PPAD-membership for Problems with Exact Rational Solutions: A General Approach via Convex Optimization

Aris Filos-Ratsikas¹, Kristoffer Arnsfelt Hansen², Kasper Høgh², and Alexandros Hollender³

¹University of Edinburgh, UK ²Aarhus University, Denmark ³University of Oxford, UK

Abstract

We introduce a general technique for proving membership of search problems with *exact rational* solutions in PPAD, one of the most well-known classes containing total search problems with polynomial-time verifiable solutions. In particular, we construct a "pseudogate", coined the *linear-OPT-gate*, which can be used as a "plug-and-play" component in a piecewise-linear (PL) arithmetic circuit, as an integral component of the "Linear-FIXP" equivalent definition of the class. The linear-OPT-gate can solve several convex optimization programs, including quadratic programs, which often appear organically in the simplest existence proofs for these problems. This effectively transforms existence proofs to PPAD-membership proofs, and consequently establishes the existence of solutions described by rational numbers.

Using the linear-OPT-gate, we are able to significantly simplify and generalize almost all known PPAD-membership proofs for finding exact solutions in the application domains of game theory, competitive markets, auto-bidding auctions, and fair division, as well as to obtain new PPAD-membership results for problems in these domains.

1 Introduction

Total search problems, i.e., search problems for which a solution is always guaranteed to exist, have been studied extensively over the better part of the last century, in the intersection of mathematics, economics and computer science. Famous examples of such problems are finding Nash equilibria in games [Nash, 1950], competitive equilibria in markets [Arrow and Debreu, 1954] and envy-free divisions of resources [Stromquist, 1980]. While the classic works in mathematics and economics have been primarily concerned with establishing the existence as well as desirable properties of these solutions, the literature of computer science over the past 35 years has been instrumental in formulating and answering questions about the computational complexity of finding them.

More precisely, Megiddo and Papadimitriou [1991] defined the class TFNP to include all total search problems for which a solution is verifiable in polynomial time. To capture the computational complexity of many problems including the aforementioned ones, several subclasses of TFNP were subsequently defined. Among those, one that has been extremely successful in this regard is the class PPAD of Papadimitriou [1994], which was proven to characterize the complexity of computing Nash equilibria in games [Daskalakis et al., 2009; Chen et al., 2009], as well as competitive equilibria for several types of markets [Chen et al., 2017], among many others.

In reality, when making statements like the above, i.e., general statements of the form, "finding a Nash equilibrium is in PPAD", or similarly for a solution to some other total search problem, it is most often meant that what lies in the class is the problem of finding *approximate* solutions. For strategic games

for example, that would mean strategy profiles which are *almost* Nash equilibria, up to some additive parameter ε . This is actually quite often necessary, as it has been shown that for many of these problems, there are cases where all of their solutions can only be described by irrational numbers, and hence we can not hope to compute them exactly on a computer.

Still, there is a large number of important variants of these domains for which *exact rational* solutions exist. For example, several strategic games always have equilibria in rational numbers, and so do certain markets for their competitive equilibria. There are also examples from fair division where rational partitions of the resources can be achieved. In all of those cases, PPAD-membership results for their approximate versions are unsatisfactory; we would like to place the *exact* problems in PPAD instead.

Indeed, coming up with proofs of existence that also guarantee rationality of solutions has been a topic of interest in the area since the very early days, way before the introduction of the relevant computational complexity classes, e.g., see [Eaves, 1976; Lemke and Howson, 1964; Lemke, 1965; Howson, 1972]. Driven by those classic results, a significant literature in computer science has attempted, and quite often has succeeded in placing the corresponding computational problems in PPAD, for several of the application domains mentioned above, including games [Sørensen, 2012; Hansen and Lund, 2018; Kintali et al., 2013; Klimm and Warode, 2020; Meunier and Pradeau, 2019], markets [Vazirani and Yannakakis, 2011; Garg and Vazirani, 2014; Garg, 2017; Garg et al., 2018], as well as the more recent domain of auto-bidding auctions [Chen et al., 2021a].

While these PPAD-membership proofs typically do follow one of a few common approaches, in essence they are rather domain-specific and require reconstructing a set of arguments again for each application at hand (see Section 1.2.1 below for a detailed discussion). Instead, we would like to have *one general technique* for proving PPAD-membership of problems with exact solutions, and ideally one that arises "organically" as the computational equivalent of the standard proofs of existence. To do this, a very promising avenue seems to be via a characterization of PPAD, coined *Linear-FIXP*, due to Etessami and Yannakakis [2010], which defines the class in terms of fixed points of problems represented by piecewise-linear arithmetic circuits. This is because a standard existence proof, e.g., via the Kakutani fixed point theorem [Kakutani, 1941] or via Brouwer's fixed point theorem [Brouwer, 1911], often obtains the solution as a fixed point of a set of local optimization problems, in which each agent or player is independently maximizing a piecewise utility/payoff function. If we could "embed" these optimization problems into a piecewise-linear circuit, that would essentially translate the existence proof into a PPAD-membership proof. This is crisply captured in the following quote from Vazirani and Yannakakis [2011], in the context of proving PPAD-membership for competitive equilibria in certain markets:

"There are very few ways for showing membership in PPAD. A promising approach for our case is to use the characterization of PPAD of Etessami and Yannakakis [2010] as the class of exact fixed-point computation problems for piecewise-linear, polynomial time computable Brouwer functions. [...] Unfortunately, we do not see how to do this [...] it is not clear how to transfer the piecewise-linearity of the utility functions to the Brouwer function." [Vazirani and Yannakakis, 2011].

Recently, Filos-Ratsikas et al. [2023] in fact developed a general technique along those lines: they designed an *optimization gate*, which can be used as part of a circuit to substitute the aforementioned optimization problems and obtain membership results. Crucially however, their membership results are *not* for the class PPAD, but rather for the class FIXP [Etessami and Yannakakis, 2010], a superclass of Linear-FIXP in which the main computational device is a (general) arithmetic circuit, not a piecewise-linear one. These circuits are particularly powerful and can capture solutions with irrationalities. Using their "OPT-gate for FIXP", Filos-Ratsikas et al. [2023] showed the FIXP-membership of several very general problems related to strategic games, markets and fair division.

While FIXP is certainly a natural class, it has not enjoyed the same success as PPAD, even in the context of classifying problems with exact solutions. Besides, in the standard (Turing) model of computation, a FIXP-membership result can be interpreted as finding a point that is *close* to a solution (e.g., in the max norm). This is often a stronger guarantee than an approximate solution as described earlier, but it it still very much only an approximation. Again, this is unsatisfactory for those problems with exact rational solutions that should be in PPAD.

Optimization Program $\mathcal C$	Feasibility Program Q
$\min f(x;c)$	$h_i(y) > 0 \implies a_i^T x \le b_i$
s.t. $Ax \leq b$	$x \in [-R, R]^n$
$x \in [-R, R]^n$	

Figure 1: The optimization programs and feasibility programs that can be solved by the linear-OPT-gate.

Could we hope to use Filos-Ratsikas et al.'s optimization gate to obtain PPAD-membership? This is actually practically impossible, for reasons which are deeply rooted in the definitions of the classes; we highlight those in Section 1.3 below. In short, the power of general arithmetic circuits over piecewise-linear ones lies in their capability to multiply and divide input variables, and this is of vital importance in the design of the OPT-gate for FIXP in [Filos-Ratsikas et al., 2023]. What we really need is *a new gate*, one which avoids such multiplications/divisions and hence can be used in a piecewise-linear arithmetic circuit. Designing such a gate poses significant technical challenges, which we highlight in Section 1.3 and present in more detail in Section 3. Additionally, clearly, the gate cannot capture the generality of applications that the OPT-gate for FIXP does, as, as we said earlier, problems with irrational solutions cannot be in PPAD. It should however be general enough to capture any problem for which exact rational solutions are possible.

This is the main technical contribution of our paper. We introduce the linear-OPT-gate, which can be used as a general purpose tool for proving PPAD-membership of problems with exact rational solutions. We demonstrate its strength and generality on a host of different applications in game theory, markets, auctions and fair division. Via its use, we are able to *significantly simplify* or *generalize* virtually all of the PPAD-membership proofs for problems with exact solutions in the literature, as well as to prove *new* membership results for problems for which PPAD-membership was not known; we offer more details in the following subsection.

1.1 A Powerful Tool for PPAD-membership: The linear-OPT-gate

We introduce the linear-OPT-gate for proving membership of problems in PPAD. The linear-OPT-gate can be used as a "plug-and-play" component in a PL arithmetic circuit, i.e., similarly to any of the other gates $\{+, -, \max, \min, \times \zeta\}$ of the circuit (see Definition 2.3). The gate is guaranteed to work correctly at a fixed point of the function that the circuit encodes, which, for the purposes of proving PPAD-membership of a problem, is equivalent to a standard gate.

The linear-OPT-gate allows us to compute solutions to optimization programs of a certain form, like those shown in the left-hand side of Figure 1. In particular, these are optimization programs with a non-empty and bounded feasible domain given by a set of linear inequalities, and the subgradient of the convex objective function (in the variables x) is given by a PL (piecewise-linear) arithmetic circuit. In particular, the linear-OPT-gate can compute the solution to any linear program, but also to more general convex programs, e.g., those with quadratic objective functions. The inherent strength of the technique lies in the fact that these types of optimization programs arise naturally in several of the applications in game theory, competitive markets and fair division. Now, for the purpose of showing membership in PPAD, they may effectively be substituted by linear-OPT-gates.

From the ability of the linear-OPT-gate to solve optimization programs of the form \mathcal{C} of Figure 1, we can also derive *feasibility programs* with *conditional constraints*, like the program \mathcal{Q} on the right-hand side of Figure 1. These feasibility programs also often arise naturally in the context of existence proofs, and can be also thought of as being solved in a black-box manner by a gate, which is constructed using the linear-OPT-gate.

Our linear-OPT-gate has a wealth of applications, which we discuss below.

¹The term "linear" here refers to piecewise-linear functions, as in the class Linear-FIXP.

1.2 Applications of the linear-OPT-gate

We apply our linear-OPT-gate to a plethora of different domains, and obtain PPAD-membership for finding solutions in several strategic games, competitive markets, auto-bidding auctions, as well as problems in fair division. We detail those applications in the corresponding sections below. Our results achieve the following three desired objectives simultaneously:

- Proofs of existence of solutions.
- Proofs of rationality of solutions.
- PPAD-membership of the corresponding problems.

For some of these domains, PPAD-membership results for the corresponding problems were known; still, the proofs to establish those were often rather involved. With the employment of our linear-OPT-gate, they become *conceptually and technically significantly simpler*. In essence, the linear-OPT-gate allows us to turn a simple existence proof into a PPAD-membership result. For some of our applications such simple existence proofs already existed, and are transformed to PPAD-membership proofs via the linear-OPT-gate. For others, developing these simpler existence proofs is also part of our contribution; we provide more details in the sections below. The linear-OPT-gate also allows us to *straightforwardly* obtain generalizations of some of the known PPAD-membership results, to cases beyond what was known in the literature. Finally, we also obtain the PPAD-membership of some problems whose complexity had not been studied in the literature before.

