Extensibility implies polynomial time algorithm] There is a polynomial time algorithm that computes a market equilibrium allocation X and prices $\boldsymbol{p}$ for any market $\mathcal{M}$ that satisfies extensibility.*

Since the algorithm is quite involved, we first convey the main ideas through a special case of *scheduling* in Section 2.2. We show the run of the main algorithm on an example in Section 2.4, Figure 2.4.

2.1.3 Applications

We show that the extensibility condition is sufficiently rich by demonstrating how it can capture scheduling and routing problems. As a consequence of the above theorem we get polynomial time algorithms for the following special cases (The proofs that these satisfy extensibility are in Section 2.7.)

Scheduling. There are d different types of machines, and the set of time slots on a machine of type k is M_k; a pair (machine type, time slot) defines a good in the market. Each agent needs $r_{ik} \in \mathbb{R}_+$ units of time on machines of type k, which is captured by the covering constraint $\sum_{j \in M_k} x_{ijk} \geq r_{ik},\ \forall k \in [d]$. All agents experience the same delay d_{jk} from time slot j on type k. Assume that the number of time slots of each machine type k is greater than the total requirements of the agents, $\sum_{i \in A} r_{ik}$.

The main motivation for this problem is scheduling, but it also captures other client-server scenarios such as crowdsourcing. Think of different machine types as machines with different configurations, i.e., with different combinations of CPU, memory, hard disk, etc. For instance, most providers of cloud computing offer a wide range of virtual machine configurations. In a crowdsourcing scenario, different machine types could correspond to different demographics of workers.

It is easy to see that our delay function can capture the flow time objective.

Flow time is an appropriate objective when a job is comprised of many small tasks, and the results of these tasks are useful immediately upon completion. An alternate objective is the completion time, but the buyer optimization problem, i.e., the problem of finding an allocation for a single agent that minimizes the completion time given prices for all the slots, is *NP-hard*. Flow time is indeed a reasonable alternative even if the true objective is completion time: it is by now standard to design (approximation) algorithms for the flow time objective, and argue that it also

approximates completion time; see [20, 58] for examples.[1]

Even for this simple case with only one type of machine, we observe that the set of equilibria may form a connected non-convex set. The non-convexity example shown in Section 2.4, Tables 2.1 and 2.2, are instances of this setting.

Restricted assignment with laminar families – Different arrival times. The above basic scheduling setting can be generalized to the following restricted assignment case, where job i is allowed to be processed only on a subset of time slots $S_{ik} \subseteq M_k$ on machines of type k. We need the S_{ik}s to form a laminar[2] family within each type, and in addition, we require that the slots in a larger subset have lower delays. That is, if for some two agents $i, i' \in A$, $S_{i'k} \subset S_{ik}$ then $\max_{j \in S_{ik} \setminus S_{i'k}} d_{jk} \leq \min_{j' \in S_{i'k}} d_{j'k}$ for each type k.

This captures the scenario where jobs may arrive at different times, as explained next. Recall that the goods in the scheduling model represent pairs of (machine, time slot). Let's define S_{kt} to be the set of goods corresponding to machine type k after time slot t. It is easy to see that S_{kt}s form a laminar family. Therefore, different arrival times can be captured by allowing each job i with arrival time t_i to use goods of S_{kt_i} only for each machine type k.

Network flows. The goods are edges in a network, where each edge e has a certain (fixed) delay d_e. Each agent i wants to send r_i units of flow from a source s_i to a sink t_i, and minimize her own delay (which is a min-cost flow problem). We show that if the network is series-parallel and the source-sink pair is common to all agents, then the instance satisfies extensibility. This is similar to the basic scheduling example in that there is a sequence of paths of increasing delay, but the difference here is that we need to price edges and not paths. The difficulty is that paths share edges and hence the edge prices should be co-ordinated in such a way that the path prices are as desired. There are networks that are not series-parallel but still the instance satisfies extensibility. We show

[1] Flow time occurs as the objective of the natural LP relaxation for problems of minimizing completion time.

[2] A family of subsets is said to be *laminar* if any two sets S and T in the family are either disjoint, $S \cap T = \emptyset$, or contained in one another, $S \subseteq T$ or $T \subseteq S$.

one such network (and also how our algorithm runs on it) in Figure 2.4 on page 37. For a general network, an equilibrium may not exist; we give such an example in Figure 2.2.

A generalization of all these special cases that still satisfies extensibility is as follows: take any number of independent copies of any of these special cases above. E.g., each agent might want some machines for job processing, as well as send some flows through a network or a set of networks, but have a common budget for both together. Our algorithm works for all such cases.

2.1.4 Properties of equilibria

Fairness. We first discuss an application of our market model to fair division of goods, where there are no monetary transfers involved. This captures scenarios where the goods to be shared are commonly owned, such as the computing infrastructure of a large company to be shared among its users. A standard fair division mechanism is competitive equilibrium from equal incomes (CEEI) [81]. This mechanism uses an equilibrium allocation corresponding to an instance of the market where all the agents have the same budget. This can be generalized to a weighted version, where different agents are assigned different budgets based on their importance.

The fairness of such an allocation mechanism follows from the following properties of equilibria shown in Section 2.10.1 for the general model. 1. The equilibrium allocation is *Pareto optimal*; this an analog of the first welfare theorem for our model. 2. The allocation is envy-free; since each agent gets the optimal bundle given the prices and the budget, he doesn't envy the allocation of any other agent. 3. Each agent gets a "fair share": the equilibrium allocation Pareto-dominates an "equal share" allocation, where each agent gets an equal amount of each resource. This property is also known as *sharing incentive* in the scheduling literature [55]. 4. Incentive compatibility (IC): the equilibrium allocation is incentive compatible "in the large", where no single agent is large enough to significantly affect the equilibrium prices. In this case, the agents are essentially price takers, and hence the allocation is IC. We also show a version of IC when the market is not large. We discuss this in more detail below.

Incentive Compatibility. In the quasi-linear utility model, an agent maximizes the valuation of the goods she gets minus the payment. In the presence of budget constraints, [42] show that no anonymous[3] IC mechanism can also be Pareto optimal, even when there are just two different goods. In the context of our model, a quasi-linear utility function specifies an "exchange rate" between delay and payments, and the agent wants to minimize a linear combination of the two. We show in Section 2.10.3 that the impossibility extends to our model via an easy reduction to the case of [42].

In the face of this impossibility, we show the following second best guarantee in Section 2.10.2. For the scheduling application mentioned above, we show that our algorithm as a market based mechanism is IC in the following sense: non-truthful reporting of m_i and r_{ik}s can never result in an allocation with a lower delay. A small modification to the payments, keeping the allocation the same, makes the entire mechanism incentive compatible for the model in which agents want to first minimize their delay and subject to that, minimize their payments.

The first incentive compatibility assumes that utility of the agents is only the delay, and does not depend on the money spent (or saved). Such utility functions have been considered in the context of online advertising [14, 46, 82]. It is a reflection of the fact that companies often have a given budget for procuring compute resources, and the agents acting on their behalf really have no incentive to save any part of this budget. In the fair allocation context (CEEI), this gives a truly IC mechanism, since the m_is are determined exogenously, and hence are not private information.

The second incentive compatibility does take payments into account, but gives a strict preference to delay over payments. Such preferences are also seen in the online advertising world, where advertisers want as many clicks as possible, and only then want to minimize payments. The modifications required for this are minimal, and essentially change the payment from a "first price" to a "second price" wherever required.

[3] Anonymity is a very mild restriction, which disallows favoring any agent based on the identity.

2.2 Scheduling on a Single Machine

Our algorithm for the general setting is quite involved, therefore we first present it for a very special case in a scheduling setting mentioned in Section 2.1.3. The basic building blocks and the structure of the algorithm and the analysis are reflected in this case. In Section 2.4, we describe the run of this algorithm on the example in Table 2.1. We note that the formal proofs are given only for the general case and not for this section.

Suppose that there is just one machine and a good is this machine at a certain time $t \in \mathbb{Z}_+$, which we refer to as slot t. The set of goods is therefore $G = \mathbb{Z}_+$ and we index the goods by t instead of j as before. Further, assume that the delay of slot t is just t, i.e., $\forall\ i \in A, d_{it} = t$. Each agent i requires a certain number of slots to be allocated to her, as captured by the covering constraint $\sum_{t\in\mathbb{Z}_+} x_{it} \geq r_i$, for some $r_i \in \mathbb{Z}_+$. We denote the sum of the requirements over a subset $S \subseteq A$ of agents as $r(S) := \sum_{i\in S} r_i$. Recall that the budget of agent i is m_i, and similarly $m(S) := \sum_{i\in S} m_i$. We will show that equilibrium prices are characterized by the following conditions.[4]

1. The prices form a piecewise linear convex decreasing curve. Let the linear pieces (segments) of this curve be numbered $1, 2, \ldots, k, \ldots$, from *right to left*.

2. There is a partitioning of the agents into sets $S^1, S^2, \ldots, S^k, \ldots$, where the number of slots in kth segment is $r(S^k)$. Note that since r_is are integers so are the $r(S^k)$s.

3. The sum of the prices of slots in kth segment equals $m(S^k)$.

4. For any $S \subset S^k$, the total price of the first $r(S)$ slots of the segment is at least $m(S)$, since otherwise these slots would be over demanded. This is equivalent to saying that the total price of the last $r(S)$ slots in this segment is at most $m(S)$.

[4]For how this equilibrium characterization leads to an analogy with Myerson's ironing for a special case of this setting, with $r_i = 1$ for all $i \in A$, see Section 2.11.

Algorithm 1 Algorithm to compute market equilibrium for scheduling

Input: $A, (m_i)_{i\in A}, (r_i)_{i\in A}$
Initialize $A' \leftarrow A$, $p_{\text{low}} \leftarrow 0, T^0 \leftarrow r(A)+1, \forall\, t \geq T^0, p_t \leftarrow 0$ and $k \leftarrow 1$
while $A' \neq \emptyset$ **do**
    $S^k \leftarrow \text{NextSeg}(p_{\text{low}}, A', (m_i)_{i\in A'}, (r_i)_{i\in A'})$
    $\lambda_{S^k} \leftarrow 2\frac{m(S^k)-p_{\text{low}}r(S^k)}{r(S^k)(r(S^k)+1)}$
    $T^k \leftarrow T^{k-1} - r(S^k)$
    $\forall\, t \in [T^k, T^{k-1}]$, set $p_t \leftarrow p_{\text{low}} + (T^{k-1}-t)\lambda_{S^k}$
    Compute allocations $\boldsymbol{x}_i$ for all $i \in S^k$ by solving LP (2.1)
    Update $p_{\text{low}} \leftarrow p_{T^k}$, $A' \leftarrow A' \setminus S^k$, and $k \leftarrow k+1$
end while
Output allocations X and prices $\boldsymbol{p}$.

Condition 1 essentially comes from the optimal bundle condition of equilibrium. It makes sure that if an agent buys goods of segment s then (1) she cannot afford earlier segments, (2) the later segments increase her delay, and (3) for any combination of these the delay-per-dollar spent is more.

The above only characterizes equilibrium *prices*. We will show that Conditions 3 and 4 imply that there exists an *allocation* of the slots in segment k to the agents in S^k such that both their requirements and budget constraints are satisfied. Such allocations can then be found by solving the following feasibility LP (2.1). In this LP, segment k corresponds to the interval $[T^k, T^{k-1}]$.

$$\begin{aligned} \forall\, i \in S^k: &\quad \textstyle\sum_{t\in[T^k,T^{k-1}]} x_{it} \geq r_i \\ \forall\, i \in S^k: &\quad \textstyle\sum_{t\in[T^k,T^{k-1}]} p_t x_{it} \leq m_i \\ \forall\, t \in [T^k, T^{k-1}]: &\quad \textstyle\sum_{i\in S^k} x_{it} \leq 1 \\ \forall\, i \in S^k, \forall\, t \in [T^k, T^{k-1}]: &\quad x_{it} \geq 0. \end{aligned} \tag{2.1}$$

We now describe the algorithm, which is formally defined in Algorithm 1. It iteratively computes S^k, starting from $k = 1$: the last segment that corresponds to the latest slots is computed first, and then the segment to its left, and so on. Inductively, suppose we have computed segments numbered 1 up to $k-1$. Let p_{low} be the price of the earliest slot in segment $k-1$, and let

$A' = A \setminus \{S^1 \cup \cdots \cup S^{k-1}\}$. For any $S \subseteq A'$, consider the sum of the prices of $r(S)$ consecutive slots to the left of this slot, forming a line segment with slope $-\lambda$ (see Figure 2.1); this sum is

$$p_{\text{low}} r(S) + \lambda \frac{r(S)(r(S)+1)}{2}.$$

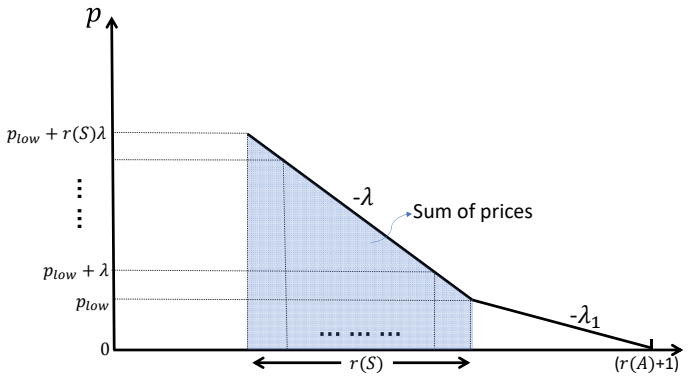


Figure 2.1: Prices on a Segment for set $S \subseteq A'$

Then for any S, one can solve for the λ where this would be equal to $m(S)$; we define this as a function of S.

$$\lambda_S := 2\frac{m(S) - p_{\text{low}} r(S)}{r(S)(r(S)+1)}.$$

The next segment is defined to be the one with the smallest slope:

$$S^k = \text{NextSeg}(p_{\text{low}}, A', (m_i)_{i \in A'}, (r_i)_{i \in A'}) := \arg\min_{S \subseteq A'} \lambda_S.$$

With this definition of the next segment, and with prices for the corresponding slots set to be linear with slope $-\lambda_{S^k}$, it follows that Conditions 3 and 4 are satisfied.

It is not immediately clear how to minimize λ_S; the function need not be submodular, for instance. The main idea here is to do a binary search over λ, as defined in Algorithm 2. Consider the function $f_{p_{\text{low}},\lambda}$ as defined in line 2 of this algorithm, and notice that $f_{p_{\text{low}},\lambda}$ is decreasing in λ. From the preceding discussion, it follows that the segment S^* we seek is such that $f_{p_{\text{low}},\lambda_{S^*}}(S^*) = 0$

Algorithm 2 Subroutine NextSeg$(p_{\text{low}}, A', (m_i)_{i\in A'}, (r_i)_{i\in A'})$

Initialize $\lambda^0 \leftarrow 0, \lambda^1 \leftarrow \max_{i\in A'} m_i$
Define $f_{p_{\text{low}},\lambda}(S) := m(S) - p_{\text{low}} r(S) - \lambda r(S)(r(S)+1)/2$
Set $S^0 \in \arg\min_{S\subseteq A', S\neq\emptyset} f_{p_{\text{low}},\lambda^0}(S)$ and $S^1 \in \arg\min_{S\subseteq A', S\neq\emptyset} f_{p_{\text{low}},\lambda^1}(S)$
while $S^0 \neq S^1$ **do**
    Set $\lambda^* \leftarrow \frac{\lambda^0+\lambda^1}{2}$ and $S^* \in \arg\min_{S\subseteq A', S\neq\emptyset} f_{p_{\text{low}},\lambda^*}(S)$
    if $f_{p_{\text{low}},\lambda^*}(S^*) > 0$ **then** Set $\lambda^0 \leftarrow \lambda^*$ and $S^0 \leftarrow S^*$
    else Set $\lambda^1 \leftarrow \lambda^*$ and $S^1 \leftarrow S^*$
    end if
end while
Return S^0

and $\forall S \subset A$, $f_{p_{\text{low}},\lambda_{S^*}}(S) \geq 0$. This implies that S^* must minimize $f_{p_{\text{low}},\lambda_{S^*}}$ over all subsets of A'. Thus, given any λ and a minimizer of $f_{p_{\text{low}},\lambda}$, we can tell whether the desired λ_{S^*} is above or below this λ, and a binary search gives us the desired segment. A minimizer of $f_{p_{\text{low}},\lambda}$ can be found efficiently since this is a submodular function.

In addition to the feasibility of LP (2.1), the main technical aspect of proving the correctness of the algorithm is to show that each agent gets an optimal allocation. This follows essentially from showing Condition 1, that the prices indeed form a piecewise linear convex curve, or equivalently, that the λ^ks form an increasing sequence. It is fairly straightforward to see that the running time of the algorithm is polynomial.

2.3 Algorithm under Extensibility

In this section we present the algorithm that proves Theorem 2; we will mimic the presentation in Section 2.2, but we have to deal with significant additional difficulties. We first present equilibrium characterization for the general model in Theorem 3 (complete proof is in Section 2.8), and then describe the key ideas in designing the algorithm (the missing proofs and other details of this part are in Section 2.9). We describe a run of our algorithm on a network flow example in Section 2.4, Figure 2.4.

Recall that in our general model, each agent has a delay function and a set of constraints on

the bundle of goods she gets. Unlike in Section 2.2, there is no simple ordering among the goods that enables a geometric description of an equilibrium, therefore some parts that are immediate in that setting require a proof here. Recall that the first step in Section 2.2 is to find an equilibrium characterization only in terms of prices. This used the geometry of the instance in order to partition the time slots into segments. For the general case, the right thing to do is to consider a partition of agents rather than a partition of goods. By abuse of terminology, in this section, by "segment" we refer to a subset of agents. Each agent i in A has a parameter λ_i, that previously corresponded to the slope of the segment they were in. Similarly, now too, all agents in a segment S^k have the same λ_i. This will also correspond to the reciprocal of the optimal dual variable for Budget constraint(i) in the agent's optimization problem at equilibrium.

A new issue that arises here is that some agents may have so much money that they don't spend all of it in equilibrium. These agents are guaranteed to get a bundle that absolutely minimizes the delay, even without any budget or supply constraints, i.e., their allocation is an optimal solution to LP (Delay LP(i)). We refer to this as "getting an optimal bundle under zero prices".

Given a vector of λ_is, denoted by $\boldsymbol{\lambda} \in \mathbb{R}_+^{|A|}$, we now define a parameterized linear program and its dual. Intuition for this definition comes from the optimal allocation LP for each agent at given prices. In the following, $LP(\boldsymbol{\lambda})$ has allocation variables x_{ij}s, the constraint CC(i) for each agent i, and the supply respecting constraint for each good j. The corresponding dual variables are respectively α_{ik}s and p_js, where p_j can be thought of as the price for good j.

$$
\begin{array}{ll}
\multicolumn{2}{c}{LP(\boldsymbol{\lambda}):} \\
\min: & \sum_i \lambda_i \sum_j d_{ij} x_{ij} \\
s.t. & \sum_j a_{ijk} x_{ij} \geq r_{ik},\ \forall (i,k) \\
& \sum_i x_{ij} \leq 1,\ \forall j \\
& x_{ij} \geq 0,\ \forall (i,j)
\end{array}
\qquad\qquad
\begin{array}{ll}
\multicolumn{2}{c}{DLP(\boldsymbol{\lambda}):} \\
\max: & \sum_{i,k} r_{ik} \alpha_{ik} - \sum_j p_j \\
s.t. & \lambda_i d_{ij} \geq \sum_k a_{ijk} \alpha_{ik} - p_j,\ \forall (i,j) \\
& p_j \geq 0,\ \forall j; \quad \alpha_{ik} \geq 0,\ \forall (i,k)\ .
\end{array}
\tag{2.2}
$$

Remarkably, the next theorem shows that the problem of computing an equilibrium reduces to

solving the above LP and its dual for a *right* parameter vector $\boldsymbol{\lambda} \in \mathbb{R}_+^{|A|}$. Recall that $OPT_i(\boldsymbol{p})$ denotes the set of optimal bundles of agent i at prices $\boldsymbol{p}$ (Section 2.1).

Theorem 3. *For a given $\boldsymbol{\lambda} > 0$ if an optimal solution X of $LP(\boldsymbol{\lambda})$ and an optimal solution $(\boldsymbol{\alpha}, \boldsymbol{p})$ of $DLP(\boldsymbol{\lambda})$ satisfy Budget constraint(i) for all agents $i \in A$, and at $(X, \boldsymbol{p})$ every agent i either spends all her budget, or $\boldsymbol{x}_i \in OPT_i(\mathbf{0})$, then $(X, \boldsymbol{p})$ constitute an equilibrium of market $\mathcal{M}$.*

We note that the proof of Theorem 3 uses only the complementary slackness conditions for the optimal allocation LP for each buyer, and therefore the theorem holds for the most general model, i.e., without any of the *extensibility* or *strong feasibility* assumptions. Theorem 3 gives us the "geometry" of an equilibrium outcome, and is roughly equivalent to Condition 1 from Section 2.2. It reduces the problem to one of finding a *right* parameter vector $\boldsymbol{\lambda}$; however there is still the entire $\mathbb{R}_+^{|A|}$ to search from. As was done in Section 2.2, our main goal is to further reduce this task to a sequence of single parameter searches, each involving submodular minimization and binary search.

Note that there is really nothing that is equivalent to Condition 2 from Section 2.2, since some goods may be allocated across agents in different segments. This is the source of many of the difficulties we face. Next, in Lemma 2, we derive a condition that is (approximately) equivalent to Conditions 3 and 4 from Section 2.2. This guarantees the existence of (allocation, prices) that satisfy the budget constraints of agents. One difference here is that this is going to be a global condition that involves the entire vector $\boldsymbol{\lambda}$, rather than a local condition that we could apply to a single segment like in Section 2.2. For this we need a number of properties of optimal solutions of $LP(\boldsymbol{\lambda})$ and $DLP(\boldsymbol{\lambda})$ that we show in Lemma 1 next.

Let us define $\text{delay}_i(X) = \sum_j d_{ij} x_{ij}$ and $\text{pay}_i(\boldsymbol{p}, X) = \sum_j p_j x_{ij}$. Similarly, for a subset of agents $S \subseteq A$, $\text{delay}_S(X) = \sum_{i \in S} \sum_j d_{ij} x_{ij}$ and $\text{pay}_S(\boldsymbol{p}, X) = \sum_{i \in S} \sum_j p_j x_{ij}$. Let $[d]$ denote

the set $\{1, \ldots, d\}$ of indices. By abuse of notation, let us define

$$\lambda(S) = \begin{cases} \text{the } \lambda \text{ value of agents in } S & \text{if all agents in } S \text{ have the same } \lambda_i. \\ \text{undefined} & \text{otherwise.} \end{cases}$$

Using *extensibility*, in the next lemma we show that optimal solutions of $LP(\lambda)$ and $DLP(\lambda)$ satisfy some invariants regarding delays and payments of agents. (1) The higher the λ is, the better the delay gets in a primal optimal solution. (2) For a fixed dual optimal solution, the total payment of a segment remains fixed at all optimal allocations. (3) As the delay of a subset decreases, its payment increases. (The lemma uses Definition 2 for "jointly optimal for".)

Lemma 1. *Given $\boldsymbol{\lambda}$, partition agents by equality of λ_i into sets $S_1, \ldots, S_d$ such that $\lambda(S_1) < \cdots < \lambda(S_d)$.*

1. *At any optimal solution X of $LP(\boldsymbol{\lambda})$, the delay is minimized first for set S_d, then for $S_{(d-1)}$, and so on, finally for S_1. This is equivalent to X being jointly optimal for each $T_g, \forall g \in [d]$ where $T_g = \cup_{q=g}^{d} S_q$, and for any other optimal solution Y we have $\text{delay}_{S_g}(Y) = \text{delay}_{S_g}(X),\ \forall g \in [d]$.*

2. *Given two dual optimal solutions $(\boldsymbol{\alpha}, \boldsymbol{p})$ and $(\boldsymbol{\alpha}', \boldsymbol{p}')$, if the first part of the dual objective is the same at both solutions for some $g \in [d]$,* i.e., *$\sum_{i \in S_g, k} r_{ik}\alpha_{ik} = \sum_{i \in S_g, k} r_{ik}\alpha'_{ik}$, then for any optimal solution X of $LP(\boldsymbol{\lambda})$, $\text{pay}_{S_g}(X, \boldsymbol{p}) = \text{pay}_{S_g}(X, \boldsymbol{p}')$.*

3. *Given two optimal solutions X and X' of $LP(\boldsymbol{\lambda})$, and an optimal solution $(\boldsymbol{\alpha}, \boldsymbol{p})$ of $DLP(\boldsymbol{\lambda})$, if for any subset $S \subseteq S_g$ for $g \in [d]$, $\text{delay}_S(X) \leq \text{delay}_S(X')$, then $\text{pay}_S(X, \boldsymbol{p}) \geq \text{pay}_S(X', \boldsymbol{p})$. The former is strict iff the latter is strict too.*

In the above lemma, the first claim follows from extensibility. The second and third claim follow from the first claim together with the fact that any pair of primal and dual optimal solutions satisfy complementary slackness conditions.

Recall Conditions 3 and 4 of Section 2.2 that respectively require *budget balanceness*, and that when a subset of agents in a segment are given their jointly optimal allocation, their total payment should be at least their total budget (or else they will over demand some good). Using the first and last part of Lemma 1 the latter can be roughly translated to saying that when the rest of the agents are given their "worst" allocation, the rest underpay in total. Based on this intuition we define the following conditions: *budget balance (*BB*)* and *subset condition (*SC*)*.

Definition 4. Given $(\boldsymbol{\lambda}, \boldsymbol{p})$, and a set $S \subseteq A$, we say that

- BB is satisfied: If either for every solution X of $LP(\boldsymbol{\lambda})$ we have $\text{pay}_S(X, \boldsymbol{p}) = m(S)$, or for every solution X of $LP(\boldsymbol{\lambda})$ we have $\text{pay}_S(X, \boldsymbol{p}) \leq m(S)$ and $\forall i \in S,\ \boldsymbol{x}_i \in OPT_i(\mathbf{0})$.
- SC is satisfied: $\forall\, T \subseteq S$ let X be an optimal solution to $LP(\boldsymbol{\lambda})$ where delay_T is maximized. Then, $m(T) \geq \text{pay}_T(X, \boldsymbol{p})$.

We will show that if BB and SC are satisfied for each "segment" at any given $\boldsymbol{\lambda} > 0$ then $\boldsymbol{\lambda}$ is the *right* parameter vector. We will call such a $(\boldsymbol{\lambda}, \boldsymbol{p})$ *proper* , which is formally defined next.

Definition 5. We say that pair $(\boldsymbol{\lambda}, \boldsymbol{p})$ is *proper* if there exists $\boldsymbol{\alpha}$ such that $(\boldsymbol{\alpha}, \boldsymbol{p})$ is an optimal solution to $DLP(\boldsymbol{\lambda})$, and pair $\boldsymbol{\lambda}, \boldsymbol{p}$ satisfies BB and SC for subsets $S_g,\ \forall g \leq d$, where $S_1, \ldots, S_d$ is the partition of A by equality of $\boldsymbol{\lambda}_i$.

The next lemma shows that the parameter vector $\boldsymbol{\lambda}$ corresponding to a *proper* pair would ensure the existence of an allocation where no agent spends more than her budget, and an agent who does not spend her entire budget gets an absolute best bundle, *i.e.,* an optimal bundle at zero prices, and thereby it gives an equilibrium using Theorem 3.

Lemma 2. *If a pair $(\boldsymbol{\lambda}^*, \boldsymbol{p}^*)$ is proper for $\boldsymbol{\lambda}^* > 0$ then there exists an optimal solution X^* to the primal $LP(\boldsymbol{\lambda}^*)$ such that* $\text{pay}_i(X^*, \boldsymbol{p}^*) \leq m_i\ \forall i \in A$*, and for every agent i either* $\text{pay}_i(X^*, \boldsymbol{p}^*) = m_i$ *or we have $\boldsymbol{x}_i \in OPT_i(\mathbf{0}),\ \forall X \in LP(\boldsymbol{\lambda}^*)$.*

Algorithm 3 Algorithm to compute a market equilibrium under extensibility

Input: $A, (m_i)_{i \in A}$, (Delay LP(i))$_{i \in A}$
Initialize $A' \leftarrow A$, $\boldsymbol{p}^{cur} \leftarrow 0, \boldsymbol{\lambda}^{cur} \leftarrow 0$ and $k \leftarrow 1$
while $A' \neq \emptyset$ **do**
    $(S^k, \boldsymbol{\lambda}^{new}, \boldsymbol{p}^{new}) \leftarrow$ NextSeg$(\boldsymbol{\lambda}^{cur}, \boldsymbol{p}^{cur}, A', (m_i)_{i \in A}$, (Delay LP$(i)$)$_{i \in A})$
    $A' \leftarrow A' \setminus S^k$, and $k \leftarrow k + 1$
    $\boldsymbol{\lambda}^{cur} \leftarrow \boldsymbol{\lambda}^{new}$, and $\boldsymbol{p}^{cur} \leftarrow \boldsymbol{p}^{new}$.
end while
Compute allocations $\boldsymbol{x}_i$ satisfying Budget constraint(i) for all $i \in A$, by solving LP (2.3) for $\boldsymbol{p}^* = \boldsymbol{p}^{cur}$ and $\boldsymbol{\lambda}^* = \boldsymbol{\lambda}^{cur}$.
Output allocations X and prices $\boldsymbol{p}^{cur}$.

Given such a $\boldsymbol{\lambda}^*$ and a solution $(\boldsymbol{\alpha}^*, \boldsymbol{p}^*)$ to $DLP(\boldsymbol{\lambda}^*)$ that satisfy conditions of Lemma 2, the lemma ensures existence of an allocation that simultaneously optimizes $LP(\boldsymbol{\lambda}^*)$ as well as satisfies Budget constraint(i), $\forall i \in A$. In other words, it guarantees that the following LP in X variables is feasible, and we can therefore compute such an allocation efficiently.

$$
\begin{aligned}
&\hat{X} \text{ is any solution of } LP(\boldsymbol{\lambda}^*): && \textstyle\sum_i \lambda_i^* \sum_j d_{ij} x_{ij} = \sum_i \lambda_i^* \sum_j d_{ij} \hat{x}_{ij} \\
&\forall (i,k): && \textstyle\sum_j a_{ijk} x_{ij} \geq r_{ik} \\
&\forall j: && \textstyle\sum_i x_{ij} \leq 1 \\
&\forall i, && \text{if } \hat{x}_i \in OPT_i(\mathbf{0}) \text{ then } \textstyle\sum_j p_j^* x_{ij} \leq m_i, \\
& && \quad \text{otherwise} \qquad \textstyle\sum_j p_j^* x_{ij} = m_i \\
&\forall (i,j): && x_{ij} \geq 0\,.
\end{aligned}
\tag{2.3}
$$

Now our goal has reduced to finding a *proper* $(\boldsymbol{\lambda}, \boldsymbol{p})$ pair. That is, if we think of the partition of agents by equality of λ_i as "segments", then we wish to find a vector $\boldsymbol{\lambda}$ such that BB and SC are satisfied for each "segment". Our algorithm, defined in Algorithm 3, tries to fulfill exactly this goal. At a high level, like in Section 2.2, our algorithm will build the segments bottom up, *i.e.,* lowest to highest λ segments. We start by setting all the λs to the same value, and find the lowest λ value where BB and SC are satisfied for a subset. Once found, we freeze this subset as a segment and continue increasing the λ for the rest to find the *next segment*, and repeat.

In this process of finding the *next segment* we need to make sure that the BB and SC conditions are maintained for the previous segments. In Section 2.2 we were able to do this by simply *fixing the prices* of the goods in earlier segments, because goods were not shared across segments. Here, some of the goods allocated to agents in the earlier segments may also be allocated to agents in the later segments, and additionally these allocations are not fixed and may keep changing during the algorithm. (We fix the allocation only at the end.) Furthermore, the prices are required to be dual optimal w.r.t. the $\boldsymbol{\lambda}$ vector that we eventually find. On the other hand in order to maintain BB and SC conditions for the previous segments we need to ensure that the total payments of the previous segments do not change.

The next lemma shows that this is indeed possible by proving that prices of goods bought by agents in the previous segments can be held fixed. In fact, we will be able to fix α_{ik}s as well, for agents in the previous segments. The proof involves an application of Farkas' lemma, leveraging extensibility. During the computation of the *next segment*, we hold fixed the λs of the agents in the segments found so far, and increase the λs of the remaining agents. To facilitate this we define $\mathbf{1}_S \in \{0,1\}^A$ as the indicator vector of $S \subseteq A$, i.e., $\mathbf{1}_S(i) = 1$ if $i \in S$, and is 0 otherwise.

Lemma 3. *Given a $\boldsymbol{\lambda}$, partition agents into $S_1, \ldots, S_d$ by equality of λ_i, where $\lambda(S_1) < \cdots < \lambda(S_d)$. For $R \subseteq S_d$ consider primal optimal $\hat{X}$ that is jointly optimal for R, and let $(\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{p}})$ be a dual optimal. Consider for some $a > 0$, the vector $\boldsymbol{\lambda}' = \boldsymbol{\lambda} + a\mathbf{1}_R$. Then $\hat{X}$ is optimal in $LP(\boldsymbol{\lambda}')$ and there exists an optimal solution $(\boldsymbol{\alpha}', \boldsymbol{p}')$ of $DLP(\boldsymbol{\lambda}')$ such that,*

$$\begin{aligned} &\forall j: && p'_j \geq \hat{p}_j \quad \textit{and} \quad \textstyle\sum_{i \notin R} \hat{x}_{ij} > 0 \Rightarrow p'_j = \hat{p}_j \\ &\forall i \notin R,\ \forall k, && \alpha'_{ik} = \hat{\alpha}_{ik}\ . \end{aligned}$$

As discussed above, our algorithm builds segments *inductively* from the lowest to highest λ value, by increasing λ of only the "remaining" agents. Suppose, we have built segments S_1 through S_{k-1}, and let $A' = A \setminus \cup_{g=1}^{k-1} S_g$ be the remaining set of agents. Let $\boldsymbol{\lambda}^{cur}$ be the current $\boldsymbol{\lambda}$ vector

where $\boldsymbol{\lambda}^{cur}(S_1) < \cdots < \boldsymbol{\lambda}^{cur}(S_{k-1}) < \boldsymbol{\lambda}^{cur}(A')$. Let $\boldsymbol{p}^{cur}$ be the corresponding dual price vector which is optimal for $DLP(\boldsymbol{\lambda}^{cur})$. For ease of notation we define the following.

$$\text{For any } a \geq 0, \text{ define } \boldsymbol{\lambda}^a = \boldsymbol{\lambda}^{cur} + a\mathbf{1}_{A'}. \tag{2.4}$$

Fix an allocation X^{cur} that is an optimal solution to $LP(\boldsymbol{\lambda}^{cur})$. We call an optimal solution $(\boldsymbol{\alpha}, \boldsymbol{p})$ to $DLP(\boldsymbol{\lambda}^a)$ *valid* if prices are monotone w.r.t. $\boldsymbol{p}^{cur}$ and X^{cur}, in the sense as guaranteed by Lemma 3 (where prices of goods allocated to previous segments are held fixed and prices of the rest of the goods are not decreased), and $\boldsymbol{\alpha}_i$s are fixed for agents outside A'. We will call the corresponding prices *valid prices*. For simplicity we will assume *uniqueness of valid prices*.[5]

$$\text{Define } \boldsymbol{p}^a \text{ to be the } \textit{valid} \text{ price vector at an optimal solution to } DLP(\boldsymbol{\lambda}^a). \tag{2.5}$$

Since the correctness is proved by induction, the *inductive hypothesis* is that w.r.t. $(\boldsymbol{\lambda}^{cur}, \boldsymbol{p}^{cur})$, both SC and BB are satisfied for $S_1, \ldots S_{k-1}$, and SC is satisfied for the remaining agents A'. The base case is easy with the λ_is all set to 0. Our next goal is to find the next segment $S_k \subseteq A'$, a new vector $\boldsymbol{\lambda}^{new}$ and a new price vector $\boldsymbol{p}^{new}$ such that the following properties hold.

1. Parameter vector $\boldsymbol{\lambda}^{new}$ is obtained from $\boldsymbol{\lambda}^{cur}$ by fixing λ_is of agents outside A', increase λ_is of agents in S_k by the same amount, and those of agents $A' \setminus S_k$ by some more. The latter increase is to separate S_k from A'. That is for some $a \geq 0$ and $\epsilon > 0$, $\boldsymbol{\lambda}^{new} = \boldsymbol{\lambda}^a + \epsilon\mathbf{1}_{A' \setminus S_k}$.
2. Price vector $\boldsymbol{p}^{new}$ is valid and optimal for $DLP(\boldsymbol{\lambda}^{new})$.
3. W.r.t. $(\boldsymbol{\lambda}^{new}, \boldsymbol{p}^{new})$, $S_1, \ldots, S_k$ satisfy both BB and SC, and $A' \setminus S_k$ satisfies SC.

The computation of the next segment S^k satisfying the above properties is done by the subroutine NextSeg, which is formally defined in Algorithm 4. As in Section 2.2, the basic idea is to

[5]This is without loss of generality since perturbing the parameters of the market ensures this. A typical way to simulate perturbation is by lexicographic ordering [99].

Algorithm 4 Subroutine NextSeg($\boldsymbol{\lambda}^{cur}, \boldsymbol{p}^{cur}, A', (m_i)_{i\in A}$, (Delay LP$(i)$)$_{i\in A}$)

Initialize $a^0 \leftarrow 0, a^1 \leftarrow \Delta$, where $\Delta = (\sum_{i,j,k} |a_{ijk}| + \sum_{i,k} |r_{ik}| + \sum_{i,j} |d_{ij}| + \sum_i |m_i|)^{2mn|C|}$.
Define function f_a as in (2.7).
Set $S^0 \in \arg\min_{S\subseteq A', S\neq\emptyset} f_{a^0}(S)$ and $S^1 \in \arg\min_{S\subseteq A', S\neq\emptyset} f_{a^1}(S)$
if $f_{a^1}(S^1) > 0$ **then** Return($A', \boldsymbol{\lambda}^{cur}, \boldsymbol{p}^{cur}$) // Note that $g(\Delta) = f_{a^1}(S^1) > 0$.
end if
while $S^0 \neq S^1$ **do**
    Set $a^* \leftarrow \frac{a^0+a^1}{2}$ and $S^* \in \arg\min_{S\subseteq A', S\neq\emptyset} f_{a^*}(S)$
    if $f_{a^*}(S^*) > 0$ **then** Set $a^0 \leftarrow a^*$ and $S^0 \leftarrow S^*$
    else Set $a^1 \leftarrow a^*$ and $S^1 \leftarrow S^*$
    end if
end while
$S^* \leftarrow S^0$.
Compute a^* by solving the feasibility LP for S^* mentioned in Lemma 7 such that $f_{a^*}(S^*) = 0$
// Next, we compute a maximal minimizer of the function f_{a^*} containing the set S^*.
$A' \leftarrow A' \setminus S^*$.
while $A' \neq \emptyset$ **do**
    $S \leftarrow \arg\min_{T\subseteq A', T\neq\emptyset} f_{a^*}(T \cup S^*)$
    if $f_{a^*}(S \cup S^*) > 0$ **then break**
    else set $S^* \leftarrow S^* \cup S$, $A' \leftarrow A' \setminus S$
end while
Set $\boldsymbol{\lambda}^{new} \leftarrow \boldsymbol{\lambda}^{a^*}$, $\lambda_i^{new} \leftarrow \lambda_i^{new} + \epsilon \mathbf{1}_{A'}$, and $\boldsymbol{p}^{new} \leftarrow$valid price at $\boldsymbol{\lambda}^{new}$, where $\epsilon \leftarrow \frac{1}{\Delta}$
Return $(S^*, \boldsymbol{\lambda}^{new}, \boldsymbol{p}^{new})$

reduce this problem to a single parameter binary search. Since SC is satisfied for the remaining agents A' at $(\boldsymbol{\lambda}^{cur}, \boldsymbol{p}^{cur})$ while BB is not, the total payment of agents in A' is less than their total budget $m(A')$. In order to keep track of this surplus budget, consider the following function on $S \subseteq A'$.

$$f_{\boldsymbol{\lambda},\boldsymbol{p}}(S) = m(S) - \text{pay}_S(X, \boldsymbol{p}), \text{ where } X \text{ is an optimal solution to } LP(\boldsymbol{\lambda}) \text{ that maximizes delay}_S. \tag{2.6}$$

We translate Property 3 above in terms of this function, in the following lemma, which essentially reduces the problem to a single parameter search.

Lemma 4. *Suppose that for some $a \geq 0$,*

$$S_k \in \arg\min_{S \subseteq A', S \neq \emptyset} \{f_{\boldsymbol{\lambda}^a, \boldsymbol{p}^a}(S)\}.$$

Further, suppose that $f_{\boldsymbol{\lambda}^a, \boldsymbol{p}^a}(S_k) = 0$, and let S_k be a maximal such set. Then there exists a rational number $\epsilon > 0$ of polynomial-size such that, w.r.t. $(\boldsymbol{\lambda}^{new}, \boldsymbol{p}^{new})$ as defined above, $S_1, \ldots, S_k$ satisfy both BB *and* SC*, and $A' \setminus S_k$ satisfies* SC*.*

The above lemma reduces the task of finding the next segment to that of finding an appropriate a such that the minimum value of $f_{\boldsymbol{\lambda}^a, \boldsymbol{p}^a}$ is zero under the *valid* price $\boldsymbol{p}^a$. This requires two things: (1) we need to find a minimizer of $f_{\boldsymbol{\lambda}^a, \boldsymbol{p}^a}$ for a given $a > 0$, and (2) we need to find the right value of a. The next lemma shows that the first task can be done using an algorithm for submodular minimization, and therefore in a polynomial time [98]. For convenience of notation, we define the following functions.

$$f_a(S) := f_{\boldsymbol{\lambda}^a, \boldsymbol{p}^a}(S) \quad \text{and} \quad g(a) := \min_{S \subseteq A', S \neq \emptyset} f_a(S)\,. \tag{2.7}$$

Lemma 5. *Given $a \geq 0$, function f_a is submodular over set A'.*

Now the following question remains: how does one find an a such that the minimum value is 0, *i.e.,* $g(a) = 0$. We do binary search for this. In the next two lemmas we derive a number of properties of g that facilitates binary search, while crucially using Lemmas 1 and 3.

Lemma 6. *If $g(a) > 0$ for all $a \geq 0$ then A' satisfies the BB and SC conditions at $(\boldsymbol{\lambda}^{cur}, \boldsymbol{p}^{cur})$ at the start of Algorithm 4.*

Lemma 7. *Function g satisfies the following: (i) $g(0) \geq 0$. (ii) $f_a(S)$ is continuous and monotonically decreasing in a, $\forall S \subseteq A'$, therefore g is continuous and monotonically decreasing. (iii) either $g(\Delta) \leq 0$ for $\Delta = (\sum_{i,j,k} |a_{ijk}| + \sum_{i,k} |r_{ik}| + \sum_{i,j} |d_{ij}| + \sum_i |m_i|)^{2mn|C|}$ or $g(a) > 0$ for*

all $a \geq 0$. (iv) *Given a set* $S \subseteq A'$*, if* $f_a(S) > 0$ *and* $f_{a'}(S) < 0$ *for* $a' > a > 0$*, then* $\exists a^* \geq 0$ *such that* $f_{a^*}(S) = 0$ *and such an* a^* *can be computed by solving a feasibility linear program of polynomial-size.*

The first part follows essentially from the fact that A' satisfies the SC condition w.r.t. $(\boldsymbol{\lambda}^{cur}, \boldsymbol{p}^{cur})$. For the second part, we show that for any $S \subseteq A'$, the function $f_a(S)$ is monotonically decreasing and continuous in a. Since the minimum of many continuous and decreasing functions is also continuous and decreasing, we get the same property for g. If g indeed becomes zero at some a then we know that there exists $S \subseteq A'$ such that $f_a(S) = 0$. We can show that for any S such an a has to be at most Δ, and thereby we get the third part. Finally, for the fourth part, the existence of a^* uses the monotonicity and the continuity of $f_a(S)$ in a. Using the fact that complementary slackness ensures optimality, we construct a feasibility linear program to compute a^*, given $S \subseteq A'$ such that $f_{a^*}(S) = 0$.