We summarize our results in Table 1, where we indicate which results were known in the literature before, which are generalizations, and which concern problems for which we did not know any results about their computational complexity.

Before we proceed with the applications, we present the main techniques that have been used previously for proving PPAD-membership results, and highlight the main technical challenges of using those techniques as opposed to the "plug-and-play" nature of our linear-OPT-gate.

1.2.1 Main Previous Approaches

Linear Complementarity Programs and Lemke's Algorithm. The first main approach for establishing rationality of solutions and PPAD-membership is that of *linear complementarity programs (LCPs)* [Cottle and Dantzig, 1968; Cottle et al., 2009]. Given an $n \times n$ matrix \mathbf{M} and a vector \mathbf{q} , an LCP seeks to find two vectors \mathbf{y} and \mathbf{v} satisfying:

$$\mathbf{M} \cdot \mathbf{y} + \mathbf{v} = \mathbf{q}, \ \mathbf{y} \ge 0, \ \mathbf{v} \ge 0, \ \text{and} \ \mathbf{y}^\mathsf{T} \cdot \mathbf{v} = 0$$

The term "complementarity" stems from the fact that in a solution, we may have either $\mathbf{y}_i > 0$ or $\mathbf{v}_i > 0$, but not both. Lemke [1965] designed an algorithm (based on the previously designed Lemke-Howson algorithm [Lemke and Howson, 1964]) to solve LCPs via a series of *complementary pivoting* steps, i.e., steps in which when a variable enters the basis, a complementary variable exits. Interestingly, the algorithm was designed in the context of computing Nash equilibria in bimatrix games, long before the associated computational complexity classes were defined. LCP-based formulations of equilibria and other fixed point problems have in fact been a subject of study in classic works (e.g, see Eaves [1976]; Howson [1972]) as a means to obtain existence proofs that guarantee the rationality of solutions. PPAD membership can be obtained by pairing the algorithm with an appropriate local orientation of its complementarity paths [Todd, 1976].

Quite importantly, Lemke's algorithm terminates with either finding a solution to the LCP, or without finding a solution, in what is referred to as a *secondary ray*. This feature of the algorithm is well-documented (e.g., see [Savani, 2006] for an excellent exposition) and is known as *ray termination*. In terms of proving PPAD-membership, it seems almost inevitable that every PPAD-membership proof that uses this approach has to argue against ray termination. As Garg and Vazirani [2014] pointedly remark, in the context of a succession of papers on equilibrium computation in competitive markets:

"In the progression of these three works, the LCPs have become more involved and proving the lack of secondary rays has become increasingly harder." [Garg and Vazirani, 2014].

This is not particular to markets either. For example, in Hansen and Lund's [2018] generalization of the results of Sørensen [2012] from bimatrix to polymatrix games, those concerning ε -proper equilibria, a new LCP needs to be devised, together with a new argument against ray termination. Additionally, there are often significant challenges in even appropriately formulating the problems in question as LCPs. In some cases, the naive formulations may lead to inefficient representations, e.g., see [Sørensen, 2012]. In other cases, all known formulations lead to *nonstandard* LCPs, which cannot be handled by the "vanilla" version of Lemke's algorithm, and require variants of the algorithm to be devised, e.g., see [Garg et al., 2018; Meunier and Pradeau, 2019]. Finally in some cases, it is not known if the derived LCPs can be solved via any variant of Lemke's algorithm, thus leading to the development of entirely new pivoting algorithms [Klimm and Warode, 2020]. These characteristics of the LCP approach make it somewhat insufficient as a general purpose PPAD-membership technique.

One advantage of LCP-based approaches is that they have been shown to perform well in practice, e.g., see [Garg et al., 2018] and references therein. However, for the purpose of proving PPAD-membership, we do not see any general advantage of the LCP method over our linear-OPT-gate.

Approximation and Rounding. The second general technique that has been used in several applications to prove the PPAD-membership of exact solutions is that of *approximation and rounding*. This generally consists of the two following main steps:

- consider an *approximation* or a *relaxation* of the solution (e.g., ε-approximate equilibria) and prove that the approximate version is in PPAD, and
- devise a rounding procedure to transform approximate solutions to exact solutions, while maintaining membership in the class.

This rather indirect approach certainly suffers in terms of elegance. More importantly however, it is very much domain-specific. First, showing the PPAD-membership for the approximate version typically still requires a non-trivial proof, often even a rather involved one, e.g., via some reduction to one of the well-known problems in PPAD, like END-OF-LINE (see Definition 2.1) or the computational version of Sperner's lemma [Sperner, 1928]. Also, the rounding procedure itself may be rather complicated, and of an ad hoc nature. For certain applications, there is a general linear programming-based technique developed by Etessami and Yannakakis [2010] to transform ε -approximate solutions to exact ones, for sufficiently small values of ε . Still, this does not apply to all problems, and it may need to be used in conjunction with other tailor-made rounding steps, e.g., see [Chen et al., 2021a; Vazirani and Yannakakis, 2011].

The linear-OPT-gate as a "plug-and-play" component. As we will explain in the following, and as it will be evident via inspection of our proofs throughout the paper, the linear-OPT-gate allows us to develop proofs which are very simple and streamlined, essentially mimicking the easiest proofs of existence. Clearly, most of the technical complications are "hidden" in the "inner workings" of the linear-OPT-gate. This is the advantage of having a "plug-and-play" component readily available for the proofs: one does not need to even be concerned about how the linear-OPT-gate works, but only to understand what kind of optimization programs it can solve. We consider this to be a significant advantage over the two aforementioned techniques, which require to devise application-specific arguments (be it arguments about ray termination or appropriate approximation and rounding). These arguments may be of a standard general nature, but they have to be devised anew for each application, as evidenced by all the different PPAD-membership results that employ these techniques.

1.2.2 Implicit Functions and Correspondences

As a final remark before we present our applications, we point out that, via machinery that we develop in Section 3.4, our linear-OPT-gate can be used to show the PPAD-membership of problems for which the

Applications to Game Theory		
Games with PL Best Response Oracles (PLBRO)	[Our Work]	
PL Concave Games	[Our Work]	
Bilinear Games	[Koller et al., 1996], implicitly	
General Threshold Games	[Our Work]	
Bimatrix Games	[Papadimitriou, 1994]	
	[Cottle and Dantzig, 1968], implicitly	
Polymatrix Games	[Howson, 1972], implicitly	
PL Succinct Games	[Our Work]	
Multi-class Congestion games with piecewise-linear latency functions		
Non-atomic Network Congestion Games	[Our Work]	
	linear latencies [Meunier and Pradeau, 2019]	
Atomic Splittable Network Congestion Games	[Our Work]	
	linear latencies [Klimm and Warode, 2020]	
Congestion Games with Malicious Players	[Our Work]	
Other equilibrium notions		
ε-proper Equilibria in Bimatrix Games	[Sørensen, 2012]	
ε-proper Equilibria in Polymatrix Games	[Hansen and Lund, 2018]	
ε-proper Equilibria in PL Succinct Games	[Our Work]	
Personalized Equilibria	[Kintali et al., 2013]	
Applications to Competitive Markets		
Exchange Markets with Linear Utilities	[Eaves, 1976], implicitly	
Arrow-Debreu Markets with SPLC Utilities	[Garg et al., 2015]	
Arrow-Debreu Markets with SPLC Utilities/Productions	[Vazirani and Yannakakis, 2011]	
	[Garg and Vazirani, 2014]	
Arrow Debreu Markets with Leontief-free	Arrow-Debreu Markets with Succinct SPLC	
Utilities/Productions [Garg et al., 2018]	Utilities/SPLC Productions [Our Work]	
Applications to Auto-Bidding Auctions		
Pacing Equilibria in Second-Price Auctions with Budgets	[Chen et al., 2021a]	
Applications to Fair Division		
Envy-free Cake Cutting	[Goldberg et al., 2020], implicitly	
Rental Harmony	[Our Work]	

Table 1: A summary of our PPAD-membership results - for other complementary results please see the respective sections/paragraphs in the introduction. Classes of domains that are within the same frame in the table (i.e., not separated by borders) are of increasing generality from top to bottom. Domains that appear in the same row of a frame are incomparable in terms of their generality. For the applications to game theory, all of the domains are special cases of PL concave games which in turn are a special case of PLBRO games. For those applications, the PPAD-membership extends to *generalized equilibria*. For all of the results in the table, regardless of whether we obtain entirely new results, generalizations, or simply results which were known in the literature, we obtain *significant simplifications* in the proofs.

inputs (e.g., utilities or latency functions) are given *implicitly* in the input. In particular, we show how we can construct PL arithmetic circuits computing these functions, when those are inputted *succinctly* via Boolean circuits. In terms of the applications, this allows us to effectively consider functions of exponential size (in the size of the circuits), e.g., piecewise-linear utility functions with exponentially-many pieces. We provide details on how this capability of the linear-OPT-gate is used in each application in the corresponding sections below. We present applications for which the aforementioned techniques of Section 1.2.1 are *inherently insufficient* for obtaining PPAD-membership results for those implicit functions, when these results are in fact enabled by the use of the linear-OPT-gate.

1.2.3 PPAD-membership for Strategic Games

We start our discussion from the applications of the linear-OPT-gate to the problem of computing (exact) equilibria in strategic games. To provide some initial intuition, before the technical sections of the paper, we provide an informal example of the use of the linear-OPT-gate to compute mixed Nash equilibria in bimatrix games; this is exposed in more detail in Section 4.1.

An Example: Bimatrix Games

A bimatrix game is a game played between two players, in which the payoffs are given by two matrices A_1 and A_2 , one for each player, denoting the payoff of the players when they each choose certain actions. Each player chooses a *mixed strategy*, i.e., a probability distribution over actions in the game, aiming to maximize their expected payoff, against the choice of the opponent. A mixed Nash equilibrium is a pair of mixed strategies for which every player is *best responding*, i.e., she is maximizing her payoff, given the strategy of the other player. The existence of mixed Nash equilibria for bimatrix games follows from Nash's general existence theorem [Nash, 1950]. The proof of the theorem that employs the Kakutani fixed point theorem [Kakutani, 1941] constructs a fixed point of a function F from the domain of mixed strategies to itself, for which each coordinate F_i is a best response for player i in the game. These best responses can be captured by optimization programs of the form C in Figure 1 and in particular for the case of bimatrix games, these are linear programs in which the subgradients of the objective functions are linear functions. The existence proof then immediately yields a PPAD-membership proof if one substitutes those programs with linear-OPT-gates that compute them.

We remark that for bimatrix games, the original PPAD-membership proof of Papadimitriou [1994] adopts the "LCP approach" that we mentioned earlier, i.e., it appeals to an alternative proof of Nash equilibrium existence due to Cottle and Dantzig [1968] (see also [Lemke and Howson, 1964]) that formulates the problem as an LCP. This is a good example of what we mentioned earlier; the linear-OPT-gate allows us to organically retrieve PPAD-membership from the standard, textbook existence proof of Nash [1950].