We initialize our binary search with a lower pivot of $a_0 = 0$ and a higher pivot of $a_1 = \Delta$. The third part of Lemma 7 guarantees that Δ is such that either the binary search will find an a as required, or no set will ever go tight and we have found our highest segment where agents don't spend all their money. Finally, since submodular minimization, binary search over a polynomial-sized range, and solving a linear program can all be done in polynomial-time, we get our main result, Theorem 2, using Lemmas 2, 4, 6 and 7, and Theorem 3.

2.4 Examples

In this section we show several interesting examples that illustrate important properties of our market and of equilibria. In addition we demonstrate a run of our algorithm on a routing example.

Non-existence of equilibria. Consider the networks (typically used to show Braess' paradox) in Figure 2.2, where the label on each edge specifies its (capacity, delay cost). There are two agents, each with a requirement of 1 from s to t. Their m_is are 100 and 1 respectively. The network on the

left has enough capacity to route two units of flow, but does not satisfy *strong feasibility* condition and does not have an equilibrium. This demonstrates importance of *strong feasibility* condition, without which even a simple market may not have an equilibrium.

The network in the middle does satisfy *strong feasibility* but not *extensibility* and has an equilibrium. The network on the right satisfies *extensibility* but not *strong feasibility* and has an equilibrium. This demonstrates that conditions of *strong feasibility* and *extensibility* are incomparable.

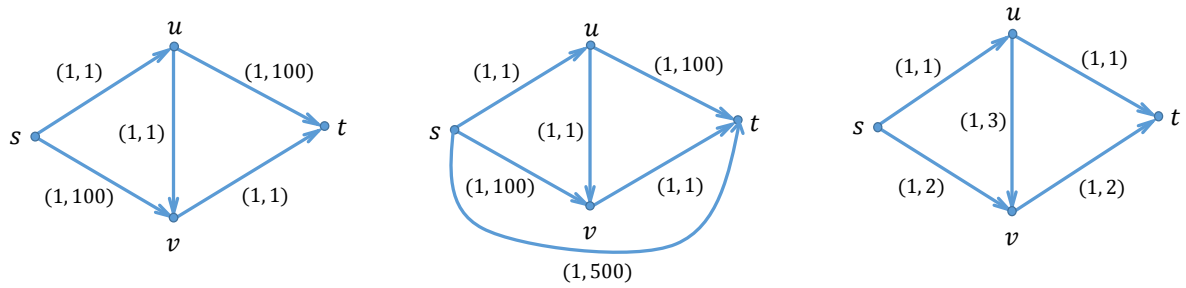
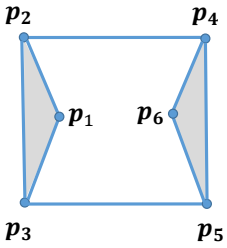


Figure 2.2: Non-existence of equilibria.
The network on the left does not satisfy *strong feasibility* and has no equilibria. Edge labels specify (capacity, delay cost), there are two agents, each with a requirement of 1 from s to t. The m_is are $100, 1$. The same network with an additional edge with huge cost in the middle, does satisfy *strong feasibility* but not *extensibility* and has an equilibrium. The network on the right with different edge labels satisfies the *extensibility* but not *strong feasibility*. It has an equilibrium and our algorithm will find one.

Non-convexity of equilibria. Consider a market in the scheduling setting of Section 2.2, with 6 agents, each with a requirement of 1. Their m_is are $30, 17, 9, 4, 3, 1$. Table 2.1 depicts some of the equilibrium prices for this instance. A bigger example with 9 agents is in Table 2.2. These examples show that the equilibrium set is not convex, but forms a connected set. Since these are in high dimension, it is not easy to determine the exact shape of the entire equilibrium set, but one can see that it is quite complicated.

A run of the algorithm. We describe the run of our algorithms on simple examples here. The run of Algorithm 1 on the example in Table 2.1 is as follows. Each row below depicts one iteration,

Table 2.1: An example in the scheduling setting of Section 2.2 where the set of equilibrium prices is non-convex.



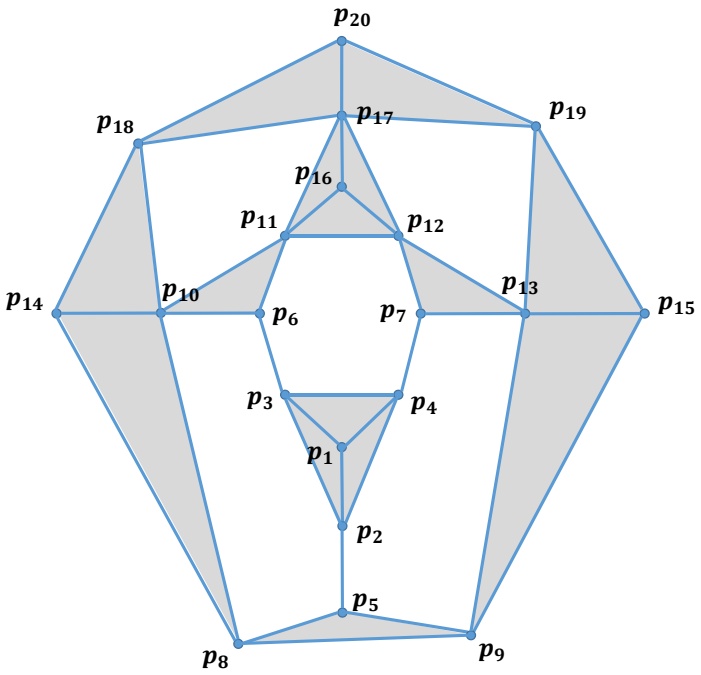
$\boldsymbol{p}_1$:	$(30, 17, 9, 5, 2, 1)$
$\boldsymbol{p}_2$:	$(30, 17, 9, 4\frac{1}{3}, 2\frac{2}{3}, 1)$
$\boldsymbol{p}_3$:	$(34, 13, 9, 5, 2, 1)$
$\boldsymbol{p}_4$:	$(34, 13, 9, 5\frac{1}{3}, 2\frac{2}{3}, 0)$
$\boldsymbol{p}_5$:	$(35, 13, 8, 4\frac{1}{3}, 2\frac{2}{3}, 1)$
$\boldsymbol{p}_6$:	$(35, 13, 8, 5\frac{1}{3}, 2\frac{2}{3}, 0)$

There are 6 agents, each with a requirement of 1. Their m_is are $30, 17, 9, 4, 3$ and 1. We depict only a subset of all equilibria here. In particular, we depict 6 equilibrium prices, $\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_6$. All prices either along solid lines connecting any two of these points, or in the shaded region are equilibria. However, if any two of these prices are not connected by a solid line, then none of the points on the line joining them is an equilibrium. For example, none of the prices on the line joining $\boldsymbol{p}_1$ and $\boldsymbol{p}_6$, $\boldsymbol{p}_2$ and $\boldsymbol{p}_5$, or $\boldsymbol{p}_3$ and $\boldsymbol{p}_4$ is an equilibrium. There are more equilibrium points not depicted here. As far as we can tell, the shape of the equilibrium set is something akin to a cup, with empty space inside, but forming a single connected region.

where we find a new segment. We first give the set of agents in this new segment, then the corresponding λ, and then the prices of the slots determined in this iteration. The last column shows the sets which give the second and third lowest λs in that iteration, and hence were not selected.

$S^1 = \{6\},$	$\lambda_{S^1} = 1,$	$p_6 = 1$	$(\lambda_{\{5,6\}} = 1\frac{1}{3}, \lambda_{\{4,5,6\}} = 1\frac{1}{3}, \dots)$
$S^2 = \{4, 5\},$	$\lambda_{S^2} = 1\frac{2}{3},$	$p_5 = 2\frac{2}{3},\ p_4 = 4\frac{1}{3}$	$(\lambda_{\{5\}} = 2, \lambda_{\{3,4,5\}} = 2\frac{1}{6}, \dots)$
$S^3 = \{3\},$	$\lambda_{S^3} = 4\frac{2}{3},$	$p_3 = 9$	$(\lambda_{\{2,3\}} = 5\frac{7}{9}, \lambda_{\{1,2,3\}} = 7\frac{1}{6})$
$S^4 = \{2\},$	$\lambda_{S^4} = 8,$	$p_2 = 17$	$(\lambda_{\{1,2\}} = 9\frac{2}{3})$
$S^5 = \{1\},$	$\lambda_{S^5} = 13,$	$p_1 = 30$	

Table 2.2: An example in the scheduling setting of Section 2.2 where the set of equilibrium prices is non-convex.



p_1 :	$(57, 44, 33, 24, 16, 10, 5, 2, 1)$
p_2 :	$(57, 44^2/3, 32^1/3, 24, 16, 10, 5, 2, 1)$
p_3 :	$(57, 44, 33, 24, 16^2/3, 9^1/3, 5, 2, 1)$
p_4 :	$(57, 44, 33, 24, 16, 10, 5, 2^2/3, 1/3)$
p_5 :	$(57^1/3, 44^2/3, 32, 24, 16, 10, 5, 2, 1)$
p_6 :	$(57, 44, 33, 24^1/3, 16^2/3, 9, 5, 2, 1)$
p_7 :	$(57, 44, 33, 24, 16, 10, 5^1/3, 2^2/3, 0)$
p_8 :	$(57^1/3, 44^2/3, 32, 24, 16^2/3, 9^1/3, 5, 2, 1)$
p_9 :	$(57^1/3, 44^2/3, 32, 24, 16, 10, 5, 2^2/3, 1/3)$
p_{10} :	$(57, 44^2/3, 32^1/3, 24^1/3, 16^2/3, 9, 5, 2, 1)$
p_{11} :	$(57, 44, 33, 24^1/3, 16^2/3, 9, 5, 2^2/3, 1/3)$
p_{12} :	$(57, 44, 33, 24, 16^2/3, 9^1/3, 5^1/3, 2^2/3, 0)$
p_{13} :	$(57, 44^2/3, 32^1/3, 24, 16, 10, 5^1/3, 2^2/3, 0)$
p_{14} :	$(57^1/3, 44^2/3, 32, 24^1/3, 16^2/3, 9, 5, 2, 1)$
p_{15} :	$(57^1/3, 44^2/3, 32, 24, 16, 10, 5^1/3, 2^2/3, 0)$
p_{16} :	$(57, 44, 33, 24^1/3, 16^2/3, 9, 5^1/3, 2^2/3, 0)$
p_{17} :	$(57, 44^2/3, 32^1/3, 24^1/3, 16^2/3, 9, 5^1/3, 2^2/3, 0)$
p_{18} :	$(57^1/3, 44^2/3, 32, 24^1/3, 16^2/3, 9, 5, 2^2/3, 1/3)$
p_{19} :	$(57^1/3, 44^2/3, 32, 24, 16^2/3, 9^1/3, 5^1/3, 2^2/3, 0)$
p_{20} :	$(57^1/3, 44^2/3, 32, 24^1/3, 16^2/3, 9, 5^1/3, 2^2/3, 0)$

There are 9 agents, each with a requirement of 1. Their m_is are $56, 45, 33, 23, 17, 10, 4, 3$ and 1. We depict only a subset of all equilibria here. In particular, we depict 20 equilibrium prices, $\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_{20}$. All prices either along solid lines connecting any two of these points, or in the shaded region are equilibria. However, if any two of these prices are not connected by a solid line, then none of the points on the line joining them is an equilibrium.

The equilibrium price found in this run is the point $\boldsymbol{p}_2$ in Table 2.1. This price curve is shown in Figure 2.3. The allocation obtained by solving the feasibility LP (2.3) is as follows: $x_{11} = 1, x_{22} = 1, x_{33} = 1, x_{44} = 4/5, x_{45} = 1/5, x_{54} = 1/5, x_{55} = 4/5, x_{66} = 1$.

We next describe the run of the algorithm on a network flow example, described in Figure 2.4. The figure shows the network structure and the edge labels specify (capacity, delay cost). There are five agents with requirements $10, 11, 12, 13, 14$ from s to t respectively. Their m_is are $12, 10, 4, 2, 2$. This network is not series-parallel, yet it satisfies the *extensibility* condition, so our algorithm finds an equilibrium.

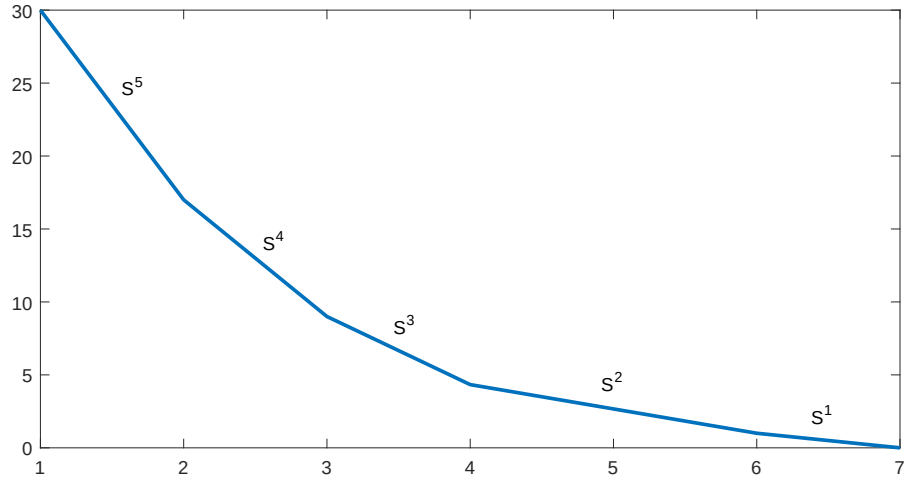


Figure 2.3: Piecewise linear convex decreasing curve of equilibrium prices obtained by the algorithm for the example in Table 2.1.

The run of Algorithm 3 on this example (in Figure 2.4) is as follows. Once again, each row below depicts one iteration, where we find a new segment. We first give the set of agents in the new segment, then the corresponding λ, and then the prices of the edges that are fixed in this iteration. The last column shows the second and third lowest λs in that iteration.

$$S^1 = \{3,4,5\}, \quad \lambda_{S^1} = {}^{8}\!/_{37}, \quad p_{sx} = p_{xt} = p_{wt} = 0, p_{st} = {}^{8}\!/_{37}, p_{sw} = {}^{16}\!/_{37}$$
$$(\lambda_{\{4,5\}} = {}^{4}\!/_{13}, \lambda_{\{2,3,4,5\}} = {}^{9}\!/_{35})$$
$$S^2 = \{1,2\}, \quad \lambda_{S^2} = {}^{478}\!/_{1147}, \quad p_{wu} = {}^{478}\!/_{1147}, \; p_{vt} = {}^{478}\!/_{1147}, p_{sv} = p_{uv} = p_{ut} = 0 \quad (\lambda_{\{2\}} = {}^{194}\!/_{407})$$

The allocation from the feasibility LP (2.3):

- Agent 1 sends ${}^{2012}\!/_{239}$ units of flow on path $s - w - u - v - t$ and ${}^{378}\!/_{239}$ units of flow on path $s - w - u - t$.

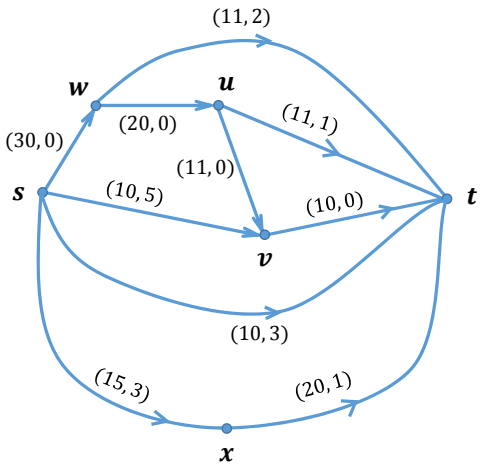


Figure 2.4: A network flow example.

There are 5 agents with requirements $10, 11, 12, 13, 14$ from s to t respectively. Their m_i's are $12, 10, 4, 2, 2$ respectively. The network satisfies the *extensibility* condition, so our algorithm finds an equilibrium.

- Agent 2 sends $^{2251}/_{239}$ units of flow on path $s - w - u - t$ and $^{378}/_{239}$ units of flow on path $s - w - u - v - t$.
- Agent 3 sends 8 units of flow on path $s - w - t$, $^{5}/_{2}$ units of flow on path $s - t$ and $^{3}/_{2}$ units of flow on path $s - x - t$.
- Agent 4 sends 2 units of flow on path $s - w - t$, $^{21}/_{4}$ units of flow on path $s - t$ and $^{23}/_{4}$ units of flow on path $s - x - t$.
- Agent 5 sends 2 units of flow on path $s - w - t$, $^{21}/_{4}$ units of flow on path $s - t$ and $^{27}/_{4}$ units of flow on path $s - x - t$.

2.5 Related work on computation and applications of market equilibrium

Computation and Complexity: The computational complexity of market equilibrium has been extensively studied for the tradition models in the past decade and a half. This investigation has involved many algorithmic techniques, such as primal-dual and flow based methods [40, 41, 88,

106, 107], auction algorithms [53], ellipsoid [62] and other convex programming based techniques [27], cell-decomposition [37, 38, 103], distributed price update rules [26, 30, 108], and complementary pivoting algorithms [48, 50], to name some of the most prominent. The algorithms have been complemented by hardness results, either for PPAD [25, 29] or for FIXP [52, 68], pretty much closing the gap between the two. Most of these papers focus on traditional utility functions used in the economics literature. A notable exception that considers *combinatorial* utility functions is [65], that study a market where agents want to send flow in a network, motivated by rate control algorithms governing the traffic in the Internet.

Beyond being an important component in the complexity theory of total functions [78], the computation of market equilibria has been studied by economists for much longer [16, 96, 102]. The classic case for the use of equilibrium *computation* is counter-factual evaluation of policy or design changes [100], based on the assumption that markets left to themselves operate at an equilibrium.

Fair allocation: Recently, market equilibrium outcomes have been used for fair allocation. Market equilibrium conditions are often considered inherently fair, therefore equilibrium outcomes have been used to allocate resources by a central planner seeking a fair allocation even when there is no actual market or monetary transfers. E.g., the proportional fair allocation, which is well known to be equivalent to the equilibrium allocation in a Fisher market [70], is widely used in the design of computer networks. Exchange of bandwidth in a bittorrent network is modeled as a process that converges to a market equilibrium by [108]. [17] proposes "competitive outcome from equal incomes" (CEEI) as a way to allocate courses to students: the allocation is an equilibrium in a market for courses in which the students participate with equal budgets (with random perturbations to break ties). This scheme has been successfully used at the Wharton business school [18]. [31] show that a suitable modification of the Fisher market equilibrium allocation can be used as a solution to a problem of fair resource allocation, without money. The mechanism is truthful,

and satisfies an approximate per-agent welfare guarantee. Truthful mechanisms have also been designed for scheduling, where it is the auctioneer who has jobs to be scheduled and the agents are the one providing the required resources e.g., see [72, 85]. This is in contrast to our setting where the agents have scheduling requirements.

Market based mechanisms: There is also a long history of "market based mechanisms", where a mechanism (with monetary transfers) implements an equilibrium outcome. The New York Stock Exchange uses such a mechanism to determine the opening prices, and copper and gold prices in London are fixed using a similar procedure [94]. There are different ways to do this: use a sample (either historic or random) or a probabilistic model of the population to compute the equilibrium price, and offer these prices to new agents. This is preferable to asking the bidders to report their preferences, computing the equilibrium on reported preferences and offering the equilibrium prices back. The latter leads to obvious strategic issues; [61] shows that strategic behavior by agents participating in such a mechanism can lead to inefficiencies. [2] show price of anarchy bounds on such mechanisms. In any case, such mechanisms are "incentive compatible in the large", meaning that as the market size grows and each agent becomes insignificant enough to affect prices on his own, his best strategy is to accept the equilibrium outcome. Nonetheless such mechanisms have been proposed and used in practice, e.g., for selling TV ads [86].

Budget constraints: Budget constraints in auctions has gained popularity in the last decade due to ad auctions [9, 42, 47, 79], but has been studied for quite some time [21, 71]. There has also been a recent line of work considering budget constraints in a procurement setting [4, 101].

2.6 Existence of Equilibrium under Strong Feasibility

In this section we show existence of equilibrium for market instances satisfying *strong feasibility* (Definition 1 in Section 2.1). Given such an instance $\mathcal{M}$ with set A of n agents and set G of m

goods, let us create another instance $\mathcal{M}'$ by adding an extra good s with "large quantity" and very high delay cost. Recall that the number of goods and agents in market $\mathcal{M}$ be m and n respectively.

Set of agents and goods in market $\mathcal{M}'$ are respectively A and $G' = G \cup \{s\}$. After normalizing to get quantity 1 for good s, we set coefficient of variable x_{is} in all the constraints of CC(i) to $a_s = (n+1)r_{max}$ for each agent, and set the delay cost for good s and for every agent to $d_s = m^{(m+1)} d_{max} a_s \left(\frac{a_{max}}{a_{min}}\right)^m$, where $a_{max} = \max_{i,j,k} |a_{ijk}|, a_{min} = \min\{1, \min_{i,j,k} |a_{ijk}|\}$, and $r_{max} = \max_{i,k} r_{ik}$. Thus, given prices $(p_1, \ldots, p_m, p_s)$ of goods in $\mathcal{M}'$, the optimal bundle of agent i at these prices can be found by solving the following linear program.

$$\begin{aligned} OPT_i(\boldsymbol{p}) = \arg\min: \quad & \textstyle\sum_j d_{ij} x_{ij} + d_s x_{is} \\ s.t. \quad & \textstyle\sum_j a_{ijk} x_{ij} + a_s x_{is} \geq r_{ik}, \ \forall k \in C \\ & \textstyle\sum_j p_j x_{ij} + p_s x_{is} \leq m_i \\ & x_{ij} \geq 0, \ \forall j \in G'. \end{aligned} \tag{2.8}$$

The next lemma follows from the construction of market $\mathcal{M}'$.

Lemma 8. *If $\mathcal{M}$ satisfies strong feasibility then so does $\mathcal{M}'$.*

Price vector $\boldsymbol{p}$ is said to be at equilibrium, if when every agent is given its optimal bundle, there is no excess demand of any good, and goods with excess supply have price zero. That is, $(X, \boldsymbol{p})$ such that,

$$\forall i \in A, \ \boldsymbol{x}_i \in OPT_i(\boldsymbol{p}), \quad \text{and} \quad \forall j \in G', \ \sum_{i \in A} x_{ij} \leq 1; \quad p_j > 0 \Rightarrow \sum_{i \in A} x_{ij} = 1. \tag{2.9}$$

Lemma 9. *If $\boldsymbol{x}_i \in OPT_i(\boldsymbol{p}), \ \forall i \in A$ at prices $\boldsymbol{p} \geq 0$ for $\mathcal{M}'$, then $\sum_i x_{is} < 1$. That is $p_s = 0$ at equilibrium.*

Proof. It is easy to see that $x_{is} \leq \frac{1}{n+1}, \ \forall i$, and hence the proof follows. □

Next we show that equilibria of $\mathcal{M}$ and $\mathcal{M}'$ are related.

Lemma 10. *If $\mathcal{M}$ satisfies strong feasibility, then every equilibrium of $\mathcal{M}'$ gives an equilibrium of $\mathcal{M}$.*

Proof. Let $(X^*, \boldsymbol{p}^*)$ respectively be an equilibrium allocation and prices of $\mathcal{M}'$. From Lemma 9, we know that $p^*_s = 0$. It suffices to show that $x^*_{is} = 0,\ \forall i \in A$, for the lemma to follow.

To the contrary suppose for some agent u, $x^*_{us} > 0$. We will construct another bundle $\boldsymbol{x}'_u$ that is affordable to agent u at prices $\boldsymbol{p}^*$, satisfies CC(u), and has a lower delay than $\boldsymbol{x}^*_u$, contradicting optimal bundle condition at equilibrium.

Due to *strong feasibility*, after all $i \neq u$ is given their bundle $\boldsymbol{x}^*_i$, there will be a bundle $\boldsymbol{x}'_u$ left for u to buy among goods in G such that $CC(u)$ constraints are satisfied. Clearly, one way to construct such $\boldsymbol{x}'_u$ is that the agent keeps buying all goods $j \neq s$ as in $\boldsymbol{x}^*_u$, and starts decreasing x^*_{us} and increasing allocation for some other available good. Note that all such available goods are under-sold at equilibrium and therefore has zero price in $\boldsymbol{p}^*$. Thus payment for $\boldsymbol{x}'_u$ and $\boldsymbol{x}^*_u$ are the same at prices $\boldsymbol{p}^*$. In other words $\boldsymbol{x}'_u$ is affordable at prices $\boldsymbol{p}^*$.

If good j is increased by δ_j as we go from $\boldsymbol{x}^*_u$ to $\boldsymbol{x}'_u$, then we claim that $\delta_j \leq m! a_s \left(\frac{a_{max}}{a_{min}}\right)^m x^*_{us}$, where $m = |G|$. This is because, in a constraint even if coefficient of variable x_{uj} is minimum possible, and it needs to compensate for increase in other goods due to their negative coefficients, this cascade could at most harm by a factor of $m! \left(\frac{a_{max}}{a_{min}}\right)^m$. Difference in delay is

$$\begin{aligned}
\textstyle\sum_j d_{uj}(x'_{uj} - x^*_{uj}) - d_s x^*_{us} &= \textstyle\sum_j d_{uj}\delta_j - d_s x^*_{us} \\
&\leq m d_{max}(\max_j \delta_j) - d_s x^*_{us} \\
&\leq \left(m^{(m+1)} d_{max} a_s \left(\frac{a_{max}}{a_{min}}\right)^m - d_s\right) x^*_{us} < 0.
\end{aligned}$$

□

Due to Lemma 10, to show existence of equilibrium for market $\mathcal{M}$ it suffices to show one for $\mathcal{M}'$. Next to show existence of equilibrium for market $\mathcal{M}'$ it suffices to consider price vectors

where $p_s = 0$ due to Lemma 9, and therefore we consider the following set of possible price vectors.

$$P = \{\boldsymbol{p} \in \mathbb{R}_+^{(m+1)} \mid p_s = 0;\ \sum_{j \in G} p_j \leq M\} \quad \text{where } M = \textstyle\sum_i m_i.$$

Let us first handle trivial instances. It is easy to see that the feasible set of $\boldsymbol{x}_i$s in LP (2.8) at $\boldsymbol{p} = \mathbf{0}$ is a superset of the feasible set at any other prices $\boldsymbol{p}$. Therefore, for agent i if $x_i = \mathbf{0} \in OPT_i(\mathbf{0})$, then she will not buy anything at any prices. In that case, it is safe to discard her from the market. Further, if there is an allocation X satisfying Supply constraints for market $\mathcal{M}'$ such that $\boldsymbol{x}_i \in OPT_i(\mathbf{0}),\ \forall i \in A$, then we get a trivial equilibrium of $\mathcal{M}'$ where all the prices are set to zero; note that in this case zero prices also constitute an equilibrium of market $\mathcal{M}$ by Lemma 10. To show existence for non-trivial instances, w.l.o.g. now on we assume the following for market $\mathcal{M}'$.

Enough Demand (ED): If X is such that $\forall i \in A,\ \boldsymbol{x}_i \in OPT_i(\mathbf{0})$, then $\boldsymbol{x}_i \neq \mathbf{0},\ \forall i$, and there exists a good $j \in G'$ such that $\sum_i x_{ij} > 1$. Clearly, $j \neq s$ due to Lemma 9.

Lemma 11. *For any $\boldsymbol{p} \in P$, $OPT_i(\boldsymbol{p})$ is non-empty, and assuming enough demand (ED), $\mathbf{0} \notin OPT_i(\boldsymbol{p}),\ \forall i \in A$.*

Proof. The first part of the proof is easy to see due to the extra good s whose price is zero in P. For the second part, to the contrary suppose for $\boldsymbol{x}_i \in OPT_i(\boldsymbol{p})$ we have $\boldsymbol{x}_i = \mathbf{0}$. By ED assumption we know that for any $\boldsymbol{x}_i^0 \notin OPT_i(\mathbf{0})$, we have $\boldsymbol{x}_i^0 \neq \mathbf{0}$. Further, feasible set of LP (2.8) is at prices $\boldsymbol{p}$ is a subset of the feasible set of this LP at zero prices. Hence we know that $\sum_j d_{ij} x_{ij}^0 < \sum_j d_{ij} x_{ij} = 0$.

It must be the case that $\boldsymbol{x}_i^0$ is not affordable at prices $\boldsymbol{p}$, i.e., $m_i < \sum_j x_{ij}^0 p_j$. For $\lambda = m_i / \sum_j x_{ij}^0 p_j$, set $\boldsymbol{x}_i' = \lambda \boldsymbol{x}_i^0 + (1-\lambda)\boldsymbol{x}_i$. Since $\boldsymbol{x}_i$ and $\boldsymbol{x}_i^0$ both satisfy CC(i), so does $\boldsymbol{x}_i'$. And since $\boldsymbol{x}_i = \mathbf{0}$ bundle $\boldsymbol{x}_i'$ is affordable at prices $\boldsymbol{p}$, thereby $\boldsymbol{x}_i'$ is feasible in $OPT_i(\boldsymbol{p})$. The delay at

$\boldsymbol{x}'_i$ is $\sum_i d_{ij}x'_{ij} = \sum_j d_{ij}(\lambda x^0_{ij} + (1-\lambda)x_{ij}) = \lambda \sum_j d_{ij}x^0_{ij} < 0 = \sum_j d_{ij}x_{ij}$, a contradiction to $\boldsymbol{x}_i$ being optimal bundle at prices $\boldsymbol{p}$. □

Next we will construct a correspondence whose fixed points are exactly the market equilibria of $\mathcal{M}'$. Let c_j^{max} be the maximum possible demand of good j; we can compute c_j^{max} by maximizing $\sum_i x_{ij}$ over the CC(i) constraints of all agents $i \in A$. Define domain

$$D = \{(X, \boldsymbol{p}) \mid \boldsymbol{p} \in P; \quad X \geq 0; \quad \forall j, \sum_i x_{ij} \leq c_j^{max}\}$$

Let $\delta = \min_i m_i$. Define correspondence $F : D \to D$ as follows where for a given $(\bar{X}, \bar{\boldsymbol{p}}) \in D$, we have $(X', \boldsymbol{p}') \in F(\bar{X}, \bar{\boldsymbol{p}})$,

$$\forall i \in A,\ \boldsymbol{x}'_i \in OPT_i(\bar{\boldsymbol{p}}), \quad \text{and} \quad \boldsymbol{p}' \in \underset{\boldsymbol{p} \in P,\ \delta \leq \sum_j p_j \leq \max\{\delta, \sum_{i\in A, j\in G'} \bar{x}_{ij}\bar{p}_j\}}{\arg\max} \sum_{i\in A, j\in G'} \bar{x}_{ij}p_j. \tag{2.10}$$

The correspondence is well defined due to Lemma 11. If $F(\bar{X}, \bar{\boldsymbol{p}})$ is a convex set, and graph of F is closed, then Kakutani's Theorem [69] implies that F has a fixed point, i.e., $\exists (X^*, \boldsymbol{p}^*) \in D$ such that $(X^*, \boldsymbol{p}^*) \in F(X^*, \boldsymbol{p}^*)$. Next we show the same.

Lemma 12. *Correspondence F has a fixed point.*

Proof. Clearly, $F(\bar{X}, \bar{\boldsymbol{p}})$ is a convex set since it is a cross product of solution sets of LPs. The lemma follows using Kakutani's fixed-point Theorem [69] if graph of F is closed.

Let $(\bar{X}^t, \bar{\boldsymbol{p}}^t)$ for $t = 1, 2, \ldots$ be a sequence of points in D, and let $(X^t, \boldsymbol{p}^t)$ be another sequence such that $(X^t, \boldsymbol{p}^t) \in F(\bar{X}^t, \bar{\boldsymbol{p}}^t)$. Then essentially, X^t and $\boldsymbol{p}^t$ are solutions of LPs that are continuously changing with $(\bar{X}, \bar{\boldsymbol{p}})$. Therefore, if $\lim_{t\to\infty}(\bar{X}^t, \bar{\boldsymbol{p}}^t) = (\bar{X}^*, \bar{\boldsymbol{p}}^*)$ and $\lim_{t\to\infty}(X^t, \boldsymbol{p}^t) = (X^*, \boldsymbol{p}^*)$, then by continuity of parameterized LP solutions, we get that $(X^*, \boldsymbol{p}^*) \in F(\bar{X}^*, \bar{\boldsymbol{p}}^*)$, implying graph of F is closed. □

Lemma 13. *If $(X^*, \boldsymbol{p}^*)$ is a fixed-point of F then $\forall j \in G', \sum_{i \in A} x^*_{ij} \leq 1$.*

Proof. Since $\boldsymbol{x}^*_i \in OPT_i(\boldsymbol{p}^*),\ \forall i \in A, \sum_{i \in A} x^*_{is} \leq 1$ follows using Lemma 9. Among the rest of the goods, suppose, for $j' \neq s$ we have $\sum_{i \in A} x^*_{ij'} > 1$. Let $U = \max\{\delta, \sum_{i,j} x^*_{ij} p^*_j\}$. Then the optimization problem of (3.3) is

$$\max_{\boldsymbol{p} \in P,\ \delta \leq \sum_j p_j \leq U} \sum_{i,j} x^*_{ij} p_j = \max_{\boldsymbol{p} \in P,\ \delta \leq \sum_j p_j \leq U} \sum_j p_j \sum_i x^*_{ij}.$$

The above quantity can be made more than U by setting $p_{j'} = U$, and therefore optimal value is strictly more than U. However, due to the fixed-point condition $\boldsymbol{p}^*$ is a solution of the above, which implies $\sum_{i,j} x^*_{ij} p^*_j > U$, a contradiction. □

Lemma 14. *Assuming* enough demand (ED)*, if $(X^*, \boldsymbol{p}^*)$ is a fixed-point of F then $\boldsymbol{p}^* \neq 0$ and $\sum_{i,j} x^*_{ij} p^*_j \geq \delta$.*

Proof. For the first part, to the contrary suppose $\boldsymbol{p}^* = 0$. Then $\forall i \in A, \boldsymbol{x}^*_i \in OPT_i(\mathbf{0})$ since $(X^*, \boldsymbol{p}^*)$, and therefore by ED condition $\exists j \neq s,\ x^*_{ij} > 1$. However $\sum_{i,j} x^*_{ij} p^*_j = 0$. Therefore,

$$\max_{\boldsymbol{p} \in P,\ \delta \leq \sum_j p_j \leq \max\{\delta, \sum_{i,j} x^*_{ij} p^*_j\}} \sum_{i,j} x^*_{ij} p_j = \max_{\boldsymbol{p} \in P,\ \sum_j p_j = \delta} \sum_{i,j} x^*_{ij} p_j > 0.$$

This contradicts the fact that $\boldsymbol{p}^*$ is a maximizer of the above.

For the second part, to the contrary suppose $\sum_{i,j} x^*_{ij} p^*_j < \delta$, then $\max\{\delta, \sum_{i,j} x^*_{ij} p^*_j\} = \delta$. Further, due to Lemma 11 (together with the ED assumption) we have $\forall i \in A, \boldsymbol{x}^*_i \neq \mathbf{0}$, and therefore at maximum $\sum_j p_j = \delta$. Since $\boldsymbol{p}^*$ is a maximizer of the above, $\sum_j p^*_j = \delta$.

However since $\sum_{i,j} x^*_{ij} p^*_j < \delta$, we get $\forall i, \sum_j x^*_{ij} p^*_j < m_i$ where $\boldsymbol{x}^*_i$ is feasible in LP (2.8) at prices $\boldsymbol{p}^*$. Let X^0 be a demand vector when all the prices are zero. Due to Lemma 13 and ED assumption we get that for some agent i, $\boldsymbol{x}^*_i \notin OPT_i(\mathbf{0})$. Let i' be this agent. This implies $\sum_j d_{i'j} x^0_{i'j} < \sum_j d_{i'j} x^*_{i'j}$. Despite lower cost at $\boldsymbol{x}^0_{i'}$ she demands $\boldsymbol{x}^*_{i'}$ at prices $\boldsymbol{p}^*$, hence it should be the case that she can not afford $\boldsymbol{x}^0_{i'}$ at those prices. However, since i' is also not spending all the

money at prices $\boldsymbol{p}^*$, there exists some $0 < \tau < 1$ such that she can afford $X'_{i'} = \tau \boldsymbol{x}^*_{i'} + (1-\tau)\boldsymbol{x}^0_{i'}$ at $\boldsymbol{p}^*$. Since both $\boldsymbol{x}^0_{i'}$ and $\boldsymbol{x}^*_{i'}$ satisfy the $CC(i')$ constraints of LP (2.8) for agent i', so does $X'_{i'}$. Thus, $X'_{i'}$ is a feasible point in $OPT_{i'}(\boldsymbol{p}^*)$ LP, and $\sum_j d_{i'j} x'_{i'j} < \sum_j d_{i'j} x^*_{i'j}$, a contradiction to $\boldsymbol{x}^*_{i'} \in OPT_{i'}(\boldsymbol{p}^*)$. □

Next we show the main result using Lemmas 12, 13 and 14.

Theorem 4. *If $\mathcal{M}'$ satisfies strong feasibility then it has an equilibrium.*

Proof. Due to Lemma 12, we know that there exists a fixed-point of correspondence F. Let this be $(X^*, \boldsymbol{p}^*)$. We will show that it is a market equilibrium of $\mathcal{M}'$. Clearly, optimal allocation condition is satisfied because $\boldsymbol{x}^*_i \in OPT_i(\boldsymbol{p}^*)$. Market clearing remains to be shown, which requires: (a) $\forall j,\ \sum_i x^*_{ij} \leq 1$, and (b) $p^*_j > 0 \Rightarrow \sum_i x^*_{ij} = 1$.

(a) follows from Lemma 13. For (b), let $U = \max\{\delta, \sum_{i,j} x^*_{ij} p^*_j\}$, then due to Lemma 14, $U = \sum_{i,j} x^*_{ij} p^*_j$. The optimization problem of (3.3) is

$$\max_{\boldsymbol{p} \in P,\ \delta \leq \sum_j p_j \leq U} \sum_{i,j} x^*_{ij} p_j = \max_{\boldsymbol{p} \in P,\ \delta \leq \sum_j p_j \leq U} \sum_j p_j \sum_i x^*_{ij}.$$

Clearly, at optimal solution of the above p_j is non-zero only where $\sum_i x^*_{ij}$ is maximum. Since $\boldsymbol{p}^*$ is a solution, if $\exists j,\ \sum_i x^*_{ij} = 1$, then (b) follows.

On the other hand suppose for all j we have $\sum_i x^*_{ij} < 1$, then clearly the optimal value of the above is strictly less than U. However since $\boldsymbol{p}^*$ is a maximizer it implies that $\sum_{i,j} x^*_{ij} p^*_j < U$, a contradiction to Lemma 14. □

The next theorem follows using Lemmas 8, 10, and Theorem 4.

Theorem 1. *[Strong feasibility implies the existence of an equilibrium] If* $(\text{CC}(i))_{i \in A}$ *of market $\mathcal{M}$ satisfies strong feasibility, then $\exists$ an allocation X and prices $\mathbf{p}$ that constitute a market equilibrium of $\mathcal{M}$.*

Remark 1. By similar argument, we can show existence of equilibrium for market instances satisfying only extensibility condition (see Definitions 3). However, since our algorithm returns an equilibrium of such a market, it already gives a constructive proof of existence.

2.6.1 Quasi-concave utility functions

In this section, we show that the preferences of agents in our model can be captured by quasi-concave utility functions. Notation: the symbol $\leq$ when used for vectors represents a co-ordinate wise relation, and $<$ represents that at least one of the inequalities is strict. Define the utility of an agent i for an allocation $\boldsymbol{x}_i$ to be the smallest delay of a feasible allocation dominated by $\boldsymbol{x}_i$, times -1:

$$U_i(\boldsymbol{x}_i) = -\min\left\{\boldsymbol{d}_i \cdot \boldsymbol{x}'_i : \boldsymbol{x}'_i \leq \boldsymbol{x}_i \;\&\; \boldsymbol{x}'_i \text{ is feasible for Delay LP}(i)\right\}.$$

If there is no $\boldsymbol{x}'_i \leq \boldsymbol{x}_i$ that is feasible for Delay LP(i) then the utility is $-\infty$. It is easy to check that this utility function is quasi-concave, and induces the same preferences as in our model.

2.7 Special Cases

In this section, we show how the applications mentioned in Section 2.1.3 satisfy the extensibility condition (Definition 3 in Section 2.1).

Scheduling. Recall the scheduling problem from Section 2.1.3. The agents are jobs that require d different types of machines, and the set of time slots on machines of type k is M_k; pair (machine type, time slot) defines a good in the market. Each agent needs $r_{ik} \in \mathbb{R}_+$ units of time on machine type k, which is captured by the covering constraint $\sum_{j \in M_k} x_{ijk} \geq r_{ik},\ \forall k \in [d]$. All agents experience the same delay d_{jk} from time slot j on type k machine.

Lemma 15. *Scheduling problem satisfies the extensibility condition.*

Proof. Consider an arbitrary set S of agents and an agent $\hat{i}$ outside of this set. Let $(\boldsymbol{x}_i)_{i \in S}$ be a

feasible allocation that minimizes the total delay of S, i.e., $\sum_{i\in S, j\in M_k, k\in[d]} d_{jk}x_{ijk}$. Since the delay values are the same for each agent the total delay minimizes when the agents in S get $\sum_{i\in S} r_{ik}$ units of machines of type k with the smallest delay. Therefore, if we assign the next r_{ik} units of machines of type k with the smallest delay to agent $\hat{i}$ then $(x_i)_{i\in(S\cup\hat{i})}$ would be the feasible allocation that minimizes the total delay. Therefore, this problem satisfies the extensibility condition. □

Restricted assignment with laminar families. Recall this setting from Section 2.1.3. The above basic scheduling setting can be generalized to the following restricted assignment case, where job i is allowed to be processed only on a subset of all the time slots $S_{ik} \subseteq M_k$ on the machine type k. We need the S_{ik}s to form a laminar[6] family within each type, and in addition, we require that the machines in a larger subset have lower delays. That is, if for some two agents $i, i' \in A$, $S_{i'k} \subset S_{ik}$ then $\max_{j\in S_{ik}\setminus S_{i'k}} d_{jk} \leq \min_{j'\in S_{i'k}} d_{j'k}$ for each type k.

Lemma 16. *Restricted assignment with laminar families satisfies the extensibility condition if the following assumption holds for the instance*

- *(Monotonicity)* $\forall i, i' \in A$*, such that* $S_{i'} \subset S_i$ *then* $\max_{j\in S_i\setminus S'_i} d_{jk} \leq \min_{j'\in S'_i} d_{j'k}$ *for each type* $k \in [d]$.