Best Response Oracles, PL Concave Games and Generalized Equilibria

PL Best Response Oracles. The approach that we highlighted above is not restricted to bimatrix games, but it actually captures a large class of strategic games. In Section 4.3 we provide a technical definition for a very general class of games, in which the best response of each agent is given by an oracle that can be computed by a PL arithmetic circuit. We refer to these games as games with PL best response oracles (PLBRO games). An equilibrium of any PLBRO game can straightforwardly be formulated as a fixed point of a function like the function F above, where each coordinate F_i computes the best response of player i via the oracle. By using linear-OPT-gates as oracles, we immediately obtain PPAD-membership results for a wealth of different games.

PL Concave Games. The class of *concave games* is a very large class of games, studied notably by Rosen [1965] and Debreu [1952]. These are games with continuous strategy spaces, for which the existence of an equilibrium is guaranteed under certain continuity and concavity assumptions on the utility functions. This

was proven by Rosen [1965] but also earlier independently by Debreu [1952], Fan [1952], and Glicksberg [1952], and for that reason the existence result is often referred to as the Debreu-Fan-Glicksberg theorem for continuous games.

In Section 4.3.1 we prove that as long as the supergradient of the (concave) utility function can be computed by a PL arithmetic circuit, concave games are PLBRO games, and hence finding an equilibrium is in PPAD. We refer to those games as PL concave games, and emphasize again that the utility function does not have to be piecewise linear, but only its (super)gradient; in particular, it could for example be a quadratic function. Bimatrix games are PL concave games, and so are *polymatrix games* [Janovskaja, 1968; Howson, 1972], *bilinear games* [Garg et al., 2011], as well as generalizations of (*digraph*) threshold games [Papadimitriou and Peng, 2021], and thus we obtain membership of finding equilibria in all of these games in PPAD. The latter two games have continuous strategy spaces, and thus the equilibria that we compute are pure, whereas for polymatrix games (and as a result, for bimatrix games) we compute equilibria in mixed strategies.

PL succinct games. In fact, we define a large class of games, which generalize polymatrix games, one which we coin PL succinct games. In these games, the expected utility of a player, given a pure strategy j and a mixed strategy \mathbf{x}_{-i} of the other players, can be computed by a PL arithmetic circuit. These are PL concave games, and the PPAD-membership of finding their mixed Nash equilibria is a corollary of the results mentioned above.

We draw parallels between PL succinct games and those defined in Daskalakis et al. [2006] and Papadimitriou and Roughgarden [2008]. Those works define classes of succinct games for which there is an oracle for computing the expected utility of the player. In [Papadimitriou and Roughgarden, 2008], this oracle is referred to as the *polynomial expectation property* and is used to show that correlated equilibria [Aumann, 1974] of games with this property can be computed in polynomial time. In [Daskalakis et al., 2006], it is shown that if the oracle is given by a *bounded division free straight-line program of polynomial length*, then these games are in PPAD. Crucially, this latter result concerns *approximate equilibria*. One could view our result as a complement to those two results, one which concerns *exact* equilibria in *rational* numbers.

Our PPAD-membership result for PL concave games captures the limits of the class of concave games for which rational equilibria exist, and thus membership in PPAD is possible. The only other known complexity results for the general class of concave games are a FIXP-completeness result due to Filos-Ratsikas et al. [2023], and a very recent PPAD-membership result for *approximate* equilibria due to Papadimitriou et al. [2023].

Generalized Equilibrium. Debreu [1952] did not only consider concave games, but in fact a more general equilibrium notion, one in which the strategy space of each player is dependent on the set of strategies chosen by the other players. This was coined a "social equilibrium" by Debreu [1952] (see also Dasgupta and Maskin [2015]) but over the years has been better known by the term *generalized equilibrium*. For our purposes, the dependence on other strategies can be embedded in the constraints of the optimization programs that we use as oracles in PLBRO games, in a way that can be handled by the linear-OPT-gate. As a corollary, we obtain all of the aforementioned PPAD-membership results for generalized equilibria (rather than standard equilibria) as well, see Section 4.3.1. To the best of our knowledge, these are the first PPAD-membership results for generalized equilibria in the literature.

Personalized Equilibria

The notion of *personalized equilibrium* was introduced by Kintali et al. [2013] in the context of games played on hypergraphs, with an equivalent strategic form. Intuitively speaking, these equilibria allow players to "match" their strategies with those of their opponents, without obeying a product distribution. Kintali et al. [2013] showed the PPAD-membership (and as a result, rationality of equilibria) of personalized equilibria via the "relaxation and rounding approach" (see Section 1.2.1). In particular, they first define an approximate version of the problem (the ε -personalized equilibrium), and reduce that problem to END-OF-LINE (see

Definition 2.1 in Section 2), via a relatively involved construction. To obtain PPAD-membership for the exact problem (i.e., when $\varepsilon=0$) Kintali et al. [2013] construct an elaborate argument that appeals to linear programming compactness, by first showing that for sufficiently small ε , ε -personalized equilibria "almost satisfy" the constraints of the linear programs, and then carefully rounding the solution to obtain an exact equilibrium.

The use of the linear-OPT-gate allows us to obtain the PPAD-membership of the problem via an extremely simple argument. Essentially, each player computes their best response via a linear program which is computed by the linear-OPT-gate, which reduces the problem to finding an equilibrium of an PLBRO game, see Section 4.3.2.

ε-proper Equilibria

We also consider an alternative equilibrium notion, that of ε -proper equilibria. This notion was introduced by Myerson [1978] to refine the notion of ε -perfect equilibrium of Selten [1975], and captures situations in which the players can make small mistakes ("trembles") in the choice of their mixed strategies. The PPAD-membership of computing ε -proper equilibria was known for bimatrix games due to Sørensen [2012] and for polymatrix games due to Hansen and Lund [2018]. Both of these works adopt the LCP approach, which means that they need to go through the hassle of establishing the properties of Lemke's algorithm, as discussed in Section 1.2.1 above. Additionally, formulating the problem as an LCP in this case is far from trivial, and requires an extended formulation of the generalized permutahedron due to Goemans [2015], to make sure that the LCP has polynomially-many constraints.

The use of our linear-OPT-gate distinctly avoids all this labor. We formulate the problem of computing a best response for each player (where the best response is defined with respect to the ε -proper equilibrium notion) as a feasibility program of the form Q in Figure 1, which can be solved by the linear-OPT-gate. This essentially renders the game a PLBRO game, and the PPAD-membership follows simply as a corollary of our main theorem for PLBRO games.

Network Congestion Games

Our last application in the area of game theory is to multi-class congestion games. In particular, we will consider two models, non-atomic congestion games and atomic splittable congestion games. In the former case, there is a continuum of players who collectively form a class controlling a certain load allocation to different resources. In the latter case, each class is represented by a single (atomic) player, who controls the load and distributes it to the resources. For both of those settings, we will also consider the subclass of network congestion games, where the strategies can be represented more succinctly using flows over a directed network.

The existence of equilibria in those games was established in classic works, e.g., see [Schmeidler, 1973] or [Milchtaich, 2000], originally via the employment of the Debreu-Fan-Glicksberg theorem [1952] for continuous games, assuming that the latencies on the resources are concave functions. Relevant to us are the works on their computational complexity, namely [Meunier and Pradeau, 2019] (for non-atomic network congestion games) and [Klimm and Warode, 2020] (for atomic splittable network congestion games). Both papers showed the PPAD-membership of finding pure equilibria in their respective settings, when the latency functions are *linear*. We remark that these games are different from atomic (non-splittable) congestion games, for which finding pure Nash equilibria is known to be in the class PLS defined by Johnson et al. [1988].

Meunier and Pradeau obtain their PPAD-membership result via the "LCP approach" mentioned in Section 1.2.1. Interestingly, their LCP formulation turns out to not be amenable to the use of Lemke's algorithm, so they have to devise a "Lemke-like" complementary pivoting algorithm, tailored to their problem. As in the case of Lemke's algorithm, they argue explicitly against ray termination. Klimm and Warode note that in their case, the problem of finding an equilibrium can be formulated as an LCP, however, it is not known or clear whether this LCP can be solved using any known variant of Lemke's algorithm. For that, they devise a rather involved proof, based on a new homotopy method, essentially a new pivoting

algorithm. Their algorithm solves the problem of finding a Nash equilibrium as a system of linear equations involving notions such as *excess flows*, *vertex potentials* and *block Laplacians*. At a very high level, the authors use the excess and potentials to define an undirected version of the END-OF-LINE graph (see Definition 2.1 in Section 2), and the determinant of the block Laplacians to define a unique orientiation of the edges, effectively reducing the problem to END-OF-LINE.

The linear-OPT-gate allows us to avoid all of the technical complications of the proofs of Meunier and Pradeau [2019] and Klimm and Warode [2020] (which are rather involved, especially the latter), and essentially obtain the PPAD-membership for both of these problems as simple corollaries of our main results for PLBRO games or concave games. In fact, we obtain *generalizations* of those PPAD-membership results to games with more general latency functions, notably piecewise-linear latency functions (implicitly or explicitly represented). In exactly the same fashion, we can use the linear-OPT-gate to obtain the PPAD-membership of *congestion games with malicious players*, a setting studied by Babaioff et al. [2009], for which computational complexity results had not been previously proven.

All of our results on congestion games are presented in Section 5.

1.2.4 PPAD-membership for Competitive Markets

We now move on to the application of our technique to the domain of competitive markets. The standard market model in the literature is that of the Arrow-Debreu market [Arrow and Debreu, 1954], where a set of consumers compete for goods endowed by them and other consumers and goods produced by a set of firms. A *competitive equilibrium* of the market is a set of allocations of goods to the consumers, a set of production quantities and a set of prices, such that at those prices, (a) all consumers maximize their individual utilities, (b) all firms produce optimal amounts, and (c) the market clears, i.e., supply equals demand. The existence of an equilibrium for the general market model was established by Arrow and Debreu [1954] via the employment of Debreu's social equilibrium theorem [1952], under some standard assumptions on the utilities of the consumers and the production sets of the firms.