Proof. Since the requirement and variables for each machines type is separate, it is enough to show this for $k = 1$. Consider an arbitrary set of agents T and an agent outside of this set $\hat{i}$. Let $T' = T \cup i$. Let $(\boldsymbol{x}_i)_{i\in T}$ to be a feasible allocation that minimizes total delay, i.e., $\sum_{i\in T} \boldsymbol{d}.\boldsymbol{x}_i$. For simplicity let $S_{\hat{i}} = \{1, 2, \ldots, m\}$ such that $d_1 \leq \cdots \leq d_m$. Consider two agent i and i'. Note that if $j \in S_i$ and $j \in S_{i'}$ then $\forall j' \in S_i$ such that $d_{j'} \leq d_j$ we have $j \in S_{i'}$ and vice versa because S_is form a laminar family and the monotonicity condition. Therefore, an optimal allocation allocates only a prefix of time slots in $S_{\hat{i}}$. Let's assign to $\hat{i}$ the next $r_{\hat{i}}$ slots with smallest delay in subset $S_{\hat{i}}$. We claim the new allocation $(\boldsymbol{x}_i)_{i\in S'}$ is minimizing the total delay, i.e., $\sum_{i\in S'} \boldsymbol{d}.\boldsymbol{x}_i$. Let's prove the

[6]A family of subsets is said to be *laminar* if any two sets S and T in the family are either disjoint, $S \cap T = \emptyset$, or contained in one another, $S \subseteq T$ or $T \subseteq S$.

claim by contradiction. Suppose the allocation is not optimal. Therefore there exists an optimal allocation $(\boldsymbol{x}'_i)_{i\in S'\cup\{\hat{i}\}}$ with less total delay. Suppose allocation X' has allocated slots 1 through l' in set $S_{\hat{i}}$ and allocation X has allocated slots 1 through l in set $S_{\hat{i}}$. Note that in allocation X' we can assume agent $\hat{i}$ is getting the last slot in X' among the machines that has been allocated in S_i because if there exists agent $\bar{i}$ that has allocation on the right side of the first slot that is allocated to $\hat{i}$ then we can swap the allocations so the total delay wouldn't change. If $l = l'$ then the delay of agent $\hat{i}$ is the same in X and X'. This is a contradiction because then it would conclude that the total delay of allocation $(\boldsymbol{x}'_i)_{i\in S}$ is less than total delay of $(\boldsymbol{x}_i)_{i\in S}$ but we assumed X is an optimal allocation for S. There are two cases.

Case1. $l < l'$. Consider allocation X' after removing agent $\hat{i}$. Since $\hat{i}$ is getting the last slots it is easy to see the remaining allocation is optimal for S. Since $l < l'$ there are agents that have less allocation in S_i in X compare to X'. Because of *monotonic delay* assumption they have been allocated to machines with highest delay instead of available machines in S_i. This is a contradiction with the fact X is a optimal allocation for S.

Case2. $l > l'$. This case is very similar to the last case. With the same argument we can argue that this case has contradiction with the fact X' is an optimal allocation. □

Network flows. Recall that in this setting agent i wants to send r_i units of flow from s to t in a directed (graph) network where each edge has a capacity and cost per unit flow specified. Here edges are goods, and the covering constraints of agent i has variable f_{ie} for each edge e representing her flow on edge e. The constraints impose flow conservation at all nodes except s and t, and that net outgoing and incoming flow at s and t respectively is r_i.

Lemma 17. *A series-parallel network satisfies the extensibility condition.*

Proof. Consider an arbitrary set of agents S and an agent outside of this set $\hat{i}$. Let $(f_i)_{i\in S}$ to be a

feasible min cost flow. Let's remove the allocated capacities from the graph and allocate min cost flow of size $r_{\hat{i}}$ to agent $\hat{i}$ in the remaining graph. It is known that this greedy algorithm gives a min cost flow of size $\sum_{i \in S \cup \hat{i}} r_i$ [8]. □

We can consider independent copies of any of these special cases above. E.g., each agent might want some machines for job processing, as well as send some flows through a network, but have a common budget for both together. Note that it would remain extensible because each copy is independent and extensible itself.

2.8 Equilibrium Characterization

In this section we characterize equilibria of the most general market instances. Recall that allocation $\boldsymbol{x}_i$ of agent i has to satisfy its covering constraints CC(i). Next we derive sufficient conditions for prices $\boldsymbol{p}$ and allocation X to be an equilibrium. As we discussed in Section 2.1, given prices $\boldsymbol{p}$ the optimal bundle of each agent i is captured by

$$\begin{array}{lll} \min & \sum_{j \in G} d_{ij} x_{ij} & \\ s.t. & \sum_{j \in G} a_{ijk} x_{ij} \geq r_{ik} & \forall k \in C \\ & \sum_{j \in G} p_j x_{ij} \leq m_i & \\ & x_{ij} \geq 0, \ \forall j \in G \, . & \end{array} \tag{OB-LP(i))}$$

It is well know that the solutions of a linear program are exactly the ones that satisfy the complementary slackness conditions [99]. Let β_{ik} and γ_i be the dual variables of the first and second of constraints in OB-LP(i). Then, the corresponding complementary slackness conditions are

$$\begin{array}{lllll} \forall k \in C: & \sum_{j \in G} a_{ijk} x_{ij} \geq r_{ik} & \perp & \beta_{ik} \geq 0 \\ \forall j \in G: & d_{ij} \geq \sum_k a_{ijk} \beta_{ik} - \gamma_i p_j & \perp & x_{ij} \geq 0 \\ & \sum_{j \in G} p_j x_{ij} \leq m_i & \perp & \gamma_i \geq 0 \, . \end{array} \tag{2.11}$$

Here $\perp$ symbol between two inequalities means that both inequalities should be satisfied, and at least one of them has to hold with equality. Let's define $Z = \{i \mid \gamma_i = 0\}$, For all i in Z and for all k define $\alpha_{ik} = \beta_{ik}$, and for all i in $A \setminus Z$ define $\lambda_i = 1/\gamma_i$ and $\alpha_{ik} = \beta_{ik}/\gamma_i$ for each constraint k. Then the above conditions of (2.11)can be rewritten as:

$$\begin{array}{lllll}
\forall i \in Z, \forall k \in C: & \sum_{j \in G} a_{ijk} x_{ij} \geq r_{ik} & \perp & \alpha_{ik} \geq 0 \\
\forall i \in Z, \forall j \in G: & d_{ij} \geq \sum_k a_{ijk} \alpha_{ik} & \perp & x_{ij} \geq 0 \\
\forall i \in A \backslash Z, \ \forall k \in C: & \sum_{j \in G} a_{ijk} x_{ij} \geq r_{ik} & \perp & \alpha_{ik} \geq 0 \\
\forall i \in A \backslash Z, \forall j \in G: & \lambda_i d_{ij} \geq \sum_k a_{ijk} \alpha_{ik} - p_j & \perp & x_{ij} \geq 0 \\
\forall i \in A \backslash Z: & \sum_{j \in G} p_j x_{ij} = m_i & \text{and} & \lambda_i > 0
\end{array} \tag{2.12}$$

At equilibrium prices, every agent should get an optimal bundle and market should clear, i.e., Supply constraints are satisfied and every good with positive price should be fully sold (see Section 2.1 for the formal definition of market equilibrium). Since optimal allocations at given prices are solutions of OB-LP(i) for each i, they must satisfy (2.12). This follows from the fact that primal-dual feasibility and complementary slackness conditions are necessary and sufficient for the solutions of a linear program. We get the following characterization.

Lemma 18. *If* $(\hat{\boldsymbol{\lambda}}, \hat{X}, \hat{\boldsymbol{p}}, \hat{\boldsymbol{\alpha}})$ *satisfies (2.12), and* $\forall j \in G, \ \sum_{i \in A} \hat{x}_{ij} \leq 1 \ \perp \ \hat{p}_j \geq 0$*, then* $(\hat{X}, \hat{\boldsymbol{p}})$ *constitutes an equilibrium allocation and prices.*

Motivated from Lemma 18 we next define a parameterized LP that captures complementary slackness conditions of OB-LP(i) for all the agents together. Suppose we are given λ_i's, let us define the following linear program parameterized by the $\boldsymbol{\lambda}$ vector, that we call $LP(\boldsymbol{\lambda})$, and its dual $DLP(\boldsymbol{\lambda})$ (same as (2.2) defined in Section 2.3):

$$
\begin{array}{lll}
\multicolumn{3}{c}{LP(\boldsymbol{\lambda})} \\
\min: & \sum_{i,j} \lambda_i d_{ij} x_{ij} & \\
s.t. & \sum_{j \in G} a_{ijk} x_{ij} \geq r_{ik} & \forall i \in A, k \in C \\
& \sum_{i \in A} x_{ij} \leq 1 & \forall j \in G \\
& x_{ij} \geq 0 & \forall i \in A, j \in G
\end{array}
\qquad
\begin{array}{lll}
\multicolumn{3}{c}{DLP(\boldsymbol{\lambda})} \\
\max: & \sum_{i \in A, k \in C} r_{ik} \alpha_{ik} - \sum_j p_j & \\
s.t. & \lambda_i d_{ij} \geq \sum_k a_{ijk} \alpha_{ik} - p_j & \forall i \in A, j \in G \\
& \alpha_{ik}, p_j \geq 0 & \forall j \in G, k \in C\ .
\end{array}
$$

Using the equilibrium characterization of Lemma 18 together with the complementary slackness conditions between constraints of $LP(\boldsymbol{\lambda})$ and $DLP(\boldsymbol{\lambda})$, next we show that the solutions of $LP(\boldsymbol{\lambda})$ and $DLP(\boldsymbol{\lambda})$ exactly capture the equilibria if given appropriate value of parameter vector $\boldsymbol{\lambda}$.

Theorem 3. *For a given $\boldsymbol{\lambda} > 0$ if an optimal solution X of $LP(\boldsymbol{\lambda})$ and an optimal solution $(\boldsymbol{\alpha}, \boldsymbol{p})$ of $DLP(\boldsymbol{\lambda})$ satisfy Budget constraint(i) for all agents $i \in A$, and at $(X, \boldsymbol{p})$ every agent i either spends all her budget, or $\boldsymbol{x}_i \in OPT_i(\mathbf{0})$, then $(X, \boldsymbol{p})$ constitute an equilibrium of market $\mathcal{M}$.*

Proof. It suffices to show that $(\boldsymbol{\lambda}, \boldsymbol{a}, \boldsymbol{p}, X)$ satisfies the conditions of Lemma 18. Let Z be the set of agents with $\boldsymbol{x}_i \in OPT_i(\mathbf{0})$. Clearly, $(\boldsymbol{a}, \boldsymbol{p}, X)$ satisfies first two conditions of (2.12) just by definition of set Z. The last one of (2.12) is already assumed in the hypothesis since agents not in Z must spend all their budget.

For the remaining conditions, let us write the complementary slackness conditions for $LP(\boldsymbol{\lambda})$.

$$\sum_j a_{ijk} x_{ij} \geq r_{ik} \quad \perp \quad \alpha_{ik} \geq 0, \ \ \forall i, k. \tag{2.13}$$

$$\lambda_i d_{ij} \geq \sum_k a_{ijk} \alpha_{ik} - p_j \quad \perp \quad x_{ij} \geq 0, \ \ \forall i, j. \tag{2.14}$$

$$\sum_i x_{ij} \leq 1 \quad \perp \quad p_j \geq 0, \ \ \forall j. \tag{2.15}$$

Conditions (2.13) and (2.14) are exactly the third and fourth conditions of (2.12), and (2.15) ensures market clearing. Thus the proof follows using Lemma 18. □

Next we show the converse of the above theorem under the assumption that $\gamma_i > 0, \forall i \in A$.

Lemma 19. *Given an equilibrium $(\hat{X}, \hat{\boldsymbol{p}})$ and the value of corresponding dual variables $\hat{\beta}_{ik}$ and $\hat{\gamma}_i$ in (2.11) for market $\mathcal{M}$ such that $\hat{\gamma}_i > 0, \forall i$ then $\hat{X}$ and $(\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{p}})$ give a solution of $LP(\hat{\boldsymbol{\lambda}})$ and $DLP(\hat{\boldsymbol{\lambda}})$ respectively for some $\hat{\boldsymbol{\lambda}}$ and $\hat{\boldsymbol{\alpha}}$.*

Proof. It is easy to see using (2.12) that for $Z = \emptyset$, $\hat{\lambda}_i = 1/\hat{\gamma}_i, \ \forall i \in A$ and $\hat{\alpha}_{ik} = \hat{\beta}_{ik}/\hat{\gamma}_i, \ \forall i \in A, k \in C$, $(\hat{X}, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{p}})$ satisfies the complementary slackness conditions of $LP(\hat{\boldsymbol{\lambda}})$ and $DLP(\hat{\boldsymbol{\lambda}})$. □

Using the equilibrium characterization given by Theorem 3 crucially, we design a polynomial-time algorithm to find an equilibrium for markets with extensibility (Definition 3) in Section 2.3.

2.9 Missing Proofs and Details of Section 4

The proof of first theorem of Section 2.3, namely Theorem 3 is in Section 2.8 where we characterize market equilibria. Next we give proof of Lemma 1. For this we basically use the fact that any pair of primal, dual solutions of a linear program has to satisfy complementary slackness [99].

Recall the $LP(\boldsymbol{\lambda})$ and $DLP(\boldsymbol{\lambda})$ of (2.2).

$$
\begin{array}{lll}
 & LP(\boldsymbol{\lambda}): & \\
\min: & \sum_i \lambda_i \sum_j d_{ij} x_{ij} & \\
s.t. & \sum_j a_{ijk} x_{ij} \geq r_{ik},\ \forall k & \\
 & \sum_i x_{ij} \leq 1,\ \forall i & \\
 & x_{ij} \geq 0,\ \forall (i,j) &
\end{array}
\qquad
\begin{array}{lll}
 & DLP(\boldsymbol{\lambda}) & \\
\max: & \sum_{i,k} r_{ik}\alpha_{ik} - \sum_j p_j & \\
s.t. & \lambda_i d_{ij} \geq \sum_k a_{ijk}\alpha_{ik} - p_j,\ \forall (i,j) & \\
 & p_j \geq 0,\ \forall j; \quad \alpha_{ik} \geq 0,\ \forall (i,k). &
\end{array}
\tag{2.16}
$$

For any given $\boldsymbol{\lambda} > 0$, the optimal solutions of the $LP(\boldsymbol{\lambda})$, namely X, and $DLP(\boldsymbol{\lambda})$, namely $(\boldsymbol{\alpha}, \boldsymbol{p})$ has to satisfy the following complementary slackness conditions.

$$
\begin{array}{lllll}
\forall (i,k): & \alpha_{ik} \geq 0 & \perp & \sum_j a_{ijk} x_{ij} \geq r_{ik} \\
\forall j: & p_j \geq 0 & \perp & \sum_i x_{ij} \leq 1 \\
\forall (i,j): & x_{ij} \geq 0 & \perp & \lambda_i d_{ij} \geq \sum_k a_{ijk}\alpha_{ik} - p_j.
\end{array}
\tag{2.17}
$$

Recall that the $\perp$ symbol between two inequalities means that both inequalities should be satisfied, and at least one of them has to hold with equality. Also recall that for subset $S \subseteq A$, we defined $\text{delay}_S(X, \boldsymbol{p}) = \sum_{i \in S, j} d_{ij} x_{ij}$ and $\text{pay}_S(X, \boldsymbol{p}) = \sum_{i \in S, j} x_{ij} p_j$, and for ease of notation we use delay_i when $S = \{i\}$ and similarly pay_i. Using this we first show a relation between delay_i and pay_i next.

Lemma 20. *For a given $\boldsymbol{\lambda} > 0$ if $\hat{X}$ and $(\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{p}})$ are optimal solutions of $LP(\boldsymbol{\lambda})$ and $DLP(\boldsymbol{\lambda})$ respectively, while X' and $(\boldsymbol{\alpha}', \boldsymbol{p}')$ are feasible in $LP(\boldsymbol{\lambda})$ and $DLP(\boldsymbol{\lambda})$ then, such that,*

$$
\begin{aligned}
&\forall i: \quad \lambda_i \text{delay}_i(\hat{X}) = \textstyle\sum_k r_{ik}\hat{\alpha}_{ik} - \text{pay}_i(\hat{X}, \hat{\boldsymbol{p}}) \\
&\forall i: \quad \lambda_i \text{delay}_i(X') \geq \textstyle\sum_k r_{ik}\alpha'_{ik} - \text{pay}_i(X', \boldsymbol{p}').
\end{aligned}
$$

Proof. For feasible solutions we have $\forall (i,k),\ \alpha'_{ik} \geq 0$ and $\sum_j a_{ijk} x'_{ij} \geq r_{ik}$. Multiplying the

two and summing over k for each i gives, $\sum_k \alpha'_{ik} \sum_j a_{ijk} x'_{ij} \geq \sum_k \alpha'_{ik} r_{ik},\ \forall i$. Another pair of constraints are $\forall (i,j),\ x'_{ij} \geq 0$ and $\lambda_i d_{ij} \geq \sum_k a_{ijk} \alpha'_{ik} - p'_j$. Multiplying the two gives,

$$\begin{aligned}
\forall i:\ \lambda_i \text{delay}(X') &= \lambda_i \textstyle\sum_j x'_{ij} d_{ij} \\
&\geq \textstyle\sum_j x'_{ij} (\sum_k a_{ijk} \alpha'_{ik}) - \sum_j p'_j x'_{ij} \\
&= \textstyle\sum_k \alpha'_{ik} \sum_j a_{ijk} x'_{ij} - \text{pay}_i(X', \boldsymbol{p}') \\
&\geq \textstyle\sum_k \alpha'_{ik} r_{ik} - \text{pay}_i(X', \boldsymbol{p}').
\end{aligned}$$

This gives the second part. Since optimal solutions satisfy complementary slackness, all inequalities satisfy with equality in the above for $(\hat{X}, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{p}})$ and we get the first part. □

Recall notation $[n] = \{1, \ldots, n\}$ for any positive integer n, and Definition 2 of an allocation X being jointly optimal for a subset of agents S.

Lemma 1. *Given $\boldsymbol{\lambda}$, partition agents by equality of λ_i into sets $S_1, \ldots, S_d$ such that $\lambda(S_1) < \cdots < \lambda(S_d)$.*

1. *At any optimal solution X of $LP(\boldsymbol{\lambda})$, the delay is minimized first for set S_d, then for $S_{(d-1)}$, and so on, finally for S_1. This is equivalent to X being jointly optimal for each $T_g, \forall g \in [d]$ where $T_g = \cup_{q=g}^{d} S_q$, and for any other optimal solution Y we have $\text{delay}_{S_g}(Y) = \text{delay}_{S_g}(X),\ \forall g \in [d]$.*

2. *Given two dual optimal solutions $(\boldsymbol{\alpha}, \boldsymbol{p})$ and $(\boldsymbol{\alpha}', \boldsymbol{p}')$, if the first part of the dual objective is the same at both solutions for some $g \in [d]$, i.e., $\sum_{i \in S_g, k} r_{ik} \alpha_{ik} = \sum_{i \in S_g, k} r_{ik} \alpha'_{ik}$, then for any optimal solution X of $LP(\boldsymbol{\lambda})$, $\text{pay}_{S_g}(X, \boldsymbol{p}) = \text{pay}_{S_g}(X, \boldsymbol{p}')$.*

3. *Given two optimal solutions X and X' of $LP(\boldsymbol{\lambda})$, and an optimal solution $(\boldsymbol{\alpha}, \boldsymbol{p})$ of $DLP(\boldsymbol{\lambda})$, if for any subset $S \subseteq S_g$ for $g \in [d]$, $\text{delay}_S(X) \leq \text{delay}_S(X')$, then $\text{pay}_S(X, \boldsymbol{p}) \geq \text{pay}_S(X', \boldsymbol{p})$. The former is strict iff the latter is strict too.*

Proof. Note that, $T_1 = A$, $T_d = S_d$, and $S_g = T_g \setminus T_{g+1},\ \forall g \in [d-1]$. For the first part, let us rewrite the objective function of $LP(\boldsymbol{\lambda})$ as

$$\sum_{g=2}^{d}(\lambda(S_g) - \lambda(S_{g-1})) \sum_{i\in T_g, j} d_{ij}x_{ij} + \lambda(S_1) \sum_{i\in T_1, j} d_{ij}x_{ij}$$

Since $\mathcal{M}$ satisfies extensibility, we can construct a minimum delay allocation X^* where S_d gets the best, then next best to S_{d-1}, and so on to finally S_1. In other words, X^* is jointly optimal for $T_g,\ \forall g \leq d$. Let X' be an arbitrary optimal solution of $LP(\boldsymbol{\lambda})$, not constructed as X^*. Then,

$$\forall 1 \leq g \leq d,\ \sum_{i\in T_g, j} d_{ij}x^*_{ij} \leq \sum_{i\in T_g, j} d_{ij}x'_{ij}.$$

To the contrary suppose at least one is strict inequality. Then, since $\lambda(S_d) > \lambda(S_{d-1}) > \cdots > \lambda(S_1) > 0$ we have

$$\begin{aligned}&\lambda(S_1)\textstyle\sum_{i\in T_1, j} d_{ij}x^*_{ij} + \sum_{g=2}^{d}(\lambda(S_g) - \lambda(S_{g-1}))\sum_{i\in T_g, j} d_{ij}x^*_{ij} <\\ &\lambda(S_1)\textstyle\sum_{i\in T_1, j} d_{ij}x'_{ij} + \sum_{g=2}^{d}(\lambda(S_g) - \lambda(S_{g-1}))\sum_{i\in T_g, j} d_{ij}x'_{ij}.\end{aligned}$$

A contradiction to X' being optimal solution of $LP(\boldsymbol{\lambda})$. Since any minimum delay allocation for a subset gives the same total delay, the first part follows.

The second part essentially follows by applying Lemma 20 twice. For optimal pair X and $(\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{p}})$, and pair X and $(\boldsymbol{\alpha}', \boldsymbol{p}')$ we get, $\forall g$

$$\lambda(S_g)\text{delay}_{S_g}(X) = \sum_{i\in S_g, k} r_{ik}\hat{\alpha}_{ik} - \text{pay}_{S_g}(X, \hat{\boldsymbol{p}}), \quad \text{and} \quad \lambda(S_g)\text{delay}_{S_g}(X) = \sum_{i\in S_g, k} r_{ik}\alpha'_{ik} - \text{pay}_{S_g}(X, \boldsymbol{p}').$$

implying, $\text{pay}_{S_g}(X, \hat{\boldsymbol{p}}) = \text{pay}_{S_g}(X, \boldsymbol{p}')$ if and only if $\sum_{i\in S_g, k} r_{ik}\hat{\alpha}_{ik} = \sum_{i\in S_g, k} r_{ik}\alpha'_{ik}$.

For the third part, applying Lemma 20 to each agent $i \in S$ and then taking the sum gives $\lambda(S_g)\text{delay}_S(\hat{X}) = \sum_{i\in S, k} r_{ik}\alpha_{ik} - \text{pay}_S(\hat{X}, \boldsymbol{p})$ and $\lambda(S_g)\text{delay}_S(X') = \sum_{i\in S, k} r_{ik}\alpha_{ik} - \text{pay}_S(X', \boldsymbol{p})$.

Combining the two, we get $\lambda(S_g)(\text{delay}_S(\hat{X}) - \text{delay}_S(X')) = \text{pay}_S(X', \boldsymbol{p}) - \text{pay}_S(\hat{X}, \boldsymbol{p})$, thereby the lemma follows. □

Given that the BB (budget balance) and SC (subset condition) of Definition 4 are satisfied at the current values of $\boldsymbol{\lambda}$ and $\boldsymbol{p}$, next we show the existence of a primal-dual solution where no agent spends more than her budget, and an agent who does not spend her entire budget gets an absolute best bundle, *i.e.,* an optimal bundle at zero prices. Recall the definition of proper pair $(\boldsymbol{\lambda}, \boldsymbol{p})$ (Definition 5).

Lemma 2. *If a pair $(\boldsymbol{\lambda}^*, \boldsymbol{p}^*)$ is proper for $\boldsymbol{\lambda}^* > 0$ then there exists an optimal solution X^* to the primal $LP(\boldsymbol{\lambda}^*)$ such that* $\text{pay}_i(X^*, \boldsymbol{p}^*) \leq m_i \; \forall i \in A$*, and for every agent i either* $\text{pay}_i(X^*, \boldsymbol{p}^*) = m_i$ *or we have $\boldsymbol{x}_i \in OPT_i(\mathbf{0}), \; \forall X \in LP(\boldsymbol{\lambda}^*)$.*

Proof. Let $S_1, \ldots, S_d$ be the partition of agent set A by equality of λ_i^*. If there is a set S_g for $g \in [d]$ such that $\text{pay}_{S_g}(X, \boldsymbol{p}^*) < m(S_g)$ for some $X \in LP(\boldsymbol{\lambda}^*)$, then from the BB condition we know that for every solution X of $LP(\boldsymbol{\lambda})$ we have $\text{pay}_{S_g}(X, \boldsymbol{p}) \leq m(S_g)$ and $\forall i \in S_g, \; \boldsymbol{x}_i \in OPT_i(\mathbf{0})$. Therefore, the delay of every agent in S_g remains the same at all optimal solutions of $LP(\boldsymbol{\lambda}^*)$. Therefore, using Lemma 1, we get that payment $\text{pay}_i(X, \boldsymbol{p}^*)$ of every agent i in S_g remains unchanged at all $X \in LP(\boldsymbol{\lambda}^*)$. In otherwords their essential spending at $(\boldsymbol{\lambda}^*, \boldsymbol{p}^*)$ is $\text{pay}_i(X, \boldsymbol{p}^*)$. Therefore, it is wlog to reset $m_i = \text{pay}_i(X, \boldsymbol{p}^*), \; \forall i \in S_g$ for any $X \in LP(\boldsymbol{\lambda}^*)$. Note that even after this reset both BB and SC will be satisfied for S_g at $(\boldsymbol{\lambda}^*, \boldsymbol{p}^*)$.

Repeate the above process for all such sets with $\text{pay}_{S_q}(X, \boldsymbol{p}^*) < m(S_q)$ for some $X \in LP(\boldsymbol{\lambda}^*)$. In the rest we show there exist an allocation such that $\text{pay}_i(X^*, \boldsymbol{p}^*) = m_i, \forall i$ where m_i is the modified budget.

Let $\tau_i(X) = m_i - \text{pay}_i(X, \boldsymbol{p}^*)$. Without loss of generality suppose agents are ordered such that $\tau_1(X) \geq \tau_2(X) \geq \cdots \geq \tau_n(X)$. Define $T_k(X) = \sum_{1 \leq i \leq k} \tau_i(X)$. Let's define the following potential function for every allocation X. The potential function is $f(X) = \sum_k T_k(X)$.

Let X^* be an optimal solution of $LP(\boldsymbol{\lambda}^*)$ that minimizes f. Order the agents such that

$\tau_1(X^*) \geq \tau_2(X^*) \geq \cdots \geq \tau_n(X^*)$. Note that $T_n(X^*) = 0$ (condition BB and new modified budgets). Therefore, $T_i(X^*) \geq 0\ \forall i$ since τ_is are in decreasing order. Therefore, $f(X^*) \geq 0$. If $f(X^*) = 0$ then it must be the case that $T_i(X^*) = 0,\ \forall i \in [n]$ and $\tau_i(X^*) = 0, \forall i \in [n]$. This gives $m_i - \text{pay}_i(X^*, \boldsymbol{p}^*) = 0,\ \forall i$ as we desired.

To the contrary suppose $f(X^*) > 0$. Let $\hat{X}$ be an optimal allocation of $LP(\boldsymbol{\lambda}^*)$ where delay of agent 1 is minimum, then of agent 2, and so on, finally of agent n.

Claim 5. $\sum_{i \leq r} \tau_i(\hat{X}) \leq 0,\ \forall r \in [n]$.

Proof. Fix an $r \in [n]$ and define $S = \{1, \ldots, r\}$ and $\bar{S} = \{r+1, \ldots, n\}$. Since the total delay of all the agents is same at both X^* and $\hat{X}$, their total payment is also same (first and third part of Lemma 1). Therefore it suffices to show $\sum_{i \in \bar{S}} \tau_i(\hat{X}) \geq 0$ because $\sum_{1 \leq i \leq n} \tau_i(\hat{X}) = 0$. Let's define $L_g = \bar{S} \cap S_g$. Note that $\hat{X}$ is an optimal allocation in which delay of $\bar{S}$ is maximized. We will show that in an optimal allocation if delay of $\bar{S}$ is maximized then delay of L_g is maximized for all g and so $m(L_g) \geq \text{pay}_{L_g}(\hat{X}, \boldsymbol{p}^*)$ (SC condition). Therefore, $m(\bar{S}) \geq \text{pay}_{\bar{S}}(\hat{X}, \boldsymbol{p}^*)$. That completes the proof.

In the following we show that if the delay of $\bar{S}$ is maximized then delay of L_g maximized for all g in an optimal allocation. Consider an optimal allocation X' of $LP(\boldsymbol{\lambda}^*)$ which is constructed by first optimizing for $S_d \backslash L_d$, then L_d, then $S_{d-1} \backslash L_{d-1}$ then L_{d-1} and so on. This is a valid construction due to the extensibility property. We claim that X' is an optimal allocation which maximizes delay of $L_g, \forall g$ individually. Total delay of $S_g\ \forall g$ is the same for all optimal allocations (Lemma 1) and delay of $(\cup_{q=g+1}^{d} S_q) \cup (S_g \backslash L_g)$ is minimized in X'. Therefore, delay of $S_g \backslash L_g$ is minimized in X' and so delay of L_g is maximized since sum of delays of $S_g \backslash L_g$ and L_g is constant among all optimal allocations. □

Using the above claim we get $\exists \hat{r}$ such that $\sum_{i \leq \hat{r}} \tau_i(\hat{X}) < 0$ because otherwise $\tau_i(\hat{X}) = 0, \forall i$ and $f(\hat{X}) = 0$. Therefore,

$$\sum_{r \leq n} \sum_{i \leq r} \tau_i(\hat{X}) < 0. \tag{2.18}$$

Let's define $X(\delta) = (1-\delta)X^* + \delta\hat{X}$. Since optimal solutions of an LP forms a convex set, $X(\delta)$ is an optimal solution of $LP(\boldsymbol{\lambda}^*)$ for all $\delta \in [0\ 1]$. For every pair i and j such that $i < j$ and $\tau_i(X^*) = \tau_j(X^*)$, we assume $\frac{\partial \tau_i(X(\delta))}{\partial \delta} > \frac{\partial \tau_j(X(\delta))}{\partial \delta}$. Note that the assumption is without loss of generality. Therefore, there exists δ small enough such that the order of τ_i's is the same for $X(\delta)$ as at X^*. Considering that small δ, we have the following

$$f(X(\delta)) = \sum_r \sum_{i \le r} \tau_i(X(\delta))$$

$$= \sum_r \sum_{i \le r} (\tau_i(X^*) - \delta(\tau_i(X^*) - \tau_i(\hat{X})))$$

$$= f(X^*) - \delta(f(X^*) - \sum_{r \le n} \sum_{i \le r} \tau_i(\hat{X}))$$

$$= (1-\delta) f(X^*) + \delta(\sum_{r \le n} \sum_{i \le r} \tau_i(\hat{X})))$$

$$< f(X^*) \qquad \text{(Using (2.18))}$$

Therefore, we get $f(X(\delta)) < f(X^*)$ which is a contradiction to X^* being optimal solution where f is minimized. □

While searching for the next segment during the algorithm, we would like to fix total payment of the segments that already have been created. Note that unlike scheduling on a single machine (Section 2.2), in this general setting we are not able to fix the allocation of the agents on the previous segments. In fact the allocation will heavily depend on what segments are created later on. However from Lemma 1 we know that the total delay of previous segments will remain unchanged. If delay is fixed then payments can be controlled using the last part of Lemma 1 if prices of goods they buy also remain unchanged. Therefore the only way to ensure that total payments are fixed seems to be, fixing prices of goods they are buying, and also hold their dual variables α_{ik}s. Next lemma shows that this is indeed possible. In a number of proofs that follows we will use the

following version of Farkas' lemma.

Lemma 21 (Farkas' lemma [98])**.** *Given a matrix A, if the system $A\boldsymbol{z} = b$ and $\boldsymbol{z} \geq 0$ is infeasible then there exists vector $\boldsymbol{y}$ such that $\boldsymbol{y}^T A \geq 0$ and $\boldsymbol{y}^T b < 0$.*

Recall notation $\mathbf{1}_S \in \{0,1\}^{|A|}$ for a subset $S \subseteq A$ denoting indicator vector of set S.

Lemma 3. *Given a $\boldsymbol{\lambda}$, partition agents into $S_1, \ldots, S_d$ by equality of λ_i, where $\lambda(S_1) < \cdots < \lambda(S_d)$. For $R \subseteq S_d$ consider primal optimal $\hat{X}$ that is jointly optimal for R, and let $(\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{p}})$ be a dual optimal. Consider for some $a > 0$, the vector $\boldsymbol{\lambda}' = \boldsymbol{\lambda} + a\mathbf{1}_R$. Then $\hat{X}$ is optimal in $LP(\boldsymbol{\lambda}')$ and there exists an optimal solution $(\boldsymbol{\alpha}', \boldsymbol{p}')$ of $DLP(\boldsymbol{\lambda}')$ such that,*

$$\begin{aligned} &\forall j: && p'_j \geq \hat{p}_j \quad \textit{and} \quad \textstyle\sum_{i \notin R} \hat{x}_{ij} > 0 \Rightarrow p'_j = \hat{p}_j \\ &\forall i \notin R,\ \forall k, && \alpha'_{ik} = \hat{\alpha}_{ik}\ . \end{aligned}$$

Proof. By construction of $\hat{X}$ and $\boldsymbol{\lambda}'$, $\hat{X}$ is feasible in $LP(\boldsymbol{\lambda}')$, it is jointly optimal for R as well as for all $\cup_{q=g}^{d} S_q,\ g \in [d]$, and $\boldsymbol{\lambda}'(S_1) < \cdots < \boldsymbol{\lambda}'(S_d \setminus R) < \boldsymbol{\lambda}'(R)$. Therefore, using the first property of Lemma 1 it follows that $\hat{X}$ is an optimal solution of $LP(\boldsymbol{\lambda}')$.

Pair $(\boldsymbol{p}', \boldsymbol{\alpha}')$ is an optimal for $DLP(\boldsymbol{\lambda}')$ if and only if it satisfies the complementary slackness conditions with $\hat{X}$. Using this the second part of the lemma would follow if $\boldsymbol{p}'$ is set to $\hat{\boldsymbol{p}} + \boldsymbol{\delta}$ where $\boldsymbol{\delta}$ and $\boldsymbol{\alpha}'$ satisfy the following.

$$\boldsymbol{\delta} \geq 0$$

$$\forall j \quad s.t. \quad \text{either } \textstyle\sum_{i \notin R} \hat{x}_{ij} > 0 \text{ or } \sum_i \hat{x}_{ij} < c_j: \quad \delta_j = 0. \tag{2.19}$$

$$\forall i \in R, \forall j \quad s.t. \quad \hat{x}_{ij} > 0: \quad \lambda'_i d_{ij} = \textstyle\sum_k a_{ijk} \alpha'_{ik} - \hat{p}_j - \delta_j. \tag{2.20}$$

$$\forall i \in R, \forall j \quad s.t. \quad \hat{x}_{ij} = 0: \quad \lambda'_i d_{ij} \geq \textstyle\sum_k a_{ijk} \alpha'_{ik} - \hat{p}_j - \delta_j. \tag{2.21}$$

The proof is by contradiction. Suppose the system is infeasible. We will show a contradiction using Farkas' lemma (Lemma 21). To convert (2.21) to equality we add slack variable γ_{ij}. In addition, we remove all δ_j that are set to zero in (2.19) from (2.20) and (2.21), and remove (2.19) itself from the system. Let T denote the set of goods j such that $\sum_{i \notin R} \hat{x}_{ij} > 0$ or $\sum_i \hat{x}_{ij} < c_j$ and $\bar{T}$ denote the set of good not in T. The remaining system can be written as follows in $A\boldsymbol{z} = \boldsymbol{b}$ form in variables δ_js and γ_{ij}s:

$$\begin{aligned}
&\forall i \in R, \forall j \quad s.t. \quad \hat{x}_{ij} > 0 \;\&\; (j \in \bar{T}): \quad \lambda'_i d_{ij} + \hat{p}_j = \textstyle\sum_k a_{ijk}\alpha'_{ik} - \delta_j \\
&\forall i \in R, \forall j \quad s.t. \quad \hat{x}_{ij} = 0 \;\&\; (j \in \bar{T}): \quad \lambda'_i d_{ij} + \hat{p}_j = \textstyle\sum_k a_{ijk}\alpha'_{ik} - \delta_j + \gamma_{ij} \\
&\forall i \in R, \forall j \quad s.t. \quad \hat{x}_{ij} > 0 \;\&\; (j \in T): \quad \lambda'_i d_{ij} + \hat{p}_j = \textstyle\sum_k a_{ijk}\alpha'_{ik} \\
&\forall i \in R, \forall j \quad s.t. \quad \hat{x}_{ij} = 0 \;\&\; (j \in T): \quad \lambda'_i d_{ij} + \hat{p}_j = \textstyle\sum_k a_{ijk}\alpha'_{ik} + \gamma_{ij}.
\end{aligned} \tag{2.22}$$

Due to Farkas' lemma if the above system is infeasible then there exists $\boldsymbol{y}$ such that $\boldsymbol{y}^T A = 0$ and $\boldsymbol{y}^T \boldsymbol{b} < 0$. That is for variables $y_{ij},\ \forall i \in R,\ \forall j$:

$$\boldsymbol{y}^T A \geq 0 \Rightarrow \begin{cases}
\forall i \in R, \forall k & : \sum_j a_{ijk} y_{ij} \geq 0 \\
\forall i \in R, \forall j \quad s.t. \quad x_{ij} = 0 & : y_{ij} \geq 0 \\
\forall j \in \bar{T} & : \sum_{i \in R} y_{ij} \leq 0.
\end{cases} \tag{2.23}$$

$$y^T b < 0 \Rightarrow \ \textstyle\sum_{i \in R, j} \lambda'_i d_{ij} y_{ij} + \sum_{i \in R, j} \hat{p}_j y_{ij} < 0. \tag{2.24}$$

Let's consider two cases.

Case 1. $\sum_{i \in R, j} d_{ij} y_{ij} \geq 0$. Since $\forall i \in R,\ \lambda(S_d) = \lambda_i < \lambda'_i = \lambda(S_d) + \tau$ and $y^T b < 0$ we get

$$\lambda(S_d) \textstyle\sum_{i \in R, j} d_{ij} y_{ij} + \sum_{i \in R, j} \hat{p}_j y_{ij} < 0. \tag{2.25}$$

On the other hand, note that

$$\begin{aligned}\lambda(S_d)\sum_{i\in R,j} d_{ij}y_{ij} + \sum_{i\in R,j}\hat{p}_j y_{ij} &= \sum_{i\in R,j} y_{ij}(\lambda_i d_{ij} + \hat{p}_j) \\ &\geq \sum_{i\in R,j} y_{ij}(\sum_k a_{ijk}\hat{\alpha}_{ik}) \qquad (\because\ (\hat{p},\hat{\alpha}) \text{ is a solution for } DLP(\lambda)) \\ &= \sum_{i\in R}\sum_k \hat{\alpha}_{ik}\sum_j y_{ij}a_{ijk} \geq 0 \quad \text{(Using (2.23))}.\end{aligned}$$

That is a contradiction.

Case 2. $\sum_{i\in R,j} d_{ij}y_{ij} < 0$. We will show that $\hat{X}$ does not give min-cost allocation to agents of R. Consider $x_{ij} = \hat{x}_{ij} + \epsilon y_{ij},\ \forall i \in R,\ \forall j$ for a small amount $\epsilon > 0$ and $x_{ij} = \hat{x}_{ij},\ \forall i \notin R,\ \forall j$. Using (2.23) it follows that X is feasible in $LP(\boldsymbol{\lambda})$. In addition,

$$\sum_i \lambda_i \sum_j d_{ij}(x_{ij} - \hat{x}_{ij}) = \lambda(S_g)\sum_{i\in R,\ j} d_{ij}(\epsilon y_{ij}) < 0$$

This contradicts $\hat{X}$ being optimal solution of $LP(\boldsymbol{\lambda})$. □

While creating next segment we need to maintain BB and SC conditions for all the previous segments. Lemma 3 ensures that prices of goods bought by agents in previous segments is fixed. If we also manage to ensure that total delay remains unchanged for previous segments, then we will be able to leverage properties from Lemma 1 to show BB and SC do remain satisfied for previous segments. The next lemma establishes exactly this.

Lemma 22. *Given $\boldsymbol{\lambda} > 0$ let the partition of agents by equality of λ_i be $S_1, \ldots, S_{k-1}, A'$ such that $\lambda(S_1) < \cdots < \lambda(S_{k-1}) < \lambda(A')$. For an $S_k \subset A'$, let $\boldsymbol{\lambda}' \geq \boldsymbol{\lambda}$ be such that the induced partition is $S_1, \ldots, S_k, A' \setminus S_k$ and $\lambda(S_1) < \cdots < \lambda(S_k) < \lambda(A' \setminus S_k)$. Then for any group $g < k$, we have* $\text{delay}_{S_g}(X) = \text{delay}_{S_g}(X')$ *where X and X' are solutions of $LP(\boldsymbol{\lambda})$ and $LP(\boldsymbol{\lambda}')$ respectively. And for any subset $T \subset S_g$,*

$$\max_{X \text{ optimal of } LP(\boldsymbol{\lambda})} \text{delay}_T(X) = \max_{X \text{ optimal of } LP(\boldsymbol{\lambda}')} \text{delay}_T(X)$$

Proof. An optimal solution of both $LP(\boldsymbol{\lambda})$ and $LP(\boldsymbol{\lambda}')$ first minimizes total delay of A' then of S_{k-1} and so on finally of S_1 (Lemma 1). Within A', $LP(\boldsymbol{\lambda}')$ may first minimize for $A' \setminus S_k$ and then of S_k. Thus, the optimal solution set may shrink as we go from $\boldsymbol{\lambda}$ to $\boldsymbol{\lambda}'$. However, due to extensibility, if X and X' are optimal solution of $LP(\boldsymbol{\lambda})$ and $LP(\boldsymbol{\lambda}')$ then, $\text{delay}_{S_g}(X) = \text{delay}_{S_g}(X')$. Further, by Lemma 1 the optimal solution of both $LP(\boldsymbol{\lambda})$ and $LP(\boldsymbol{\lambda}')$ where delay_T is maximized essentially minimizes total delay of $A' \cup_{q=g+1}^{k-1} S_q \cup (S_g \setminus T)$ and then of T, then of $\cup_{q=1}^{g-1} S_q$. By extensibility condition delay_T at any such allocation remains the same. □

The next lemma shows that if before we start our search for next segment, already created segments $S_1, \ldots, S_{k-1}$ satisfies BB and SC w.r.t. $(\boldsymbol{\lambda}^{cur}, \boldsymbol{p}^{cur})$, and the remaining agents satisfy SC, then for the value of a where minimum of f_a is zero, the minimizer gives the next segment without ruining the former. Recall that, for a given $a \geq 0$ and $\epsilon > 0, \boldsymbol{\lambda}^{new} = \boldsymbol{\lambda}^a + \epsilon \mathbf{1}_{A' \setminus S_k}$, and the prices $\boldsymbol{p}^{new}$ are valid and optimal for $DLP(\boldsymbol{\lambda}^{new})$, where $\mathbf{1}_{A' \setminus S_k}$ is an indicator vector of set $A' \setminus S_k$, and $\boldsymbol{\lambda}^a$ as defined in (2.4). We will also use notation $\boldsymbol{p}^a$ to denote prices at the valid optimal solution of $DLP(\boldsymbol{\lambda}^a)$ (See (2.5)), i.e., in the sense guaranteed by Lemma 3.