Previous results and proofs. It has been well-known since the early works in the area [Eaves, 1976] that in general Arrow-Debreu markets, competitive equilibria may be irrational. A significant literature, starting with the work of Eaves [1976] aimed at identifying special cases of the Arrow-Debreu market for which *exact rational* solutions are always possible. When computer science took over in this quest, the related question of establishing the PPAD-membership of finding those exact solutions was also brought forward. Most of the PPAD-membership proofs that were developed through the years followed the "LCP approach", see Section 1.2.1. We present them here in succession:

- Eaves [1976] considered the simplest case of exchange markets (no production) with linear utilities for the consumers and devised an LCP that can be solved by Lemke's algorithm. To establish the latter fact, he argued against ray termination, a characteristic of this approach that we emphasized in Section 1.2.1. A PPAD-membership proof is implicit in his result.²
- Garg et al. [2015] considered exchange markets with *separable piecewise-linear concave (SPLC)* utilities, a generalization of linear utilities in which every agent has a piecewise linear concave utility for the amount of a good *j* that she receives, and her total utility for her bundle is additive over goods. The authors proved the PPAD-membership of finding competitive equilibria in those markets via devising an LCP that was "quite complex" [Garg et al., 2015], and naturally had to argue against ray termination, to establish that Lemke's algorithm will terminate on this LCP with a valid solution.
- Garg and Vazirani [2014] considered Arrow-Debreu markets with SPLC utilities as well as SPLC production functions. This is in fact the work from which the quote of Section 1.2.1 is taken. The quote highlights the increasing challenge of developing these LCPs and establishing their successful

²Note that for exchange markets with linear utilities and no production the problem is in fact known to be polynomial-time solvable [Jain, 2007].

termination. Indeed, for this LCP, Garg and Vazirani [2014] devise a set of linear programs, and then use the complementary slackness and their feasibility conditions to develop the LCP needed for production. The non-homogeneity of the resulting LCP for the equilibrium problem is dealt with in a manner which is different from previous works [Eaves, 1976; Garg et al., 2015] and, naturally, since the developed LCP is different, Garg and Vazirani again need to argue against ray termination.

- The most general class of utility/production functions for which a PPAD-membership of exact competitive equilibria was proven is that of *Leontief-free* functions [Garg et al., 2018], which generalize SPLC functions. For this, the authors devise yet another LCP formulation, which turns out to be even more complex than those of previous works. This is because it has to differentiate between "normal" and "abnormal" variables, the latter preventing the employment of Lemke's algorithm. To circumvent this, they exploit some additional structure of their *nonstandard* LCP, and then they also *modify* Lemke's algorithm, to account for the possibility of abnormal variables becoming zero. Finally, as they devise a new LCP, they also have to argue once again against ray termination.

Besides those works, the first work in computer science to prove PPAD-membership for markets with SPLC utilities/productions was [Vazirani and Yannakakis, 2011]. The approach in that paper is not the "LCP approach" but the "approximation and rounding approach" (again, see Section 1.2.1). An issue with this method is that very small changes in the prices may result in drastic changes in the optimal bundles of the consumers, which makes the proof quite challenging. To deal with this, Vazirani and Yannakakis [2011] devise a set of technical lemmas that allow them to "force" certain allocations over others.

Our results. Our results in this section are twofold.

- **Simplified proofs.** First, we employ the linear-OPT-gate to recover all of the aforementioned PPAD-membership results via proofs which are conceptually and technically quite simpler. In particular, we formulate the optimal consumption and the optimal production as linear programs similar to program \mathcal{C} of Figure 1, which can be effectively substituted by linear-OPT-gates in a PL arithmetic circuit. We also apply a standard variable change which was first used by Eaves [1976], and which we refer to as *Gale's substitution*, see Remark 6.2. For the prices, we develop a feasibility program, similar to program \mathcal{Q} of Figure 1. In a fixed point of the circuit, the optimality of consumption and production follows almost immediately by design. The main technical challenge of the proofs lies in arguing the market clearing of the outputted prices, which however still requires a relatively short proof.

To introduce the reader gently to our proof technique, we first apply it to the simple setting of exchange markets with linear utilities in Section 6.1, then to the setting of Arrow-Debreu markets with linear utilities and productions in Section 6.2, and finally to the general case of Arrow-Debreu markets with Leontief-free utilities and productions in Section 6.3.

- **PPAD-membership for** *Succinct* **SPLC (SSPLC) utilities.** In Section 6.4 we introduce a new class of utility functions, which we coin *succinct separable piecewise-linear (SSPLC) utilities*. These are SPLC utilities in which the different segments of the utility function need not be given explicitly in the input (as in the case of (explicit) SPLC utilities), but can be accessed implicitly via a boolean circuit. Effectively, this allows us to *succinctly* represent SPLC functions with exponentially many pieces, where the input size is the size of the given circuits. We remark that the "LCP-approach" developed in the aforementioned papers is inherently limited in providing PPAD-membership results for this class. Indeed, one could formulate the problem as a large LCP in exponentially-many variables, and that would establish the existence of rational solutions. However, this formulation would no longer be a polynomial time reduction (since now we do not have explicit input parameters u_{jk}^i for the utility of each piece) and hence it would not imply the PPAD-membership of the problem. In contrast, using our machinery from Section 3.4 we can make sure that our linear-OPT-gate can be used to obtain PPAD-membership for markets with SSPLC utilities as well. In our result we also add (explicit) SPLC production, which our technique clearly can handle. We provide a discussion on the challenges of extending our results to also capture SSPLC production

functions at the end of Section 6.4. Note that the SSPLC functions and the Leontief-free functions are of incomparable generality (and hence they appear on the same line of Table 1). Whether we can prove PPAD-membership for a class of "succinct Leontief-free functions", which would generalize both settings, is an interesting technical question.

1.2.5 PPAD-membership for Auto-bidding Auctions

Our next application is on the domain of auto-bidding auctions, which has received a lot of attention recently, due to its applicability in real-world scenarios [Balseiro et al., 2021b,c,a; Balseiro and Gur, 2019; Conitzer et al., 2022b,a; Li and Tang, 2023; Chen et al., 2021a; Borgs et al., 2007]. In particular, in Section 7 we consider the settings studied by Conitzer et al. [2022b,a], Chen et al. [2009] and Li and Tang [2023], in which buyers participate in several parallel single-item auctions, via scaling their valuations by a chosen parameter α , called the *pacing multiplier*. The buyers do that while facing constraints on their feasible expenditure, typically provided by budgets or return-on-investment (ROI) thresholds. The objective is to find a *pacing equilibrium*, i.e., pacing multipliers and allocations for the buyers that are consistent with the format of the auction run (e.g, first-price or second-price) and satisfy the expenditure constraints of all the buyers simultaneously. Pacing equilibria have a similar flavor to the competitive equilibria discussed earlier, but are sufficiently different, and thus require separate handling.

Our proof vs the previous approach. We prove that computing pacing equilibria in parallel second-price auctions with budgets is in PPAD. The problem was already known to be in PPAD (in fact, PPAD-complete) by the recent results of Chen et al. [2021a], building on the original existence result of Conitzer et al. [2022b]. Chen et al.'s proof rather heavily applies the "approximation and rounding" paradigm highlighted in Section 1.2.1. In particular, Chen et al. define a (δ, γ) -approximate variant of the pacing equilibrium, where $\delta, \gamma > 0$ are two approximation parameters. Intuitively, this equilibrium corresponds to an "almost equilibrium" (i.e., the expenditure constraints are "almost" satisfied) of an "almost second-price auction" (i.e., an auction in which the set of winners is those with "almost" the highest bid). The authors prove that finding these approximate equilibria is in PPAD, via a reduction to a computational version of Sperner's lemma [Sperner, 1928], and then devise an intrictate rounding procedure to convert (δ, γ) -equilibria into γ -equilibria. The final step in their proof applies the aforementioned technique of Etessami and Yannakakis [2010] (see Section 1.2.1) to further round these equilibria to pacing equilibria (i.e., where $\gamma = 0$).

Our proof employs the linear-OPT-gate and is conceptually and technically much simpler, without needing to use approximations. Instead, we again apply the standard variable change in Gale's substitution (see Remark 6.2) which we also used for the case of competitive markets, to work with the expenditures rather than the allocations directly. From there, we can formulate the task of finding the optimal expenditures as a set of linear programs (one for each buyer), and the pacing multipliers will be obtained as a fixed point solution of a single simple equation. These linear programs can be solved by linear-OPT-gates which essentially establishes the PPAD-membership of the problem. The proof is detailed in Section 7.1.

ROI-constrained buyers. We observe that the existence proof underlying our PPAD-membership proof in this section can in fact almost straightforwardly be modified to yield the existence of pacing equilibria for a different setting in auto-bidding auctions, that of second-price auctions with average return-on-investment (ROI) constraints, studied by Li and Tang [2023]. Li and Tang established the existence of pacing equilibria via a rather indirect proof, which first reduces the problem to a somewhat convoluted concave game and applies the Debreu-Fan-Glicksberg theorem [Debreu, 1952] to obtain Nash equilibrium existence, and then recovers a pacing equilibrium as a limit point of such a Nash equilibrium. This proof in fact closely follows the original proof of Conitzer et al. [2022b] for the budgeted setting, and clearly does not have any implications on the computational complexity of the problem.

Our proof, besides its advantages in terms of simplicity, also allows us for the first time to obtain computational membership results for pacing equilibria in the ROI-constrained buyer case. It turns out that for this setting, all pacing equilibria may be irrational (see Example 7.1 in Section 7.2.3), and hence

membership in PPAD is not possible. Instead, we employ the OPT-gate for FIXP developed by Filos-Ratsikas et al. [2023] to easily transform our existence proof into a FIXP-membership proof.

1.2.6 PPAD-membership for Fair Division

The last applications of our linear-OPT-gate are related to the task of fairly partitioning a resource among a set of agents with different preferences over its parts. In particular, we show the PPAD-membership of computing *exact* envy-free solutions in two fundamental problems, namely *envy-free cake cutting* [Gamow and Stern, 1958] and *rental harmony* [Su, 1999], when the preferences of the agents ensure the existence of rational partitions.

Envy-free cake cutting. The envy-free division of a continuous resource (metaphorically, a "cake") is one of the most fundamental and well-studied mathematical problems of the last century. The origins of the theory of the problem can be traced back to the pioneering work of Steinhaus [1949], with different variants being studied over the years in a large body of literature in mathematics, economics, and computer science; see [Brams and Taylor, 1996; Robertson and Webb, 1998; Procaccia, 2013] for some excellent textbooks on the topic. The existence of an envy-free division was established in 1980 independently by Stromquist [1980], by Woodall [1980], and by Simmons (credited in [Su, 1999]), even when the division is required to be *contiguous*, i.e., when each agent receives a single, connected piece of the resource. These proofs proceed by first establishing the existence of divisions that are *approximately* envy-free and then obtaining exact solutions as limit points of these approximations.

It is known that in general, envy-free divisions might be irrational (e.g., see [Bei et al., 2012], or Example 8.1 for a simpler example), and hence the problem of computing them cannot be in PPAD. Filos-Ratsikas et al. [2023] showed that envy-free cake cutting is in the class FIXP, which, recall, is appropriate for capturing the complexity of such problems. Still, there are interesting cases for which rational divisions always exist. This is the case for example when the agents' preferences are captured by piecewise constant density functions [Goldberg et al., 2020], a class of functions which is general enough to capture many problems of interest. A FIXP-membership result for these variants is unsatisfactory, and we would like to obtain a PPAD-membership result instead.

Without the convenience of using our linear-OPT-gate, one can establish such a membership result via the "approximation and rounding" technique, see Section 1.2.1. Deng et al. [2012] showed that *approximately* envy-free cake cutting is in PPAD, by transforming Simmons' proof into a computational reduction. Goldberg et al. [2020] showed how to "round" the approximate solution to obtain an exact envy-free division for preferences captured by piecewise-constant densities, as long as ε is sufficiently small.