Lemma 4. *Suppose that for some* $a \geq 0$*,*

$$S_k \in \arg\min_{S \subseteq A', S \neq \emptyset} \{f_{\boldsymbol{\lambda}^a, \boldsymbol{p}^a}(S)\}.$$

Further, suppose that $f_{\boldsymbol{\lambda}^a, \boldsymbol{p}^a}(S_k) = 0$*, and let* S_k *be a maximal such set. Then there exists a rational number* $\epsilon > 0$ *of polynomial-size such that, w.r.t.* $(\boldsymbol{\lambda}^{new}, \boldsymbol{p}^{new})$ *as defined above,* $S_1, \ldots, S_k$ *satisfy both* BB *and* SC*, and* $A' \setminus S_k$ *satisfies* SC*.*

Proof. First part of Lemma 1 implies that every optimal solution of $LP(\boldsymbol{\lambda}^{new})$ minimizes delay of sets $A' \setminus S_k$, S_k, …, S_1 in that sequence, while optimal of $LP(\boldsymbol{\lambda}^{cur})$ minimizes delay of sets in sequence A', S_{k-1},…, S_1. Therefore clearly the set of optimal solutions of $LP(\boldsymbol{\lambda}^{new})$ is a subset of the optimal solutions of $LP(\boldsymbol{\lambda}^{cur})$. This together with the fact that at $(\boldsymbol{\lambda}^{cur}, \boldsymbol{p}^{cur})$ BB and SC

are satisfied for each $g \leq k-1,\ S_g$, for any optimal X' of $LP(\boldsymbol{\lambda}^{new})$, we have $\text{pay}_{S_g}(X', \boldsymbol{p}^{cur}) = m(S_g)$. Let $\boldsymbol{\alpha}^{cur}$ be corresponding dual variable, then Lemma 20 gives $\lambda(S_g)\text{delay}_{S_g}(X') = \sum_{i \in S_g, k} r_{ik}\alpha^{cur}_{ik} - \text{pay}_{S_g}(X', \boldsymbol{p}^{cur})$

Since $\boldsymbol{p}^{new}$ is a valid solution of $DLP(\boldsymbol{\lambda}^{new})$, at corresponding valid $(\boldsymbol{\alpha}^{new}, \boldsymbol{p}^{new})$ value of ${\boldsymbol{\alpha}^{new}}_i$ for each $i \notin A'$ is same as $\boldsymbol{\alpha}^{cur}_i$. Using Lemma 20 we get $\lambda(S_g)\text{delay}_{S_g}(X') = \sum_{i \in S_g, k} r_{ik}{\alpha^{new}}_{ik} - \text{pay}_{S_g}(X', \boldsymbol{p}^{new}) = \sum_{i \in S_g, k} r_{ik}{\alpha^{cur}}_{ik} - \text{pay}_{S_g}(X', \boldsymbol{p}^{new})$. This together with the above equality gives $\text{pay}_{S_g}(X', \boldsymbol{p}^{new}) = \text{pay}_{S_g}(X', \boldsymbol{p}^{cur}) = m(S)$. Hence BB is satisfied by $S_1,\ldots,S_{k-1}$ at $(\boldsymbol{\lambda}^{new}, \boldsymbol{p}^{new})$. By the same reasoning, and using Lemma 22 and third part of Lemma 1 we get that they also satisfy SC.

Note that $\boldsymbol{\lambda}^{new}$ is $\boldsymbol{\lambda}^a$ with ϵ added to λ_is of agents in $A' \setminus S_k$. And $\boldsymbol{p}^{new}$ is a valid optimal of $DLP(\boldsymbol{\lambda}^{new})$ obtained starting from $\boldsymbol{p}^a$ where α_{ik}s of agents not in $A' \setminus S_k$, and prices of goods "bought by them" are held fixed (Lemma 3). Since function $f_{\boldsymbol{\lambda},\boldsymbol{p}}$ keeps track of surplus budget, and S_k is the minimizer of $f_{\boldsymbol{\lambda}^a,\boldsymbol{p}^a}$ where surplus budget of S_k is zero at $\boldsymbol{p}^a$ in addition, it follows that for every subset of S_k the surplus budget is non-negative. Thus we get SC for S_k at $(\boldsymbol{\lambda}^{new}, \boldsymbol{p}^{new})$ using Lemma 22. For BB note that we have $\boldsymbol{\lambda}^{new}(S_k) < \boldsymbol{\lambda}^{new}(A' \setminus S_k)$. Hence, due to first part of Lemma 1, optimal allocations X of $LP(\boldsymbol{\lambda}^{new})$ will give first to $A' \setminus S_k$ minimum delay, and then next minimum to S_k. This is exactly same as maximizing $\text{delay}_{S_k}(X)$ among optimal of $LP(\boldsymbol{\lambda}^a)$ (where $\boldsymbol{\lambda}^a(S^*) = \boldsymbol{\lambda}^a(A' \setminus S^*) > \boldsymbol{\lambda}^a(S_g),\ \forall g \leq k-1$). Due to the fact that $f_{\boldsymbol{\lambda}^a,\boldsymbol{p}^a}(S_k) = 0$, at such an allocation we also have $m(S_k) = \text{pay}_{S_k}(X, \boldsymbol{p}^a)$. This will be same as $\text{pay}_{S_k}(X, \boldsymbol{p}^{new})$ due to construction of $\boldsymbol{p}^{new}$ from $\boldsymbol{p}^a$ in Lemma 3. Since at every such allocation $\text{delay}_{S_k}(X)$ remains the same, $\text{pay}_{S_k}(X, \boldsymbol{p}^{new})$ remains the same (Lemma 22).

For the second part, namely $A' \setminus S_k$ satisfies SC w.r.t. $(\boldsymbol{\lambda}^{new}, \boldsymbol{p}^{new})$, it suffices to show existence of $\epsilon > 0$ such that $f_{\boldsymbol{\lambda}^{new},\boldsymbol{p}^{new}}(T) \geq 0,\ \forall T \subset A' \setminus S_k$. We will show this in Lemma 27 below. □

Above lemma implies that if the minimizer of f_a gives zero value, then it forms the next segment. One crucial task therefore is to find a minimizer of f_a efficiently. Next lemma and (2.7) show that f_a is a submodular function, implying its minimizer can be found in polynomial time.

Lemma 5. *Given $a \geq 0$, function f_a is submodular over set A'.*

Proof. For ease of notation let us use f to denote function f_a, $\boldsymbol{\lambda} = \boldsymbol{\lambda}^a$, $\boldsymbol{p} = \boldsymbol{p}^a$, and α be the dual vector that forms *valid* solution of $DLP(\boldsymbol{\lambda}^a)$ together with $\boldsymbol{p}^a$. Recall that agent set A is partitioned by equality of λ_i^a into sets $S_1, \ldots, S_{(k-1)}, A'$ such that $\boldsymbol{\lambda}^a(S_1) < \cdots < \boldsymbol{\lambda}^a(S_{(k-1)}) < \boldsymbol{\lambda}^a(A')$.

Let $S \subset T \subset A'$ and $a \notin T$. Define $S' = S \cup \{a\}$ and $T' = T \cup \{a\}$. It suffices to show the following

$$f(S') - f(S) \geq f(T') - f(T).$$

Let's recall the following two complementary slackness conditions.

$$\begin{array}{lll} \forall i \in A,\ \forall k \in \mathcal{C}: & \sum_{j \in G} a_{ijk} x_{ij} \geq r_{ik} & \perp \quad \alpha_{ik} \geq 0 \\ \forall i \in A, \forall j \in G: & \lambda_i d_{ij} \geq \sum_k a_{ijk} \alpha_{ik} - p_j & \perp \quad x_{ij} \geq 0. \end{array}$$

Using these it is easy to get the following, where λ^* is the λ_i of agents in A' which we know is the same.

$$\lambda^* \text{delay}_S(X) = \text{pay}_S(X, \boldsymbol{p}) + \sum_{i \in S, k} \alpha_{ik} r_{ik}. \tag{2.26}$$

For set $S \subseteq A'$ let us denote its complement within A' by $\bar{S} = A' \setminus S$. From the first part of Lemma 1 we know that optimal solution of $LP(\boldsymbol{\lambda})$ will first minimize delay of set A' since it has highest λ value. Note that, $\bar{S} = \bar{S'} \cup \{a\}$. If X^S is an optimal solution of $LP(\boldsymbol{\lambda})$ where delay of S is maximized then by extensibility it can constructed as follows: within minimization for set A' first minimizing delay for $\bar{S'}$, then for a, and lastly for S. Then we have:

$$\begin{aligned} f(S') - f(S) &= m_a - \text{pay}_{S'}(X^S, \boldsymbol{p}) + \text{pay}_S(X^S, \boldsymbol{p}) \\ &= m_a - \textstyle\sum_{i \in S', k} \alpha_{ik} r_{ik} + \lambda^* \text{delay}_{S'}(X^S) + \sum_{i \in S, k} \alpha_{ik} r_{ik} - \lambda^* \text{delay}_S(X^S) && \text{(Using (2.26))} \\ &= m_a - \textstyle\sum_k \hat{\alpha}_{ak} r_{ak} + \lambda^* (\text{delay}_{S'}(X^S) - \text{delay}_S(X^S)) \\ &= m_a - \textstyle\sum_k \hat{\alpha}_{ak} r_{ak} + \lambda^* \text{delay}_a(X^S) && (X^S \text{ construction}). \end{aligned}$$

Similarly we get,

$$f(T') - f(T) = m_a - \sum_k \hat{\alpha}_{ak} r_{ak} + \lambda^* \text{delay}_a(X^T)$$

where X^T is an optimal solution where delay of T is maximized, which can be constructed by first minimizing delay for $\bar{T}'$, next for a, and last for T. Recall that $S \subset T$ and so $\bar{T} \subset \bar{S}$. We constructed X^S and X^T by first optimizing for $\bar{S}'$ and $\bar{T}'$ and then adding a. Therefore, $\text{delay}_a(X^S) \leq \text{delay}_a(X^T)$ and so

$$f(S') - f(S) \geq f(T') - f(T).$$

□

The NextSeg subroutine does binary search on the value of a to find the one where minimizer of f_a gives zero. In next few lemmas we show why binary search is the right tool to find this critical value of a. Essentially we show Lemma 7 which has four parts and we will show them in separate lemmas.

Lemma 23. *If A' satisfies SC condition of Definition 4 at $(\boldsymbol{\lambda}^{cur}, \boldsymbol{p}^{cur})$ then $f_{\boldsymbol{\lambda}^{cur}, \boldsymbol{p}^{cur}}(T) \geq 0$, $\forall T \subset A'$. In other words $g(0) \geq 0$.*

Proof. Since A' satisfies SC w.r.t. $(\boldsymbol{\lambda}^{cur}, \boldsymbol{p}^{cur})$, by definition of SC condition (Definition 4), it follows that for any $T \subset A'$, $m(T) - \text{pay}_T(X^T, \boldsymbol{p}^{cur}) \geq 0$, where X^T is an optimal solution of $LP(\boldsymbol{\lambda}^{cur})$ where $\text{delay}_T(X)$ is maximized. Thus, $f_{\boldsymbol{\lambda}^{cur}, \boldsymbol{p}^{cur}}(T) = m(T) - \text{pay}_T(X^T, \boldsymbol{p}^{cur}) \geq 0$. □

Lemma 24. *For any $T \subset A'$, value $f_a(T)$ monotonically decreases with increase in a. Furthermore, if there exists $c > b$ such that $f_c(T) < f_b(T)$ then $f_a(T)$ strictly decreases as we go from $a = b$ to $a = c$.*

Proof. There is a unique valid price vector $\boldsymbol{p}^a$ and $\boldsymbol{p}^{a'}$ constituting optimal solution of $DLP(\boldsymbol{\lambda}^a)$ and $DLP(\boldsymbol{\lambda}^{a'})$. If we apply Lemma 3 for $\boldsymbol{\lambda} = \boldsymbol{\lambda}^a$ and $\boldsymbol{\lambda}' = \boldsymbol{\lambda}^{a'}$, then we get that $\boldsymbol{p}^a \leq \boldsymbol{p}^{a'}$.

Note that the partition of agents by equality of λ_i is the same at both $\boldsymbol{\lambda}^a$ and $\boldsymbol{\lambda}^{a'}$ and further their ordering by the value of $\lambda(S)$ is also same. Therefore, due to the first part of Lemma 1 every optimal solutions of $LP(\boldsymbol{\lambda}^a)$ are the same $LP(\boldsymbol{\lambda}^{a'})$. Hence, among them the ones minimizing $\text{delay}_T(X)$ for any given $T \subset A'$ are the same, say X^T is one of them. Note that delay_T remains the same at all such allocations.

$$f_a(T) = m(T) - \text{pay}_T(X^T, \boldsymbol{p}^a) = m(T) - \sum_{i \in T, j} x_{ij}^T p_j^a \geq m(T) - \sum_{i \in T, j} x_{ij}^T p_j^{a'} = f_{a'}(T).$$

Since solutions of linear programs change continuously with change in parameters, the first part follows.

For the second part, we know that solutions of $LP(\boldsymbol{\lambda}^a)$ is the same for all $b \leq a \leq c$, and therefore the ones where total delay of agents in T is maximized also remains the same. Let $\boldsymbol{p}^b$ and $\boldsymbol{p}^c$ be the valid prices at $\boldsymbol{\lambda}^b$ and $\boldsymbol{\lambda}^c$ respectively. And let $\boldsymbol{\alpha}^b$ and $\boldsymbol{\alpha}^c$ be corresponding $\boldsymbol{\alpha}$ vectors at the dual optimal. Then clearly, for any $X \in LP(\boldsymbol{\lambda}^b)$, both $(\boldsymbol{\alpha}^b, \boldsymbol{p}^b)$ at $\boldsymbol{\lambda} = \boldsymbol{\lambda}^b$ and $(\boldsymbol{\alpha}^c, \boldsymbol{p}^c)$ at $\boldsymbol{\lambda} = \boldsymbol{\lambda}^c$ satisfy complementary slackness conditions. Furthermore, for any convex combination $d = \tau b + (1-\tau)c$ where $\tau \in [0,1]$, we have $\boldsymbol{\lambda}^d = \tau\boldsymbol{\lambda}^b + (1-\tau)\boldsymbol{\lambda}^c$ and it is easy to see that $(\boldsymbol{\alpha}', \boldsymbol{p}') = \tau(\boldsymbol{\alpha}^b, \boldsymbol{p}^b) + (1-\tau)(\boldsymbol{\alpha}^c, \boldsymbol{p}^c)$ satisfies complementary slackness with X at $\boldsymbol{\lambda} = \boldsymbol{\lambda}^d$. And $\boldsymbol{p}'$ is valid, and the lemma follows. □

Lemma 25. *For any given $a' > 0$ with $g(a') > 0$, the following are equivalent:*

(a) $g(a) > 0$ *for all* $a \geq a'$.

(b) $\forall X \in LP(\boldsymbol{\lambda}^{a'})$ *and* $\forall i \in A'$ $\boldsymbol{x}_i \in OPT_i(\mathbf{0})$.

(c) $f_a(A') = f_{a'}(A')$ *and* $\boldsymbol{p}^a = \boldsymbol{p}^{a'}$ *for all* $a \geq a'$.

Proof. If $g(a') > 0$ then $f_{a'}(A') > 0$ as well. This implies $\lim_{a \to \infty} \boldsymbol{p}^a < \infty$. At valid solution of the dual α_{ik} for agents outside A' is held fixed, therefore $\sum_k r_{ik}\alpha_{ik}$ for all $i \notin A'$ is fixed at

all $DLP(\boldsymbol{\lambda}^a)$. Furthermore, complementary slackness condition for $i \notin A'$ ensures that for all of them $\lambda_i \sum_j d_{ij}x_{ij} = \sum_k r_{ik}\alpha_{ik} - \sum_j x_{ij}p_j$ at every solution of $LP(\boldsymbol{\lambda}^a)$ for all $a \geq 0$.

Let $(\boldsymbol{\alpha}^a, \boldsymbol{p}^a)$ be the valid solution of $DLP(\boldsymbol{\lambda}^a)$, then we have $(\boldsymbol{\lambda}(A') + a)\sum_{i\in A',j} d_{ij}x_{ij} = \sum_{i\in A',k} r_{ik}\alpha^a_{ik} - \sum_{i\in A',j} x_{ij}p^a_j$ at every optimal of $LP(\boldsymbol{\lambda}^a)$. This implies

$$\sum_{i\in A',j} d_{ij}x_{ij} = \sum_{i\in A',k} r_{ik}\alpha^a_{ik}/(\boldsymbol{\lambda}(A') + a) - \sum_{i\in A',j} x_{ij}p^a_j/(\boldsymbol{\lambda}(A') + a)$$

As a goes to ∞ second term vanishes since $\boldsymbol{p}^a < \infty$. And since $r_{ik} \geq 0,\ \forall(i,k)$, we get that $\lim_{a\to\infty} \alpha^a_{ik}/(\boldsymbol{\lambda}(A') + a)$ exists for all $i \in A', k$. Let it be β_{ik}. Then from the complementary slackness conditions, for any $X \in LP(\boldsymbol{\lambda}^{a'})$, $a' = (a+1)$ we get,

$$\begin{aligned}
&\forall i \in A', j: \quad x_{ij} > 0 \quad \Rightarrow \quad d_{ij} = \textstyle\sum_k a_{ijk}\beta_{ik} \\
&\forall i \in A', j: \quad x_{ij} = 0 \quad \Rightarrow \quad d_{ij} \geq \textstyle\sum_k a_{ijk}\beta_{ik} \\
&\forall i \in A', k: \quad \beta_{ik} > 0 \quad \Rightarrow \quad \alpha^{a'}_{ik} > 0 \Rightarrow \textstyle\sum_j a_{ijk}x_{ij} = r_{ik} \\
&\forall i \in A', k: \quad \beta_{ik} = 0 \quad \Rightarrow \quad \textstyle\sum_j a_{ijk}x_{ij} \geq r_{ik}
\end{aligned} \tag{2.27}$$

It is easy to check that the above exactly implies $\boldsymbol{x}_i \in OPT_i(\mathbf{0}),\ \forall i \in A'$. Thus, (b) follows from (a).

For $(b) \Rightarrow (c)$, let $(\beta_{ik})_k$ be a dual optimal of $OPT_i(\mathbf{0})$ for every $i \in A'$, and for the rest set β_{ik} to zero. Let $(\hat{\boldsymbol{\alpha}}, \boldsymbol{p}^{a'})$ be a valid solution of $DLP(\boldsymbol{\lambda}^{a'}$. We know that by construction β_{ik}'s satisfies all the constraints of (2.27) together with every $X \in LP(\boldsymbol{\lambda}^a)$ for all $a \geq a'$. Similarly $(\hat{\boldsymbol{\alpha}}, \boldsymbol{p}^{a'})$ as well satisfies all complementary slackness conditions with X between $LP(\boldsymbol{\lambda}^{a'}$ and $DLP(\boldsymbol{\lambda}^{a'})$. Putting these two together we get that $(\hat{\boldsymbol{\alpha}}, \boldsymbol{p}^{a'}) + \epsilon(\boldsymbol{\beta}, \mathbf{0})$ is a valid optimal solution of $DLP(\boldsymbol{\lambda}^{(a'+\epsilon)}$. Note that prices do not change.

$(c) \Rightarrow (a)$ follows from the fact that if $f_a(A')$ remains constant with increase in a then $pay_a(X, \boldsymbol{p}^a)$ is constant for all $X \in LP(\boldsymbol{\lambda}^a)$. This is possible only if no price is increasing. Since no price can decrease anyway due to Lemma 3, we essentially get that $f_a(S) = f_{a'}(S)$ for

all $S \subseteq A'$ and for all $a \geq a'$. This implies $g(a) = g(a') > 0,\ \forall a \geq a'$. $\square$

Lemma 26. *Given existence of $a > 0$ for set $S \subseteq A'$ such that $f_a(S) = 0$, such an a can be found by solving a feasibility LP of polynomial-size. Furthermore, if there exists $a_h \geq 0$ such that $g(a_h) \leq 0$, then $g(\Delta) \leq 0$ for $\Delta = (\sum_{i,j,k} |a_{ijk}| + \sum_{i,k} |r_{ik}| + \sum_{i,j} |d_{ij}| + \sum_i |m_i|)^{2mn|C|}$.*

Proof. For $S \subseteq A'$, existence of $a \geq 0$ with $f_a(S) = 0$ implies that valid $(\boldsymbol{p}^a = \hat{\boldsymbol{p}} + \boldsymbol{\delta}, \boldsymbol{\alpha}^a)$ of $DLP(\boldsymbol{\lambda}^a)$ is a feasible point in the following where $\hat{X}$ is the optimal solution of $LP(\boldsymbol{\lambda}^a)$ where delay of S is maximized:

$$
\begin{aligned}
\forall j: \quad & \textstyle\sum_{i \notin A'} \hat{x}_{ij} > 0 \text{ or } \sum_i \hat{x}_{ij} < 1 \Rightarrow \delta_j = 0 \\
\forall i \in A', \forall j: \quad & \hat{x}_{ij} > 0 \Rightarrow (\hat{\lambda} + a) d_{ij} = \textstyle\sum_k a_{ijk} \alpha_{ik} - \hat{p}_j - \delta_j \\
\forall i \in A', \forall j: \quad & \hat{x}_{ij} = 0 \Rightarrow (\hat{\lambda} + a) d_{ij} \geq \textstyle\sum_k a_{ijk} \alpha_{ik} - \hat{p}_j - \delta_j \\
\forall i \in A', \forall k: \quad & \textstyle\sum_j a_{ijk} \hat{x}_{ij} > r_{ik} \Rightarrow \alpha_{ik} = 0 \\
& \textstyle\sum_{i \in S, j} (\hat{p}_j + \delta_j)(\sum_{i \in S} \hat{x}_{ij}) = m(S) \\
& a \geq 0,\ \boldsymbol{\delta} \geq \mathbf{0},\ \boldsymbol{\alpha} \geq \mathbf{0}.
\end{aligned}
$$

Hence existence of rational such a of polynomial-size, and that it can be found by solving a feasibility LP follow. Cleraly such an a would be at most Δ.

For the second part we know that $\exists S \subseteq A'$, $f_{a_h}(S) = g(a_h) \leq 0$. From Lemma 24 we know that for some $a \in [0, a_h]$ $f_a(S) = 0$ and from the first part of this lemma we know that $a \leq \Delta$. By definition of g we have $g(a) \leq 0$ as well. Since $f_a(S)$ is monotonicaly decreasing in a for all $S \subseteq A'$ so is $g(a)$. The lemma follows.

$\square$

Lemma 6. *If $g(a) > 0$ for all $a \geq 0$ then A' satisfies the BB and SC conditions at $(\boldsymbol{\lambda}^{cur}, \boldsymbol{p}^{cur})$ at the start of Algorithm 4.*

Proof. From Lemma 25 we know that if $g(a) > 0$ for all $a \geq 0$ then $\boldsymbol{p}^a = \boldsymbol{p}^{cur}$ for all $a \geq 0$. Furthermore, for all $X \in LP(\boldsymbol{\lambda}^{cur})$ and for all $i \in A'$ $\boldsymbol{x}_i$ is an optimal bundle of agent i at zero

prices. That means $\text{delay}_i(X)$ is the same at all these allocations, implying $\text{pay}_i(X, \boldsymbol{p}^{cur})$ remains the same at all these allocations. For an $i \in A'$ set $T = i$, then we know that $f_a(T) \geq g(a) > 0$ implying $\text{pay}_i(X, \boldsymbol{p}^{cur}) \leq m_i$ at the allocation where T gets maximum delay among solutions of $LP(\boldsymbol{\lambda}^a)$. From previous conclusion that payment is the same at all solutions of $LP(\boldsymbol{\lambda}^{cur})$ it BB follows. And SC follows because $g(a) > 0$ and $\boldsymbol{p}^a = \boldsymbol{p}^{cur}$ at all $a \geq 0$. □

The next lemma follows essentially from Lemmas 23, 24, 25, and 26.

Lemma 7. *Function g satisfies the following: (i) $g(0) \geq 0$. (ii) $f_a(S)$ is continuous and monotonically decreasing in a, $\forall S \subseteq A'$, therefore g is continuous and monotonically decreasing. (iii) either $g(\Delta) \leq 0$ for $\Delta = (\sum_{i,j,k} |a_{ijk}| + \sum_{i,k} |r_{ik}| + \sum_{i,j} |d_{ij}| + \sum_i |m_i|)^{2mn|C|}$ or $g(a) > 0$ for all $a \geq 0$. (iv) Given a set $S \subseteq A'$, if $f_a(S) > 0$ and $f_{a'}(S) < 0$ for $a' > a > 0$, then $\exists a^* \geq 0$ such that $f_{a^*}(S) = 0$ and such an a^* can be computed by solving a feasibility linear program of polynomial-size.*

Proof. Part (i) follows from Lemma 23. Part (ii) from Lemma 24 and the fact that $g(a)$ is minimum of $f_a(S)$ over all subsets $S \subset A'$. Minimum of continuously decreasing functions is also continuously decreasing. Parts (iii) and (iv) from Lemmas 25 and 26. □

From Lemma 4 we know that at the end of NextSeg subroutine, when we have S^* a maximal minimizer of f_{a^*} and $f_{a^*}(S^*) = 0$ then S^* together with previous segments $S_1, \ldots, S_{k-1}$ will satisfy BB and SC at $(\boldsymbol{\lambda}^{new}, \boldsymbol{p}^{new})$. Next we show existence of appropriate $\epsilon > 0$ so that SC condition is satisfied for $A' \setminus S^*$. This will complete the missing part of Lemma 4.

Lemma 27. *If $a^* > 0$ such that $g(a^*) = 0$ and $S^* \subset A'$ be maximal set such that $f_{a^*}(S^*) = 0$, then $\exists \epsilon > 0$ rational of polynomial size such that SC condition is satisfied for $A' \setminus S^*$ w.r.t. $(\boldsymbol{\lambda}^{new}, \boldsymbol{p}^{new})$ where $\boldsymbol{\lambda}^{new} = \boldsymbol{\lambda}^{a^*} + \epsilon \mathbf{1}$, $\boldsymbol{p}^{new}$ is a valid optimal of $DLP(\boldsymbol{\lambda}^{new})$, and $\mathbf{1}$ is indicator vector of set $A' \setminus S^*$.*

Proof. To show SC for $A' \setminus S^*$ w.r.t. $(\boldsymbol{\lambda}^{new}, \boldsymbol{p}^{new})$, it suffices to show the following:

- $f_{\boldsymbol{\lambda}^{new},\boldsymbol{p}^{new}}(T) \geq 0, \forall T \subset A' \setminus S^*$.

Since S^* was the maximal set where value of function f_{a^*} is 0 and minimum value of f_{a^*} is zero, we have that for any $S \subset A' \setminus S^*$, $f_{a^*}(S \cup S^*) > 0$. If there is an $a^T > a^*$ such that $f_{a^T}(T) = 0$, then by applying Lemma 26 it is a rational of polynomial size and can be computed by solving a linear program, otherwise set $a^T = \infty$. Among all of these pick the least one, lets call it a^{min}. It has to be strictly more than a^*.

Let $A'' = A' \setminus S^*$, and fix set $S \subset A''$. Let $\hat{\boldsymbol{\lambda}} = \boldsymbol{\lambda}^{a^*}$, $\hat{X}$ be an optimal allocation of $LP(\hat{\boldsymbol{\lambda}})$ where $A'' \setminus S$ gets the best, then S, then S^* and then rest of the segments. Let $\hat{\boldsymbol{p}} = \boldsymbol{p}^{a^*}$. Similar to Lemma 26 solve the following LP to compute maximum value of a^S such that $m(S) \geq \text{pay}_S$ when λ_is of only A'' is increased. Here $b, \boldsymbol{\delta}$ and $\boldsymbol{\alpha}$ variables, $c_j^S = \sum_{i \in S} \hat{x}_{ij},\ \forall j$, and $\hat{\lambda} = \hat{\lambda}_i,\ i \in A''$.

$$
\begin{aligned}
&\max : b \quad \text{s.t.} \\
&\forall j : && \textstyle\sum_{i \notin A''} \hat{x}_{ij} > 0 \text{ or } \sum_i \hat{x}_{ij} < 1 \Rightarrow \delta_j = 0 \\
&\forall i \in A'', \forall j : && \hat{x}_{ij} > 0 \Rightarrow (\hat{\lambda} + b) d_{ij} = \textstyle\sum_k a_{ijk} \alpha_{ik} - \hat{p}_j - \delta_j \\
&\forall i \in A'', \forall j : && \hat{x}_{ij} = 0 \Rightarrow (\hat{\lambda} + b) d_{ij} \geq \textstyle\sum_k a_{ijk} \alpha_{ik} - \hat{p}_j - \delta_j \quad \forall (i,j) \\
&\forall i \in A'', \forall k : && \textstyle\sum_j a_{ijk} \hat{x}_{ij} > r_{ik} \Rightarrow \alpha_{ik} = 0 \\
& && \textstyle\sum_{i \in S, j} (\hat{p}_j + \delta_j) c_j^S \leq m(S) \\
& && b \geq 0,\ \boldsymbol{\delta} \geq \mathbf{0},\ \boldsymbol{\alpha} \geq \mathbf{0}.
\end{aligned}
$$

Clearly, $b = 0$, $\delta_j = 0,\ \forall j$, and $\alpha_{ik} = \hat{\alpha}_{ik}, \forall (i,k)$ is feasible where $(\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{p}})$ is the valid optimal solution of $LP(\hat{\boldsymbol{\lambda}})$. If there is a finite optimal of the above LP then set $a^S = b$ otherwise set $a^S = \infty$. Since $a^T > a^*$, we have $a^S > 0$. Taking ϵ to be less than $a^S,\ \forall S \subset A''$ will suffice, due to monotonicity of f_a function (Lemma 24). Since a^Ss are polynomial size, there is an ϵ of polynomial size. □

Putting everything together next we argue that at the end of the algorithm all the created segments satisfy BB and SC w.r.t. $(\boldsymbol{\lambda}^{cur}, \boldsymbol{p}^{cur})$. In other words, $(\boldsymbol{\lambda}^{cur}, \boldsymbol{p}^{cur})$ are *proper* .

Lemma 28. *The $\boldsymbol{\lambda}^{cur}$ and $\boldsymbol{p}^{cur}$ obtained at the end of Algorithm 3 are proper (Definition 4).*

Proof. Lemmas 4 and 27 imply that if at the beginning of k^{th} call to $NextSeg$, w.r.t. $(\boldsymbol{\lambda}^{cur}, \boldsymbol{p}^{cur})$, $S_1, \ldots, S_{k-1}$ satisfies BB and SC, A' satisfies SC, and $g(\Delta) \leq 0$, then at the end of it if $S_k \neq A'$ then, w.r.t. $(\boldsymbol{\lambda}^{new}, \boldsymbol{p}^{new})$, $S_1, \ldots, S_k$ satisfies BB and SC, and $A' \setminus S_k$ satisfies SC. On the other hand if $g(\Delta) = f_{a^1}(S^1) > 0$ on line 4 of the algorithm, then by Lemma 6 we have that $S_1, \ldots, S_{k-1}$ and A' satisfies BB and SC. Applying this inductively, starting from $k = 1$ where $A' = A$, and resetting $A' = A' \setminus S_k$ every time, the theorem follows. This is because if we made s calls in total to $NextSeg$ forming segments $S_1, \ldots, S_s$, and $A' = \emptyset$ at the end, then all s segments satisfy BB and SC w.r.t. $(\boldsymbol{\lambda}^{cur}, \boldsymbol{p}^{cur})$ at the end. Thus $(\boldsymbol{\lambda}^{cur}, \boldsymbol{p}^{cur})$ are *proper* . □

Next we show correctness of Algorithm 3 using Lemmas 2, 7 and 28, and Theorem 3, and the next theorem follows.

Theorem 6. *Given a market $\mathcal{M}$ satisfying extensibility and sufficient demand, Algorithm 3 returns its equilibrium allocation and prices in time polynomial in the size of the bit description of $\mathcal{M}$.*

Proof. The fact that if Algorithm 3 terminates in polynomial time then it returns equilibrium allocation and prices of market $\mathcal{M}$ follows from Lemmas 28 and 2, and Theorem 3.

The question is why should the algorithm terminate, and that too in polynomial-time. Note that, every call to NextSeg reduces size of active set of agents. Therefore, if the instance has n agents then algorithm makes at most n calls to NextSeg. Subroutine NextSeg does binary search for value of a between 0 and a polynomial-sized rational. In each iteration of binary search it minimizes a submodular function, and in each call to the submodular function we will be solving at most constantly many linear programs of polynomial size. Thus submodular minimization can be done in polynomial time. Due to Lemma 7 and in particular Lemma 26 value of a where $g(a)$ becomes zero for some set is rational of polynomial-size. Hence overall the binary search has to terminate in polynomial time. The next loop again takes at most $O(n)$ iterations, with sub-modular

minimization in each. Thus the NextSeg subroutine terminates in polynomial-time. Finding allocation satisfying Budget constraint(i) of all the agents $i \in A$ at the end of the algorithm, given prices $\boldsymbol{p}^{cur}$ and λ values $\boldsymbol{\lambda}^{cur}$ is equivalent to solving the feasibility linear program (2.3). Thus, overall the algorithm terminates in polynomial time. □

We get our main result using Theorem 6.

Theorem 2. *[Extensibility implies polynomial time algorithm] There is a polynomial time algorithm that computes a market equilibrium allocation X and prices $\mathbf{p}$ for any market $\mathcal{M}$ that satisfies extensibility.*

2.10 Fairness and Incentive Compatibility Properties

In this section we show fairness properties of our general model, and incentive compatibility properties of our algorithm as a mechanism in scheduling application. Yet another utility model is that of quasi-linear utilities, where the agent also specifies an "exchange rate" between delay and payments, and wants to minimize a linear combination of the two. We show in Section 2.10.3 that for such a utility model *there is no* IC mechanism that is also Pareto optimal and anonymous, even with a single good and two agents.

2.10.1 Fairness Properties

The first welfare theorem for traditional market models says that at equilibrium the utility vector of agents is Pareto-optimal among utility vectors at all possible feasible allocations. We first show a similar result for our markets. For our model the set of feasible delay cost vectors are

$$\mathcal{D} = \{(\text{delay}_1(X), \ldots, \text{delay}_n(X)) \mid \boldsymbol{x}_i \text{ is feasible in CC}(i) \text{ for each agent } i \in A, \text{ and } X \text{ satisfies Supply constraints for each good } j \in G\} .$$

Theorem 7. *Given market $\mathcal{M}$, the delay cost vector at any of its equilibrium is Pareto-optimal in set $\mathcal{D}$.*

Proof. Let $(\hat{X}, \hat{\boldsymbol{p}})$ be an equilibrium and $\hat{\beta}_{ik}$ and $\hat{\gamma}_i$ be the corresponding dual variables in (2.11). Let $Z = \{i \mid \hat{\gamma}_i = 0\}$. Note that if agent i has $\hat{\gamma}_i = 0$ then applying this to (2.11) results in

$$\begin{array}{lll} \forall k \in C: & \sum_{j \in G} a_{ijk} x_{ij} \geq r_{ik} & \perp \quad \beta_{ik} \geq 0 \\ \forall j \in G: & d_{ij} \geq \sum_k a_{ijk} \beta_{ik} & \perp \quad x_{ij} \geq 0\,. \end{array} \tag{2.28}$$

It is easy to check that the above are the corresponding complementary slackness conditions for an optimal bundle of agent i at zero prices. Therefore, each agent in Z gets an absolute best possible bundle at $\hat{X}$, and therefore the delay of these agents can not be improved.

Let's remove these agents and their allocation from the market and equilibrium. It is easy to see the remaining is an equilibrium for the remaining market. In this equilibrium we have $\hat{\gamma}_i > 0, \forall i \notin Z$.

Using Lemma 19 we know that for some vector $\boldsymbol{\lambda}^* > 0$, $\hat{X}$ is a solution of $LP(\boldsymbol{\lambda}^*)$. Let $d_i^* = \text{delay}_i(\hat{X}),\ \forall i \in A$. If there is a $\boldsymbol{d} \in \mathcal{D}$ such that $d_i \leq d_i^*,\ \forall i \in A$ with at least one strict inequality, then the strict inequality has to be for an $i \in A \setminus Z$. Furthermore, the allocation corresponding to $\boldsymbol{d}$ restricted to agents in $A \setminus Z$ is feasible in $LP(\boldsymbol{\lambda}^*)$ and would give strictly lower objective value, a contradiction. □

The next theorem establishes envy-freeness for the general model and follows directly from the equilibrium condition that every agent demands an optimal bundle at given prices.

Theorem 8. *Equilibrium allocation of a given market $\mathcal{M}$ is envy-free.*

Next we show that at equilibrium each agent gets a "fair share": the equilibrium allocation Pareto-dominates an "equal share" allocation, where each agent gets an equal amount of each resource. This property is also known as *sharing incentive* in the scheduling literature [55].

Theorem 9. *Given a market $\mathcal{M}$, let X be an allocation where agent i gets $\frac{m_i}{\sum_{i\in A} m_i}$ amount of each good, i.e., $x_{ij} = \frac{m_i}{\sum_{i\in A} m_i},\ \forall i \in A, \forall j \in G$. Then at any equilibrium $(X^*, \boldsymbol{p}^*)$ of market $\mathcal{M}$,* $\text{delay}_i(X^*) \leq \text{delay}_i(X),\ \forall i \in A$.

Proof. At equilibrium under-sold goods have price zero, and no agent spends more than its budget. This gives

$$\sum_{i,j} x^*_{ij} p^*_j \leq \sum_i m_i \Rightarrow \sum_j p^*_j \sum_i x^*_{ij} \leq \sum_i m_i \Rightarrow \sum_j p^*_j \leq \sum_i m_i\ .$$

Therefore, for agent i, bundle $\boldsymbol{x}_i$ where amount of every good is exactly $\frac{m_i}{\sum_{i\in A} m_i}$ costs money $\sum_j x_{ij} p^*_j = \frac{m_i}{\sum_i m_i} \sum_j p^*_j \leq m_i$. Thus, it is affordable at prices $\boldsymbol{p}^*$. However, she preferred $\boldsymbol{x}^*_i$ instead, which implies either $\boldsymbol{x}'_i$ is not feasible in $\text{CC}(i)$ in which case $\text{delay}_i(\boldsymbol{x}'_i)$ is infinity, or she prefers $\boldsymbol{x}^*_i$ to $\boldsymbol{x}'_i$. In either case we get $\text{delay}_i(X^*) \leq \text{delay}_i(X)$. □

2.10.2 Scheduling: Algorithm as a Truthful Mechanism

Market based mechanisms are usually not (dominant strategy) *incentive compatible* (IC), except in the large market assumption where each individual agent is too small to influence the price, and therefore can be assumed to act as a price taker. Somewhat surprisingly, we can show IC, in a certain sense, of the market based mechanism for the special case of our market that corresponds to the scheduling setting presented in Section 2.2, and its generalization described in Section 2.1.3 with multiple machine types.

We show that our market based mechanism is IC in the following sense: non-truthful reporting of m_i and r_{ik}s can never result in an allocation with a lower delay cost. A small modification to the payments, keeping the allocation the same, makes the entire mechanism incentive compatible for the setting in which agents want to first minimize their delay and subject to that, minimize their payments.

The first incentive compatibility assumes that utility of the agents is only the delay, and does

not depend on the money spent (or saved). Such utility functions have been considered in the context of online advertising [14, 46, 82]. It is a reflection of the fact that companies often have a given budget for procuring compute resources, and the agents acting on their behalf really have no incentive to save any part of this budget. Our model could also be applied to scenarios with virtual currency in which case the agents truly don't have any incentive to minimize payments.

The second incentive compatibility does take payments into account, but gives a strict preference to delay over payments. Such preferences are also seen in the online advertising world, where advertisers want as many clicks as possible, and only then want to minimize payments. The modifications required for this are minimal, and essentially change the payment from a "first price" to a "second price" wherever required.

Yet another utility model is that of quasi-linear utilities, where the agent also specifies an "exchange rate" between delay and payments, and wants to minimize a linear combination of the two. We show in Section 2.10.3 that for such a utility model *there is no* IC mechanism that is also Pareto optimal and anonymous, even with a single good and two agents. Pareto optimality is a benign notion of optimality that has been used as a benchmark for designing combinatorial auctions with budget constraints [42, 47, 57]. Anonymity is also a reasonable restriction, which disallows favoring any agent based on the identity. In the face of this impossibility, our mechanism offers an attractive alternative.

Pure delay minimization

Suppose that there are j independent copies of the basic scheduling setting in Section 2.2, with the requirement of agent i for the j^{th} copy being r_{ik}. In this section we show that our algorithm is actually IC, i.e., the agents have no incentive to misreport m_is or r_{ik}s, assuming that agents only want to minimize their delay cost and don't care about their payments as long as they are within the budgets. Note that reporting lower m_i or higher r_{ik} are the only possible types of misreport. Fixing preferences of all agents except agent i, consider two runs of the algorithm, one where agent i is

truthful and another where she misreports her preferences. In particular, say agent i either reports a lower budget m_i', and/or a higher requirements r_{ik}' for good j.

Consider the first iteration in which the two runs differ, and let (S_1, λ_1) and (S_2, λ_2) be the segments found respectively in the truthful and non-truthful runs in this iteration. For any $\boldsymbol{\lambda}$, any $\boldsymbol{p}$, and any set S that does not contain i, $f_{\boldsymbol{p},\boldsymbol{\lambda}}(S)$ remains the same between the two runs; for any set S that contains i, $f_{\boldsymbol{p},\boldsymbol{\lambda}}(S)$ is strictly smaller in the non-truthful run. Hence, i does not belong to any of the segments found in earlier iterations, and S_2 necessarily contains i.[7] Further, $\lambda_2 < \lambda_1$.

Let A' be the set of agents who are not in one of the segments found prior to the current iteration. By definition A' is the same for both the runs, and includes i, as argued in the previous paragraph. Let X^1 and X^2 be respectively the allocations output by the algorithm for the truthful and the non-truthful runs. We will show the existence of a weakly feasible allocation X' such that (1) For every agent $i' \in A', i' \neq i$, his delay in X' is no higher than his delay in X^1, and (2) For agent i, his allocation in X' is the same as his allocation in X^2.

This implies that i is no better off in the non-truthful run, because of the following reasoning. The total delay of all the agents in A' is minimized in X^1, therefore the total delay of all the agents in A' cannot be lower in allocation X', even when the delay for agent i is calculated using only his actual requirements. Since no other agent has a higher delay in X', it is impossible for i to get a lower delay.

It remains to show the existence of X' as claimed. We define X' differently based on whether the agent is in S_2 or not.

Case 1: $i' \in S_2$**:** In this case, $\boldsymbol{x}'_{i'} = \boldsymbol{x}^2_{i'}$. This satisfies the second requirement since $i \in S_2$. Since $\lambda_2 < \lambda_1$, every agent in S_2 faces a smaller price, for every copy j and every time slot in which she is allocated. For $i' \neq i$, given the same budget and the same requirements, this actually implies that her delay in X^2 is strictly smaller than her delay in X^1.

[7] Consider the possibilities where $i \notin S_2$ and note that S_2 cannot be the minimizer in the non-truthful run given that S_1 is the minimizer in the truthful run.

Case 2: $i' \notin S_2$**:** In this case, we first start with the allocation X^1, in the slots $[1, r_j(B \setminus S_2)]$ for each copy j. Note that these slots have not been allocated at all in Case 1. Consider the total deficit after this allocation. This must be equal to the total amount of slots in $[1, r_j(B \setminus S_2)]$ that are allocated to agents in S_2 by X^1, because of feasibility of X^1. Now re-allocate these empty slots in $[1, r_j(B \setminus S_2)]$ to make up for the remaining requirement of these agents, and note that this can only lower the delay.

Secondary preference for payments

In this section, we consider the utility model where an agent wants to first minimize her flow-time, and subject to that, wants to further minimize her payments. We keep the same allocation as Algorithm 1, but change the payments of some agents, and show that this is IC.

We first define the set of agents whose payments will be modified. Recall that Algorithm 1 outputs a sequence of segments, where each segment corresponds to a pair (λ, S). Call an agent *marginal* if he gets the latest slots in his segment. This includes agents who are in singleton segments, as well as agents who just happen to get such an allocation even though they are in a segment with other agents. We modify the payments of only the marginal agents; all non-marginal agents pay their budget.

Lemma 29. *Any non-marginal agent gets a strictly higher delay cost for any misreport of his information.*

Proof. Consider the proof of incentive compatibility for only delay cost minimization in Section 2.10.2, and the notation therein. Note that if $S_2 \neq \{i'\}$, then the delay cost of i' strictly increases. Now suppose $S_2 = \{i'\}$. In the new allocation f^2, agent i' gets the latest slots among all agents in B. Since i' is not a marginal agent, he was getting a strictly better allocation in f^1, and the lemma follows. □

This shows that the mechanism is IC for non-marginal agents, even with their payments equal

to the budgets.

We now define the modification to payments for marginal agents. As in Section 2.10.2, misreports can still not get a better delay cost for marginal agents, since the allocation remains the same. The only possibility is that misreporting can decrease payments, while keeping the delay cost the same. Marginal agents can decrease their budgets, still get the same allocation, and pay less in the equilibrium payment. This has a limit; at some lower budget declaration, they get "merged" with a previous segment, and any further lowering of the budget will strictly lower their delay cost. *The payment of a marginal agent is defined to be the infimum of all budget declarations for which the lower segments are unaltered, i.e., the run of the algorithm up to the previous segment remains unchanged.*

We now argue that this mechanism is IC, for marginal agents. We only need to consider misreports that don't change the allocation, since those that do only give a higher delay cost. Among these, misreporting the budget clearly has no effect on the payment. Finally, we argue that reporting a higher r_{ik} can only lead to a higher payment. This is because the budget at which the agent merges with the previous segment happens at a higher value, as can be seen from the formula for λ_S.

2.10.3 Quasi Linear Utility Model

In this section we consider a quasi linear utility model for the agents. In this model, agents can choose to tradeoff payment for delay cost, as specified by an "exchange rate", denoted by η_i, for agent i. We consider the design of incentive compatible (IC) auctions, that are also Pareto optimal. In the related literature of IC auctions for combinatorial auctions with budget constraints, this has been adopted as the standard notion of optimality. The usual notion of social welfare is ill fitted for the case of budgets.[8]

As in Section 2.1 let the allocation of agent i for good j denoted by x_{ij}, but now we don't have

[8]Of course, the revenue objective is also widely considered, and continues to make sense even in the presence of budgets.

prices for the slots. Instead we simply have a payment for each agent, denoted by $payment(i)$ for agent i. The allocation and the payments are together called the outcome of the auction. Agent i now wants to minimize the objective

$$\sum_j d_{ij}x_{ij} + \eta_i payment(i).$$

A type of an agent is its budget m_i, its covering constraints CC(i), and its η_i. An auction is (dominant strategy) IC if for any agent, misreporting its type does not lead to an outcome with a lower objective, no matter what the other agents report. An outcome is *Pareto optimal* if for no other outcome,

1. all agents, including the auctioneer, are at least as well off as in the given outcome, and
2. at least one agent is strictly better off.

The auctioneer's objective is to simply maximize the sum of all the payments.

We also restrict the auction to be *anonymous*, which means that the auction cannot rely on the identity of the agents. Formally, an auction is anonymous if it is invariant under all permutations of agent identities.

The main result of this section is an impossibility.

Theorem 10. *There is no IC, Pareto optimal, and anonymous auction for our scheduling problem with quasi linear utilities, for the case of a single good and two agents.*

Since a single good and two agents is the most basic case, an impossibility follows for all generalizations as well.

The theorem follows from a reduction to a combinatorial auction with additive valuations, and an impossibility result of [42]. Consider an auction for a single *divisible* item, with budget constraints. Agent i has valuation of v_i per unit quantity of the item, and a budget m_i. The outcome

of the auction is an allocation x_i and payment $payment(i)$ for agent i, such that $\sum_i x_i \leq 1$ and $x_i \in [0,1]$. The utility of agent i is $v_i x_i - payment(i)$, and the budget constraint as before is that $p_i \leq m_i$. IC and Pareto optimality are as before, and we need an additional notion of *individual rationality* (IR): $v_i x_i - payment(i) \geq 0$. [42] showed the following impossibility.

Theorem 11 ([42])**.** *There is no IC, Pareto optimal, IR and anonymous auction for auctioning a single divisible good to 2 agents with budget constraints.*

of Theorem 10. Consider an instance of the scheduling problem of Section 2.2 with a single machine and two agents, where each agent requires 1 unit of the good. Pareto optimality implies that goods are not wasted, so the entire first two slots are completely allocated. If agent i gets x_i units of the slot $t = 1$, then his delay cost is $x_i + 2(1 - x_i)$. His objective is then

$$x_i + 2(1 - x_i) + \eta_i payment(i) = 2 - \eta_i \left(\tfrac{1}{\eta_i} x_i - payment(i) \right).$$

Minimizing this objective is equivalent to maximizing $\frac{1}{\eta_i} x_i - payment(i)$, which is exactly as in the single divisible good auction with $v_i = \frac{1}{\eta_i}$. We also show that the IC constraint for the scheduling problem implies the IR constraint for the divisible good case. If the IR constraint is violated, i.e., $\frac{1}{\eta_i} x_i < payment(i)$, then the value of the objective of agent i for this outcome is strictly smaller than 2. Then the agent is better off stating a budget of 0. This will force his payment to 0. The worst delay cost he can get is 2, so his total objective value is 2.

Therefore, an IC, Pareto optimal, and anonymous auction for our scheduling problem implies an IC, Pareto optimal, IR, and anonymous auction for the divisible good case, and the theorem follows. □

2.11 Relation to Myerson's ironing

Recall the scheduling application in Section 2.1.3, which we specialize further as follows. There is a single machine ($d = 1$). Each agent requires only one unit of time slot, i.e., $r_{i1} = 1, \forall\, i \in A$.

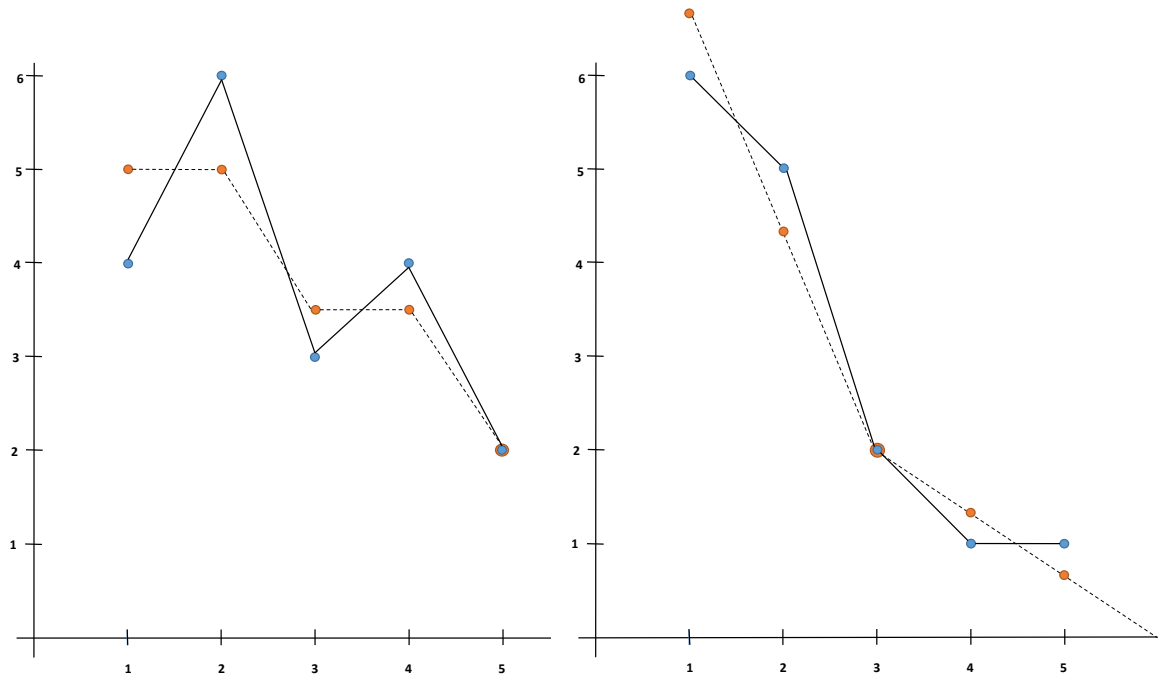


Figure 2.5: Example of Myerson's ironing
On the left is an example of Myerson's ironing. The solid curve with the blue dots is the given curve, which is non-monotone. The dashed curve with the orange dots is the ironed curve, which is monotone. On the right is an example of our problem. The solid curve is the money function, which is monotone but not convex. The dashed one is the price function, which is convex. Both dashed curves are such that their "area under the curve" is higher than that for the solid curves, and satisfy a minimality condition among all such curves.

The delay of slot j is j, i.e., $d_{j1} = j$ for all j. For this special case, we show that equilibrium conditions are equivalent to a set of conditions that are reminiscent of the *ironing* procedure used in the characterization of optimal auctions by [83]. It is in fact "one higher derivative" analog of Myerson's ironing.

Let's first restate Myerson's ironing procedure for the case of a uniform distribution over a discrete support. Suppose that we plot on the x-axis the quantiles, in the decreasing order of value, and on the y-axis the corresponding virtual values. This is possibly a non-monotone function, and Myerson's ironing asks for an ironed function that is *monotone non-increasing*, and is such that the area under the curve (starting at 0) of the ironed function is always higher than that for the given function. Further, the ironed function given by this procedure is the *minimal* among all such functions. This means that wherever the area under the curve differs for the two functions, the ironed function is *constant*. (See Figure 2.5 on page 81.)

In the special case of scheduling stated above, the equilibrium price of the good as a function of time is obtained as an ironed analog of the *money function*: the function $i \mapsto m_i$, where we assume the m_is are sorted in the decreasing order. This money function is monotone non-increasing by definition but it need not be a convex function. The price as a function of time must be a *monotone non-increasing and convex* function. The area under the curve of the price function must always be higher than that of the money function; further, wherever the two areas are different, the price function must be *linear*. One can see that the conditions are the same as that of Myerson's ironing, except each condition is replaced by a higher derivative analog. Unlike Myerson's, the solution to our problem is no longer unique and the solution set may be non-convex!

CHAPTER 3

SETTLING THE COMPLEXITY OF LEONTIEF AND PLC EXCHANGE MARKETS UNDER EXACT AND APPROXIMATE EQUILIBRIA

In this chapter we show membership in PPAD for the problem of computing approximate equilibria for an Arrow-Debreu exchange market for piecewise-linear concave (PLC) utility functions. As a corollary we also obtain membership in PPAD for Leontief utility functions. This settles an open question of Vazirani and Yannakakis (2011).

Next we show FIXP-hardness of computing equilibria in Arrow-Debreu exchange markets under Leontief utility functions, and Arrow-Debreu markets under linear utility functions and Leontief production sets, thereby settling these open questions of Vazirani and Yannakakis (2011). As corollaries, we obtain FIXP-hardness for PLC utilities and for Arrow-Debreu markets under linear utility functions and polyhedral production sets. In all cases, as required under FIXP, the set of instances mapped onto will admit equilibria, i.e., will be "yes" instances. If all instances are under consideration, then in all cases we prove that the problem of deciding if a given instance admits an equilibrium is ETR-complete, where ETR is the class Existential Theory of Reals.

As a consequence of the results stated above, and the fact that membership in FIXP has been established for PLC utilities, the entire computational difficulty of Arrow-Debreu markets under PLC utility functions lies in the Leontief utility subcase. This is perhaps the most unexpected aspect of our result, since Leontief utilities are meant for the case that goods are perfect complements, whereas PLC utilities are very general, capturing not only the cases when goods are complements and substitutes, but also arbitrary combinations of these and much more.

Finally, we give a polynomial time algorithm for finding an equilibrium in Arrow-Debreu exchange markets under Leontief utility functions provided the number of agents is a constant. This settles part of an open problem of Devanur and Kannan (2008).

3.0.1 Previous Results on Computability of Market Equilibria

The first utility functions to be studied were linear. Once polynomial time algorithms were found for markets under such functions [37, 39, 41, 43, 53, 62, 63, 87, 105, 110] and certain other cases [28, 38, 54, 64, 104], the next question was settling the complexity of Arrow-Debreu markets under separable, piecewise-linear concave (SPLC) utility functions. This problem was shown to be complete [25, 103] for PPAD. Also, when all instances are under consideration, the problem of deciding if a given SPLC market admits an equilibrium was shown to be NP-complete [103]. The notion of SPLC production sets was defined in [49] and Arrow-Debreu markets under such production sets and linear utility functions were shown to be PPAD-complete.

Previous computability results for Leontief utility functions were the following: In contrast to our result, Fisher markets under Leontief utilities admit a convex program [45] and hence their equilibria can be approximated to any required degree in polynomial time [7, 15]. Arrow-Debreu markets under Leontief utilities were shown to be PPAD-hard [29]. They reduce 2-Nash to a special case called "pairing economy" in which each agent brings one unit of a distinct good. For this case, equilibria are rational; however, in general they are irrational for Leontief markets [44], and hence their complexity is not characterized by PPAD. We note that the two complexity classes PPAD and FIXP appear to be quite disparate – whereas solutions to problems in the former are rational numbers, those to the latter are algebraic numbers. And whereas the former is contained in function classes NP $\cap$ co-NP, the latter lies somewhere between P and PSPACE, and is likely to be closer to the harder end of PSPACE [109].

Leontief utilities are a limiting case of constant elasticity of substitution (CES) utilities [75]. Finding an approximate equilibrium under the latter was also shown to be PPAD-complete [24].

3.1 Technical Contributions

Let $\mathcal{M}$ denote an Arrow-Debreu exchange market under piecewise-linear concave (PLC) utilities. Daskalakis, Goldberg and Papadimitriou [33] proved that computation of approximate fixed point of a Lipschitz continuous function, whose Lipschitz constant is polynomial sized, is in PPAD. In showing membership of the problem of computing an equilibrium of $\mathcal{M}$ in FIXP, [48] had given a particular fixed point formulation F such that the fixed points of F give equilibria of $\mathcal{M}$ and vice-versa. We start with F and to show that the problem of computing an ϵ-approximate market equilibrium for $\mathcal{M}$ is in PPAD, we need to show two things: (i) F is Lipschitz continuous with constant K, where $size(K) = poly(size(\mathcal{M}))$, and (ii) a δ-approximate fixed point of F gives an ϵ-approximate market equilibrium, where δ and ϵ are polynomially related, i.e., $size(\delta) = poly(size(\epsilon, \mathcal{M}))$.

The first step is easy to show; however, the second step is quite involved and technical because F is rather intricate. We note that even in the case of Nash equilibrium, whose fixed point formulation is relatively simple, it was non-trivial to show membership in PPAD [33]. Informally, at a market equilibrium (prices, allocation of goods to agents), each agent obtains an optimal bundle of goods and demand of each good meets its supply (market clearing). At given prices, optimal bundles of an agent can be captured through a linear program (LP). At a fixed point of F, the primal and dual constraints, and complementary slackness conditions of this LP are satisfied. This ensures that each agent receives an optimal bundle. We show that at an approximate fixed point of F, both feasibility and complementary slackness constraints are approximately satisfied. This proves that each agent gets an approximately optimal bundle. Further, we show that market clearing is also approximately satisfied at an approximate fixed point of F. Together, they imply approximate market equilibrium and hence membership in PPAD.

As stated in the Introduction, some claims made in the literature do not hold without formally proving membership in PPAD of the above-stated problem. As an example, [60] state that the

Leontief exchange market problem does not have a fully polynomial-time approximation scheme, unless PPAD $\subseteq$ P, and that the smoothed complexity of any algorithm for computing a market equilibrium in a Leontief economy, is not polynomial, unless PPAD $\subseteq$ RP.

We next move to our FIXP-hardness result and describe the difficulties encountered in obtaining reduction $\mathcal{R}$ and the ideas needed to overcome them. For this purpose, it will be instructive to draw a comparison between reduction $\mathcal{R}$ and the reduction from 2-Nash to SPLC markets given in [25]. At the outset, observe the latter is only dealing with linear functions of variables[1] and hence is much easier than the former.

Both reductions create one market with numerous agents and goods, and the amount of each good desired by an agent gets determined only after the prices are set. Yet, at the desired prices, corresponding to solutions to the problem reduced from, the supply of each good needs to be exactly equal to its demand. In the latter reduction, the relatively constrained utility functions give a lot more "control" on the optimal bundles of agents. Indeed, it is possible to create one large market with many agents and many goods and still argue how much of each good is consumed by each agent at equilibrium.

We do not see a way of carrying out similar arguments when all agents have Leontief utility functions. The key idea that led to our reduction was to create several modular units within the large market and ensure that each unit would have a very simple and precise interaction with the rest of the market. Leontief utilities, which seemed hard to manage, in fact enabled this in a very natural manner as described below.

Closed submarket: A closed submarket is a set S of agents satisfying the following: At every equilibrium of the complete market, the union of initial endowments of all agents in S exactly equals the union of optimal bundles of all these agents.

Observe that the agents in S will not be sequestered in any way — they are free to choose their

[1]Since the payoff of the row player from a given strategy is a linear function of the variables denoting the probabilities played by the column player.

optimal bundles from all the goods available. Yet, we will show that at equilibrium prices, they will only be exchanging goods among themselves. We note that the proof of PPAD-hardness for Nash equilibrium computation also uses small game gadgets to accomplish arithmetic operations of addition, multiplication and comparison [22, 33]; however, since there variables are captured through strategies of different players, these gadgets did not interfere. In our case, the primary challenge is to prevent flow of goods across gadgets and to ensure desired price dependencies even when same goods are used across gadgets. We achieve this through the notion of closed submarkets.

These closed submarkets enable us to ensure that variables denoting prices of goods satisfy specified arithmetic relations. The latter are linear function and product; we show that these two arithmetic relations suffice to encode any polynomial equation. Under linear functions, we want that $p_a = Bp_b + Cp_c + D$, where B, C and D are constants.

Under product, we want that $p_a = p_b \cdot p_c$. Designing this closed submarket, say $\mathcal{M}$, requires several ideas, which we now describe. $\mathcal{M}$ has an agent i whose initial endowment is one unit of good a and she desires only good c. We will ensure that the *amount* of good c leftover, after all other agents in the submarket consume what they want, is exactly p_b, i.e., the *price* of good b. At equilibrium, i must consume all the leftover good c, whose total cost is $p_b \cdot p_c$. Therefore the price of her initial endowment, i.e., one unit of good a, must be $p_b \cdot p_c$, hence establishing the required product relation. The tricky part is ensuring that exactly p_b amount of good c is leftover, without knowing what p_b will be at equilibrium. This is non-trivial, and this submarket needs to have several goods and agents in addition to the ones mentioned above.

Once reduction $\mathcal{R}$ is established, FIXP-hardness follows from the straightforward observation that a 3-Nash instance can be encoded via polynomials, where each variable, which represents the probability of playing a certain strategy, is constrained in the interval $[0, 1]$. To get ETR-hardness, we appeal to the result of Schaefer and Štefankovič [97] that checking if a 3-Nash instance has a solution in a ball of radius half in l_∞-norm is ETR-hard; this entails constraining the variables to be

in the interval $[0, 1/2]$. By Nash's theorem, in the former case, the market will admit an equilibrium and in the latter case, it will admit an equilibrium iff the 3-Nash instance has a solution in the ball of radius half in l_∞-norm. Membership in ETR follows by essentially showing a reduction in the reverse direction: given a Leontief market, we obtain a set of simultaneous multivariate polynomial equations whose roots capture its equilibria.

Our final result gives a polynomial time algorithm for computing an equilibrium for Arrow-Debreu exchange markets under Leontief utility functions provided the number of agents is a constant, say d. Using the property that equilibrium allocation of an agent can be written in terms of her equilibrium utility, we show that if equilibrium exists, then there is one where the number of goods with positive prices is at most d. Next we iterate over all subsets of size d of goods, and for each set we reduce the problem of checking existence of equilibrium to checking feasibility of a set of polynomial inequalities in $2d$ dimension. Since this can be done in polynomial time [5, 6], we get a polynomial time algorithm.

3.1.1 Organization of the chapter

We only give an overview of our results here and refer the reader to the full version paper for complete details and proofs. In Section 3.2.1 we define the Arrow-Debreu exchange market model, and the relevant utility functions. The definition of 3-player Nash equilibrium problem and its relation with the complexity classes FIXP and ETR are given in Section 3.2.2. Section 3.3 contains an overview of our second result where we show FIXP-hardness of computing an equilibrium in Leontief exchange markets. An overview of our first result where we show membership in PPAD for computing approximate equilibrium in exchange markets under PLC utilities is given in Section 3.4. Finally, Section 3.5 contains an overview of our third result where we give a polynomial time algorithm for computing an equilibrium for Arrow-Debreu exchange markets under Leontief utilities provided the number of agents is a constant.

3.2 Preliminaries

3.2.1 The Arrow-Debreu Market Model

An Arrow-Debreu (AD) *exchange* market[2] [67] consists of a set G of divisible goods and a set A of agents. Let g denote the number of goods in the market. Each agent i comes with an initial endowment of goods; W_{ij} is amount of good j with agent i. The preference of an agent i over bundles of goods is captured by a non-negative, non-decreasing and concave utility function $U_i : \mathbb{R}_+^g \to \mathbb{R}_+$. Non-decreasingness is due to free disposal property, and concavity captures the law of diminishing marginal returns. Each agent wants to buy a (optimal) bundle of goods that maximizes her utility to the extent allowed by her earned money from the initial endowment.

Let $\boldsymbol{p} \in \mathbb{R}_+^g$ denote the prices of goods, where p_j is the price of good j. Given $\boldsymbol{p}$, let $OPT_i(\boldsymbol{p}) = \arg\max_{\boldsymbol{y}}\{U_i(\boldsymbol{y}) \mid \sum_j y_j p_j = \sum_j W_{ij} p_j\}$ denote the set of optimal bundles of agent i. Let $\boldsymbol{x}_i \in \mathbb{R}^g$ denote the assignment of goods to agent i, where x_{ij} is the amount of good j.

If there is an assignment $\boldsymbol{x}_i \in OPT_i(\boldsymbol{p})$ to each agent i so that there is neither deficiency nor surplus of any good, then such prices are called *market clearing* or *market equilibrium* prices. Formally,

Definition 6 (Market Equilibrium)**.** $(\boldsymbol{x}, \boldsymbol{p})$ is an equilibrium of an Arrow-Debreu exchange market $\mathcal{M}$ if

$$\begin{aligned} \forall i \in A \quad &: \quad \boldsymbol{x}_i = \arg\max_{\boldsymbol{y}}\{U_i(\boldsymbol{y}) \mid \sum_j y_j p_j \leq \sum_j W_{ij} p_j\} \\ \forall j \in G \quad &: \quad \sum_{i \in \boldsymbol{\alpha}} x_{ij} = \sum_{i \in \boldsymbol{\alpha}} W_{ij}. \end{aligned}$$

The market equilibrium problem is to find such prices when they exist. In a celebrated result, Arrow and Debreu [67] proved that market equilibrium always exists under some mild conditions, however the proof is non-constructive and uses the machinery of Kakutani fixed point theorem.

[2]Refer to the full version for the definition of Arrow-Debreu markets with production firms.

We note that an arbitrary market may not admit an equilibrium.

To work under finite precision it is customary to assume that utility functions are piecewise-linear concave.

Piecewise-Linear Concave (PLC) Utility Function

The utility function U_i of agent i is said to be piecewise-linear concave (PLC) if at bundle $X_i = (x_{i1}, \ldots, x_{ig})$ it is given by

$$U_i(X_i) = \min_k \{\sum_j U_{ij}^k x_{ij} + T_i^k\},$$

where U_{ij}^k's and T_i^k's are given non-negative rational numbers. Since the agent gets zero utility when she gets nothing, we have $U_i(\mathbf{0}) = 0$, and therefore at least one T_i^k is zero.

Leontief Utility Function.

It is a special subclass of PLC, where each good is required in a fixed proportion. Formally, it is given by:

$$U_i(X_i) = \min_{j \in G: A_{ij} > 0} \{\frac{x_{ij}}{A_{ij}}\}, \quad A_{ij} \geq 0.$$

In other words the agent wants good j in A_{ij} proportion. Clearly, the agent has to spend $\sum_j A_{ij} p_j$ amount of money to get one unit of utility. Thus, optimal bundle satisfies the following condition.

$$\forall j \in G,\ x_{ij} = \beta_i A_{ij}, \quad \text{where } \beta_i = \frac{\sum_{j \in G} W_{ij} p_j}{\sum_{j \in G} A_{ij} p_j} \tag{3.1}$$

Since all market equilibria may be irrational even in the special case of Leontief utilities [44], it may not be possible to compute them exactly. Next we define a notion of approximate market equilibrium.

Definition 7 (ϵ-approximate market equilibrium)**.** $(\boldsymbol{x}, \boldsymbol{p})$ is an ϵ-approximate equilibrium of an Arrow-Debreu exchange market $\mathcal{M}$ if each agent receives an approximately optimal bundle and the aggregate demand of each good is approximately its aggregate supply, i.e.,

$$\begin{aligned}
\forall i \in \boldsymbol{\alpha} \quad &: \quad U_i(\boldsymbol{x}_i) \geq (1-\epsilon) U_i(\tilde{\boldsymbol{x}}_i), \text{where} \\
&\qquad \tilde{\boldsymbol{x}}_i = \arg\max_{\boldsymbol{y}} \{U_i(\boldsymbol{y}) \mid \sum_j y_j p_j \leq \sum_j W_{ij} p_j\} \\
\forall i \in \boldsymbol{\alpha} \quad &: \quad \sum_{j \in \mathcal{G}} x_{ij} p_j \leq (1+\epsilon) \sum_{j \in \mathcal{G}} W_{ij} p_j \\
\forall j \in \mathcal{G} \quad &: \quad \sum_{i \in \boldsymbol{\alpha}} x_{ij} \leq (1+\epsilon) \sum_{i \in \boldsymbol{\alpha}} W_{ij} \, .
\end{aligned}$$

Sufficiency conditions

Market equilibrium may not exist, and it is NP-complete to decide whether there exists an equilibrium even in the exchange markets with SPLC utility functions [103]. Arrow-Debreu [67] gave the following sufficiency conditions for the existence of equilibrium: $W > 0$ and each agent is non-satiated. In case of PLC markets, non-satiation implies that for every k, there exists a j such that $U_{ij}^k > 0$.

3.2.2 3-Player Nash Equilibrium (3-Nash)

Given a 3-player finite game, let the set of strategies of player $p \in \{1, 2, 3\}$ be denoted by $\mathcal{S}_p$. Let $\mathcal{S} = \mathcal{S}_1 \times \mathcal{S}_2 \times \mathcal{S}_3$. Such a game can be represented by 3-dimensional tensors A_1, A_2 and A_3, representing payoffs of first, second and third players respectively. If players play $\boldsymbol{s} = (s_1, s_2, s_3) \in \mathcal{S}$, then the payoffs are $A_1(\boldsymbol{s})$, $A_2(\boldsymbol{s})$ and $A_3(\boldsymbol{s})$ respectively. Without loss of generality, we assume that $0 \leq A_p(\boldsymbol{s}) \leq 1, \forall p$.

Let Δ_p denote the probability distribution over set $\mathcal{S}_p$, $\forall p \in \{1, 2, 3\}$ (the set of mixed-strategies for player p), and let $\Delta = \Delta_1 \times \Delta_2 \times \Delta_3$. Given a mixed-strategy profile $\boldsymbol{z} = (\boldsymbol{z}_1, \boldsymbol{z}_2, \boldsymbol{z}_3) \in \Delta$, let z_{ps} denote the probability with which player p plays strategy $s \in \mathcal{S}_p$, and

let $\boldsymbol{z}_{-p}$ be the strategy profile of all the players at $\boldsymbol{z}$ except p. For player $p \in \{1,2,3\}$ the total payoff and payoff from strategy $s \in \mathcal{S}_p$ at $\boldsymbol{z}$ are respectively,

$$\pi_p(\boldsymbol{z}) = \sum_{\boldsymbol{s}\in\mathcal{S}} A_p(\boldsymbol{s}) z_{1s_1} z_{2s_2} z_{3s_3} \quad \text{and}$$

$$\pi_p(s, \boldsymbol{z}_{-p}) = \sum_{\boldsymbol{t}\in\mathcal{S}_{-p}} A_p(s, \boldsymbol{t}) \prod_{q\neq p} z_{qt_q} \ .$$

Definition 8 (Nash (1951) [84])**.** A mixed-strategy profile $\boldsymbol{z} \in \Delta$ is a Nash equilibrium (NE) if no player gains by deviating unilaterally. Formally, $\forall p = 1,2,3 \ \ \pi_p(\boldsymbol{z}) \geq \pi_p(\boldsymbol{z}', \boldsymbol{z}_{-p}), \forall \boldsymbol{z}' \in \Delta_p$.

Consider the following system of multivariate polynomials, where $\boldsymbol{A} = (A_1, A_2, A_3)$:

$$F_{NE}(\boldsymbol{A}): \begin{array}{ll} \forall p \in \{1,2,3\}, & \sum_{s\in S_p} z_{ps} = 1 \\ \forall p \in \{1,2,3\},\ s \in S_p, & \pi_p(s, \boldsymbol{z}_{-p}) + \beta_{ps} = \delta_p \\ \forall p \in \{1,2,3\},\ s \in S_p, & z_{ps}\beta_{ps} = 0 \\ \forall p \in \{1,2,3\},\ s \in S_p, & 0 \leq z_{ps} \leq 1 \\ \forall p \in \{1,2,3\},\ s \in S_p, & 0 \leq \beta_{ps} \leq 1 \\ \forall p \in \{1,2,3\},\ s \in S_p, & 0 \leq \delta_p \leq 1 \ . \end{array} \tag{3.2}$$

Lemma 30. *Nash equilibria of $\boldsymbol{A}$ are exactly the solutions of system $F_{NE}(\boldsymbol{A})$, projected onto $\boldsymbol{z}$.*

Let 3-Nash denote the problem of computing a Nash equilibrium of a 3-player game. Next we describe its relation with the complexity classes FIXP and ETR.

3.2.3 The Class FIXP

The class FIXP was defined to capture complexity of the exact fixed point problems with algebraic solutions [68]. An instance I of FIXP consists of an algebraic circuit C_I defining a function $F_I : [0,1]^d \to [0,1]^d$, and the problem is to compute a fixed-point of F_I. The circuit is a finite representation of function F_I (like a formula), consisting of $\{\max, +, *\}$ operations, rational con-

stants, and d inputs and outputs. We note that a circuit representing a problem in FIXP operates on real numbers.

In order to remain faithful to Turing machine computation, [68] also defined three discrete problems on FIXP, namely $FIXP_{pc}$ (partial computation), $FIXP_d$ (decision) and $FIXP_a$ ((strong) approximation). We refer to the full version for their definitions.

Whereas FIXP is a class of, in general, real-valued search problems, whose complexity can be studied in a real computation model, e.g., [13], note that $FIXP_{pc}$, $FIXP_d$ and $FIXP_a$ are classes of discrete search problems, hence their complexity can be studied in the standard Turing machine model. This is precisely the reason to define these three classes. [68] showed the following result.

Theorem 12 (Etessami-Yannakakis (2010) [68])**.** *Given a* 3*-player game* $\boldsymbol{A} = (A_1, A_2, A_3)$*, computing its NE is FIXP-complete. In particular, the corresponding Decision, (Strong) Approximation, and Partial Computation problems are complete respectively for* $FIXP_d$*,* $FIXP_a$ *and* $FIXP_{pc}$*.*

3.2.4 Existential Theory of Reals (ETR)

The class ETR was defined to capture the decision problems arising in *existential theory of reals* [97]. An instance I of class ETR consists of a sentence of the form

$$(\exists x_1, \ldots, x_n)\phi(x_1, \ldots, x_n),$$

where ϕ is a quantifier-free $(\wedge, \vee, \neg)$-Boolean formula over the predicates (sentences) defined by signature $\{0, 1, -1, +, *, <, \leq, =\}$ over variables that take real values. The question is if the sentence is true. The size of the problem is $n + size(\phi)$, where n is the number of variables and $size(\phi)$ is the minimum number of signatures needed to represent ϕ (we refer the readers to [97] for detailed description of ETR, and its relation with other classes like PSPACE). Schaefer and Štefankovič showed the following result; the first result on the complexity of a *decision* version of

3-Nash.

Nash equilibrium always exists [84], however there are many non-trivial *decision* questions.

Definition 9 (Decision 3-Nash)**.** Decision 3-Nash is the problem of checking if a given 3-player game $\boldsymbol{A}$ admits a Nash equilibrium $\boldsymbol{z}$ such that $\boldsymbol{z} \leq 0.5$.

Theorem 13 (Schaefer-Štefankovič (2015) [97])**.** *Decision* 3*-Nash is ETR-complete.*

Note that changing the upper bound on all z_{ps}'s from 1 to 0.5 in $F_{NE}(\boldsymbol{A})$ (3.2), exactly captures the NE with $\boldsymbol{z} \leq 0.5$. Thus *Decision* 3*-Nash* can be reduced to checking if such a system of polynomials admits a solution. Next we show a construction of Leontief exchange markets to exactly capture the solutions of a system of polynomials, similar to that of $F_{NE}(\boldsymbol{A})$, at its equilibria.

3.3 Multivariate Polynomials to Leontief Exchange Market

Consider the following system of m multivariate polynomials on n variables $\boldsymbol{z} = (z_1, \ldots, z_n)$:

$$F : \{f_i(\boldsymbol{z}) = 0, \forall i \in [m]; \quad 0 \leq L_j \leq z_j \leq U_j, \ \forall j \in [n]\} \ . \tag{3.3}$$

The coefficients of f_i's, and the upper and lower bounds U_j's and L_j's are assumed to be rational numbers. In this section we show that solutions of F can be captured as equilibrium prices of a Leontief exchange market. The problems of 3-Nash and Decision 3-Nash can be characterized by a set similar to (3.3) (Lemma 30), in turn we obtain FIXP and ETR hardness results for Leontief exchange markets, from the corresponding hardness of 3-Nash (Theorems 12 and 13).

Polynomial f_i is represented as sum of monomials, and a monomial $\alpha z_1^{d_1} \ldots z_n^{d_n}$ is represented by tuple $(\alpha, d_1, ..., d_n)$; here coefficient α is a rational number.[3] Let $\mathcal{M}_{f_i}$ denote the set of monomials of f_i, and $size[f_i] = \sum_{(\alpha,\boldsymbol{d})\in\mathcal{M}_{f_i}} size(\alpha, \boldsymbol{d})$, where $size(r)$ for a rational number r is the minimum number of bits needed to represent its numerator and denominator. Degree of f_i

[3]In fact, our reduction is also applicable to a succinct representation of polynomials, e.g., $f_i = (x+1)^d$, however the monomial representation is enough to obtain the hardness result due to (3.2).

is $deg(f_i) = \max_{(\alpha,\boldsymbol{d})\in\mathcal{M}_{f_i}} \sum_j d_j$. Size of F, denoted by $size[F]$, is $m + n + \sum_j(size(U_j) + size(L_j)) + \sum_i(deg(f_i) + size[f_i])$. Given F, next we construct an exchange market in time polynomial in $size[F]$, whose equilibria correspond to solutions of F.

Preprocessing. First we transform system F into a polynomial sized equivalent system that uses only the following basic operations on *non-negative* variables (refer to the full version for more details).

$$\begin{aligned} &(\mathbf{LIN}.) \quad z_a = Bz_b + Cz_c + D, \text{ where } B, C, D \geq 0 \\ &(\mathbf{QD}.) \quad z_a = z_b * z_c \end{aligned} \tag{3.4}$$

Let $R(F)$ be a reformulation of F using these basic operations. All variables in $R(F)$ are constrained to be non-negative. In order to construct $R(F)$ from F, we need to introduce many auxiliary variables. Let the number of variables in $R(F)$ be N, and out of these let $z_1, \ldots, z_n$ be the original set of variables of F (3.3). Given a system $R(F)$ of equalities, we will construct an exchange market $\mathcal{M}$, such that value of each variable $z_j,\ j \in [N]$ is captured as price p_j of good G_j in $\mathcal{M}$. Further, we make sure that these prices satisfy all the relations in $R(F)$ at every equilibrium of $\mathcal{M}$.

Ensuring scale invariance. Since equilibrium prices of an exchange market are scale invariant, the relations that these prices satisfy have to be scale invariant too. However note that in (3.4) $(LIN.)$ and $(QD.)$ are not scale invariant. To handle this we introduce a special good G_s, such that when its price p_s is set to 1 we get back the original system.

$$\begin{aligned} &(\mathbf{LIN}.) \quad p_a = Bp_b + Cp_c + Dp_s, \text{ where } B, C, D \geq 0 \\ &(\mathbf{QD}.) \quad p_a = \tfrac{p_b * p_c}{p_s} \end{aligned} \tag{3.5}$$

Let $R'(F)$ be a system of equalities after applying the transformation of (3.5) to $R(F)$. Note that, $R'(F)$ has exactly one extra variable than $R(F)$, namely p_s, and solutions of $R'(F)$ with $p_s = 1$ are exactly the solutions of $R(F)$. Let the size of $R'(F)$ be (# variables + # relations

in $R'(F)$ + $size(B, C, D)$ in each of (LIN.)-type relations). To bound the values at a solution of $R'(F)$, define

$$H = M_{max}U_{max}^{d} + 1, \text{ where } M_{max} = \max_{i} |\mathcal{M}_{f_i}|,$$

$$d = \max_{f_i} deg(f_i), \text{ and } U_{max} = \max\{\max_{j} U_j, \max_{f_i, (\alpha, \boldsymbol{d}) \in \mathcal{M}_{f_i}} |\alpha|\}.$$

Lemma 31. $size[R'(F)] = poly(size[F])$. *Vector* $\boldsymbol{p}$ *is a non-negative solution of* $R'(F)$ *with* $p_s = 1$ *iff* $z_j = p_j, \forall j \in [n]$ *is a solution of* F. *Further,* $p_j \leq H, \forall j \in [N]$.

3.3.1 Market Construction

In this section we construct market $\mathcal{M}$ consisting of goods $G_1, \ldots, G_N$ and G_s, such that the prices $p_1, \ldots, p_N$ and p_s, satisfy all the relations of $R'(F)$ at equilibrium. To ensure $p_s > 0$ at equilibrium, we add the following agent to $\mathcal{M}$. Recall that W_{ij} is the amount of good G_j agent A_i brings to the market, X_i is the bundle of goods consumed by her, and $U_i : \mathbb{R}_+^g \to \mathbb{R}_+$ is her utility function.

$$A_s : \ W_{ss} = 1, \ W_{sj} = 0, \ \forall j \in [N]; \quad U_s(X_s) = x_{ss} \tag{3.6}$$

Lemma 32. *At every equilibrium of market* $\mathcal{M}$, *we have* $p_s > 0$, *and* $x_{ss} = W_{ss}$.

Since a price p_j may be used in multiple relations of $R'(F)$, the corresponding good has to be used in many different gadgets. When we combine all these gadgets to form market $\mathcal{M}$, the biggest challenge is to analyze the flow of goods among these gadgets at equilibrium. We overcome this all together by forming *closed submarket* for each gadget.

Definition 10 (Submarket)**.** A submarket $\widetilde{\mathcal{M}}$ of a market $\mathcal{M}$ consists of a subset of agents and goods such that endowment and utility functions of agents in $\widetilde{\mathcal{M}}$ are defined over goods only in $\widetilde{\mathcal{M}}$.

Definition 11 (Closed Submarket)**.** A submarket $\widetilde{\mathcal{M}}$ of a market $\mathcal{M}$ is said to be closed if at every equilibrium of the entire market $\mathcal{M}$, the submarket $\widetilde{\mathcal{M}}$ is locally at equilibrium, i.e., its total

demand equals its total supply. The total demand of $\widetilde{\mathcal{M}}$ is the sum of demands of agents in $\widetilde{\mathcal{M}}$ and its total supply is the sum of initial endowments of agents in $\widetilde{\mathcal{M}}$.

In other words, $\widetilde{\mathcal{M}}$ does not interfere with the rest of the market in terms of supply and demand, even if some goods in $\widetilde{\mathcal{M}}$ are used outside as well. Note that the market of (3.6) is a closed submarket (Lemma 32) with only one agent A_s and one good G_s. We will see that each submarket $\widetilde{\mathcal{M}}$ establishing a relation of $(LIN.)$ and $(QD.)$ has a set of *exclusive* goods used only in $\widetilde{\mathcal{M}}$, to achieve the *closed* property. Before describing construction of closed submarkets for more involved relations, first we describe it for a simple *equality* relation.

Submarket for relation $(EQ.)$ $p_a = p_b$

The gadget for $(EQ.)$ consists of two agents with Leontief utility functions, as given in Table 3.1, where good G_r is exclusive to this submarket. The endowment vector W_i of agent A_i should be interpreted as (amount of G_a, amount of G_b, amount of G_r), i.e., in the same order of goods as listed on the first line of the table.

Table 3.1: Closed submarket (EQ.) $p_a = p_b$

$\mathcal{M}_{\mathbf{EQ}}$:	2 Agents (A_1, A_2) and 3 Goods (G_a, G_b, G_r)
	(G_r: an exclusive good)
	A_1: $W_1 = (0, 1, 1)$ and $U_1(X) = \min\{x_a, x_r\}$
	A_2: $W_2 = (1, 0, 1)$ and $U_2(X) = \min\{x_b, x_r\}$

Lemma 33. *The market $\mathcal{M}_{EQ}$ of Table 3.1: (i) is a closed submarket, (ii) at equilibrium, it enforces $p_a = p_b$, and (iii) every non-negative solution of $p_a = p_b$ gives an equilibrium.*

Proof. Let α and β denote the utility obtained by A_1 and A_2 at equilibrium respectively. Then using (3.1) which characterizes optimal bundles for Leontief functions, the market clearing conditions of the two agents give: $p_b + p_r = \alpha(p_a + p_r)$ and $p_a + p_r = \beta(p_b + p_r)$.

Clearly these conditions imply that $\alpha\beta = 1 \Rightarrow \beta = 1/\alpha$. Note that A_1 and A_2 consume α and β amounts of good G_r respectively. And since this good is exclusive to $\mathcal{M}_r$, no other agent will

consume it. Further, there are exactly two units of G_r available in the entire market $\mathcal{M}$. Hence we get $\alpha + \beta \leq 2$. Replacing $\beta = \frac{1}{\alpha}$ gives $(\alpha - 1)^2 \leq 0 \Rightarrow \alpha = \beta = 1$. Therefore, we get that every equilibrium of $\mathcal{M}_r$ enforces $p_a + p_r = p_b + p_r \Rightarrow p_a = p_b$. Further, $\mathcal{M}_r$ is a closed submarket because at equilibrium, demand of every good *in* $\mathcal{M}_r$ is equal to its supply *in* $\mathcal{M}_r$ even though every good except G_r might participate in the rest of the market as well. For the last part, if $p_a = p_b \geq 0$, then choosing $p_r = 1$, and $x_{1a} = x_{1r} = x_{2b} = x_{2r} = 1$ gives a market equilibrium of $\mathcal{M}_r$. □

We refer the reader to full version for the $(LIN.)$ submarket, which is an extension of $(EQ.)$.

Submarket for Relation $(QD.)$ $p_a = \frac{p_b p_c}{p_s}$

The market construction is quite involved, so we simplify it using the two assumptions (refer to the full version for complete details). First, that $p_s = 1$ and second, that $p_b \neq 0$.

As mentioned earlier, establishing this relation in the market turns out to be complex, even with the above two assumptions. For this, we need to make sure that at every equilibrium price p_a of a good a is same as the product of prices of two other goods b and c. One way to establish this is by creating an agent who brings 1 unit of good a and desires only good c. For this to work, the challenge is to ensure that the amount of good c leftover, after all other agents in the submarket consume what they want, is exactly p_b, without knowing what p_b will be at equilibrium. We show that this is indeed possible, but for this we need to create several gadgets and combine them in a particular way.