Luckily, the linear-OPT-gate allows us to avoid having to do that, and instead directly obtain a PPAD-membership result without any need for approximations. In particular, we revisit the FIXP-membership proof of Filos-Ratsikas et al. [2023]; similarly to our approach in this paper, they essentially first construct an existence proof for the problem, one which involves a pair of optimization programs, and then substitute those programs with their OPT-gates for FIXP. One might wonder if, by simply following the steps of the proof and substituting those programs with linear-OPT-gates instead, we can recover the PPAD-membership of the problem, for those classes of preferences for which it is possible. This is almost true, apart from the fact that there is a step in their proof that cannot be done in a PL arithmetic circuit.

Still, we manage to substitute that part by a third optimization program, which is in fact a rather simple linear program, and can effectively be substituted by a linear-OPT-gate. This allows us to obtain the PPAD-membership of the problem for the general class of valuation functions (i.e., functions expressing the preferences via numerical values) that can be computed by a PL arithmetic circuit, see Section 8.1, capturing the aforementioned case of valuations with piecewise-constant densities.

Rental harmony. The rental harmony problem, notably studied by Su [1999], is concerned with the partition of rent among a set of tenants which have different preferences over combinations of rooms and rent partitions. In the generality studied by Su [1999], this problem is in fact equivalent to that of finding an

envy-free division of a *chore* among a set of agents. Su's existence proof is inspired by Simmons' proof for envy-free cake cutting, but employs a "dual Sperner labelling" [Sperner, 1928]. Similarly to the proofs for cake-cutting, the proof also appeals to limits of approximate solutions. In contrast to cake-cutting however, computational complexity results about this general version of the problem were not known, not even for approximate partitions.

In Section 8.2, we prove that the problem of finding a solution to rental harmony is in PPAD, as long as the valuations of the tenants for the rent partition are given by PL arithmetic circuits. Interestingly, this is established via very much the same approach as the proof for envy-free cake cutting, thus providing for the first time a unified proof of existence for those two problems. If one goes beyond the aforementioned valuation functions, all rental harmony solutions may be irrational, as we show in Example 8.2. For those cases, we explain how the existence proof can be coupled with the OPT-gate for FIXP of Filos-Ratsikas et al. [2023] to establish the FIXP-membership of the problem.

Computing Envy-free and Pareto-optimal allocations. We remark that very recently Caragiannis et al. [2023] used our linear-OPT-gates to establish that computing probabilistic envy-free and Pareto-optimal allocations of multiple divisible goods is in PPAD.

1.3 The linear-OPT-gate vs the OPT-gate for FIXP

As we mentioned in the introduction, Filos-Ratsikas et al. [2023] were the first to develop an OPT-gate for the computational class FIXP [Etessami and Yannakakis, 2010]. FIXP is the class that captures the complexity of computing a fixed point of an arithmetic circuit, i.e., a circuit over the basis $\{+, -, \max, \min, \div, *\}$ with rational constants, see Definition 2.2. FIXP is a larger class than Linear-FIXP, due to the fact that we can multiply and divide inside the circuit.

The tools that our linear-OPT-gate provides are conceptually very similar to those of the OPT-gate for FIXP of Filos-Ratsikas et al. [2023], in that they can substitute convex optimization programs within existence proofs, when constructing a circuit whose fixed points are the solutions that we are looking for. However, the design of the gate itself is much more challenging.

The reason for this is the absence of the general multiplication gate *. While we can multiply any two circuit variables in a general arithmetic circuit, we can only multiply variables by constants in a PL arithmetic circuit. The construction of the OPT-gate for FIXP by Filos-Ratsikas et al. [2023] makes extensive usage of the multiplication gate * and can thus not directly be used for creating the linear-OPT-gate. In our case, the constraint matrix A is fixed (i.e., not an input to the linear-OPT-gate) and this does help to eliminate some of the general multiplication gates, but not all of them. At a high level, the construction of Filos-Ratsikas et al. [2023] ensures that the output x of the gate satisfies

$$\mu_0 \cdot \partial f(x) + A^\mathsf{T} \mu = 0$$

where μ satisfies some standard KKT conditions. If x is feasible and if $\mu_0 > 0$, then it follows that x is an optimal solution by standard arguments (using the convexity of f). The term μ_0 is carefully constructed as a function of μ and x in order to ensure that x must be feasible and that $\mu_0 > 0$ when x is feasible. However, since both μ_0 and $\partial f(x)$ depend on x, in our case we cannot construct the term $\mu_0 \cdot \partial f(x)$, because that would entail multiplying two variables in the circuit. As a result, our construction instead ensures that the output x of the gate satisfies

$$\varepsilon \cdot \partial f(x) + A^{\mathsf{T}} \mu = 0$$

where μ again satisfies some standard KKT conditions, and where $\varepsilon > 0$ is some sufficiently small constant that is picked when constructing the gate. By standard arguments it still holds that if x is feasible, then it is an optimal solution. The challenge however is to ensure that x is indeed feasible. While the argument is relatively straightforward in the work of Filos-Ratsikas et al. [2023], because μ_0 can depend on x, here μ_0 has been replaced by a constant ε . Our main technical contribution in the construction of the linear-OPT-gate is to show that there exists a sufficiently small $\varepsilon > 0$ that forces x to be feasible, and that such ε can be

constructed efficiently given the parameters of the gate (but, importantly, not its inputs!). As a bonus, our modified construction and analysis allows us to obtain a linear-OPT-gate that does not require any constraint qualification, whereas the construction of Filos-Ratsikas et al. [2023] required an explicit Slater condition (which of course, as they show, is necessary in the case where the matrix *A* is not fixed).

From the standpoint of applications, the linear-OPT-gate can be used in almost the same direct manner as the OPT-gate for FIXP of Filos-Ratsikas et al. [2023]. In some cases, precisely because we cannot multiply within a PL arithmetic circuit, we may have to apply some standard variable changes, to "linearize" certain constraints. Still, the linear-OPT-gate can effectively substitute appropriate optimization programs in the same way that the OPT-gate for FIXP can. In a nutshell, one can view the linear-OPT-gate as a more powerful tool for those applications for which rational exact solutions exist.

1.4 Organization of the Paper

In Section 2 we provide the main definitions and terminology needed for our paper. In Section 3, we detail the construction of the linear-OPT-gate, and prove its correctness. In the same section (Section 3.4), we also develop the necessary machinery to show how the linear-OPT-gate can be used to obtain PPAD-membership of problems where certain functions are given implicitly in the input to the problem. In Section 4, we provide the first applications of the linear-OPT-gate to several important classes of games and to different equilibrium notions, besides Nash equilibria. In Section 5 we explain how to apply the machinery that we develop in Section 4 in order to obtain PPAD-membership results for equilibrium computation in nonatomic and atomic splittable congestion games. In Section 6, we present the applications of our gate to finding competitive equilibria in Arrow-Debreu markets with different utility and production functions. In Section 7 we demonstrate the applicability of the linear-OPT-gate to obtain membership results for the auto-bidding auctions with pacing strategies. In Section 8 we obtain membership results for the two fundamental fair division problems of envy-free cake cutting and rental harmony. We offer some discussion and some directions for future work in Section 9.

We would like to emphasize that while our paper is very long, this is almost exclusively due to the fact that it covers so many applications, rather than due to the proofs that we develop for those applications, which in reality range from being very short to relatively short. For each of all of the domains that we consider, (a) we provide the appropriate definition and place the setting in context within the rest of the paper, (b) we discuss the related work and possibly the previous PPAD-membership results (if any), (c) we provide detailed comparisons with those previous proofs to demonstrate the effectiveness of our linear-OPT-gate as a general-purpose proof technique, and finally (d) we develop the proofs themselves. In some cases in fact, we first apply the technique to simpler settings for a gentle introduction, and then move on to study those settings in their full generality. We believe that all of our application sections are largely self-contained, and can be read almost in isolation, even after only reading the introduction of the paper, and by referring only to certain clearly referenced parts in other sections.

2 Preliminaries

In this section, we introduce the computational class PPAD, as well as the main machinery that will be used throughout the paper. The details for the specific settings that we will consider in our applications will be defined in the corresponding sections. We start with the definitions of the relevant computational complexity classes.

2.1 The class PPAD

All of the problems that we will consider in this paper will be *total search problems*. A total search problem is one in which a solution is always guaranteed to exist. For example, finding a Nash equilibrium in a game is a total search problem, by Nash's theorem [Nash, 1950]. Similarly, competitive equilibria in markets always

exist (e.g., see [Arrow and Debreu, 1954]). The class TFNP [Megiddo and Papadimitriou, 1991] contains all total search problems *in NP*, i.e., those for which a candidate solution can be verified in polynomial time. For example, verifying whether a given set of strategies is a Nash equilibrium in a bimatrix game (see Definition 4.1) can be done in polynomial time, and hence the problem of finding Nash equilibria in bimatrix games is in TFNP. For a formal definition of the class TFNP, we refer the reader to [Papadimitriou, 1994].

The class PPAD, introduced by Papadimitriou [1994], is defined with respect to its canonical problem, called End-of-Line, see Definition 2.1 below. PPAD is the class of all problems in TFNP that are polynomial-time reducible to End-of-Line.

Definition 2.1 (END-OF-LINE). The END-OF-LINE problem is defined as: given Boolean circuits P and S with n input bits and n output bits such that $P(0) = 0 \neq S(0)$, find x such that $P(S(x)) \neq x$ or $S(P(x)) \neq x \neq 0$.

Intuitively, PPAD captures the following problem. We are given a directed graph in which every node has indegree and outdegree at most 1 and a source of this graph, and we are asked to find another source or a sink. Such a node exists by the parity argument on the degrees of the nodes, which is the underlying principle of the class PPAD. Importantly, we are not given this graph explicitly in the input (otherwise the problem would be trivially in P), but we can access the predecessor and the successor of a given node via Boolean circuits; these are the circuits *P* and *S* of Definition 2.1 above. We will be using an alternative definition of the class, via PL arithmetic circuits, which we will define in Section 2.2 next.

2.2 The classes FIXP and Linear-FIXP

We start by defining arithmetic circuits and PL arithmetic circuits.³

Definition 2.2 (Arithmetic Circuit). An *arithmetic circuit* is a circuit using gates in $\{+, -, *, \div, \max, \min\}$ as well as rational constants.

A PL arithmetic circuit is simply an arithmetic circuit were multiplication and division are not allowed.

Definition 2.3 (PL Arithmetic Circuit). A PL arithmetic circuit is a circuit using gates in $\{+, -, \max, \min, \times \zeta\}$ as well as rational constants, where $\times \zeta$ denotes multiplication by a rational constant.

We will use PL arithmetic circuits to provide an alternative definition of the class PPAD. First, we state and prove the next simple lemma, which will be useful in Section 3. For a rational number a, we let size(a) denote the number of bits needed to describe a in the standard representation, where a is written as an irreducible fraction, and the numerator and denominator are written in binary. We let size(f) denote the number of bits needed to describe a PL arithmetic circuit f (in particular, this includes the length of the description of any constants used in f).