In order to present the submarket in a modular manner, we will first define some devices. Each of these devices will be implemented via a set of agents with Leontief utility functions. Each device ensures a certain relationship between the net endowment left over by these agents and the net consumption of these agents; for convenience, we will call these the *net endowment and net consumption of the device*. Clearly, at equilibrium prices, for each device, the total worth of its net endowment and net consumption must be equal. The first device converts price of a good to

amount of another good whose price is one.

Converter (Conv(q)): The net consumption of this device is 1 unit of good G_1, whose price is p, and the net endowment is p/q units of good G_2, whose price is q. Table 3.2 and Figure 3.1 illustrate the implementation. In the figure tuple on edges represent *(amount, price)* of the goods whose number is shown in circle. Table 3.2 has two parts: Part 1 describes the market and Part 2 enforces linear relations using $(LIN.)$ submarkets.

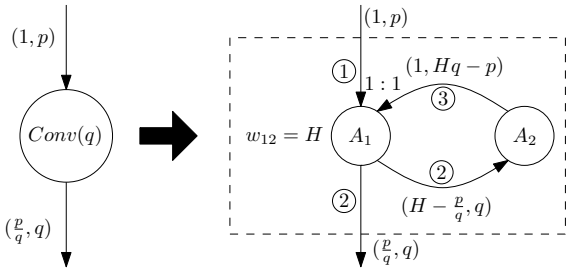


Figure 3.1: Flow of goods in Part 1 of Table 3.2 for Conv(q). Wires are numbered in circle, and wire i carries good G_i. The tuple on each wire represents (amount, price).

Table 3.2: A closed submarket for Conv(q)

Part 1:	**Input:** 1 unit of G_1 at price p **Output:** p/q units of G_2 at price q 2 Agents (A_1, A_2), 3 goods (G_1, G_2, G_3) A_1: $W_{12} = H$ and $U_1(X) = \min\{x_1, x_3\}$ A_2: $W_{23} = 1$ and $U_2(X) = x_2$
Part 2:	Closed submarkets for these linear relations $p_2 = q$ $p_3 = Hq - p$

There are two agents A_1 and A_2, and three goods G_1, G_2 and G_3. A_1 brings H units of G_2, whose price is set to q (H is a constant defined in Section 3.3). A_1 wants to consume G_1 and G_3 in the ratio of 1:1. The net consumption of this device, i.e., 1 unit of G_1 at price p, is consumed by A_1. A_2 brings 1 unit of G_3, whose price is enforced to $Hq - p$. A_2 wants to consume only

G_2, hence it consumes $H - p/q$ units of G_2 (observe that there is no need to perform the division involved in p/q explicitly). The remaining p/q units of G_2 form the net endowment of the device, as required.

We will use $Conv$ to convert price p_c to endowment of a good with price 1. In order to convert this endowment to an endowment of a good with price p_b and to make the entire submarket closed, we need the following two more devices. Their construction is a bit more involved, and we refer the reader to full version for their complete details.

Combiner (Comb(l, p_a, p_b)): The net consumption is l units each of goods G_1 and G_2, whose prices are p_a and p_b, respectively. The net endowment is l units of a good G_3, whose price is $p_a + p_b$.

Table 3.3: A closed submarket for Comb(l, p_a, p_b)

Part 1:	**Input:** l units of G_1 and G_2 at price p_a and p_b **Output:** l units of G_3 at price $p_a + p_b$ 3 Agents (A_1, A_2, A_3), 5 goods $(G_1, G_2, G_3, G_4, G_5)$ A_1: $W_{14} = 1$ and $U_1(X) = \min\{x_1, x_2\}$ A_2: $W_{23} = H$ and $U_2(X) = \min\{x_4, x_5\}$ A_3: $W_{35} = 1$ and $U_2(X) = x_3$
Part 2:	Closed submarkets for these linear relations $p_3 = p_a + p_b$ $p_5 = Hp_a + Hp_b - p_4$

Splitter (Spl(l, p_a, p_b)): The net endowment is l units each of two goods G_2 and G_3, whose prices are p_a and p_b, respectively. The net consumption is l units of Good 1, whose price is $p_a + p_b$.

Submarket construction for $p_a = p_b p_c$: Consider the submarket given in Table 3.5 and Figure 3.2. In this market, the 7 goods, $G_1, \ldots G_7$ are exclusive to the submarket. The prices of some goods are set using $(LIN.)$ relations as specified in Part 2. The submarket uses 2 Converters, 1 Combiner and 1 Splitter. Each device is specified by its (net endowment, net consumption). In

Table 3.4: A closed submarket for Spl(l, p_a, p_b)

Part 1:	**Input:** l units of G_1 at price $p_a + p_b$ **Output:** l units of G_2 and G_3 at price p_a and p_b 3 Agents (A_1, A_2, A_3), 5 goods $(G_1, G_2, G_3, G_4, G_5)$ A_1: $W_{14} = 1$ and $U_1(X) = x_1$ A_2: $W_{22} = W_{23} = H$ and $U_2(X) = \min\{x_4, x_5\}$ A_3: $W_{35} = 1$ and $U_2(X) = \min\{x_2, x_3\}$
Part 2:	Closed submarkets for these linear relation $p_2 = p_a$ $p_3 = p_b$ $p_5 = Hp_a + Hp_b - p_4$

Table 3.5: A closed submarket for $p_a = p_b p_c$

Part 1:	2 Agents (A_1, A_2), 2 Converters $(Conv_1, Conv_2)$, 1 Combiner $(Comb)$, 1 Splitter (Spl), and 7 Goods $(G_1, \ldots, G_7)$ A_1: $W_{11} = 1$ and $U_1(X) = x_4$ A_2: $W_{26} = 1$ and $U_2(X) = x_5$ $Conv_1 = Conv(1)$: (G_1, G_2) $Conv_2 = Conv(p_b)$: (G_6, G_7) $Comb(p_c, p_b, 1)$: $((G_2, G_7), G_3)$ $Spl(p_c, p_b, 1)$: $(G_3, (G_4, G_5))$
Part 2:	Closed submarkets for these linear relations $p_1 = p_c$ $p_2 = 1$ $p_4 = 1$ $p_5 = p_b$ $p_6 = p_a$ $p_7 = p_b$

addition to the agents needed for implementing these devices, the submarket requires 2 additional agents, A_1 and A_2.

Lemma 34. *The submarket of Table 3.5 enforces $p_a = p_b p_c$ and is closed under assumption $p_b \neq 0$.*

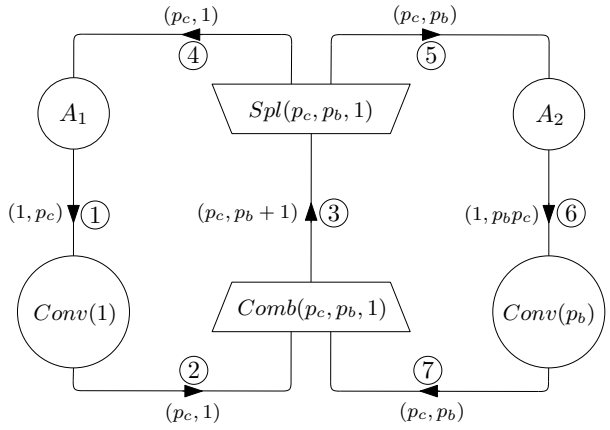


Figure 3.2: Flow of goods in Part 1 of Table 3.5. Wires are numbered, and wire i carries good G_i. The tuple on each wire represents (amount, price).

For each relation r of $R'(F)$, depending on its type, construct a closed submarket $\mathcal{M}_r$ as described in Section 3.3.1. Combine all the $\mathcal{M}_r$'s and add the agent of equation (3.6) to form one market $\mathcal{M}$.

Next we prove the main theorem of this section which will give all the desired hardness results.

Theorem 14. *Equilibrium prices of market $\mathcal{M}$, projected onto $(p_1, \ldots, p_n)$, are in one-to-one correspondence with the solutions of F. Furthermore, $size[\mathcal{M}] = poly(size[F])$.*

Theorem 14 shows that finding solutions of F can be reduced to finding equilibria of a Leontief exchange market. As discussed in Section 3.2.2, the problem of computing a NE of a 3-player game $\boldsymbol{A}$ can be formulated as finding a solution of system $F_{NE}(\boldsymbol{A})$ (3.2) of polynomials in which variables take values in $[0, 1]$ (Lemma 30). Note that $size[F_{NE}(\boldsymbol{A})] = O(size(\boldsymbol{A}))$. The next theorem follows using the formulation of (3.2), together with Lemma 30, and Theorems 12, 13 and 14.

Theorem 15. *Computing an equilibrium of an exchange market under Leontief utility functions is FIXP-hard. In particular, the corresponding Decision, (Strong) Approximation, and Partial Computation problems are hard for $FIXP_d$, $FIXP_a$ and $FIXP_{pc}$, respectively. Furthermore,*

checking existence of an equilibrium in an arbitrary Leontief exchange market (and in market with PLC utilities) is ETR-complete[4].

3.4 Membership in PPAD

In this section, we show that computing an approximate equilibrium in an exchange market with PLC utilities is in PPAD. This resolves an open question of [103].

The problem of computing an approximate market equilibrium under sufficiency conditions has been known to be PPAD-hard [25, 29, 36, 60] for more than a decade; however, membership in PPAD has not been established yet. The only fixed point formulation known for this problem was obtained in the context of proving membership in FIXP [48]. We use this formulation to show the result. Let $\mathcal{M}$ denote an exchange market with PLC utilities, and $size(\mathcal{M})$ denote the bit length of input parameters of $\mathcal{M}$. Let F be the fixed point formulation for $\mathcal{M}$ given in [48]. For this, we need to show the following:

1. F is Lipschitz continuous with constant K, where $size(K) = poly(size(\mathcal{M}))$.
2. δ-approximate fixed point of F gives an ϵ-approximate equilibrium, where $size(\delta) = poly(size(\epsilon, \mathcal{M}))$.

The first task is easy to show and it implies that finding an ϵ-approximate fixed point of F is in PPAD [89]. Showing the second step is quite involved and technical because F is rather intricate. For the second task, we need to show approximate market clearing of every good and approximately optimal bundle to each agent at an approximate fixed point of F.

There is a simple linear program (LP) that captures optimal bundles of each agent at a given price vector. Using this LP and its dual, F captures optimal bundles of each agent as feasibility of primal and dual constraints, and complementary slackness conditions. We show that at a δ-approximate fixed point of F, both feasibility and complementary slackness constraints are

[4]Membership in ETR is obtained by characterizing the set of equilibria as simultaneous solutions of a set of multivariate polynomial equations (refer to the full version for details).

approximately satisfied. This essentially proves that each agent gets an approximately optimal bundle.

Next we briefly describe the description of fixed point formulation F, given in [48]. Recall the market parameters from Section 3.2.1. Given prices $\boldsymbol{p}$, the optimal utility of agent i is a solution of the following LP, where variables $\boldsymbol{x} = \{x_{ij} \mid i \in \boldsymbol{\alpha}, j \in \mathcal{G}\}$ capture the assignment of goods to agents, and λ_i's and γ_k^i's are dual variables.

$$\begin{aligned} &\max\ u_i && &&\min\ \lambda_i \sum_j W_{ij}p_j + \sum_k T_i^k \gamma_k^i \\ &\forall k:\ u_i \le \sum_j U_{ij}^k x_{ij} + T_i^k && \xleftrightarrow{dual} && \forall j:\ \sum_k U_{ij}^k \gamma_i^k \le \lambda_i p_j \\ &\sum_j x_{ij}p_j \le \sum_j W_{ij}p_j && && \sum_k \gamma_i^k = 1 \\ &\forall j:\ x_{ij} \ge 0 && && \lambda_i \ge 0;\ \forall k:\ \gamma_k^i \ge 0 \end{aligned} \tag{3.7}$$

Without loss of generality, we may assume that the total initial endowment of every good is 1, i.e, $\sum_{i \in \boldsymbol{\alpha}} W_{ij} = 1, \forall j \in \mathcal{G}$. Let $m \stackrel{\text{def}}{=} |\boldsymbol{\alpha}|$, and $n \stackrel{\text{def}}{=} |\mathcal{G}|$. Let H denote the maximum number of hyperplanes in an agent's PLC utility function, and wlog we may assume that it is same for each agent. Let $\boldsymbol{\lambda} = \{\lambda_i \mid i \in \boldsymbol{\alpha}\}$ and $\boldsymbol{\gamma} = \{\gamma_i^k \mid i \in \boldsymbol{\alpha}, k \in [H]\}$. Let $x_{max} \stackrel{\text{def}}{=} 1.1$, $W_{min} \stackrel{\text{def}}{=} \min_{(i,j)} W_{ij}$, $U_{max} \stackrel{\text{def}}{=} \max_{(i,j,k)} U_{ij}^k$, $U_{min} \stackrel{\text{def}}{=} \min_{(i,j,k):U_{ij}^k > 0} U_{ij}^k$, $T_{max} \stackrel{\text{def}}{=} \max_{(i,k)} T_i^k$, and $\lambda_{max} \stackrel{\text{def}}{=} 2n^{(U_{max}+T_{max})/W_{min}}$. Note that $W_{min} > 0$ under sufficiency conditions.

Let $D \stackrel{\text{def}}{=} \{(\boldsymbol{p}, \boldsymbol{x}, \boldsymbol{\gamma}, \boldsymbol{\lambda}) \in \mathbb{R}_+^N \mid \sum_j p_j = 1;\ x_{ij} \le x_{max};\ \sum_k \gamma_i^k = 1;\ \lambda_i \le \lambda_{max}\}$, where N is the total number of variables, and $F : D \to D$ is a continuous function such that $(\overline{\boldsymbol{p}}, \overline{\boldsymbol{x}}, \overline{\boldsymbol{\gamma}}, \overline{\boldsymbol{\lambda}}) \stackrel{\text{def}}{=} F(\boldsymbol{p}, \boldsymbol{x}, \boldsymbol{\gamma}, \boldsymbol{\lambda})$ as given in Table 3.6.

Theorem 16 (Garg-Mehta-Vazirani (2014) [48])**.** *Assuming sufficient conditions of the existence of equilibrium, every fixed point of F gives an equilibrium of $\mathcal{M}$ and vice versa.*

Lemma 35. *F is Lipschitz continuous with constant K, where $size(K) = poly(size(\mathcal{M}))$.*

Table 3.6: FIXP circuit for exchange markets

$$\overline{p}_j = \frac{p_j + \max\{\sum_i x_{ij} - 1, 0\}}{\sum_j (p_j + \max\{\sum_i x_{ij} - 1, 0\})}$$

$$\overline{\gamma}_i^k = \frac{\gamma_i^k + \max\{u_i - \sum_j U_{ij}^k x_{ij} - T_i^k, 0\}}{\sum_k (\gamma_i^k + \max\{u_i - \sum_j U_{ij}^k x_{ij} - T_i^k, 0\})}$$

$$\overline{x}_{ij} = \min\left\{\max\left\{x_{ij} + \sum_k U_{ij}^k \gamma_i^k - \lambda_i p_j, 0\right\}, x_{max}\right\}$$

$$\overline{\lambda}_i = \min\left\{\max\left\{\lambda_i + \sum_j x_{ij} p_j - \sum_j W_{ij} p_j, 0\right\}, \lambda_{max}\right\}$$

3.4.1 Approximate Market Clearing

In this section, we show that market clears approximately at an approximate fixed point of F. Let $\boldsymbol{y} = (\boldsymbol{p}, \boldsymbol{x}, \gamma, \boldsymbol{\lambda})$. First we define ϵ-approximate fixed point of F.

Definition 12 (ϵ-Approximate Fixed Point)**.** $\boldsymbol{y}$ is an ϵ-approximate fixed point of F if $\|F(\boldsymbol{y}) - \boldsymbol{y}\|_\infty \leq \epsilon$.

Lemma 36. *At an ϵ-approximate fixed point $(\boldsymbol{x}, \boldsymbol{p}, \boldsymbol{\lambda}, \boldsymbol{\gamma})$ of F,*

- $|\sum_j x_{ij} p_j - \sum_j W_{ij} p_j| \leq \epsilon,\ \forall i \in \boldsymbol{\alpha}$
- $\sum_i x_{ij} \leq 1 + 5mn\sqrt{\epsilon n},\ \forall j \in \mathcal{G}$.

3.4.2 Approximately Optimal Bundle

In this section, we show that each agent gets an approximately optimal bundle at an approximate fixed point of F. We achieve this by showing that feasibility and complementary slackness of (3.7) are approximately satisfied.

Lemma 37. *At an ϵ-approximate fixed point $(\boldsymbol{x}, \boldsymbol{p}, \boldsymbol{\lambda}, \boldsymbol{\gamma})$ of F,*

- $\forall (i,j):$ *If* $x_{ij} > \epsilon$ *then* $\lambda_i p_j \leq \sum_k U_{ij}^k \gamma_k^i + \epsilon$
- $\forall (i,j):\ \lambda_i p_j \geq \sum_k U_{ij}^k \gamma_k^i - \epsilon$
- $\forall (i,k):\ u_i - \sum_j U_{ij}^k x_{ij} - T_i^k \ \leq\ 12nH^2 U_{max}^2 x_{max}\sqrt{\epsilon}$
- $\forall (i,k):$ *If* $\gamma_k^i > \sqrt[4]{\epsilon}$ *then*
  $u_i \geq \sum_j U_{ij}^k x_{ij} + T_i^k - 24nH^2 U_{max}^2 x_{max}(nx_{max} + \lambda_{max})\sqrt[4]{\epsilon}$.

Lemmas 36 and 37 give

Theorem 17. *At an* ϵ^8*-approximate fixed point of* F*, where* $\epsilon < 1/24mnH^2U_{max}^2 x_{max}(nx_{max}+\lambda_{max})$*,*

1. $\forall i:\ |\sum_j x_{ij}p_j - \sum_j W_{ij}p_j| \leq \epsilon$
2. $\forall j:\ \sum_i x_{ij} \leq 1 + \epsilon,\ \forall j$
3. $\forall i:\ u_i \leq \sum_j U_{ij}^k x_{ij} + T_i^k + \epsilon,\ \forall i$
4. $\forall (i,j):\ \lambda_i p_j \geq \sum_k U_{ij}^k \gamma_k^i - \epsilon,\ \forall (i,j)$
5. $\forall (i,j):$ *If* $x_{ij} > \epsilon$ *then* $\lambda_i p_j \leq \sum_k U_{ij}^k \gamma_k^i + \epsilon$
6. $\forall (i,k):$ *If* $\gamma_k^i > \epsilon$ *then* $u_i \geq \sum_j U_{ij}^k x_{ij} + T_i^k - \epsilon$.

Let u_i^1 be the optimal utility of (3.7). For a given ϵ, consider the following modified LP, where feasibility constraints of (3.7) are perturbed:

$$\begin{aligned} \max\ & u_i - \epsilon \sum_j x_{ij} \\ & \forall k:\ u_i \leq \sum_j U_{ij}^k x_{ij} + T_i^k + \epsilon \\ & \sum_j x_{ij} p_j \leq \sum_j W_{ij} p_j + \epsilon \\ & \forall j:\ x_{ij} \geq 0. \end{aligned} \tag{3.8}$$

The dual is

$$\begin{aligned}
\min\ \lambda_i \sum_j W_{ij} p_j + \sum_k \gamma_k^i (T_i^k + \epsilon) \\
\forall j:\ \sum_k U_{ij}^k \gamma_i^k \leq \lambda_i p_j + \epsilon \\
\sum_k \gamma_i^k = 1 \\
\lambda_i \geq 0;\ \forall k:\ \gamma_k^i \geq 0.
\end{aligned}$$

Let u_i^2 be the optimal utility of (3.8).

Lemma 38. $u_i^2 \geq u_i^1 - \epsilon n x_{max}, \forall i.$

Proof. Let $\dot{\boldsymbol{x}}$ and $\ddot{\boldsymbol{x}}$ be optimal solutions of (3.7) and (3.8) respectively. Since $\dot{\boldsymbol{x}}$ is a feasible point in (3.8), we have $u_i^2 - \epsilon \sum_j \ddot{x}_{ij} \ \geq\ u_i^1 - \epsilon \sum_j \dot{x}_{ij}$, that implies $u_i^2 - u_i^1 \ \geq\ \epsilon(\sum_j \dot{x}_{ij} - \sum_j \ddot{x}_{ij}) \ \geq\ \epsilon n x_{max}$. □

Suppose we have a candidate point $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{p}}, \tilde{\boldsymbol{\lambda}}, \tilde{\boldsymbol{\gamma}})$, which satisfies all feasibility constraints of (3.8) but approximately satisfies complementary slackness constraints as follows:

$$\begin{aligned}
\forall (i,j):\ \text{ If } \tilde{x}_{ij} > \epsilon \text{ then } \tilde{\lambda}_i \tilde{p}_j \leq \sum_k U_{ij}^k \tilde{\gamma}_i^k + \epsilon \\
\forall (i,k):\ \text{ If } \tilde{\gamma}_i^k > \epsilon \text{ then } u_i^3 \geq \sum_j U_{ij}^k \tilde{x}_{ij} + T_i^k - \epsilon \\
\forall i:\ \text{ If } \tilde{\lambda}_i > \epsilon \text{ then } \sum_j \tilde{x}_{ij} \tilde{p}_j \geq \sum_j W_{ij} \tilde{p}_j - \epsilon,
\end{aligned}$$

where u_i^3 is the utility obtained at this point. At prices $\boldsymbol{p} = \tilde{\boldsymbol{p}}$, let u_i^1 and u_i^2 be the optimal value of (3.7) and (3.8) respectively. Next we show that $\tilde{\boldsymbol{x}}$ gives an approximately optimal utility at $\tilde{\boldsymbol{p}}$.

Lemma 39. $\forall i:\ \ u_i^3 \ \geq\ u_i^1 - \epsilon(2 + 2\lambda_{max} + 4nx_{max} + HT_{max} + nHx_{max}U_{max})$.

Finally, using Theorem 17 and Lemma 39, we get

Theorem 18. *At an ϵ^{16}-approximate fixed point $(\boldsymbol{x}, \boldsymbol{p}, \boldsymbol{\gamma}, \boldsymbol{\lambda})$ of F, where $\epsilon < 1/(24mn^2H^2U_{max}^2x_{max}^2\lambda_{max}T_{max})$, we have*

(i) $\forall i:\ \sum_j x_{ij}p_j \leq \sum_j W_{ij}p_j + \epsilon$

(ii) $\forall j:\ \sum_i x_{ij} \leq 1 + \epsilon$

(iii) $\forall i:\ u_i \geq u_i^{opt} - \epsilon$,

where u_i^{opt} is the optimal utility of agent i at prices $\boldsymbol{p}$.

Lemma 40. *For an ϵ, if we have a solution $(\boldsymbol{x}, \boldsymbol{p})$ such that*

(i) $\forall i:\ \sum_j x_{ij}p_j \leq \sum_j W_{ij}p_j + \epsilon$

(ii) $\forall j:\ \sum_i x_{ij} \leq 1 + \epsilon$

(iii) $\forall i:\ u_i \geq u_i^{opt} - \epsilon$,

then it gives an ϵ'-approximate equilibrium, where $\epsilon' = U_{min}W_{min}\epsilon$.

From Theorem 18, Theorem 16 and Lemma 40, we get

Theorem 19. *Assuming sufficiency conditions for the existence of equilibrium, for any $0 < \epsilon < 1$, an ϵ-approximate equilibrium of exchange markets with piecewise-linear concave utilities can be obtained from a δ-approximate fixed point of F, where $\delta = (\frac{\epsilon}{\Lambda})^{16}$ and $\Lambda = \frac{24mn^2H^2U_{max}^2x_{max}^2\lambda_{max}T_{max}}{W_{min}U_{min}}$.*

Note that finding an ϵ-approximate fixed point of a Lipschitz-continuous function from a convex compact domain to itself is in PPAD [89]. Using Lemma 35 and Theorem 19, together with [89], we get

Theorem 20. *Assuming sufficiency conditions for the existence of equilibrium, finding an approximate equilibrium in exchange markets with PLC utilities is in PPAD.*

Since finding an n^{-13}-approximate equilibrium of SPLC markets is PPAD-hard [25], we get

Theorem 21. *Finding an n^{-13}-approximate equilibrium of exchange markets with PLC utilities is PPAD-complete, where number of agents and goods are $O(n)$.*

3.5 Leontief Utilities under Constant Number of Agents

In this section, we show that there is a polynomial time algorithm for finding an equilibrium in Arrow-Debreu exchange markets under Leontief utility functions provided the number of agents is a constant. This settles part of an open problem of Devanur and Kannan [38]. Consider a Leontief exchange market with n goods and d agents, where d is a constant. The Leontief utility function of agent i is given by

$$U_i(\boldsymbol{x}_i) = \min_{j=1}^{n} \left\{ \frac{x_{ij}}{A_{ij}} \right\},$$

where $A_{ij} \geq 0$ is the fraction of good j that agent i wants. Let W_{ij} be the amount of good j agent i owns. We assume wlog that $\sum_i W_{ij} = 1,\ \forall j$. Let us capture the equilibrium utility of agent i in variable β_i, then the optimal bundle condition gives,

$$x_{ij} = A_{ij}\beta_i \tag{3.9}$$

at an equilibrium. Further, if $(p_1, \ldots, p_n)$ are corresponding equilibrium prices, then the market clearing conditions can be written as,

$$\begin{aligned} \forall j \in \mathcal{G}: \qquad & \textstyle\sum_i x_{ij} = \sum_i A_{ij}\beta_i \leq 1 \\ & \text{if } p_j > 0 \text{ then } \textstyle\sum_i A_{ij}\beta_i = 1\ . \end{aligned} \tag{3.10}$$

Further, since Leontief utility function is non-satiated (given any bundle, there exists another bundle where utility increases), the agents will spend all of their earned money. This gives the following relation in $\boldsymbol{\beta}$ and $\boldsymbol{p}$:

$$\forall i \in \boldsymbol{\alpha}:\ \sum_j p_j x_{ij} = \sum_j W_{ij} p_j \Rightarrow \beta_i = \frac{\sum_j W_{ij} p_j}{\sum_j A_{ij} p_j}\ .$$

First, we show that if there is an equilibrium, then there is one where at most d prices are

non-zero.

Lemma 41. *If an exchange market with Leontief utilities has an equilibrium, then there is one where at most d goods have non-zero prices.*

Due to Lemma 41, to find an equilibrium, it suffices to check for every set S of d goods if there is an equilibrium by setting the prices of goods outside S to zero. And this can be achieved by checking the feasibility of the following system, where β_is and p_js are variables.

$$\begin{aligned} &\forall j \notin S,\ p_j = 0; \quad \forall j \in S,\ p_j \geq 0. \\ &\forall j \notin S,\ \textstyle\sum_i A_{ij}\beta_i \leq 1; \quad \forall j \in S,\ \sum_i A_{ij}\beta_i = 1 \\ &\forall i,\ \beta_i = \frac{\sum_j W_{ij}p_j}{\sum_j A_{ij}p_j} \end{aligned} \tag{3.11}$$

Lemma 42. *If $\exists \boldsymbol{\beta}^*,\ \boldsymbol{p}^*$ satisfying (3.11) then they constitute an equilibrium.*

Proof. Let $x^*_{ij} = \beta^*_i A_{ij}$, then for every agent i we get

$$\sum_j x^*_{ij} p^*_j = \beta^*_i \sum_j A_{ij} p^*_j = \sum_j W_{ij} p^*_j \ ,$$

using the third condition of (3.11). This together with the fact that $x^*_{ij} = \beta^*_i A_{ij}$ it follows that X^*_i is an optimal bundle of agent i at prices $\boldsymbol{p}^*$. Market clearing for goods follows from the first two conditions of (3.11). □

Note that system (3.11) remains unchanged if we remove price variables that are set to zero. Then it will have $2d$ variables, The first two conditions are linear in these variables, while the third condition is of degree two. Since d is a constant, checking non-emptiness of (3.11) can be done in polynomial time [5, 6, 38].

If (3.11) turns out to be non-empty then by Lemma 42 we get an equilibrium. Lemma 41 implies that we need to check feasibility of this system for every subset of goods of size d. There are at most $\binom{n}{d} \leq n^d$ such systems need to be checked, which is a polynomial in number because

d is a constant. Therefore, overall we can find an equilibrium in polynomial time, and the next theorem follows:

Theorem 22. *Consider an Arrow-Debreu exchange market under Leontief utility functions in which the number of agents is a constant. Then, in polynomial time we can determine if an equilibrium exists, and if so, we can find one.*

3.6 Discussion

Is computing an equilibrium for a Fisher market under PLC utilities FIXP-hard? Clearly the problem is in FIXP since Fisher markets are a subcase of Arrow-Debreu markets. We believe that existing techniques, for example of [103] establishing hardness for Fisher markets under SPLC utilities via reduction from Arrow-Debreu markets, will not work and new ideas are needed. As stated in Section 3.0.1, finding an approximate equilibrium under CES utilities was also shown to be PPAD-complete [24]. Is computing an exact equilibrium FIXP-complete?

In economics, uniqueness of equilibria plays an important role. In this vein, we ask what is the complexity of deciding if a PLC or Leontief market has more than one equilibria. We note that the reduction given in this chapter blows up the number of equilibria and hence it will not answer this question in a straightforward manner.

CHAPTER 4

$\exists\mathbb{R}$-COMPLETENESS FOR MULTI-PLAYER NASH EQUILIBRIA

As a result of a series of important works [22, 32, 33, 56, 89], the complexity of 2-player Nash equilibrium is by now well understood, even when equilibria with special properties are desired and when the game is symmetric. However, for multi-player games, when equilibria with special properties are desired, the only result known is due to Schaefer and Štefankovič [97]: that checking whether a 3-player Nash Equilibrium (3-Nash) instance has an equilibrium in a ball of radius half in l_∞-norm is $\exists\mathbb{R}$-complete, where $\exists\mathbb{R}$ is the class of decision problems which can be reduced in polynomial time to Existential Theory of the Reals.

In this chapter, we show that the following decision versions of 3-Nash are also $\exists\mathbb{R}$-complete: checking whether (i) there are two or more equilibria, (ii) there exists an equilibrium in which each player gets at least h payoff, where h is a rational number, (iii) a given set of strategies are played with non-zero probability, and (iv) all the played strategies belong to a given set.

Next, we give a reduction from 3-Nash to symmetric 3-Nash, hence resolving an open problem of Papadimitriou [90]. This yields $\exists\mathbb{R}$-completeness for symmetric 3-Nash for the last two problems stated above as well as completeness for the class FIXP$_a$, a variant of FIXP for strong approximation. All our results extend to k-Nash, for any constant $k \geq 3$.

4.1 Technical Overview

We first give the main idea behind our reduction from 3-Nash to symmetric 3-Nash (Theorem 35). We will reduce the given game (A, B, C), where each tensor is of size $m \times n \times p$, to a symmetric game, D, of size $l \times l \times l$, where $l = m + n + p$ (see Section 4.2.1 for the description of (symmetric) games). In this game, under each symmetric NE, the strategy of each player can

be decomposed into three vectors, say $X, Y, \boldsymbol{z}$, of dimension m, n, p, respectively. An essential condition for recovering a Nash equilibrium for the original game (A, B, C) is that each of these three vectors be non-zero; this is also the most difficult part of the reduction.

To achieve this we construct a $3 \times 3 \times 3$ symmetric game G all of whose symmetric NE are of full support, even though it is only partially specified (see (4.6)). We "blow up" G to derive D, which is of size $l \times l \times l$, and the unspecified entries of G create room where tensors A, B, C are "inserted". Now, if $(X, Y, \boldsymbol{z})$ is a symmetric NE of D then so is $(\sum_i x_i, \sum_j y_j, \sum_k z_k)$ of G. As a result, each vector, $X, Y, \boldsymbol{z} \neq 0$. Next we show that if these vectors are scaled to probability vectors, they form a NE for (A, B, C). Additional arguments yield $\exists\mathbb{R}$-completeness for **Subset** and **Superset** for symmetric k-Nash (Theorems 36 and 37).

Next we give the idea for showing that symmetric 3-Nash is complete for the class FIXP$_a$ (Theorem 39). Note that we are unable to show that symmetric 3-Nash is complete for the class FIXP itself, since we don't see how to express the solution to the given instance (A, B, C) as a rational linear projection of the solution of the reduced symmetric game D, a requirement of FIXP reductions [68].

Under FIXP$_a$, given an instance I and a rational $\epsilon > 0$, we need to compute a vector X that is within (additive) ϵ distance from some solution, i.e., $\exists X^* \in Sol(I)$ such that $|X^* - X|_\infty \leq \epsilon$, in time polynomial in $size[I]$ and $\log(1/\epsilon)$. In the above reduction, obtaining a solution of (A, B, C) involves e.g., dividing X by $\sum_i x_i$. If the latter is very small, this may give us a vector that is very far away from a solution of (A, B, C), even though x may be close to a solution of D.

We get around this problem by a small change in the above reduction, namely, we need to multiply the tensors A, B, C by a small constant ϵ' before they are "inserted" at the appropriate places in G to get symmetric game D. This ensures that vector $(\sum_i x_i, \sum_j y_j, \sum_k z_k)$ is approximately $(1/3, 1/3, 1/3)$. As a result, given a point close to a solution of D, we can get a point "close" to a solution of (A, B, C).

Next, we describe how we show $\exists\mathbb{R}$-completeness for the four decision problems, mentioned

in the previous section, for k-Nash. To show hardness in case of 3-players, we reduce **InBox**, which is known to be $\exists\mathbb{R}$-complete for 3-Nash [97], to each of **MaxPayoff**, **Subset** and **Superset**, and then from **MaxPayoff** to **NonUnique**. Hardness for the k-Nash, $k > 3$, follows since 3-Nash reduces to k-Nash trivially by introducing dummy players. To show containment in $\exists\mathbb{R}$ we give a non-linear complementarity problem (NCP) formulation that exactly captures NE of a given game (Theorems 23 and 24).

Next, we briefly explain the reduction from **InBox** to **MaxPayoff** for the 2-player case (see Section 4.4.1 for details); 3-player case is an extension of it (Section 4.4.2). Let the given game be represented by two payoff matrices (A, B) of size $m \times n$, one for each player. The **InBox** problem is to check if it has a NE in which all strategies are played with at most 0.5 probability. We reduce it to checking if another game (C, D) has a NE in which every player gets payoff at least $h > 0$ (**MaxPayoff**). Without loss of generality (wlog) we can assume that $A, B > 0$.

We construct $m(n+1) \times n(m+1)$ matrices C and D, where the top-left block is set to $A + h$ and $B + h$ respectively. This ensures that if each player gets payoff h at a NE, then strategies from this block are played with non-zero probability, and normalizing them gives a NE of (A, B). The latter follows since NE set remains invariant under additive scaling of payoffs. In order to retrieve a NE in 0.5 ball, we ensure that if any of these strategies is (relatively) played with more than 0.5 probability then a sequence of deviations leads to both players playing only among their last mn strategies where payoff is zero ($< h$).

In particular suppose the second player plays Y in the top-left block. The last mn strategies of the row player are divided into n blocks of size m, one for each $y_j,\ j \leq n$ such that if $y_j > 0.5$ then best response of the first player is to deviate to j^{th} block. The payoff of the second player is set to -1 in these blocks, so then y_j fetches -1 and second player is forced to deviate to her last mn strategies where both get zero. Similarly for the first player.

Organization: In Section 4.2 we formally define the (symmetric) k-Nash problem, their decision problems, and discuss the complexity classes $\exists\mathbb{R}$ and FIXP. Membership in $\exists\mathbb{R}$ for decision prob-

lems in (symmetric) k-Nash is shown in Section 4.3. In Section 4.4, we show that decision problems in k-player games, for any constant $k \geq 3$, are $\exists\mathbb{R}$-complete. $\exists\mathbb{R}$-completeness of decision problems in *symmetric* 3-Nash is shown in Section 4.5. In Section 4.6, we show that computing an equilibrium in symmetric 3-Nash is FIXP$_a$-complete. Since symmetric 3-Nash does not trivially reduce to symmetric k-Nash, we extend the $\exists\mathbb{R}$ and FIXP$_a$-completeness results for the latter in Section 4.7 for any constant $k \geq 3$.

4.2 Preliminaries

In this section we formally define the (symmetric) k-Nash problem, and their decision problems. Further, we discuss the complexity classes $\exists\mathbb{R}$ and FIXP.

Notations: Vectors are represented in bold-face letters, and i^{th} coordinate of vector X is denoted by x_i, and X^{-i} denotes the vector X with i^{th} coordinate removed. $\mathbf{1}$ and $\mathbf{0}$ represent all ones and all zeros vector respectively of appropriate dimension. For integers $k < l$, $X(k : l) = (x_k, x_{k+1}, \ldots, x_l)$. We use $[n]$ to denote set $\{1, \ldots, n\}$ and $[k : l]$ to denote $\{k, k+1, \ldots, l\}$. If X is of m dimensional, then by $\sigma(X)$ we mean $\sum_{i=1}^{m} x_i$, and $\eta(X) = {}^{X}/_{\sigma(X)}$. Concatenation of vectors X and Y is denoted by $(X|Y)$. Given a matrix A and $h \in \mathbb{R}$, $A + h$ denotes the matrix A with h added to each of its entries. Further, $A(i, :)$ is its i^{th} row and $A(:, j)$ is its j^{th} column.

4.2.1 (Symmetric) k-Nash

For a given k-player game let $S_i, i \in [k]$ be the set of pure strategies of player i, and let $\boldsymbol{S} = \times_{i\in[k]} S_i$. The payoffs of player i can be represented by a k-dimensional tensor A_i, such that $A_i(\boldsymbol{s})$ denotes the payoff she gets when $\boldsymbol{s} \in \boldsymbol{S}$ is played. Players may randomize among their strategies. Let Δ_i denote the set of mixed strategy profiles of player i, and let $\boldsymbol{\Delta} = \times_{i\in[k]} \Delta_i$. Expected payoff of player i from $X = (X^1, \ldots, X^k) \in \boldsymbol{\Delta}$ is $\pi_i(X) = \sum_{\boldsymbol{s}\in\boldsymbol{S}} (\Pi_{i\in[k]} X^i_{s_i}) A_i(\boldsymbol{s})$.

Definition 13. (Nash Equilibrium (NE) [84]) $X \in \mathbf{\Delta}$ is said to be a NE if no player gains by unilateral deviation. Formally, $\forall i,\ \forall X'^i \in \Delta_i,\ \pi_i(X) \geq \pi_i(X'^i, X^{-i})$.

Let $\pi_i(s, X^{-i})$ denote the payoff i receives when she plays $s \in S_i$ and others play as per X^{-i}. It is easy to see that X is a NE iff [84]

$$\forall i \in [k],\ \ \forall s \in S_i,\ \ x_s^i > 0 \ \ \Rightarrow \ \ \pi_i(s, X^{-i}) = \max_{t \in S_i} \pi_i(t, X^{-i})\ . \tag{4.1}$$

Symmetric k-Nash: In a symmetric game the players are indistinguishable. Their strategy sets are identical (S) and payoffs are symmetric represented by one tensor A. For a player, the payoff she gets by playing $s' \in S$, when others are playing $\boldsymbol{s} \in S^{(k-1)}$, is $A(s', \boldsymbol{s})$. Further, who is playing what in $\boldsymbol{s}$ does not matter. Formally, A satisfies $A(s', \boldsymbol{s}) = A(s', \boldsymbol{s}_\tau)$ for all permutations τ of $(1, \ldots, k-1)$, where $\boldsymbol{s}_\tau$ is the corresponding permuted vector.

A profile $X \in \mathbf{\Delta}$ is called *symmetric* if $X^i = X^j,\ \forall i, j \in [k]$, thus one vector $X \in \Delta$ is enough to denote a symmetric profile. At a symmetric strategy profile all the players get the same payoff, and we denote it by $\pi(X)$. The problem of computing a symmetric NE (SNE) of a symmetric game is called *symmetric k-Nash*.

Note that the description of a (symmetric) k-player game takes $O(km^k)$ space, where $m = \max_i |S_i|$, which is exponential in m and k. To keep it polynomial, we consider k as a constant. Further, wlog $(A_1, \ldots, A_k) > 0$ because adding a constant to the tensors does not change the set of NE.

2-Nash: The payoff tensors in case of 2-player game are matrices, say (A, B), A for player one and B for player two. If the first player plays i and second plays j, then their respective payoff are A_{ij} and B_{ij}. Game is said to be symmetric if $B = A^T$. A mixed strategy is $(X, Y) \in \Delta_1 \times \Delta_2$, and respective payoffs at such a strategy are $X^T AY$ and $X^T BY$. 2-Nash is the problem of finding

a Nash equilibrium of such a game, i.e., strategy (X, Y) such that

$$X^T AY \geq X'^T AY, \ \forall X' \in \Delta_1 \quad \text{and} \quad X^T BY \geq X^T BY', \ \forall Y' \in \Delta_2 \ .$$

The NE characterization of (4.1) reduces to:

$$\forall i \in S_1, \ x_i > 0 \Rightarrow (AY)_i = \max_{k \in S_1} (AY)_k; \ \forall j \in S_2, \ y_j > 0 \Rightarrow (X^T B)_j = \max_{k \in S_2} (X^T B)_k \ . \quad (4.2)$$

3**-Nash:** It is the k-Nash problem with $k = 3$ players. We will represent such a game by three 3-dimensional tensors (A, B, C); A for player one, B for player two, and C for player three. If player one plays i, two plays j and three plays k, then their respective payoffs are A_{ijk}, B_{ijk}, and C_{ijk}. If the game is symmetric then we have $A_{ijk} = A_{ikj} = B_{jik} = B_{kij} = C_{jki} = C_{kji}$. A mixed strategy is denoted by $(X, Y, \boldsymbol{z}) \in \Delta_1 \times \Delta_2 \times \Delta_3$. Thus NE characterization of (4.1) reduces to:

$$\begin{array}{rcl} \forall i \in S_1, \ x_i > 0 & \Rightarrow & \sum_{j \in S_2, k \in S_3} A_{ijk} y_j z_k = \max_{l \in S_1} \sum_{j \in S_2, k \in S_3} A_{ljk} y_j z_k \\ \forall j \in S_2, \ y_j > 0 & \Rightarrow & \sum_{i \in S_1, k \in S_3} B_{ijk} x_i z_k = \max_{l \in S_2} \sum_{i \in S_1, k \in S_3} B_{ilk} x_i z_k \\ \forall k \in S_3, \ z_k > 0 & \Rightarrow & \sum_{i \in S_1, j \in S_2} C_{ijk} x_i y_j = \max_{l \in S_3} \sum_{i \in S_1, j \in S_2} C_{ijl} x_i y_j \ . \end{array} \quad (4.3)$$

Decision Problems: Computational complexity of numerous decision problems have been studied for 2-Nash and 3-Nash [32, 56]. In this thesis, we consider the following:

- **NonUnique**: Does there exist more than one NE?
- **MaxPayoff**: Given a rational number h, does there exist a NE where every player gets payoff at least h?
- **Subset**: Given sets $T_i \subset S_i, \ \forall i \in [k]$, does there exist a NE where every strategy in T_i is played with positive probability by player i?
- **Superset**: Given sets $T_i \subset S_i, \ \forall i \in [k]$, does there exist a NE where all the strategies outside

T_i are played with zero probability by player i?

- **InBox**: Does there exists a NE where every strategy is played with probability less than or equal 0.5?

All but last problem have been shown to be NP-complete in case of 2-Nash [32, 56], and the last one is shown to be $\exists\mathbb{R}$-complete in case of 3-Nash [97]. In this chapter, we show $\exists\mathbb{R}$-completeness for the first four decision problems for k-Nash, and for third and fourth for symmetric k-Nash, where $k \geq 3$ in both cases.

We refer the reader to Section 4.4 and Section 3.2.3 for definition of existential theory of reals and the class of FIXP and its variant FIXP$_A$

4.3 (Symmetric) k-Nash: Containment in $\exists\mathbb{R}$

In this section we show that the first four decision problems described in Section 4.2.1 are in $\exists\mathbb{R}$, for k-Nash as well as symmetric k-Nash. For a k-player game $(A_1, \ldots, A_k)$, NE characterization of (4.1) can be reformulated as a set of polynomial inequalities as follows, where variable x_s^i captures the probability with which player i plays $s \in S_i$, and variable λ_i captures her best payoff. Recall function $\pi_i(s, X^{-i})$ from Section 4.2.1 representing payoff of player i when she plays $s \in S_i$ and others play as per X^{-i}.

$$\forall i \in [k],\ \ \forall s \in S_i,\ \ x_s^i \geq 0;\ \ \pi_i(s, X^{-i}) \leq \lambda_i;\quad x_s^i(\pi_i(s, X^{-i}) - \lambda_i) = 0;\quad \sum_{s \in S_i} x_s^i = 1\ . \quad (4.4)$$

It is easy to see that strategy profile $X \in \mathbf{\Delta}$ satisfies (4.1) if and only if it satisfies (4.4).

Theorem 23. *Given a k-player game $(A_1, \ldots, A_k)$, for a constant k, the problems of* **NonUnique**, **MaxPayoff**, **Subset** *and* **Superset** *are in* $\exists\mathbb{R}$.

Proof. To frame **NonUnique** as an $\exists\mathbb{R}$ problem, take two copies of (4.4) each with different sets of variables, say X and Y, and add $|X - Y|^2 > 0$ to it. This system has a feasible solution (X, Y)

if and only if the game has two NE $X \neq Y$. Thus, containment of **NonUnique** in $\exists\mathbb{R}$ follows.

For **MaxPayoff**, add $\forall i \in [k],\ \pi_i(X) \geq h$ to the system (4.4). It has a feasible solution X if and only if X is a NE of the game where payoff received by every player is at least h, implying **MaxPayoff** is in $\exists\mathbb{R}$.

Similarly, to formulate **Subset**, add $\forall i \in [k],\ \forall s \in T_i,\ x_s^i > 0$ to (4.4). And for **Superset**, add $\forall i \in [1:k],\ \forall s \in S_i \setminus T_i,\ x_s^i = 0$ to (4.4). □

Given a symmetric game A, the following system of polynomial inequalities (similar to (4.4)) exactly captures its symmetric NE, where variable x_s captures the probability of playing strategy $s \in S$ and λ captures the payoff.

$$\forall s \in S,\ \ x_s \geq 0;\ \ \pi(s, X) \leq \lambda;\quad x_s(\pi(s, X) - \lambda) = 0 \ \text{ and } \ \sum_s x_s = 1\ .$$

The proof for the next theorem follows similar to that of Theorem 23.

Theorem 24. *Given a symmetric k-player game A, for a constant k, the problems of* **NonUnique***,* **MaxPayoff***,* **Subset** *and* **Superset** *for symmetric NE are in* $\exists\mathbb{R}$.

4.4 k-Nash: $\exists\mathbb{R}$-completeness for Decision Problems

In this section we show that **MaxPayoff**, **Subset**, **Superset** and **NonUnique** are $\exists\mathbb{R}$-complete in k-player games, for any constant $k \geq 3$. Containment in $\exists\mathbb{R}$ follows from Theorem 23 from Section 4.3. We show $\exists\mathbb{R}$-hardness for these four decision problems in case of 3-player games next, and since 3-player game trivially reduces to k-player game, for $k > 3$, by adding $k - 3$ dummy players with one strategy each, the result will follow for the latter as well. To show hardness for **MaxPayoff**, **Subset** and **Superset** we reduce from **InBox** (in Section 4.4.1), and for **NonUnique** we reduce from **MaxPayoff** (in Section 4.4.3).

4.4.1 $\exists\mathbb{R}$-hardness: **InBox** to **MaxPayoff**, **Subset** and **Superset**

To convey the main ideas, we first describe the reduction in 2-player games and later generalize it to the 3-player case in Section 4.4.2. We show the reduction from **InBox** to **MaxPayoff**, and from the intermediate lemmas, reduction to **Subset** and **Superset** will follow. Let the given two player game be represented by $m \times n$ dimensional payoff matrices $(A, B) > 0$.

For $a \geq 0$, let $\mathcal{B}_a = [0,\ a]^{m+n}$ be a ball of radius a at origin in l_∞ norm. We will construct another game (C, D), with $m(n+1) \times n(m+1)$-dimensional matrices, and show that it has a NE where each player gets at least $h > 0$ payoff (**MaxPayoff**) if and only if the game (A, B) has a NE in $\mathcal{B}_{0.5}$ (**InBox**). First we define a couple of notations required for the construction.

Definition 14. Let i and j be integers where $i \in [m]$ and $j \in [n]$, and h be a real number. We define the following operators:

$A_{(i,:)+h}$: matrix A with h added to the entries in its i^{th} row, and

$A_{(:,j)+h}$: matrix A with h added to the entries in its j^{th} column.

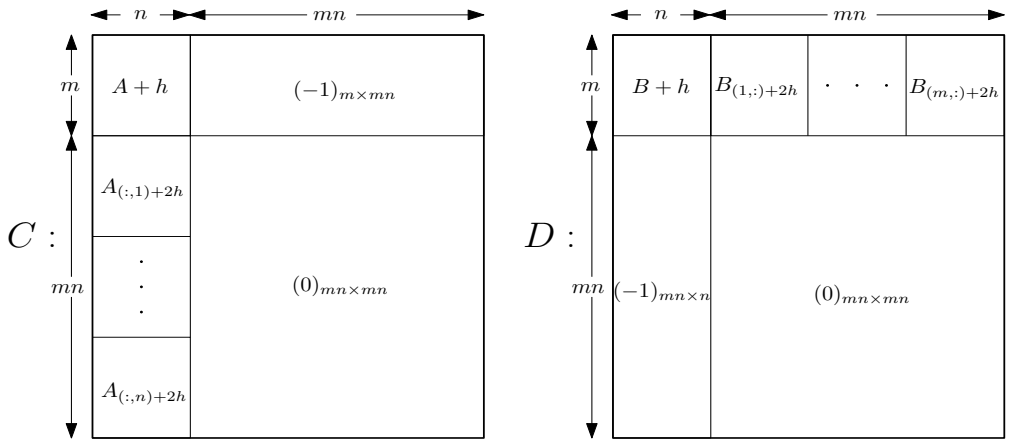
Definition 15. Given a matrix M of size $a \times b$ and integers r and s such that $a + r - 1 \leq m(n+1)$ and $b + s - 1 \leq n(m+1)$, define $[M]_{r,s}$ to be an $m(n+1) \times n(m+1)$-dimensional matrix where M is copied starting at position (r, s), and all other coordinates are set to zero.

Using the above notations we construct matrices C and D as follows, where $h > 0$.

$$C = [A + h]_{1,1} + [(-1)_{m \times mn}]_{1,n+1} + \textstyle\sum_{j \in [n]} [A_{(:,j)+2h}]_{jm+1,1} \text{ , and}$$
$$D = [B + h]_{1,1} + [(-1)_{mn \times n}]_{m+1,1} + \textstyle\sum_{i \in [m]} [B_{(i,:)+2h}]_{1,in+1} \text{ .}$$

The next lemma follows from the construction of C, D. Recall that $\sigma(X) = \sum_i x_i$.

Lemma 43. *Given a strategy (X', Y') of game (C, D), let $X = X'(1 : m)$, $Y = Y'(1 : n)$,*



$\alpha = h * \sigma(Y) - \sigma(Y'(n+1:(m+1)n))$, *and* $\beta = h * \sigma(X) - \sigma(X'(m+1:(n+1)m))$. *Then,*

$$(CY')_i = \begin{cases} \alpha + (AY)_i & \textit{if } i \in [m] \\ 2hy_{\lfloor (i-1)/m \rfloor} + (AY)_r & \textit{if } i \in [m+1, m(n+1)],\ r = ((i-1)\ mod\ m) + 1. \end{cases}$$

$$(X'^T D)_j = \begin{cases} \beta + (X^T B)_j & \textit{if } j \in [n] \\ 2hx_{\lfloor (j-1)/n \rfloor} + (X^T B)_r & \textit{if } j \in [n+1, n(m+1)], r = ((j-1)\ mod\ n) + 1. \end{cases}$$

Before the formal reduction, here is a brief intuition. Note that in (C, D) we have copied $(A + h, B + h)$ in the top-left $m \times n$ block, we call it *first block* now on. Since adding a constant does not change NE of a game, if strategies from only the first block are played with non-zero probability at a NE of (C, D), then they give a NE of (A, B) as well. Also, the payoff achieved at such a NE are at least h, a solution of **MaxPayoff**, using Lemma 43.

To guarantee a NE in $\mathcal{B}_{0.5}$ for game (A, B) (solution of **InBox**), we make use of the blocks added after the first block in both the directions. In particular, in Lemma 43, if $\exists j \in [n],\ y_j > 0.5 * \sigma(Y)$, then for the first player her first m strategies are worse than those from block $[mj + 1 : mj + m]$, forcing her to play only from her last mn strategies. This will force the second player to move away from the first block too (or else he gets negative payoff), and thereby leading to a NE where both play from the last mn strategies and both get zero payoff – also not a solution of

MaxPayoff. We will use these observations crucially in the reduction.

We show that solutions of **InBox** in game (A, B), *i.e.,* (X, Y) such that $X, Y \leq 0.5$, are retained as NE of (C, D). The proof uses the fact that in C and D, top-left block encodes A and B respectively.

Lemma 44. *(A, B) has a NE $(X, Y) \in \mathcal{B}_{0.5}$ iff $(X', Y') = ((X, 0_{mn}), (Y, 0_{mn}))$ is a NE of (C, D).*

Proof. To prove forward direction, it suffices to check if strategy profile (X', Y') satisfies (4.2) for game (C, D). We show the conditions for the first player, namely involving C, and proof for the second player follows similarly. As last mn strategies in Y' are not played at all, we have $\alpha = h \sum_{j \in [n]} y_i - \sum_{j \in [n+1, n(m+1)]} y'_j = h * 1 - 0 = h$. This together with Lemma 43 gives

$$i \in [m],\ (CY')_i = h + (AY)_i \quad \Rightarrow \quad \max_{i \in [m]} (CY')_i = h + \max_{i \in [m]} (AY)_i \ .$$

For $i \in [m+1, m(n+1)]$, let $r = ((i-1) \mod m) + 1$ and $k = \lfloor (i-1)/m \rfloor$. Then using Lemma 43 and the fact that $y_k \leq 0.5$, we have

$$(CY')_i \leq 2h(0.5) + (AY)_r = h + (AY)_r = (CY')_r \ .$$

In other words strategies $[1 : m]$ give at least as much payoff as the rest. Since (X, Y) is a NE of game (A, B), if $x'_i = x_i > 0$ then $(CY')_i = h + (AY)_i = h + \max_{k \in [m]} (AY)_k = \max_{k \in [m(n+1)]} (CY')_k$.

For the reverse direction, $\exists i \in [m]$ s.t. $x'_i > 0$ and hence $\forall j \in [n],\ (CY')_i \geq (CY')_{mj+i} \Rightarrow 2hy_j \leq h \Rightarrow y_j \leq 0.5$. Similarly $X \leq 0.5$ follows. □

Lemma 44 maps a solution of **InBox** in game (A, B) to a NE of (C, D) where players play only among their first m, n strategies respectively. Clearly, at such a NE both the players in game (C, D) get at least h payoff, therefore it is also a solution of **MaxPayoff** in (C, D). Next we show

a reverse mapping: a NE of (C, D) where both players play some of first m, n strategies, gives a NE of game (A, B). Recall that for vector X, $\eta(X) = {}^{X}/_{\sigma(X)}$.

Lemma 45. *If (X', Y') is a NE of game (C, D) s.t. $X = X'[1 : m]$ and $Y = Y'[1 : n]$ are non zero, then $(\eta(X), \eta(Y))$ is a NE for game (A, B), and $(\eta(X), \eta(Y)) \in \mathcal{B}_{0.5}$.*

Proof. As $\sigma(X), \sigma(Y) > 0$, to show $(\eta(X), \eta(Y))$ is a NE of (A, B) it suffices to show the following.

$$\forall i \in [m], x_i > 0 \;\Rightarrow\; (AY)_i = max_{k\in[m]}(AY)_k \text{ , and}$$
$$\forall j \in [n], y_j > 0 \;\Rightarrow\; (X^T B)_j = max_{k\in[n]}(X^T B)_k \text{ .}$$

We show that the first one holds and the proof for the second follows similarly. Let

$$\lambda = max_{k\in[m(n+1)]}(CY')_k \text{ and } \lambda' = max_{k\in[m]}(CY')_k = \alpha + \max_{k\in[m]}(AY)_k \text{ (Using Lemma 43).}$$

As $\exists i \in [m], x'_i > 0$ we have $\lambda' = \lambda$. Thus we get

$$\forall i \in [m], x_i > 0 \Rightarrow (CY')_i = \lambda \Rightarrow \alpha + (AY)_i = \alpha + \max_{k\in[m]}(AY)_k \Rightarrow (AY)_i = \max_{k\in[m]}(AY)_k \text{ .}$$

For the second part, to the contrary suppose $\exists j \in [n],\ (\eta(Y))_j = \frac{y_j}{\sigma(Y)} > 0.5 \Rightarrow 2y_j > \sigma(Y)$. Then for some $i \in [m]$ we have $x'_i > 0$ and $(CY')_i \leq h\sigma(Y) + (AY)_i < 2hy_j + (AY)_i = (CY')_{jm+i} \leq \lambda$, a contradiction to (X', Y') being a NE of game (C, D). □

Lemmas 44 and 45 imply that game (A, B) has a NE in $\mathcal{B}_{0.5}$ if and only if game (C, D) has a NE where both players play some of first m, n strategies respectively. If we show that to get payoff of at least h in the latter game, players have to play some of first m, n strategies, then clearly the reduction will follow.

Lemma 46. *Given a strategy profile (X', Y'), if $X'^T CY' \geq h$ and $X'^T DY' \geq h$ then $X = X'(1 : m)$ and $Y = Y'(1 : n)$ are non-zero.*

Proof. If $Y = \mathbf{0}$, then $\forall i \in [m(n+1)]$ we have $(CY')_i \leq 0$ using Lemma 43, and in turn $X'^T CY' \leq 0$. Similarly, if $X = 0$, then $\forall j \in [n(m+1)]$ we have $(X'^T D)_j \leq 0$, and then $X'^T DY' \leq 0$. Lemma follows using the fact that $h > 0$. □

The next theorem follows using Lemmas 44, 45, and 46.

Theorem 25. *Game (A, B) has a NE in $\mathcal{B}_{0.5}$ if and only if game (C, D) has a NE where every player gets payoff at least h.*

Next theorem shows reduction from **InBox** to **Superset** using Lemma 44.

Theorem 26. *Game (A, B) has a NE in $\mathcal{B}_{0.5}$ if and only if game (C, D) has a NE where all the strategies played with non-zero probability by the first and second player are from $T_1 = [1 : m]$ and $T_2 = [1 : n]$ respectively.*

Lemmas 44 and 45 imply that, one of the first m, n strategies are played with non-zero probability by respective players in game (C, D) if and only if game (A, B) has a NE in $\mathcal{B}_{0.5}$. Thus next theorem gives a Turing reduction from **InBox** to **Subset**.

Theorem 27. *Game (A, B) has a NE in $\mathcal{B}_{0.5}$ if and only if $\exists i \in [m], \exists j \in [n]$ such that for $T_1 = \{i\}$ and $T_2 = \{j\}$, game (C, D) has a NE where all strategies of T_1 and T_2 are played with non-zero probability.*

Leveraging on the intuition presented for the reduction on 2-player games, next we extend Theorems 25, 26 and 27 to 3-player games in order to get the hardness results for the same.

4.4.2 3-Nash: **InBox** to **MaxPayoff**, **Subset** and **Superset**

Like in the two player case, given a 3-player game with $m \times n \times p$-dimensional payoff tensors (A, B, C), we will create a game (D, E, F) of size $m(n+1) \times n(p+1) \times p(m+1)$ and insert the original game in the first block with h added. We start with the definitions, analogous that of 14 and 15.

Definition 16. For $i \in [m]$, $j \in [n]$, $k \in [p]$, and a real number h, define

$A_{(i,:,:)+h}$: Tensor A with h added to the entries $A_{ij'k'}$ $\forall j' \in [n], \forall k' \in [p]$,

$A_{(:,j,:)+h}$: Tensor A with h added to the entries $A_{i'jk'}$ $\forall i' \in [m], \forall k' \in [p]$, and

$A_{(:,:,k)+h}$: Tensor A with h added to the entries $A_{i'j'k}$ $\forall i' \in [m], \forall j' \in [n]$.

Definition 17. Given a tensor T of size $a \times b \times c$ and integers r, s, t s.t. $a + r - 1 \leq m(n+1), b + s - 1 \leq n(p+1)$ and $c + t - 1 \leq p(m+1)$, define $[T]_{r,s,t}$ to be an $m(n+1) \times n(p+1) \times p(m+1)$ dimensional tensor where T is copied starting at position (r, s, t), and all other coordinates are set to zero.

Construct game (D, E, F) as follows, given (A, B, C) and a scalar $h > 0$.

$$\begin{aligned} D &= [A + h]_{1,1,1} + [(-1)_{m,n(p+1),mp}]_{1,1,p+1} + \textstyle\sum_{j \in [n]} [A_{(:,j,:)+2h}]_{jm+1,1,1} \ , \\ E &= [B + h]_{1,1,1} + [(-1)_{mn,n,(m+1)p}]_{m+1,1,1} + \textstyle\sum_{k \in [p]} [B_{(:,:,k)+2h}]_{1,kp+1,1} \ , \\ F &= [C + h]_{1,1,1} + [(-1)_{m(n+1),np,p}]_{1,n+1,1} + \textstyle\sum_{i \in [m]} [C_{(i,:,:)+2h}]_{1,1,in+1} \ . \end{aligned} \tag{4.5}$$

We will mimic the proof of 2-Nash to 3-Nash next, *i.e.,* Lemmas 43, 44, and 45. In the proof of each of these lemmas, argument for the second player follows similar to that for the first player due to symmetry in the construction of the reduced game. Therefore, in what follows we will focus on the first player again, and argument for the second and third player follows similarly.

Recall that $\pi_i(X)$, for $X \in \mathbf{\Delta}$ represents the payoff of player i when played profile is X. Since we will be dealing with two games in this section, in order to resolve ambiguity we super-script it with the payoff tensor under consideration. To denote payoff from a pure-strategy i with respect to tensor A, when other two are playing $Y, \boldsymbol{z}$ we use $\pi_1^A(i, Y, \boldsymbol{z})$, even if $Y, \boldsymbol{z}$ are not probability distributions.

Next lemma follows from the construction of game (D, E, F) in (4.5).

Lemma 47. *Let Y' and $\boldsymbol{z}'$ be vectors of sizes $n(p+1)$ and $p(m+1)$ respectively. Let $Y = Y'[1:n]$,*

$\boldsymbol{z} = \boldsymbol{z}'[1:p]$ *and* $\alpha = h * \sigma(Y)\sigma(\boldsymbol{z}) - \sum_{j\in[p+1,p(m+1)]} z'_j$. *We have*

$$\pi_1^D(i, Y', \boldsymbol{z}') = \begin{cases} \alpha + \pi_1^A(i, Y, \boldsymbol{z}) & \text{if } i \in [m] \\ 2hy_{\lfloor (i-1)/m \rfloor} + \pi_1^A(r, Y, \boldsymbol{z}) & \text{if } i \in [m+1, m(n+1)], \\ & \text{where } r = ((i-1)\ mod\ m) + 1 \end{cases} .$$

Let $\mathcal{B}_{0.5} = [0, 0.5]^{m+n+p}$. Using the payoff structure in game (D, E, F) we show the next lemma.

Lemma 48. *Game* (A, B, C) *has a NE* $(X, Y, \boldsymbol{z}) \in \mathcal{B}_{0.5}$ *iff* $(X', Y', \boldsymbol{z}') = ((X, 0_{mn}), (Y, 0_{np}), (\boldsymbol{z}, 0_{mp}))$ *is a NE of the game* (D, E, F).

Proof. The proof is similar to that of Lemma 44. For the forward direction, we show the first condition of (4.3) characterizing 3-Nash, and other two follow similarly. Note that again $\alpha = h$, and hence $\max_{i\in[m]} \pi_1^D(i, Y', \boldsymbol{z}') = h + \max_{i\in[m]} \pi_1^A(i, Y, \boldsymbol{z})$ (Using Lemma 47). Further, $\forall j \in [n]$ and $\forall i \in [m]$, we have $\pi_1^D(jm+i, Y', \boldsymbol{z}') = 2hy_j + \pi_1^A(i, Y, \boldsymbol{z}) \leq h + \pi_1^A(i, Y, \boldsymbol{z}) = \pi_1^D(i, Y', \boldsymbol{z}')$. Thus, the first m strategies are at least as good as last $[m+1, m(n+1)]$. We get $\forall i \in [m(n+1)],\ x'_i > 0 \Rightarrow \pi_1^D(i, Y', \boldsymbol{z}') = \max_{s\in[m(n+1)]} \pi_1^D(s, Y', \boldsymbol{z}')$. Argument for the second and third player follows similarly using the fact that $\boldsymbol{z} \leq 0.5$ and $X \leq 0.5$ respectively.

For the reverse direction, $\exists i \in [m],\ x'_i > 0$ and hence $\forall j \in [n],\ \pi_1^D(i, Y', \boldsymbol{z}') \geq \pi_1^D(mj + i, Y', \boldsymbol{z}') \Rightarrow 2hy_j \leq h \Rightarrow y_j \leq 0.5$. Similarly $X \leq 0.5$ and $\boldsymbol{z} \leq 0.5$ follows by arguing for third and second players respectively. □

Next we obtain a solution of **InBox** for game (A, B, C) from a NE of (D, E, F) where players play some strategies from the first m, n and p strategies respectively with non-zero probability.

Lemma 49. *If* $(X', Y', \boldsymbol{z}')$ *is a NE of game* (D, E, F) *such that the vectors* $X = X'[1:m]$, $Y = Y'[1:n]$, *and* $\boldsymbol{z} = \boldsymbol{z}'[1:p]$ *are non-zero, then* $(\eta(X), \eta(Y), \eta(\boldsymbol{z}))$ *is a NE for game* (A, B, C), *and* $(\eta(X), \eta(Y), \eta(\boldsymbol{z})) \in \mathcal{B}_{0.5}$.

Proof. As $\sigma(X), \sigma(Y), \sigma(\boldsymbol{z}) > 0$, profile $(\eta(X), \eta(Y), \eta(\boldsymbol{z}))$ is well-defined. To show that it is a NE of game (A, B, C) it suffices to show the following for the first player, and similar argument follows for the other two players.

$$\forall i \in [m],\ x_i > 0 \ \Rightarrow\ \pi_1^A(i, Y, \boldsymbol{z}) = \max_{l \in [m]} \pi_1^A(l, Y, \boldsymbol{z})\ .$$

Let $\lambda = \max_{k \in [m(n+1)]} \pi_1^D(i, Y', \boldsymbol{z}')$, and $\lambda' = \max_{k \in [m]} \pi_1^D(k, Y', \boldsymbol{z}') = \alpha + \max_{k \in [m]} \pi_1^A(k, Y, \boldsymbol{z})$ (Using Lemma 47). As $\exists i \in [m], x'_i > 0$ we have $\lambda' = \lambda$. Thus we get

$$\begin{aligned}
\forall i \in [m],\ \ x_i > 0 \ &\Rightarrow\ x'_i > 0 \\
&\Rightarrow\ \pi_1^D(i, Y', \boldsymbol{z}') = \lambda \\
&\Rightarrow\ \alpha + \pi_1^A(i, Y, \boldsymbol{z}) = \alpha + \max_{k \in [m]} \pi_1^A(k, Y', \boldsymbol{z}') \\
&\Rightarrow\ \pi_1^A(i, Y', \boldsymbol{z}') = \max_{k \in [m]} \pi_1^A(k, Y', \boldsymbol{z}')\ .
\end{aligned}$$

For the second part, to the contrary suppose $\exists j \in [n],\ (\eta(Y))_j = \frac{y_j}{\sigma(Y)} > 0.5 \Rightarrow 2y_j > \sigma(Y)$. Then for some $i \in [m]$ we have $x'_i > 0$ and $\pi_1^D(i, Y', \boldsymbol{z}') \leq h\sigma(Y) + \pi_1^A(i, Y', \boldsymbol{z}') < 2hy_j + \pi_1^A(i, Y', \boldsymbol{z}') = \pi_1^D(jm + i, Y', \boldsymbol{z}') \leq \lambda$, a contradiction to $(X', Y', \boldsymbol{z}')$ being a NE of game (D, E, F). □

Now if we can relate the NE of (D, E, F) where at least one of the first m, n, and p strategies are played by the first, second, and third players respectively, and the payoff received at the NE by all the players, then **InBox** to **MaxPayoff** reduction will follow.

Lemma 50. *Given a strategy profile $\boldsymbol{d} = (X', Y', \boldsymbol{z}')$ of game (D, E, F), if $\pi_i(\boldsymbol{d}) \geq h > 0,\ i = 1, 2, 3$, then $X = X'(1:m)$, $Y = Y'(1:n)$ and $\boldsymbol{z} = \boldsymbol{z}'(1:p)$ are non-zero.*

Proof. If $Y = \mathbf{0}$, then $\forall i \in [m(n+1)]$ we have $\pi_1^D(i, Y', \boldsymbol{z}') \leq 0$ using Lemma 47, and in turn $\pi_1(\boldsymbol{d}) \leq 0$. Similarly, if $\boldsymbol{z} = 0$ then we get $\pi_2(\boldsymbol{d}) \leq 0$, and if $X = 0$ then $\pi_3(\boldsymbol{d}) \leq 0$. Lemma follows using the fact that $h > 0$. □

The next theorem, for **InBox** to **MaxPayoff** reduction, follows using Lemmas 48, 49, and 50.

Theorem 28. *Game (A, B, C) has a NE in $\mathcal{B}_{0.5}$ if and only if game (D, E, F) has a NE where every player gets payoff at least h.*

The next theorem showing reduction from **InBox** to **Superset** follows using Lemma 48.

Theorem 29. *Game (A, B, C) has a NE in $\mathcal{B}_{0.5}$ if and only if game (D, E, F) has a NE where all the strategies played with non-zero probability by players are from $T_1 = [1 : m]$, $T_2 = [1 : n]$ and $T_3 = [1 : p]$ respectively.*

Next theorem follows using Lemmas 48 and 49, and gives a Turing machine reduction (many-to-one) from **InBox** to **Subset**.

Theorem 30. *Game (A, B, C) has a NE in $\mathcal{B}_{0.5}$ if and only if $\exists i \in [m], \exists j \in [n], \exists k \in [p]$ such that for $T_1 = \{i\}$, $T_2 = \{j\}$ and $T_3 = \{k\}$, game (D, E, F) has a NE where all strategies of T_1, T_2, T_3 are played with non-zero probability.*

From Theorem 30, it follows that to solve **InBox** for game (A, B, C) we will need to solve (mnp) many instances of **Subset** in game (D, E, F), we get many-to-one reduction from **InBox** to **Subset**. Theorems 28, 29 and 30 together with $\exists\mathbb{R}$-hardness of **InBox** in 3-Nash, and Theorem 23 gives the next result.

Theorem 31. *The problems of* **MaxPayoff**, **Subset** *and* **Superset** *are $\exists\mathbb{R}$-complete in 3-player games.*

A 3-player game can be reduced to a k-player game for $k > 3$ trivially, without changing its set of NE, by adding $k - 3$ dummy players with one strategy each (and payoff tensor $A_i = [h]$ to get reduction for **MaxPayoff**). And therefore, the next theorem follows from Theorem 31.

Theorem 32. *Given a k-player game $(A_1, \ldots, A_k)$, for a constant $k \geq 3$, the problems of* **MaxPayoff**, **Subset** *and* **Superset** *are $\exists\mathbb{R}$-complete.*

In the next section we show $\exists\mathbb{R}$-completeness for **NonUnique**, by reducing **MaxPayoff** to **NonUnique** in 3-player games.

4.4.3 **MaxPayoff** to **NonUnique**

In this section we reduce **MaxPayoff** to **NonUnique** in a 3-player game. Let (A, B, C) be a given game, and for a given rational number $h > 0$, we are asked to check if it has a NE where all three players get payoff at least h. We will reduce this problem to checking if game (D, E, F) has more than one equilibrium. Tensors A, B, C are of size $m \times n \times p$, where m, n, p are number of strategies of player $1, 2, 3$ respectively. Let $m' = m + 1, n' = n + 1, p' = p + 1$, and D, E, F be of size $m' \times n' \times p'$, where

$$\begin{aligned}
&\forall i \in [m], j \in [n], k \in [p], && D_{ijk} = A_{ijk},\ E_{ijk} = B_{ijk},\ F_{ijk} = C_{ijk} \\
&\forall j \in [n'], k \in [p'], && D_{m'jk} = h \\
&\forall i \in [m'], k \in [p'], && E_{in'k} = h \\
&\forall i \in [m'], j \in [n'], && F_{ijp'} = h\ .
\end{aligned}$$

Rest of the entries in D, E, F are set to zero. Basically, we added one extra strategy for each player and made sure that the player gets payoff h when she plays this extra strategy regardless of what others play.

Lemma 51. *Let $(X', Y', \boldsymbol{z}')$ be a strategy profile for game (D, E, F), and $X = X'(1 : m), Y = Y'(1 : n)$ and $\boldsymbol{z} = \boldsymbol{z}'(1 : p)$. Then,*

- $\pi_1^D(m', Y', \boldsymbol{z}') = h$, $\pi_2^E(X', n', \boldsymbol{z}') = h$, *and* $\pi_3^F(X', Y', p') = h$.
- $\forall i \in [1 : m],\ \pi_1^D(i, Y', \boldsymbol{z}') = \pi_1^A(i, Y, \boldsymbol{z})$. $\forall j \in [1 : n],\ \pi_2^E(X', j, \boldsymbol{z}') = \pi_2^B(X, j, \boldsymbol{z})$. $\forall k \in [1 : p],\ \pi_3^F(X', Y', k) = \pi_3^C(X, Y, k)$.

Proof. The first part follows by construction. For the second part, we show $\forall i \in [1 : m],\ \pi_1^D(i, Y', \boldsymbol{z}') =$

$\pi_1^A(i, Y, \boldsymbol{z})$. The rest can be proven similarly. Recall that $n' = n + 1$ and $p' = p + 1$. We have

$$\begin{aligned}
\pi_1^D(i, Y', \boldsymbol{z}') &= \textstyle\sum_{j\in[n'],k\in[p']} D_{ijk} y'_j z'_k \\
&= \textstyle\sum_{j\in[n],k\in[p]} D_{ijk} y_j z_k + \sum_{k\in[p']} D_{in'k} y'_{n'} z'_k + \sum_{j\in[n']} D_{ijp'} y'_j z'_{p'} \\
&= \textstyle\sum_{j\in[n],k\in[p]} A_{ijk} y_j z_k \\
&= \pi_1^A(i, Y, \boldsymbol{z}) \,,
\end{aligned}$$

where the third equality holds because $D_{ijk} = A_{ijk},\ \forall j \in [n], \forall k \in [p]$, and $D_{ijk} = 0$ if either $j = n'$ or $k = p'$. □

Next we show that game (D, E, F) has a trivial pure NE where all players play their extra strategy.

Lemma 52. *Pure-strategy profile (m', n', p') is a NE of game (D, E, F).*

Proof. When players two and three are playing strategy n' and p' respectively, then $\forall i \in [m]$ payoff $D_{in'p'}$ of the first player is zero, while $D_{m'n'p'} = h > 0$. Therefore, playing m' is the best response for her. Similarly, we can argue for players two and three. □

Except for the trivial NE established in Lemma 52 if game (D, E, F) has another equilibrium, then we need to construct a solution of **MaxPayoff** in game (A, B, C).

Lemma 53. *If $(X', Y', \boldsymbol{z}') \neq (m', n', p')$ is a NE of game (D, E, F), then $(\eta(X), \eta(Y), \eta(\boldsymbol{z}))$ is a NE of game (A, B, C) with payoff at least h to each player, where $X = X'(1 : m), Y = Y'(1 : n)$ and $\boldsymbol{z} = \boldsymbol{z}'(1 : p)$.*

Proof. First we show that $\sigma(X), \sigma(Y), \sigma(\boldsymbol{z}) > 0$. To the contrary, suppose $\boldsymbol{z} = \mathbf{0}$ and wlog $X \neq \mathbf{0}$. Then, $z'_{p'} = 1$, and $\exists i \in [m],\ x'_i > 0$ with payoff $\pi_1^D(i, Y', \boldsymbol{z}') = \pi_1^A(i, Y, \boldsymbol{z}) = 0$ (Lemma 51), a contradiction because player one will deviate to m' that always fetches payoff $h > 0$. Similar contradiction can be derived if $\sigma(Y) = 0$ or $\sigma(X) = 0$.

We will show that $\eta(X)$ is a best response of the first player when other two are playing $\eta(Y)$ and $\eta(\boldsymbol{z})$ respectively in (A, B, C), and that her payoff is at least h. Argument for other players follow similarly. Let $\lambda = \max_{s\in[m]} \sum_{j\in[n],k\in[p]} A_{sjk}y_j z_k$. It suffices to show that $\forall i \in [m],\ x_i > 0 \Rightarrow \sum_{j\in[n],k\in[p]} A_{ijk}y_j z_k = \lambda$ and $\lambda \geq h$, because normalization will increase the payoff of all the pure-strategies, and that too by the same factor.

Let $\lambda' = \max_{i\in[m']} \pi_1^D(i, Y', \boldsymbol{z}')$, then $\lambda = \lambda'$ because $\exists i \in [m], x_i > 0$ and payoff at i is λ.

$$x_i > 0 \Rightarrow x'_i > 0 \Rightarrow \pi_1^D(i, Y', \boldsymbol{z}') = \lambda' \Rightarrow \sum_{j\in[n],k\in[p]} A_{sjk}y_j z_k = \lambda\,.$$

Now since each player gets payoff h from their last strategy in game (D, E, F) (Lemma 51), other strategies played with non-zero probabilities have to fetch payoff at least h and hence $\lambda = \lambda' \geq h$ follows. □

We also need to establish that if game (A, B, C) has a feasible solution for **MaxPayoff** then game (D, E, F) has more than one equilibrium.

Lemma 54. *If $(X, Y, \boldsymbol{z})$ is a Nash equilibrium of (A, B, C) where every player gets payoff at least h, then $((X|0), (Y|0), (\boldsymbol{z}|0))$ is a NE of game (D, E, F).*

Proof. Let $X' = (X|0), Y' = (Y|0)$ and $\boldsymbol{z}' = (\boldsymbol{z}|0)$. We will show that X' is a best response for player one against $Y', \boldsymbol{z}'$ in (D, E, F), and cases for other two players follow similarly. Let $\lambda = \max_{i\in[m]} \pi_1^A(i, Y, \boldsymbol{z})$ and $\lambda' = \max_{i\in[m']} \pi_1^D(i, Y', \boldsymbol{z}')$. Since $\lambda \geq h$ and $\pi_1^D(m', Y', \boldsymbol{z}') = h$ (Lemma 51) we get $\lambda = \lambda'$, and the lemma follows. □

Using Lemmas 52, 53 and 54, we get the next theorem.

Theorem 33. *Game (A, B, C) has a NE where every player gets at least h payoff iff game (D, E, F) has more than one equilibrium.*

As argued in Section 4.4.2, a 3-player game can be trivially reduced to a k-player game, for $k > 3$, by adding $k - 3$ dummy players. Therefore, next theorem follows using Theorems 23, 31 and 33.

Theorem 34. *Given a k-player game $(A_1, \ldots, A_k)$, for a constant $k \geq 3$, the problem of* **NonUnique** *is $\exists\mathbb{R}$-complete.*

Since there is no reduction known from 3-Nash to symmetric 3-Nash, results of this section do not follow directly for symmetric 3-Nash, or symmetric k-Nash for that matter. In the next section we show $\exists\mathbb{R}$-completeness results for symmetric Nash equilibria in 3-player symmetric games.

4.5 Symmetric 3-Nash: $\exists\mathbb{R}$-Completeness

Containment in $\exists\mathbb{R}$ for various decision versions of symmetric 3-Nash is shown in Section 4.3 (Theorem 24). In this section, we show $\exists\mathbb{R}$-hardness for **Subset** and **Superset** for symmetric 3-Nash, by giving a reduction from 3-Nash to symmetric 3-Nash.

Let the given game be (A, B, C), where each tensor is of size $m \times n \times p$. Let D denote the reduced symmetric game, which will be of size $l \times l \times l$, where $l = m + n + p$. Let $(X, Y, \boldsymbol{z})$ be a NE of (A, B, C). We will show that there are positive numbers α, β, γ such that $(\boldsymbol{d}, \boldsymbol{d}, \boldsymbol{d})$ is a NE of the reduced game, where $\boldsymbol{d}$ is a l-dimensional vector $(\alpha X | \beta Y | \gamma \boldsymbol{z})$. Furthermore, let $(\boldsymbol{d}, \boldsymbol{d}, \boldsymbol{d})$ be a NE of the reduced game, where $\boldsymbol{d}$ decomposes into vectors $X', Y', \boldsymbol{z}'$ of dimension m, n, p respectively. Scaling these vectors gives a NE $(X, Y, \boldsymbol{z})$ of game (A, B, C). This will yield mapping in both directions.

Essential to this reduction is the $3 \times 3 \times 3$ symmetric game $G(a, b, c)$ given below. We represent the payoff tensor of the first player by three 3×3 matrices, one for each of her pure strategy. Here

a, b, c are any non-negative reals.

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & a \\ 0 & a & 0 \end{bmatrix}, \qquad \begin{bmatrix} 0 & 0 & b \\ 0 & 0 & 0 \\ b & 0 & 1 \end{bmatrix}, \qquad \begin{bmatrix} 1 & c & 0 \\ c & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{4.6}$$

Lemma 55. *If (α, β, γ) is a symmetric NE of game G, then $\alpha, \beta, \gamma > 0$.*

Proof. We will first show that G has no symmetric NE of support one or two. This involves a case analysis of which we present one representative case each. First observe that $(\alpha, \beta, \gamma) = (1, 0, 0)$ cannot be a symmetric NE, since player 1 should play $(0, 0, 1)$ if the other two players play the given strategy. Next consider the strategy $(\alpha, \beta, 0)$ with the first two components non-zero. Because the matrix corresponding to second strategy of the first player has all zeros in the upper left 2×2 sub-matrix, she will be strictly better off playing the third strategy instead of the second. Hence any symmetric NE of G must be of full support. □

From G, we derive a symmetric game D, which is of size $l \times l \times l$, by blowing up each of the three strategies of G to m, n, p number of strategies respectively. Copy 0s and 1s to their respective blocks, and replace a, b, c with tensors A, B, C respectively after appropriate rotation. For example we have $G(1, 2, 3) = a$, which is replaced by A as is, while $G(1, 3, 2) = a$ is replaced by A after rotation so that first, second and third dimensions correspond to players one, three and two respectively. In general, $G(i_1, i_2, i_3)$ is replaced by an appropriate tensor after rotation such that first, second and third dimensions correspond to players i_1, i_2 and i_3 respectively where $i_1 \neq i_2 \neq i_3 \neq i_1$. Following is the formal description of D

$$D_{stu} = \begin{cases} A_{s(t-m)(u-m-n)} & \text{if } s \le m \ \& \ m < t \le m+n \ \& \ m+n < u \le l \\ A_{s(u-m)(t-m-n)} & \text{if } s \le m \ \& \ m < u \le m+n \ \& \ m+n < t \le l \\ B_{t(s-m)(u-m-n)} & \text{if } t \le m \ \& \ m < s \le m+n \ \& \ m+n < u \le l \\ B_{u(s-m)(t-m-n)} & \text{if } u \le m \ \& \ m < s \le m+n \ \& \ m+n < t \le l \\ C_{t(u-m)(s-m-n)} & \text{if } t \le m \ \& \ m < u \le m+n \ \& \ m+n < s \le l \\ C_{u(t-m)(s-m-n)} & \text{if } u \le m \ \& \ m < t \le m+n \ \& \ m+n < s \le l \\ 1 & \text{if } s \le m \ \& \ m < t = u \le m+n, \\ 1 & \text{if } m < s \le m+n \ \& \ m+n < t = u \le l \\ 1 & \text{if } m+n < s \le l \ \& \ t = u \le m \\ 0 & Otherwise\ . \end{cases} \tag{4.7}$$
$$1 \le s,t,u \le l$$

In the above game, suppose two players are playing mixed-strategy $\boldsymbol{d} = (X|Y|\boldsymbol{z})$, where $X, Y, \boldsymbol{z}$ are of dimensions m, n, p respectively. Then from strategy s the third player receives payoff:

$$\pi^D(s, \boldsymbol{d}) = \begin{cases} (\sigma(Y))^2 + 2\sum_{j\in[n],k\in[p]} A_{sjk} y_j z_k, & \text{if } s \le m, \\ (\sigma(\boldsymbol{z}))^2 + 2\sum_{i\in[m],k\in[p]} B_{isk} x_i z_k & \text{if } m < s \le m+n \ \cdot \\ (\sigma(X))^2 + 2\sum_{i\in[m],j\in[n]} C_{ijs} x_i y_j & \text{if } m+n < s \le l \end{cases} \tag{4.8}$$

Wlog we assume that $A, B, C \ge 0$ and hence $D \ge 0$. We consider $\frac{0}{0}$ as 0.

Lemma 56. *If $\boldsymbol{d} = (X|Y|\boldsymbol{z})$ is a SNE of game D then $(\sigma(X), \sigma(Y), \sigma(\boldsymbol{z}))$ is a NE of $G(a,b,c)$ where $a = \frac{\max_{s\le m}\sum_{jk} A_{sjk} y_j z_k}{\sigma(Y)\sigma(\boldsymbol{z})}$, $b = \frac{\max_{s\le n}\sum_{i,k} B_{isk} x_i z_k}{\sigma(X)\sigma(\boldsymbol{z})}$, $c = \frac{\max_{s\le p}\sum_{i,j} C_{ijs} x_i y_j}{\sigma(X)\sigma(Y)}$.*

Proof. Let $\alpha = \sigma(X)$, $\beta = \sigma(Y)$ and $\gamma = \sigma(\boldsymbol{z})$. Clearly, the payoffs from three strategies of G are respectively $\beta^2 + 2a\beta\gamma$, $\gamma^2 + 2b\alpha\gamma$, and $\alpha^2 + 2c\alpha\beta$. Observe that these are also the best payoffs among strategies $[1:m], [m+1:m+n]$ and $[m+n+1:l]$ respectively in game D. Let the maximum among these three be λ. Then, we have

$$\alpha = \sigma(X) > 0 \Rightarrow \exists i \leq m,\ x_i > 0 \Rightarrow \pi^D(i, \boldsymbol{d}) = \beta^2 + 2a\beta\gamma = \lambda\,.$$

Similarly, we can show that if $\beta > 0$ then payoff at the second strategy is λ, and if $\gamma > 0$ then the third gives λ. Hence (α, β, γ) is a symmetric NE of game G. □

Lemmas 55 and 56 imply that at any SNE $\boldsymbol{d} = (X|Y|\boldsymbol{z})$, all three components $X, Y, \boldsymbol{z}$ of the strategy profile are non-zero. Next we show that normalizing each gives a NE of the original game (A, B, C). Recall the notations $\sigma(X) = \sum_i x_i$ and $\eta(X) = \frac{X}{\sigma(X)}$.

Lemma 57. *If $\boldsymbol{d} = (X|Y|\boldsymbol{z})$ is a SNE of game D, then $\sigma(X), \sigma(Y), \sigma(\boldsymbol{z}) > 0$, and $(\eta(X), \eta(Y), \eta(\boldsymbol{z}))$ is a NE of game (A, B, C).*

Proof. From Lemmas 55 and 56 it follows that vectors X, Y, and $\boldsymbol{z}$ are non-zero. Thus the first part follows.