Lemma 2.1. For any PL arithmetic circuit $f: \mathbb{R}^n \to \mathbb{R}^m$ and any rational $B \ge 0$ it holds that

$$\max_{x \in [-B,B]^n} \|f(x)\|_{\infty} \le 2^{\text{poly(size}(B),\text{size}(f))}.$$

Proof. Since a PL arithmetic circuit can be evaluated efficiently (see, e.g., [Fearnley et al., 2022, Lemma 3.3]), we have $||f(0)||_{\infty} \leq 2^{\text{poly}(\text{size}(f))}$. Additionally, f is L-Lipschitz-continuous over \mathbb{R}^n with Lipschitz constant $L = 2^{\text{poly}(\text{size}(f))}$, see, e.g., [Fearnley et al., 2022, Lemma A.1]. As a result, for any $x \in [-B, B]^n$

$$||f(x)||_{\infty} \le ||f(0)||_{\infty} + L||x||_{\infty} \le 2^{\text{poly(size}(B),\text{size}(f))}.$$

³Sometimes in the literature, these are also referred to as *algebraic circuits*, e.g., see [Filos-Ratsikas et al., 2023]. We use the term "PL arithmetic circuit" for circuits over the basis $\{+, -, \max, \min, \times \zeta\}$ with rational constants. Some recent works call them "linear" arithmetic circuits instead [Deligkas et al., 2021; Fearnley et al., 2022].

We now move on to the definition of the related computational classes, in the context of arithmetic circuits. We mentioned the class FIXP in the introduction; we proceed to formally define it below. The following definitions follow those of [Filos-Ratsikas et al., 2023].

A search problem Π with real-valued search space is defined by associating to any input instance I (encoded as a string over a finite alphabet Σ) a search space $D_I \subseteq \mathbb{R}^{d_I}$ and a set of solutions Sol(I). We assume there is a polynomial time algorithm that given I computes a description of D_I .

Next, we define basic FIXP problems and basic Linear-FIXP problems.

Definition 2.4 (Basic (linear)-FIXP problem). A search problem Π is a basic (Linear)-FIXP problem if every instance I describes a nonempty compact convex domain D_I described by a set of linear inequalities with rational coefficients and a continuous map $F_I \colon D_I \to D_I$ given by a (PL) arithmetic circuit C_I , and the solution set is $Sol(I) = \{x \in D_I \mid F_I(x) = x\}$. We assume that C_I well-defined, i.e., it does not divide by zero and that it indeed represents a function F_I with $F_I(D_I) \subseteq D_I$.

Next, we define reductions between search problems. Let Π and Γ be search problems with real-valued search space. A many-one reduction from Π to Γ is a pair of maps (f,g). The instance mapping f maps instances I of Π to instances f(I) of Γ , and for any solution $y \in \operatorname{Sol}(f(I))$ the solution mapping g maps the pair (I,y) to a solution $g(I,y) \in \operatorname{Sol}(I)$ of Π . In order to avoid meaningless reductions, it is required that $\operatorname{Sol}(f(I)) \neq \emptyset$ if $\operatorname{Sol}(I) \neq \emptyset$. We require that the instance mapping f is computable in polynomial time. Etessami and Yannakakis [2010] defined the notion of SL-reductions where the solution mapping g is separable linear. This means there exists a map π : $\{1,\ldots,d_I\} \to \{1,\ldots,d_{f(I)}\}$ and rational constants $a_i,b_i,$ $i=1,\ldots,d_I$, such that for $g\in\operatorname{Sol}(f(I))$ one has that $g\in G(I,g)$ is given by $g\in G(I,g)$ for all $g\in G(I,g)$ in polynomial time.

We now define the class FIXP.

Definition 2.5 (FIXP). The class FIXP consists of all search problems with real-valued search space that SL-reduce to a basic FIXP problem for which the domain D_I is a convex polytope described by a set of linear inequalities with rational coefficients and the function F_I is defined by an arithmetic circuit C_I .

The class Linear-FIXP is the "piecewise-linear fragment of FIXP" [Etessami and Yannakakis, 2010], defined below.

Definition 2.6 (Linear-FIXP). The class Linear-FIXP consists of all search problems with real-valued search space that SL-reduce to a basic Linear-FIXP problem for which the domain D_I is a convex polytope described by a set of linear inequalities with rational coefficients and the function F_I is defined by a PL arithmetic circuit C_I .

Above we have formally defined the class Linear-FIXP as a class of search problems with real-valued search space forming a subclass of FIXP. However, Linear-FIXP also naturally defines a subclass of TFNP since for any instance I of a basic Linear-FIXP problem Π , Sol(I) always contains rational-valued solutions of polynomial bit-length. Following the convention in the literature, we will denote the class of search problems in TFNP reducible to the exact fixed-point computation defined by Π by Linear-FIXP as well.

With this convention, Etessami and Yannakakis [2010] showed the following equivalence result:

Theorem 2.1 ([Etessami and Yannakakis, 2010]). Linear-FIXP = PPAD.

Theorem 2.1 provides an alternative definition of PPAD which we will be using throughout the paper. Roughly speaking, to show that a problem is in PPAD, it suffices to show that it can be reduced to computing a fixed point of a function encoded by a PL arithmetic circuit.

⁴Given an arithmetic circuit, it is not clear how to check whether it does satisfy these properties, so we assume that it does, i.e., we consider promise problems, see also [Filos-Ratsikas et al., 2023]. For the case of basic Linear-FIXP problems, the first condition is trivially satisfied (since division is not allowed), but for the second condition we still require a promise. Note that this means that the problem is formally not a TFNP problem, but instead a promise-TFNP problem. This is however not an issue for proving PPAD-membership, since the problem ultimately reduces to the TFNP problem End-of-Line [Etessami and Yannakakis, 2010].

Remark 2.1 (Linear vs PL). The term "linear" in "linear-FIXP" might be a bit misleading, as it may suggest that it only refers to linear functions. From Definition 2.3, it should be obvious that they capture piecewise-linear functions instead, and are hence more general. We believe that the term "linear" was used in the related literature rather than "piecewise-linear" for succinctness and brevity. In this paper, we will call these arithmetic circuits PL arithmetic circuits (for "piecewise-linear") but still refer to the piecewise-linear fragment of FIXP as Linear-FIXP for consistency with the literature. We also call our new gate "linear-OPT-gate" to remain consistent with Linear-FIXP.

We conclude the section with a very useful definition, that of *PL pseudo-circuits*.

Definition 2.7 (PL pseudo-circuit). A PL pseudo-circuit with n inputs and m outputs is a PL arithmetic circuit $F: \mathbb{R}^n \times [0,1]^\ell \to \mathbb{R}^m \times [0,1]^\ell$. The output of the PL pseudo-circuit on input \mathbf{x} is any \mathbf{y} that satisfies $F(\mathbf{x},\mathbf{z}) = (\mathbf{y},\mathbf{z})$ for some $\mathbf{z} \in [0,1]^\ell$. Note that a PL pseudo-circuit can have multiple possible outputs.

Intuitively, PL pseudo-circuits are PL arithmetic circuits which are only required to work correctly at a fixed point of the encoded function (or, to be more precise, when its "auxiliary" variables **z** satisfy a fixed point condition). In particular, PL arithmetic circuits are PL pseudo-circuits, and in fact, for the purpose of proving membership in PPAD, those two are equivalent. The linear-OPT-gates that we will define in the next section are in fact PL pseudo-circuits that are used as *primitives* or *subroutines* in larger PL pseudo-circuits. The following is an example of a simple PL pseudo-circuit, that was already used as an important primitive by Filos-Ratsikas et al. [2023].

Example 2.1 (PL pseudo-circuit computing the Heaviside function). The Heaviside function is the following correspondence

$$H(x) = \begin{cases} 1 & \text{if } x > 0 \\ [0, 1] & \text{if } x = 0 \\ 0 & \text{if } x < 0 \end{cases}.$$

We can construct a PL pseudo-circuit $F : \mathbb{R} \times [0,1] \to \mathbb{R} \times [0,1]$ computing H by letting

$$F(x,z) := (z, \min\{1, \max\{0, z + x\}\}).$$

It is easy to check that *F* indeed computes H, i.e., $F(x,z) = (y,z) \implies y \in H(x)$.

Computing a fixed point of a PL pseudo-circuit corresponds to computing a fixed point of the PL arithmetic circuit representing it. Thus, a PL pseudo-circuit is guaranteed to have at least one rational fixed point, and the problem of computing such a fixed point lies in PPAD.

3 A Powerful Tool for PPAD-membership: The linear-OPT-gate

In this section, we develop our main tool, the linear-OPT-gate.

Definition 3.1 (linear-OPT-gate). The linear-OPT-gate is a gate which:

- is parameterized⁵ by $n, m, k \in \mathbb{N}$, a rational matrix $A \in \mathbb{R}^{m \times n}$, and a PL arithmetic circuit $G_{\partial f}$: $\mathbb{R}^n \times \mathbb{R}^k \times [0,1]^\ell \to \mathbb{R}^n \times [0,1]^\ell$.
- takes as input $b \in \mathbb{R}^m$, $c \in \mathbb{R}^k$, and $R \in \mathbb{R}$.

The linear-OPT-gate outputs an optimal solution of the following optimization problem (over variables $x \in \mathbb{R}^n$):

⁵The parameters of a gate determine its behavior and must be provided every time a gate of this type is used in a circuit (and thus also count towards the representation size of the circuit). For example, whenever we use a "multiplication by a constant" gate $\times \zeta$, we have to specify the constant parameter ζ of the gate. The same also applies to the linear-OPT-gate, except that it has (many) more parameters.

Optimization Program $\mathcal C$

min
$$f(x;c)$$

s.t. $Ax \le b$ (1)
 $x \in [-R,R]^n$

whenever the two following conditions hold for the given inputs $b \in \mathbb{R}^m$, $c \in \mathbb{R}^k$, and $R \in \mathbb{R}$:

- 1. The feasible domain $\{x \in [-R, R]^n : Ax \leq b\}$ is not empty.
- 2. The map $x \mapsto f(x;c)$ is a convex function on the feasible domain and its subgradient is given by the PL pseudo-circuit $G_{\partial f}$.

If the two conditions are not satisfied then the linear-OPT-gate can have arbitrary output. If the conditions are satisfied and \mathcal{C} has multiple optimal solutions, then any such optimal solution is an acceptable output for the gate.

The following theorem is our main result. It shows that linear-OPT-gates can be simulated by standard gates by adding some auxiliary inputs and outputs to the circuit.

Theorem 3.1. When constructing a PL arithmetic circuit for the purpose of proving membership in PPAD, we can also use linear-OPT-gates, in addition to the standard gates. More formally, given a PL arithmetic circuit⁶ $F: D \to D$ that uses linear-OPT-gates, we can construct in polynomial time a PL arithmetic circuit $G: D \times [0,1]^t \to D \times [0,1]^t$ that does not use linear-OPT-gates, but which is equivalent to F, in the following sense:

$$G(y,\alpha) = (y,\alpha) \implies F(x) = x$$

for all $x \in D$ and $\alpha \in [0,1]^t$.