Let $X' = \eta(X), Y' = \eta(Y)$ and $\boldsymbol{z}' = \eta(\boldsymbol{z})$; clearly these are well-defined due to the first part. We will show that $(X', Y', \boldsymbol{z}')$ satisfies conditions (4.3) characterizing NE of game (A, B, C). We do this for the first condition, the rest two follow similarly. Let λ denote the maximum payoff of a player in symmetric game D when others are playing $\boldsymbol{d}$. For strategy $s \in S_1$ of the first player, we have

$$\begin{aligned}
x'_s > 0 \ &\Rightarrow\ x_s > 0 \\
&\Rightarrow\ \pi^D(s, \boldsymbol{d}) = \lambda \ \text{(Using (4.8) and (4.3))} \\
&\Rightarrow\ \pi^D(s, \boldsymbol{d}) \geq \pi^D(s', \boldsymbol{d}),\ \ \forall s' \leq m \\
&\Rightarrow\ \textstyle\sum_{j\in[n],k\in[p]} A_{sjk} y'_j z'_k \geq \sum_{j\in[n],k\in[p]} A_{s'jk} y'_j z'_k, \forall s' \leq m\,.
\end{aligned}$$

□

The mapping from SNE of game D to NE of game (A, B, C) established in Lemma 57 implies that computing SNE in symmetric games is no easier than computing a NE in normal games. We extend this reduction to k-Nash in Section 4.7. Next, we show a mapping in reverse direction, i.e., from NE of (A, B, C) to a SNE of D, to obtain $\exists\mathbb{R}$-hardness results for a number of decision problems in symmetric 3-Nash.

Lemma 58. *Let $(X, Y, \boldsymbol{z})$ be a NE of (A, B, C), and let (α, β, γ) be a NE of game $G(a, b, c)$ where a, b, c are set to payoffs of the first, second and third players respectively at the NE of game (A, B, C). Then $\boldsymbol{d} = (\alpha X|\beta Y|\gamma \boldsymbol{z})$ is a SNE of game D.*

Proof. Clearly, $a = \max_{i \in S_1} \sum_{j,k} A_{ijk} y_j z_k$, $b = \max_{j \in S_2} \sum_{i,k} B_{ijk} x_i z_k$ and $c = \max_{i,j} C_{ijk} x_i y_j$. Let $X' = \alpha X$, $Y' = \beta Y$ and $\boldsymbol{z}' = \gamma \boldsymbol{z}$, then clearly $\boldsymbol{d} = (X'|Y'|\boldsymbol{z}')$ is a mixed-strategy, *i.e.,* $\sigma(\boldsymbol{d}) = 1$. Since $\alpha, \beta, \gamma > 0$ (Lemma 55), we have $X', Y', \boldsymbol{z}' \neq 0$. In symmetric game D, let $a' = \max_{s \leq m} \pi^D(s, \boldsymbol{d}) = \beta^2 + 2a\beta\gamma$, $b' = \max_{m < s \leq m+n} \pi^D(s, \boldsymbol{d}) = \gamma^2 + 2b\alpha\gamma$, and $c' = \max_{m+n < s \leq l} \pi^D(s, \boldsymbol{d}) = \alpha^2 + 2c\alpha\beta$. Note that a', b', c' are payoffs from the three strategies at (α, β, γ) in game G. Since (α, β, γ) is a NE of G, we have $a' = b' = c'$ (using Lemma 55).

As $(X, Y, \boldsymbol{z})$ is a NE of game (A, B, C), we get

$$\forall i \in [m],\ x'_i > 0 \Rightarrow x_i > 0 \Rightarrow \sum_{j,k} A_{ijk} y_j z_k = a \Rightarrow \pi^D(i, \boldsymbol{d}) = a' \ .$$

Similarly we get, $\forall j \in [n],\ y'_j > 0 \Rightarrow \pi^D(m + j, \boldsymbol{d}) = b'$, and $\forall k \in [p],\ z'_k > 0 \Rightarrow \pi^D(m + n + k, \boldsymbol{d}) = c'$. Lemma follows using the fact that $a' = b' = c'$. □

The next theorem summaries the relation between NE of game (A, B, C) and SNE of game D, and follows using Lemmas 57 and 58.

Theorem 35. *Profile $\boldsymbol{d} = (X|Y|\boldsymbol{z})$ is a SNE of game D iff $(\eta(X), \eta(Y), \eta(\boldsymbol{z}))$ is a NE of game (A, B, C).*

We showed a number of $\exists\mathbb{R}$-completeness results for 3-Nash in Section 4.4. Since support of a NE remains intact in the reduction from 3-Nash to symmetric 3-Nash as shown in Theorem 35, next we show $\exists\mathbb{R}$-completeness of **Subset** and **Superset** problems for symmetric 3-Nash.

Theorem 36. *Given a symmetric game D and a subset $T \subset S$, it is $\exists\mathbb{R}$-complete to check if there exists a SNE X s.t. $x_s > 0,\ \forall s \in T$ (**Subset**).*

Proof. Theorem 31 establishes that checking if game (A, B, C) has a NE where strategies in $T_i \subset S_i,\ i = 1, 2, 3$ are played with non-zero probability is $\exists\mathbb{R}$-complete. Let $l = m + n + p$. Construct a symmetric game D of size $l \times l \times l$ from G of (4.6) by blowing it up and replacing a, b and c with A, B, and C respectively. Construct D as given in (4.7).

Let $T = T_1 \cup \{j + m \mid j \in T_2\} \cup \{k + m + n \mid k \in T_3\}$. Using Theorem 35 it follows that game (A, B, C) has a NE where strategies of T_i are played with positive probability if and only if game D has a symmetric NE where strategies of T are played with positive probability. Since size of D is $O(size(A, B, C))$, $\exists\mathbb{R}$-hardness follows.

Containment in $\exists\mathbb{R}$ follows from Theorem 24. □

The next theorem follows similarly using Theorems 31 and 35.

Theorem 37. *Given a symmetric game D and a subset $T \subset S$, it is $\exists\mathbb{R}$-complete to check if there exists a SNE X s.t. $x_s = 0,\ \forall s \in S \setminus T$ (**Superset**).*

4.6 Symmetric 3-Nash: FIXP$_a$-completeness

Even though Theorem 35 reduces 3-Nash, which is known to be FIXP-complete [68], to symmetric 3-Nash, we do not get FIXP-harness for the latter. This is because to obtain a solution, say X, of former requires *division* among the coordinates of a solution, say $\boldsymbol{d}$, of the latter. While FIXP reduction requires that every x_i is a linear function of some d_j, with rational coefficients [68] (in order to handle irrational solutions under Turing reduction). Since there always exists a strong

approximate solution that constitutes only rational numbers, such a requirement is not needed for FIXP$_a$. Leveraging on this, we give a reduction for strong approximation in 3-Nash to strong approximation in symmetric 3-Nash in this section, to get FIXP$_a$-completeness for the latter.

First, for containment in FIXP$_a$ we show a more general result, namely for symmetric k-Nash, $k \geq 3$. And later show the hardness for the 3-player case. We show that symmetric k-Nash, for a constant k, is in FIXP, and consequently strong approximation is in FIXP$_a$. Let the given game be represented by tensor A and let the set of pure strategies of players be S. At a symmetric NE all players play the same mixed-strategy. Consider a function $F : \Delta \to \Delta$ as follows, where $X' = F(X)$ for an $X \in \Delta$:

$$\forall s \in S, \quad x'_s = \frac{x_s + \max\{\pi^A(s, X) - \pi^A(X), 0\}}{1 + \sum_s \max\{\pi^A(s, X) - \pi^A(X), 0\}} \,. \tag{4.9}$$

Nash [84] proved that fixed-points of F are exactly the symmetric NE of game A.

Theorem 38. *The problem of computing a symmetric NE in a symmetric k-player game, for a constant k, is in FIXP, and corresponding strong approximation is in* FIXP$_a$.

Proof. The operations used in defining F are $+, -, *, /$ and $\max$. Further, domain of F is convex and compact, and function is well-defined over the domain. Thus, finding fixed-points of F is in FIXP by definition. Since description of F is $O(size(A))$, this together with Nash's result [84] imply that finding a symmetric NE of A is also in FIXP. Further, for a given $\epsilon > 0$ if X is ϵ-near to an actual fixed-point X^*, i.e., $|X - X^*|_\infty < \epsilon$, then X is also a strong approximate symmetric NE of game A. Containment in FIXP$_a$ follows. □

For FIXP$_a$-hardness result we need to compute a strategy profile $(X', Y', \boldsymbol{z}')$ that is ϵ-near to an actual equilibrium of (A, B, C), given a symmetric profile $\boldsymbol{d}$ ϵ'-near to a symmetric NE $\boldsymbol{d}^*$ of D, where distances are measured in l_∞ norm.

In reduction of Theorem 35, obtaining solution of (A, B, C) involves e.g., dividing X by $\sigma(X)$.

If the latter is very small, this may give us a vector that is very far from a solution of (A, B, C), even when $\boldsymbol{d}$ may be close to $\boldsymbol{d}^*$. To get around this, next we make sure that $\sigma(X)$ is big enough.

Wlog, we assume that all entries of $A, B, C \in [0, 0.1]$, as adding constants to A, B, C or scaling them by positive constants does not change its set of NE. In that case, payoffs of a player at its NE is in $[0, 0.1]$. The a, b, c of Lemma 56 are also in $[0, 0.1]$. Thus, if we can lower bound the NE strategy (α, β, γ) of game G with such a, b, c then we get a lower bound on $\sigma(X)$, $\sigma(Y)$ and $\sigma(\boldsymbol{z})$ as desired.

Lemma 59. *If (α, β, γ) is a NE of game $G(a, b, c)$, where $a, b, c \in [0, 0.1]$, then $\frac{1}{4} \leq \alpha, \beta, \gamma \leq \frac{1}{2}$.*

Proof. Note that $\alpha, \beta, \gamma > 0$ because of Lemma 55. Therefore each of the three strategies fetch the same payoff, i.e., $\beta^2 + 2a\beta\gamma = \gamma^2 + 2b\alpha\gamma = \alpha^2 + 2c\alpha\beta$. We show that none of $\alpha, \beta, \gamma < 1/4$, and the upper bound follows because $\alpha + \beta + \gamma = 1$. There are two cases for each, and we show them for α. For β and γ they follow similarly.

Case I: $\alpha < 1/4$, and $\beta, \gamma \geq 1/4$.

As $\beta+\gamma \geq 3/4$, wlog let $\beta \geq 3/8$. Then, we have $\beta^2+2a\beta\gamma \geq 9/64+3a/16$, and $\alpha^2+2c\alpha\beta \leq 1/16+c/2$. The above equality gives $9/64 + 3a/16 \leq 1/16 + c/2 \Rightarrow 5/64 \leq c/2 - 3a/16 \Rightarrow c \geq 10/64 \geq 0.1$, a contradiction.

Case II: $\alpha, \gamma < 1/4$, and $\beta > 1/2$.

$\beta^2+2a\beta\gamma \geq 1/4$ and $\gamma^2+2c\alpha\gamma \leq 1+2c/16$. Thus, we have $4 \leq 1+2c \Rightarrow c \geq 3/2$, a contradiction. □

Next we show that strong approximate symmetric NE of game D maps to a strong approximate NE of (A, B, C), under the mapping of Theorem 35.

Lemma 60. *Let $\boldsymbol{d}^* = (X^*|Y^*|\boldsymbol{z}^*)$ be a symmetric Nash equilibrium of game D, and $\boldsymbol{d} = (X|Y|\boldsymbol{z})$ be such that $|\boldsymbol{d} - \boldsymbol{d}^*|_\infty \leq \epsilon$. Then, $\left|\frac{x_i}{\sigma(X)} - \frac{x_i^*}{\sigma(X^*)}\right| \leq \epsilon'$, $\forall i$; $\left|\frac{y_j}{\sigma(Y)} - \frac{y_j^*}{\sigma(Y^*)}\right| \leq \epsilon'$, $\forall j$; and $\left|\frac{z_k}{\sigma(\boldsymbol{z})} - \frac{z_k^*}{\sigma(\boldsymbol{z}^*)}\right| \leq \epsilon'$, $\forall k$, where $\epsilon = \frac{\epsilon'}{20l}$.*

Proof. Lemmas 56 and 59 give us $\frac{1}{4} \leq \sigma(X^*) \leq \frac{1}{2}$. Using this we obtain bounds on $\sigma(X)$.

$$\forall i \leq m,\ |x_i - x_i^*| \leq \epsilon \Rightarrow |\sigma(X) - \sigma(X^*)| \leq m\epsilon \Rightarrow \sigma(X^*) - m\epsilon \leq \sigma(X) \leq \sigma(X^*) + m\epsilon$$

Assuming $\epsilon < \frac{1}{20m}$, we get that $\frac{1}{5} \leq \sigma(X) \leq \frac{2}{3}$. Next consider the quantity we wish to bound.

$$\begin{aligned}
\left|\frac{x_i}{\sigma(X)} - \frac{x_i^*}{\sigma(X^*)}\right| &\leq 20|x_i \textstyle\sum_k x_k^* - x_i^* \sum_k x_k| \\
&\leq 20|x_i(m\epsilon + \textstyle\sum_k x_k) - (x_i - m\epsilon)\sum_k x_k| \\
&\leq 20(m\epsilon(x_i + \textstyle\sum_k x_k)) \\
&\leq 20(m+1)\epsilon \leq \epsilon' .
\end{aligned}$$

Similar argument suffices to show $\forall j,\ \left|\frac{y_j}{\sigma(Y)} - \frac{y_j^*}{\sigma(Y^*)}\right| \leq \epsilon'$, and $\forall k,\ \left|\frac{z_k}{\sigma(\boldsymbol{z})} - \frac{z_k^*}{\sigma(\boldsymbol{z}^*)}\right| \leq \epsilon'$. □

From Theorem 35 we know that a symmetric NE $\boldsymbol{d}^* = (X^*|Y^*|\boldsymbol{z}^*)$ maps to a NE $(X'^*, Y'^*, \boldsymbol{z}'^*) = (\eta(X^*), \eta(Y^*), \eta(\boldsymbol{z}^*))$of game (A, B, C). Lemma 60 implies that finding a profile $(X', Y', \boldsymbol{z}')$ that is ϵ' near to $(X'^*, Y'^*, \boldsymbol{z}'^*)$, for any $\epsilon' < 1$ reduces to finding a symmetric profile $\boldsymbol{d}$ that is $\frac{\epsilon'}{20l}$ near to $\boldsymbol{d}^*$. Clearly, there is such a $\boldsymbol{d}$ with size $poly\{size(A, B, C), \log(\frac{1}{\epsilon'})\}$, and therefore it can be mapped to a solution of (A, B, C) in polynomial time. Since, such an approximation in 3-Nash is FIXP$_a$-hard [68], and symmetric 3-Nash is in FIXP (Theorem 38), the next theorem follows.

Theorem 39. *Symmetric* 3*-Nash is* FIXP$_a$*-complete.*

Since there is no trivial reduction from symmetric 3-player game to symmetric k-player game, in the next section we extend Theorems 36, 37 and 39 to symmetric k-Nash, to obtain all the results for the latter.

4.7 Symmetric k-Nash: $\exists\mathbb{R}$ and FIXP$_a$ Completeness

Building on the construction of Section 4.5, in this section we reduce k-Nash to symmetric k-Nash. Given a k-player game $A = (A_1, \ldots, A_k)$ we construct a symmetric game D where the

set of strategies of each player is $S = \cup_i S_i$, such that NE of game A maps to symmetric NE of game D, and vice-versa. Note that D will be a k-dimensional tensor with $l = \sum_i m_i$ coordinates in each dimension, where $m_i = |S_i|$. First we construct a symmetric game G (similar to that of (4.6)), which has now k-players each with k strategies. As players are identical in symmetric games, the payoff of a player from her pure-strategy depends on which strategies are played by how many players; it doesn't matter who played what. Therefore, the non-zero entries of G may be represented as follows, where $a_1, \ldots, a_k$ are non-negative numbers.

$G(i, i+1, \ldots, i+1) = 1, \ \forall i < k; \quad G(k, 1, \ldots, 1) = 1;$

$G(i, \{1, \ldots, i-1, i+1, \ldots, k\}) = a_i, \forall i \leq k, \forall$ permutations of $\{1, \ldots, i-1, i+1, \ldots, k\}$;

Set the rest of entries of G to zero.

Similar to Lemma 55, it follows that all symmetric NE of G are of full support. Next, we can blow up G to construct D. Note that G is a k-dimensional tensor with length k in each dimension, *i.e.,* for any $(i_1, \ldots, i_k)^{th}$ entry of G each $i_j \in [k]$. In every dimension, j^{th} element will represent j^{th} player of game A when mapped to D, and therefore in game D it will be replaced by $m_j = |S_j|$ many elements. Thus D will be k-dimensional tensor with length $l = \sum_{j=1}^{k} m_j$ in each dimension.

If $G(i_1, \ldots, i_k)$ is zero or one, then we replace it by a k-dimensional tensor of all zeros or all ones respectively of dimension $m_{i_1} \times \cdots \times m_{i_k}$. Note that if $G(i_1, \ldots, i_k) = a_i$ for some $i \in [k]$, then set $\{i_1, \ldots, i_k\} = [k]$, *i.e.,* every player is represented. We will replace this entry in G by tensor A_i from game A after appropriate rotation so that its j^{th} dimension corresponds to j^{th} player.

Like Lemma 56 we can show that if $\boldsymbol{d} = (X^1 | \ldots | X^k)$ is a symmetric NE of game D then $(\sigma(X^1), \ldots, \sigma(X^k))$ is a symmetric NE of game G, thereby showing that each of these sums are strictly positive. Here a_i is set to the best payoff achieved among the strategies of X^i divided by $\Pi_{j \neq i} \sigma(X^j)$. Further, $\boldsymbol{d}$ being a NE it ensures that if a coordinate j of X^i is non-zero then payoff from j^{th} strategy, among strategies corresponding to X^i is the best. This sets the stage to obtain

NE of game A from $\boldsymbol{d}$, namely, $(\eta(X^1), \ldots, \eta(X^k))$ (Similar to Lemma 57).

For the reverse mapping, let $X = (X^1, \ldots, X^k)$ be a NE of game A, and let $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_k)$ be a symmetric NE of G where a_i is set to the payoff player i receives at the given NE of A. Then, it follows that $\boldsymbol{d} = (\alpha_1 X^1 | \ldots | \alpha_k X^k)$ is a symmetric NE of D. The brief reason is as follows: the best payoff from ith block of strategies is $a'_i = \alpha_{i+1}^{k-1} + (k-1)! a_i \Pi_{j \neq i} \alpha_i$, and X being a NE of A, non-zero strategies of X^i fetch best payoff to player i, namely a_i. Hence, in $\boldsymbol{d}$ the strategies played with non-zero probability within block i fetch payoff a'_i. Since a_i is the maximum payoff of player i from any of its pure strategies, a'_i is also maximum among the payoffs from the strategies within the block. Furthermore, a'_i is also the payoff from ith strategy in game G, and $\boldsymbol{\alpha}$ being a NE with full support, it ensures that all a'_is are same. Thus, in $\boldsymbol{d}$ best payoffs are the same across blocks, and therefore it is a symmetric NE of game D.

The next theorem follows from the above discussion (of this section).

Theorem 40. *Profile $\boldsymbol{d} = (X^1 | \ldots | X^k)$ is a symmetric NE of game D if and only if $(\eta(X^1), \ldots, \eta(X^k))$ is a NE of game $(A_1, \ldots, A_k)$.*

Using Theorem 40 together with Theorems 24 and 31 we get the following $\exists\mathbb{R}$-completeness results.

Theorem 41. *For symmetric k-Nash, problems* **Subset** *and* **Superset** *are $\exists\mathbb{R}$-complete, where $k \geq 3$ is a constant.*

A normal form k-player game can be reduced to $k+1$-player game trivially by adding a dummy player with one strategy and any payoff, and therefore FIXP$_a$-hardness of Theorem **??** extends to k-Nash for $k \geq 3$. However, such a reduction is not possible in case of symmetric games, because the resulting game has to satisfy the symmetry conditions (see Section 4.2.1). Therefore, FIXP$_a$-hardness for symmetric 3-Nash does not extend to symmetric k-Nash for $k > 3$. We show this result using the fact that k-Nash is FIXP$_a$-hard together with Theorem 40.

As done in Section 4.6, we need to lower bound each $\sigma(X^i)$ for a given symmetric NE $\boldsymbol{d} = (X^1|\ldots|X^k)$ of game D. Similar to Lemma 59, we show the following. Define,

$$\Gamma = \frac{1}{(k-1)!}\left(\left(\frac{k}{k-1}\right)^{(k-1)} - 1\right).$$

Lemma 61. *Let $(\alpha_1, \ldots, \alpha_k)$ be a NE of game G with $a_i \in [0,\ \Gamma],\ \forall i \in [k]$, then $\frac{1}{k+1} \leq \alpha_i \leq \frac{1}{k-1},\ \forall i \in [k]$.*

Proof. Suppose not, and wlog let $\alpha_1 < \frac{1}{k+1}$. Then $\exists i \neq 1,\ \alpha_i > \frac{k}{k^2-1}$, let it be $i = 2$ (wlog). Then the payoff of first player from strategy one is:

$$\alpha_2^{(k-1)} + (k-1)!a_1\Pi_{i=2}^k\alpha_i \geq \left(\frac{k}{k^2-1}\right)^{(k-1)}.$$

Given that d non-negative numbers sum up to one, then their product is maximized when each number is 1/d. Using this, from strategy k, player one gets

$$\alpha_1^{(k-1)} + (k-1)!a_k\Pi_{i=1}^{(k-1)}\alpha_i < \frac{1}{(k+1)^{(k-1)}} + \frac{(k-1)!\ a_k}{(k+1)(k-1)^{(k-2)}}.$$

Since at equilibrium both the strategies are played with nonzero probability we get

$$\left(\frac{k}{k^2-1}\right)^{(k-1)} < \frac{1}{(k+1)^{(k-1)}} + \frac{(k-1)!\ a_k}{(k+1)(k-1)^{(k-2)}} \Rightarrow \frac{1}{(k-1)!}\left(\left(\frac{k}{k-1}\right)^{(k-1)} - 1\right) < a_k,$$

which is a contradiction to $a_k \leq \Gamma$. □

We can wlog assume that $A_1, \ldots, A_k \in [0, \Gamma]$, since NE remains unchanged when all the payoffs are scaled additively, or multiplicatively by a positive constant. This will ensure that payoff of each player in A at any NE is in $[0, \Gamma]$.

Since we know that if $\boldsymbol{d} = (X^1|\ldots|X^k)$ is a symmetric NE of game D then $(\sigma(X^1), \ldots, \sigma(X^k))$ is a symmetric NE of game G, from Lemma 61 we get that each of $\sigma(X^i)$ is lower bounded by

$\frac{1}{k+1}$. Finally, using this lower bound we can show that if $|\boldsymbol{d} - \boldsymbol{d}^*|_\infty < \epsilon$ where $\boldsymbol{d}^*$ is a symmetric NE, then $|x_s^i/\sigma(X^i) - {x_s^*}^i/\sigma({X^*}^i)| < \epsilon',\ \forall i \in [1:k], \forall s \in S_i$, where $\epsilon = \frac{\epsilon'}{2(k+1)^2(\sum_i |S_i|)}$ and $\epsilon < \frac{1}{5(k+1)(\max_{i\in[k]}|S_i|)}$ (similar to Lemma 60). In other words the strategy profile $(\eta(X^1), \ldots, \eta(X^k))$ obtained from $\boldsymbol{d}$ is ϵ'-near to NE $(\eta({X^*}^1), \ldots, \eta({X^*}^k))$ obtained from $\boldsymbol{d}^*$ for game A. Thus, FIXP$_a$-hardness follows for symmetric k-Nash, and we get the next result using Theorem 38.

Theorem 42. *For a constant $k \geq 3$, symmetric k-Nash is* FIXP$_a$*-complete.*

4.8 Discussion

There is a reduction from symmetric 2-Nash to 2-Nash using the notion of imitation games [77]. Is there an analogous reduction from symmetric k-Nash to k-Nash, for $k \geq 3$? For the case of 2-player games, Papadimitriou [91] asked the complexity of finding a non-symmetric equilibrium in a symmetric game. This was recently shown to be NP-complete [80]. What is the complexity of the analogous question for k-player games, for $k \geq 3$? For the case of 2-player games, the question of counting the number of equilibria, even those satisfying special properties, is typically #P-complete. What is the complexity of analogous questions for k-player games, for $k \geq 3$? Are they PSPACE-complete? Another question is whether our reduction from 3-Nash to symmetric 3-Nash creates a one-to-one correspondence between solutions of the two problems. If so, intractability of counting 3-Nash solutions will carry over to counting symmetric 3-Nash solutions.

For k-player games, $k \geq 3$, finding an ϵ-approximate Nash equilibrium was shown to be in the class PPAD by [33]. Equilibrium questions that are in this class have admitted complementary pivot algorithms that are practical, e.g., for 2-Nash [73] and for market equilibrium under separable, piecewise-linear concave utility functions [50]. Are there practical algorithms for finding an ϵ-approximate Nash equilibrium in k-player games, $k \geq 3$?

We next come to other results on NE satisfying certain properties for two-player games. First, [32] showed that finding a exact NE that approximately maximizes properties such as social wel-

fare is NP-hard. Next, [59] showed that finding an approximately Nash equilibrium which maximizes the social welfare is as hard as finding a planted clique in a random graph $G(n, 1/2)$, and [1] showed the same hardness for the following three problems: finding an approximate NE with payoff more than v, finding two approximate Nash equilibria that are far apart and finding an approximate NE with a small support. Resolving the complexity of analogous problems for 3-player games is open.

REFERENCES

[1] Per Austrin, Mark Braverman, and Eden Chlamtác. "Inapproximability of NP-complete variants of Nash equilibrium". In: *Theory of Computing* 9 (2013), pp. 117–142.

[2] Moshe Babaioff et al. "On the efficiency of the Walrasian mechanism". In: *ACM Conference on Economics and Computation, EC*. 2014, pp. 783–800.

[3] Yakov Babichenko. "Query complexity of approximate Nash equilibria". In: *Journal of the ACM* 63.4 (2016).

[4] Ashwinkumar Badanidiyuru, Robert Kleinberg, and Yaron Singer. "Learning on a budget: posted price mechanisms for online procurement". In: *Proceedings of the 13th ACM Conference on Electronic Commerce*. ACM. 2012, pp. 128–145.

[5] S. Basu, R. Pollack, and M.-F. Roy. "On the combinatorial and algebraic complexity of quantifier elimination". In: *JACM* 43(6) (1996), pp. 1002–1045.

[6] S. Basu, R. Pollack, and M.-F. Roy. *Quantifier Elimination and Cylindrical Algebraic Decomposition*. Springer, 2004.

[7] M. S. Bazarra, H. D. Sherali, and C. M. Shetty. *Nonlinear Programming: Theory and Algorithms*. John Wiley & Sons, 2006.

[8] Wolfgang W. Bein and Peter Brucker. "Minimum cost flow algorithms for series-parallel networks". In: *Discrete Applied Mathematics* 10.2 (1985), pp. 117–124.

[9] Sayan Bhattacharya et al. "Incentive compatible budget elicitation in multi-unit auctions". In: *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*. SIAM. 2010, pp. 554–572.

[10] Vittorio Bilò and Marios Mavronicolas. "A catalog of EXISTS-R-complete decision problems about Nash equilibria in multi-player games". In: *Proceedings of STACS*. 2016, 17:1–17:13.

[11] Vittorio Bilò and Marios Mavronicolas. "Existential-R-complete decision problems about symmetric Nash equilibria in symmetric multi-player games". In: *Proceedings of STACS*. 2017, 13:1–13:14.

[12] Vittorio Bilò and Marios Mavronicolas. “The complexity of decision problems about Nash equilibria in win-lose games”. In: *Proceedings of SAGT*. 2012, pp. 37–48.

[13] Lenore Blum et al. *Complexity and Real Computation*. Springer-Verlag, 1998.

[14] Christian Borgs et al. “Dynamics of bid optimization in online advertisement auctions”. In: *Proceedings of the 16th international conference on World Wide Web*. ACM. 2007, pp. 531–540.

[15] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2009.

[16] W. C. Brainard and H. E. Scarf. “How to compute equilibrium prices in 1891”. In: *Cowles Foundation Discussion Paper* 1270 (2000).

[17] Eric Budish. “The Combinatorial Assignment Problem: Approximate Competitive Equilibrium from Equal Incomes”. In: *Journal of Political Economy* 119.6 (2011), pp. 1061–1103.

[18] Eric B Budish and Judd B Kessler. “Changing the Course Allocation Mechanism at Wharton”. In: *Chicago Booth Research Paper* 15-08 (2014).

[19] John Canny. “Some algebraic and geometric computations in PSPACE”. In: *Proceedings of STOC*. 1988, pp. 460–467.

[20] Jivitej S Chadha et al. “A competitive algorithm for minimizing weighted flow time on unrelatedmachines with speed augmentation”. In: *Proceedings of the forty-first annual ACM symposium on Theory of computing*. ACM. 2009, pp. 679–684.

[21] Yeon-Koo Che and Ian Gale. “The optimal mechanism for selling to a budget-constrained buyer”. In: *Journal of Economic theory* 92.2 (2000), pp. 198–233.

[22] X. Chen, X. Deng, and S.-H. Teng. “Settling the complexity of computing two-player Nash equilibria”. In: 56(3) (2009).

[23] X. Chen and S.-H. Teng. “Spending is not easier than trading: on the computational equivalence of Fisher and Arrow-Debreu equilibria”. In: *ISAAC: International Symposium on Algorithms and Complexity*. 2009, pp. 647–656.

[24] Xi Chen, Dimitris Paparas, and Mihalis Yannakakis. “The complexity of non-monotone markets”. In: *ACM Symposium on the Theory of Computing*. 2013, pp. 181–190.

[25] X. Chen et al. “Settling the complexity of Arrow-Debreu equilibria in markets with additively separable utilities”. In: *FOCS*. 2009.

[26] Yun Kuen Cheung, Richard Cole, and Nikhil Devanur. “Tatonnement beyond gross substitutes?: gradient descent to the rescue”. In: *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*. ACM. 2013, pp. 191–200.

[27] Bruno Codenotti, Benton McCune, and Kasturi Varadarajan. “Market Equilibrium via the Excess Demand Function”. In: 2005.

[28] Bruno Codenotti, Sriram Pemmaraju, and Kasturi Varadarajan. “On the polynomial time computation of equilibria for certain exchange economies”. In: 2005, pp. 72–81.

[29] B. Codenotti et al. “Leontief economies encode two-player zero-sum games”. In: *SODA*. 2006.

[30] Richard Cole and Lisa Fleischer. “Fast-converging tatonnement algorithms for one-time and ongoing market problems”. In: *Proceedings of the fortieth annual ACM symposium on Theory of computing*. 2008, pp. 315–324.

[31] Richard Cole, Vasilis Gkatzelis, and Gagan Goel. “Mechanism design for fair division: allocating divisible items without payments”. In: *ACM Conference on Electronic Commerce, EC ’13*. 2013, pp. 251–268.

[32] Vincent Conitzer and Tuomas Sandholm. “New complexity results about Nash equilibria”. In: *Games and Economic Behavior* 63.2 (2008), pp. 621–641.

[33] C. Daskalakis, P. W. Goldberg, and C. H. Papadimitriou. “The Complexity of Computing a Nash Equilibrium”. In: 39(1) (2009), pp. 195–259.

[34] Ruchira S. Datta. “Universality of Nash equilibria”. In: 28.3 (2003), pp. 424–432.

[35] Ruchira S. Datta. “Universality of Nash equilibria”. In: *Mathematics of Operations Research* 28.3 (2003), pp. 424–432.

[36] X. Deng and Y. Du. “The computation of approximate competitive equilibrium is PPAD-hard”. In: *Inform. Proc. Letters* 108 (), pp. 369–373.

[37] Xiaotie Deng, Christos Papadimitriou, and Shmuel Safra. “On the Complexity of Equilibria”. In: *Proceedings of ACM Symposium on Theory of Computing*. 2002.

[38] N. Devanur and R. Kannan. “Market Equilibria in Polynomial Time for Fixed Number of Goods or Agents”. In: *FOCS*. 2008, pp. 45–53.

[39] N. Devanur and V. Vazirani. “An improved approximation scheme for computing Arrow-Debreu prices for the linear case”. In: *Proceedings of 23rd FSTTCS*. 2003.

[40] N. Devanur and V. V. Vazirani. “The spending constraint model for market equilibrium: Algorithmic, Existence and uniqueness results”. In: *Proceedings of 36th STOC*. 2004.

[41] N. Devanur et al. “Market Equilibrium via a Primal-Dual Algorithm for a Convex Program”. In: *JACM* 55.5 (2008).

[42] Shahar Dobzinski, Ron Lavi, and Noam Nisan. “Multi-unit auctions with budget limits”. In: *Games and Economic Behavior* 74.2 (2012), pp. 486–503.

[43] Ran Duan and Kurt Mehlhorn. “A combinatorial polynomial algorithm for the linear Arrow-Debreu market”. In: *ICALP*. 2013.

[44] B. C. Eaves. “A Finite Algorithm for the Linear Exchange Model”. In: *Journal of Mathematical Economics* 3 (1976), pp. 197–203.

[45] E. Eisenberg. “Aggregation of Utility Functions”. In: *Management Sciences* 7 (1961), pp. 337–350.

[46] Jon Feldman et al. “Budget optimization in search-based advertising auctions”. In: *Proceedings of the 8th ACM conference on Electronic commerce*. ACM. 2007, pp. 40–49.

[47] Amos Fiat et al. “Single valued combinatorial auctions with budgets”. In: *Proceedings of the 12th ACM conference on Electronic commerce*. ACM. 2011, pp. 223–232.

[48] Jugal Garg, Ruta Mehta, and Vijay V. Vazirani. “Dichotomies in Equilibrium Computation, and Complementary Pivot Algorithms for a New Class of Non-Separable Utility Functions”. In: *STOC*. 2014.

[49] Jugal Garg and Vijay V. Vazirani. “On equilibrium computation in markets with production”. In: *SODA*. 2014.

[50] Jugal Garg et al. “A Complementary Pivot Algorithm for Market Equilibrium under Separable Piecewise-Linear Concave Utilities”. In: 2012.

[51] Jugal Garg et al. “ETR-completeness for decision versions of multi-player (symmetric) Nash equilibria”. In: *Proceedings of ICALP(1)*. 2015, pp. 554–566.

[52] Jugal Garg et al. “Settling the Complexity of Leontief and PLC Exchange Markets Under Exact and Approximate Equilibria”. In: 2017, pp. 890–901.

[53] R. Garg and S. Kapoor. “Auction algorithms for market equilibrium”. In: *Proceedings of 36th STOC*. 2004.

[54] R. Garg, S. Kapoor, and V. V. Vazirani. “An Auction-Based Market Equilbrium Algorithm for the Separable Gross Substitutibility Case”. In: *Proceedings, APPROX*. 2004.

[55] Ali Ghodsi et al. “Dominant Resource Fairness: Fair Allocation of Multiple Resource Types”. In: *Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation*. NSDI’11. Boston, MA, 2011, pp. 323–336.

[56] Itzhak Gilboa and Eitan Zemel. “Nash and correlated equilibria: Some complexity considerations”. In: *Games and Economic Behavior* 1 (1989), pp. 80–93.

[57] Gagan Goel, Vahab Mirrokni, and Renato Paes Leme. “Polyhedral clinching auctions and the adwords polytope”. In: *Journal of the ACM (JACM)* 62.3 (2015), p. 18.

[58] Leslie A Hall et al. “Scheduling to minimize average completion time: Off-line and on-line approximation algorithms”. In: *Mathematics of operations research* 22.3 (1997), pp. 513–544.

[59] Elad Hazan and Robert Krauthgamer. “How hard is it to approximate the best Nash equilibrium?” In: *SIAM Journal on Computing* 40.1 (2011), pp. 79–91.

[60] Li-Sha Huang and Shang-Hua Teng. “On the approximation and smoothed complexity of Leontief market equilibria”. In: *Frontiers in Algorithmics, First Annual International Workshop, FAW*. 2007, pp. 96–107.

[61] Leonid Hurwicz. “On Informationally Decentralized Systems”. In: *Decision and Organization: A Volume in Honor of Jacob Marschak*. Ed. by C. B. McGuire and Roy Radner. Amsterdam: North-Holland, 1972.

[62] K. Jain. “A polynomial time algorithm for computing the Arrow-Debreu market equilibrium for linear utilities”. In: 37.1 (2007), pp. 306–318.

[63] K. Jain, M. Mahdian, and A. Saberi. “Approximating market equilibrium”. In: *Proceedings of 6th APPROX*. 2003.

[64] K. Jain and K. Varadarajan. “Equilibria for economies with production: Constant-returns technologies and production planning constraints”. In: 2006, pp. 688–697.

[65] Kamal Jain and Vijay V Vazirani. “Eisenberg–Gale markets: Algorithms and game-theoretic properties”. In: *Games and Economic Behavior* 70.1 (2010), pp. 84–106.

[66] Albert Xin Jiang and Kevin Leyton-Brown. “Polynomial-time computation of exact correlated equilibrium in compact games”. In: *Games and Economic Behavior* 91 (2015), pp. 347–359.

[67] K. Arrow and G. Debreu. “Existence of an Equilibrium for a Competitive Economy”. In: *Econometrica* 22 (1954), pp. 265–290.

[68] K. Etessami and M. Yannakakis. “On the complexity of Nash equilibria and other fixed points”. In: 39(6) (2010), pp. 2531–2597.

[69] Shizuo Kakutani. “A generalization of Brouwer’s fixed point theorem”. In: *Duke Mathematical Journal* 8.3 (1941), pp. 457–459.

[70] F. P. Kelly and V. V. Vazirani. “Rate Control as a Market Equilibrium”. Unpublished manuscript. 2002.

[71] Jean-Jacques Laffont and Jacques Robert. “Optimal auction with financially constrained buyers”. In: *Economics Letters* 52.2 (1996), pp. 181–186.

[72] Ron Lavi and Chaitanya Swamy. “Truthful mechanism design for multidimensional scheduling via cycle monotonicity”. In: *Games and Economic Behavior* 67.1 (2009).

[73] C. E. Lemke and Jr. J. T. Howson. “Equilibrium Points of Bimatrix Games”. In: *SIAM J. on Applied Mathematics* 12.2 (1964), pp. 413–423.

[74] Yehuda John Levy. “Projections and functions of Nash equilibria”. In: *International Journal of Game Theory* 45.1 (2016), pp. 435–459.

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

[76] Andreu Mas-Colell, Michael Dennis Whinston, and Jerry R Green. *Microeconomic theory*. Vol. 1. Oxford university press New York, 1995.

[77] Andrew McLennan and Rabee Tourky. “Simple complexity from imitation games”. In: *Games and Economic Behavior* 68.2 (2010), pp. 683–688.

[78] Nimrod Megiddo and Christos H Papadimitriou. “On total functions, existence theorems and computational complexity”. In: *Theoretical Computer Science* 81.2 (1991), pp. 317–324.

[79] A. Mehta et al. “AdWords and Generalized On-line Matching”. In: *FOCS*. 2005.

[80] R. Mehta, V. V. Vazirani, and S. Yazdanbod. “Settling Some Open Problems on Symmetric Nash Equilibria”. Manuscript. 2013.

[81] Hervé Moulin. *Fair division and collective welfare*. MIT press, 2004.

[82] S Muthukrishnan, Martin Pál, and Zoya Svitkina. “Stochastic models for budget optimization in search-based advertising”. In: *Algorithmica* 58.4 (2010), pp. 1022–1044.

[83] Roger B. Myerson. “Optimal Auction Design”. In: *Mathematics of Operations Research* 6.1 (1981), pp. 58–73.

[84] John Nash. “Non-cooperative Games”. In: *Annals of Mathematics* 54.2 (Sept. 1951), pp. 289–295.

[85] N. Nisan and A. Ronen. “Algorithmic mechanism design”. In: *Games and Economic Behavior* 35.1–2 (2001), pp. 166–196.

[86] Noam Nisan et al. “Google$x$fffd$x$fffd$x$fffds auction for TV ads”. In: *Automata, Languages and Programming* (2009), pp. 309–327.

[87] J. B. Orlin. “Improved algorithms for computing Fisher’s market clearing prices”. In: *ACM Symposium on the Theory of Computing*. 2010, pp. 291–300.

[88] James B Orlin. “Improved algorithms for computing Fisher’s market clearing prices: computing Fisher’s market clearing prices”. In: *Proceedings of the forty-second ACM symposium on Theory of computing*. ACM. 2010, pp. 291–300.

[89] C. Papadimitriou. “On the complexity of the parity argument and other inefficient proofs of existence”. In: *JCSS* 48(3) (1994), pp. 498–532.

[90] Christos H. Papadimitriou. http://www.cs.berkeley.edu/~christos/agt11/notes/lect3.pdf. 2011.

[91] Christos H. Papadimitriou. “The complexity of finding Nash equilibria”. In: *Chapter 2, Algorithmic Game Theory, eds. N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani* (2007).

[92] Christos H. Papadimitriou and Tim Roughgarden. “Computing equilibria in multi-player games”. In: *Proceedings of SODA*. 2005, pp. 82–91.

[93] A. Rubinstein. Personal communication. 2016.

[94] Aldo Rustichini, Mark A Satterthwaite, and Steven R Williams. “Convergence to efficiency in a simple market with incomplete information”. In: *Econometrica: Journal of the Econometric Society* (1994), pp. 1041–1063.

[95] R. Savani and B. von Stengel. “Exponentially Many Steps for Finding a Nash Equilibrium in a Bimatrix Game”. In: *FOCS*. 2004, pp. 258–267.

[96] H. Scarf. *The Computation of Economic Equilibria*. Yale University Press, 1973.

[97] Marcus Schaefer and Daniel Štefankovič. “Fixed points, Nash equilibria, and the existential theory of the reals”. Theory of Computing Systems. 2015.

[98] A. Schrijver. *Combinatorial Optimization*. Springer-Verlag, 2003.

[99] A. Schrijver. *Theory of Linear and Integer Programming*. John Wiley & Sons, New York, NY, 1986.

[100] J. B. Shoven and J. Whalley. *Applying general equilibrium*. Cambridge University Press, 1992.

[101] Yaron Singer. “Budget feasible mechanisms”. In: *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*. IEEE. 2010, pp. 765–774.

[102] S. Smale. “A convergent process of price adjustment and global Newton methods”. In: *Journal of Mathematical Economics* 3(2) (1976), pp. 107–120.

[103] Vijay V. Vazirani and Mihalis Yannakakis. “Market equilibrium under separable, piecewise-linear, concave utilities”. In: *Journal of ACM* 58.3 (2011), 10:1–10:25.

[104] Laszlo A. Vegh. “Concave generalized flows with applications to market equilibria”. In: *IEEE Annual Symposium on Foundations of Computer Science*. 2012.

[105] Laszlo A. Vegh. “Strongly polynomial algorithm for a class of minimum-cost flow problems with separable convex objectives”. In: *ACM Symposium on the Theory of Computing*. 2012.

[106] László A Végh. “Concave Generalized Flows with Applications to Market Equilibria”. In: *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*. IEEE. 2012, pp. 150–159.

[107] László A Végh. “Strongly polynomial algorithm for a class of minimum-cost flow problems with separable convex objectives”. In: *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*. ACM. 2012, pp. 27–40.

[108] Fang Wu and Li Zhang. “Proportional response dynamics leads to market equilibrium”. In: *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*. ACM. 2007, pp. 354–363.

[109] M. Yannakakis. Personal communication. 2012.

[110] Yinyu Ye. “A path to the Arrow-Debreu competitive market equilibrium”. In: *Math. Program.* 111(1-2) (2008), pp. 315–348.