Remark 3.1. Note that G must have a fixed point by Brouwer's fixed point theorem and the problem of computing such a fixed point lies in PPAD by Theorem 2.1. By Theorem 3.1 it follows that F must also have a fixed point and computing one also lies in PPAD. The expression "F(x) = x" in the theorem should be understood as: "On input x, there exists a valid assignment to all the gates of F such that the output of F is x". The value assigned to a standard gate is fully specified by its inputs. However, for a linear-OPT-gate there might be multiple acceptable assignments (e.g., if it solves an LP that has multiple solutions), and the expression "F(x) = x" states that there exists at least one acceptable assignment such that the output of F is F(x) = x0. It does not say that any acceptable assignment will work, but merely that there exists one that does.

The proof of Theorem 3.1 can be found in Section 3.3. We continue with some additional remarks about the linear-OPT-gate.

Remark 3.2. Note that the objective function f does not need to be computable by a PL arithmetic circuit. We only require that its subgradient (on the feasible domain) is given as a PL pseudo-circuit. In particular, f can be a (convex) quadratic polynomial.

In most of our applications, it will suffice for optimization program \mathcal{C} to be a linear program, i.e., for f to be linear function. We will use \mathcal{P} to refer to the general form of this linear program, and we will reference that in our applications.

Linear Program
$$\mathcal{P}$$

min
$$c^{\mathsf{T}}x$$

s.t. $Ax \le b$
 $x \in [-R, R]^n$ (2)

⁶Here *D* is a nonempty compact convex domain represented by linear inequalities.

Note that here $f(x;c) = c^{\mathsf{T}}x$, and its subgradient is simply computed by the PL pseudo-circuit $G_{\partial f}$: $\mathbb{R}^n \times \mathbb{R}^k \to \mathbb{R}^n$, $(x;c) \mapsto c$. The PL pseudo-circuit $G_{\partial f}$ is in fact just a normal PL arithmetic circuit here, i.e., $\ell = 0$, since no auxiliary fixed point variables are needed to compute the subgradient.

Remark 3.3. The linear-OPT-gate can of course also solve maximization problems max f(x) where the objective function f is concave, since this is equivalent to the problem min -f(x). In that case we have to provide a PL pseudo-circuit computing the supergradient of f, or equivalently the subgradient of -f.

Remark 3.4. Note that we require that the constraint matrix *A* be fixed, whereas the right-hand side of the constraints *b* can be given as an input to the gate. This is in fact necessary, as the following example shows. If the linear-OPT-gate could solve the LP

min
$$x$$

s.t. $a \cdot x \ge 1$
 $x \in [-2, 2]$

where a is not fixed, then we would obtain a PL pseudo-circuit computing 1/a for $a \in [1,2]$. But then, we would be able to construct a PL pseudo-circuit $F: [1,2] \to [1,2]$ computing $y \mapsto \min\{2, \max\{1,2/y\}\}$. The only fixed point of F is at $y = \sqrt{2}$, which is a contradiction, since PL pseudo-circuits always have at least one rational fixed point.

3.1 Feasibility Program with Conditional Constraints

Using the linear-OPT-gate, we can also solve feasibility programs, which will be very useful throughout our applications. In particular, when constructing a PL arithmetic circuit *for the purpose of proving membership in PPAD*, we can assume without loss of generality that we have access to an additional gate solving feasibility programs with conditional constraints, which:

- is parameterized by $n, m, k \in \mathbb{N}$, a rational matrix $A \in \mathbb{R}^{m \times n}$, and PL arithmetic circuits $h_i : \mathbb{R}^k \to \mathbb{R}$ for i = 1, ..., m.
- takes as input $b \in \mathbb{R}^m$, $y \in \mathbb{R}^k$, and $R \in \mathbb{R}$.

The gate outputs a feasible solution of the following feasibility problem (over variables $x \in \mathbb{R}^n$):

Feasibility Program Q

$$h_i(y) > 0 \implies a_i^\mathsf{T} x \le b_i$$

 $x \in [-R, R]^n$ (3)

whenever it is feasible. Note that we may add *unconditional* constraints to feasibility program Q above by simply setting $h_i(y) = 1$ in the conditional constraints above.

Construction. We can solve the feasibility problem Q in (3) by solving the following optimization problem (over variables $x \in \mathbb{R}^n$)

$$\min \sum_{i=1}^{m} \max\{0, h_i(y)\} \cdot \max\{0, a_i^{\mathsf{T}} x - b_i\}$$
s.t. $x \in [-R, R]^n$ (4)

Note that if x is an optimal solution for (4) with objective function value 0, then x is feasible for Q in (3). Furthermore, if Q is feasible, then the optimal value of (4) is 0. Thus, if Q is feasible, then any optimal solution to (4) will also be a feasible solution to Q in (3).

As a result, it suffices to show that we can use the linear-OPT-gate to solve (4). Clearly, the feasible domain of (4) is nonempty. Thus, it remains to show that we can construct a PL pseudo-circuit computing the subgradient of $x \mapsto f(x; y, b)$, where

$$f(x;y,b) = \sum_{i=1}^{m} \max\{0, h_i(y)\} \cdot \max\{0, a_i^{\mathsf{T}} x - b_i\}.$$

Note that this function is indeed convex in x.

The subgradient of $x \mapsto \max\{0, a_i^\mathsf{T} x - b_i\}$ can be expressed as $H(a_i^\mathsf{T} x - b_i) \cdot a_i$, where we recall that H is the Heaviside function defined as

$$H(z) = \begin{cases} 1 & \text{if } z > 0 \\ [0, 1] & \text{if } z = 0 \\ 0 & \text{if } z < 0 \end{cases}.$$

Thus, the subgradient of $x \mapsto f(x;y,b)$ can be written as $\sum_{i=1}^m \max\{0,h_i(y)\} \cdot \operatorname{H}(a_i^\mathsf{T} x - b_i) \cdot a_i$. For each $i \in [m]$, we can compute the term $\max\{0,h_i(y)\} \cdot \operatorname{H}(a_i^\mathsf{T} x - b_i)$ by using Lemma 3.1 below. Since the vectors a_i are fixed, we can then compute the product $\max\{0,h_i(y)\} \cdot \operatorname{H}(a_i^\mathsf{T} x - b_i) \cdot a_i$. Doing this for every $i \in [m]$ and then summing up yields an element in the subgradient of $x \mapsto f(x;y,b)$. Thus, we have successfully constructed a PL pseudo-circuit computing this subgradient. It remains to prove the following lemma we used, which will also be useful later.

Lemma 3.1. For the purpose of proving PPAD-membership, we can construct a PL pseudo-circuit computing $(x,y) \mapsto H(x) \cdot y$.

Proof. Note that $H(x) \cdot y$ can be obtained by computing $H(x) \cdot \max\{0, y\} - H(x) \cdot \max\{0, -y\}$. Thus, it suffices to prove that we can compute $H(x) \cdot \max\{0, y\}$. This can indeed be achieved by using the linear-OPT-gate to solve the following LP (in variable $v \in \mathbb{R}$):

$$\max v \cdot x$$
s.t. $0 \le v \le \max\{0, y\}$

Note that the feasible domain is nonempty, and the gradient of the objective function is x, which can trivially be computed by a PL arithmetic circuit. It is straightforward to verify that any optimal solution v satisfies $v \in H(x) \cdot \max\{0, y\}$, as desired.

3.2 Using the linear-OPT-gate in applications

In our applications in subsequent sections we will be constructing PL arithmetic circuits containing several linear-OPT-gates, corresponding to multiple optimization programs like the program \mathcal{C} above, as well as feasibility programs like the program \mathcal{Q} . It will be helpful to be able to reference the inputs to those linear-OPT-gates as opposed to the variables of the corresponding programs, particularly because variables for one program would be inputs to the linear-OPT-gate corresponding to another program and vice versa. We will use the term *gate inputs* to refer to those inputs.

Definition 3.2 (Gate Inputs). Consider an optimization program in the form of C or a feasibility program in the form of Q and let C be its corresponding linear-OPT-gate. We will refer to the inputs of C as *gate inputs* of the program C or Q.

Using this terminology, we can argue that a specific program can be solved by the linear-OPT-gate as follows.

For optimization programs of the form C we need to argue

- Conditions 1 and 2 in the definition of the linear-OPT-gate for \mathcal{C} above, namely that the domain is non-empty and that the subgradient of the convex objective function is given by a PL pseudo-circuit $G_{\partial f}$,
- that gate inputs appear only on the right-hand side of the constraints, but not on the left-hand side.

For feasibility programs of the form Q we need to argue that

- the feasibility program is solvable (i.e., feasible),
- the gate inputs appear only on the right-hand side of the constraints $a_i^T x \leq b_i$,
- only gate inputs appear on the left-hand side of the conditional constraints, i.e., in the function $h_i(y) > 0$.

The conditions above are obviously equivalent to the optimization and feasibility programs having the form of (1) and (3), since the gate inputs are the inputs to the linear-OPT-gate. Here the conditions are simply "spelled-out", because it is easier to refer to them in subsequent sections. What is not obvious is how one may argue about the solvability of a feasibility program Q.

Solvability of Q**.** The feasibility programs that will appear in most of our applications will have the following same general form; it will be easy to argue the solvability of those that do not.

Feasibility Program Q_{app}

$$h_k(y) - h_{k'}(y) > 0 \implies w_k \le \rho \cdot w_{k'} \text{ for all } k, k' \in [m]$$

$$\sum_{i=1}^m w_i = 1, \quad w_i \ge \frac{\rho^m}{m}, \text{ for all } i \in [m]$$

for some $0 < \rho \le 1$, where $w \in \mathbb{R}^m$ are the variables. For this type of feasibility program Q_{app} , we can define the notion of a *feasibility graph*.

Definition 3.3 (The feasibility graph G_Q). Consider a feasibility program of the form Q_{app} . Let $G_{Q_{app}}$ be the graph that has nodes for each $k \in [m]$, and a directed edge (k, k') if and only if $h_k(y) - h_{k'}(y) > 0$. We will refer to $G_{Q_{app}}$ as the *feasibility graph* of Q_{app} .

The following lemma provides a general condition for solvability of Q_{app} .

Lemma 3.2 (Solvability of Q_{app}). A feasibility program of the form Q_{app} is solvable as long as its feasibility graph $G_{Q_{app}}$ is acyclic.

Proof. Assume that $G_{Q_{app}}$ is acyclic, and let d_k be the length of the *longest* path from node k to a sink node in Q_{app} . Let

$$w_k = \frac{\rho^{d_k}}{\sum_{j=1}^m \rho^{d_j}}$$
, for all $k \in [m]$.

We will argue that these values of w_k , for $k \in [m]$, satisfy the constraints of the feasibility program \mathcal{Q}_{app} . Obviously, $\sum_{k=1}^m w_k = 1$ by definition. Since the graph has m nodes, it holds that $d_k \leq m$. Since $\rho \leq 1$, this implies that $\rho^{d_k} \geq \rho^m$ and that $\sum_{j=1}^n \rho^{d_j} \leq m$. Therefore, we obtain that $w_k \geq \frac{\rho^m}{m}$. It remains to show these values of w_k satisfy the conditional constraints. Indeed, consider an edge (k,k') in the feasibility graph \mathcal{Q}_{app} , which, recall, corresponds to a constraint where $h_k(y) - h_{k'}(y) > 0$. Since (k,k') is an edge in \mathcal{Q}_{app} ,

we have that $d_k \ge d_{k'} + 1$, as there is a path from k to a sink of Q_{app} that starts with the edge (k, k'). This implies that

$$w_k = \frac{\rho^{d_k}}{\sum_{j=1}^m \rho^{d_j}} \le \frac{\rho^{d'_k+1}}{\sum_{j=1}^m \rho^{d_j}} = \frac{\rho \cdot \rho^{d'_k}}{\sum_{j=1}^m \rho^{d_j}} = \rho \cdot w_{k'},$$

and hence the corresponding conditional constraint is satisfied.

Thus, in our applications in which the feasibility programs that we construct are of the form Q_{app} above, it suffices to show that their corresponding feasibility graph Q_{app} is acyclic, in order to establish their solvability by Lemma 3.2.

3.3 Construction and proof for the linear-OPT-gate

In this section, we prove our main result stated earlier, Theorem 3.1. The theorem follows from the following proposition, which proves that a single linear-OPT-gate can be simulated by a standard PL arithmetic circuit.

Proposition 3.1. Given $n, m, k \in \mathbb{N}$, a rational matrix $A \in \mathbb{R}^{m \times n}$, rational bounds R > 0 and C > 0, as well as a PL arithmetic circuit $G_{\partial f} : \mathbb{R}^n \times \mathbb{R}^k \times [0,1]^\ell \to \mathbb{R}^n \times [0,1]^\ell$, we can construct a PL arithmetic circuit $F : \mathbb{R}^m \times \mathbb{R}^k \times [0,1]^t \to [-R,R]^n \times [0,1]^t$ in time

$$poly(n, m, k, size(A), size(R), size(C), size(G_{\partial f}))$$

which satisfies

$$F(b,c,\alpha)=(x,\alpha) \implies x$$
 is an optimal solution to optimization problem C in (1) at (b,c)

whenever b and c satisfy the following three conditions:

- 1. The feasible domain $\{x \in [-R, R]^n : Ax \leq b\}$ is not empty.
- 2. The map $x \mapsto f(x;c)$ is a convex function on the feasible domain and its subgradient is given by the PL pseudo-circuit $G_{\partial f}$.
- 3. $||c||_{\infty} \leq C$.

Theorem 3.1 follows from Proposition 3.1 by simply repeatedly replacing every linear-OPT-gate by its corresponding standard PL arithmetic circuit with auxiliary inputs and outputs, until no linear-OPT-gates are left in the circuit.

The attentive reader might have noticed that the statement of Proposition 3.1 does not completely correspond to what was claimed in the definition of the linear-OPT-gate (Definition 3.1). Namely, the proposition assumes that we are given R as a parameter when we construct the gadget, whereas the original definition allowed R to be an input to the linear-OPT-gate. Furthermore, the proposition also asks for an upper bound C on the length of C to be given as a parameter when constructing the linear-OPT-gate, whereas no such C was mentioned earlier. This is without loss of generality, as we argue next.

Instead of assuming that R is a fixed parameter of the linear-OPT-gate, we can assume that R is an input, but that we are also given an upper bound R' on R. This can easily be achieved by applying Proposition 3.1 using R' as the value of the parameter R, and explicitly adding constraints $x_i \leq R$, $-x_i \leq R$ (where R can now indeed be given as an input, since it only appears on the right hand side of constraints). Note that as long as we indeed have $R \leq R'$, the new optimization problem is equivalent to the previous one, and the linear-OPT-gate will correctly solve it.

We still have to provide upper bounds R' and C when constructing the linear-OPT-gate. The crucial observation here is that the statement of Theorem 3.1 explicitly mentions that linear-OPT-gates can only be used *for the purpose of proving membership in PPAD*. More formally, this should be interpreted as saying:

as long as we are ultimately only using the linear-OPT-gates inside a PL arithmetic circuit with bounded domain (which is the case in Theorem 3.1), we can assume that we do not need to explicitly provide upper bounds R' and C. The reason for this stems from Lemma 2.1, which states that we can bound the magnitude of any value inside a PL arithmetic circuit with bounded domain, in terms of the size of the description of the circuit, and the bound on the domain. Thus, whenever we use a linear-OPT-gate in a PL arithmetic circuit with bounded domain, since the inputs R and C are computed by a PL arithmetic circuit with bounded domain, we can compute corresponding upper bounds R' and C for the actual construction of the linear-OPT-gate gadget according to Proposition 3.1. An important but subtle point is that the linear-OPT-gate is always guaranteed to output an element in $[-R', R']^n$. As a result, even in a PL arithmetic circuit that uses multiple linear-OPT-gates, we can efficiently compute upper bounds on the magnitudes of numbers given as input to the linear-OPT-gate, even before replacing the linear-OPT-gates by the actual gadgets implementing them (namely the construction of Proposition 3.1).

3.3.1 Construction of the PL arithmetic circuit *F*

We begin with the description of how the circuit F is constructed. For this we use a sufficiently small value $\varepsilon > 0$, which we construct in the next section. We let $t := n + \ell + m$ and write $[0,1]^t = [0,1]^n \times [0,1]^{\ell+m}$.

On input $(b, c, \alpha, \beta) \in \mathbb{R}^m \times \mathbb{R}^k \times [0, 1]^n \times [0, 1]^{\ell + m}$, the circuit *F* proceeds as follows:

- 1. Compute $x := 2R\alpha R$. In other words, we scale $\alpha \in [0,1]^n$ into a point $x \in [-R,R]^n$.
- 2. Compute $(v, \overline{\beta}_1, \dots, \overline{\beta}_{\ell}) := G_{\partial f}(x, c, \beta_1, \dots, \beta_{\ell}).$
- 3. Compute $\mu_i \in H(a_i^\mathsf{T} x b_i)$ for $i = 1, \ldots, m$, using auxiliary variables $\beta_{\ell+1}, \ldots, \beta_{\ell+m}$ and corresponding outputs $\overline{\beta}_{\ell+1}, \ldots, \overline{\beta}_{\ell+m}$. More formally, compute $\overline{\beta}_{\ell+i} := \min\{1, \max\{0, \beta_{\ell+i} + a_i^\mathsf{T} x b_i\}\}$ and $\mu_i := \beta_{\ell+i}$ for $i = 1, \ldots, m$.
- 4. Compute

$$\overline{x} := \Pi_R \left(x - \varepsilon \cdot v - A^\mathsf{T} \mu \right)$$

where Π_R denotes projection to $[-R, R]^n$.

- 5. Compute $\overline{\alpha} := (\overline{x} + R)/2R$ (i.e., scale $\overline{x} \in [-R, R]^n$ into a point $\overline{\alpha} \in [0, 1]^n$).
- 6. Output $(\overline{x}, \overline{\alpha}, \overline{\beta})$.

Here $a_i \in \mathbb{R}^n$ denotes the *i*th row of matrix A.

Note that we can construct a PL arithmetic circuit computing *F* in time

$$poly(n, m, k, size(A), size(R), size(C), size(G_{\partial f})),$$

assuming that ε can computed in polynomial time in these quantities (which we argue in the section). In particular, the projection $\Pi_R(y)$ of some vector y can be obtained by computing $\min\{R, \max\{-R, y_i\}\}$ for each coordinate of y. Furthermore, note that the first output \overline{x} of F always satisfies $\overline{x} \in [-R, R]^n$, even when the fixed point constraints $\alpha = \overline{\alpha}$ and $\beta = \overline{\beta}$ are not satisfied.

3.3.2 Construction of ε

We now describe how ε is constructed. Since $G_{\partial f}$ is a PL arithmetic circuit, by Lemma 2.1 we can compute in time poly(size(R), size($G_{\partial f}$)) a rational K > 0 such that

$$\max_{(x,c,\beta)\in[-R,R]^n\times[-C,C]^n\times[0,1]^{\ell}}\|G_{\partial f}(x,c,\beta)\|_2 \le K.$$
 (5)

For the construction of ε we will also require the following notation. We define $\widetilde{A} \in \mathbb{R}^{(m+2n)\times n}$ and $\widetilde{b} \in \mathbb{R}^{m+2n}$ such that the system " $\widetilde{A}x \leq \widetilde{b}$ " corresponds to the system " $Ax \leq b$ " with the additional constraints " $x_i \leq R$ " and " $-x_i \leq R$ " for $i=1,\ldots,n$. In particular, the first m rows of \widetilde{A} correspond to A and the first m entries in \widetilde{b} correspond to B. We use \widetilde{a}_i to denote the Bth row of B.

We set $\varepsilon := \gamma^*/K$, where $\gamma^* > 0$ is computed in time poly(size(A)) using the following lemma.

Lemma 3.3. Given $A \in \mathbb{R}^{m \times n}$, we can construct in polynomial time a sufficiently small number $\gamma^* > 0$ such that for any nonempty $I \subseteq [m+2n]$ and for every partition of I into I_0 and I_1 , such that $I_1 \neq \emptyset$, if the following optimization problem (in variables $u \in \mathbb{R}^n$ and $\lambda \in \mathbb{R}^{|I|}$)

min
$$\sum_{i \in I_{1}} \widetilde{a}_{i}^{\mathsf{T}} u$$
s.t.
$$\widetilde{a}_{i}^{\mathsf{T}} u = 0 \quad \forall i \in I_{0}$$

$$\widetilde{a}_{i}^{\mathsf{T}} u \geq 0 \quad \forall i \in I_{1}$$

$$\|u\|_{2} = 1$$

$$u = \sum_{i \in I} \lambda_{i} \widetilde{a}_{i}$$

$$\lambda_{i} \geq 0 \quad \forall i \in I$$
(6)

is feasible, then its optimal value γ satisfies $\gamma > \gamma^*$.

Proof. Let I, I_0 , and I_1 be as specified in the statement of the lemma, and such that optimization problem (6) is feasible. Letting γ denote the optimal value of (6), note that $\gamma \geq \gamma'/\sqrt{n}$, where γ' is the optimal value of the same optimization problem, except that we replace the constraint " $\|u\|_2 = 1$ " by " $\|u\|_{\infty} = 1$ ", namely:

min
$$\sum_{i \in I_1} \widetilde{a}_i^{\mathsf{T}} u$$
s.t.
$$\widetilde{a}_i^{\mathsf{T}} u = 0 \quad \forall i \in I_0$$

$$\widetilde{a}_i^{\mathsf{T}} u \ge 0 \quad \forall i \in I_1$$

$$\|u\|_{\infty} = 1$$

$$u = \sum_{i \in I} \lambda_i \widetilde{a}_i$$

$$\lambda_i \ge 0 \quad \forall i \in I$$
(7)

This is due to the fact that all other constraints are invariant to scaling of (u, λ) . In particular, note that (7) is also feasible.

The optimal value γ' of (7) satisfies $\gamma' = \min_{k \in I, s \in \{+1, -1\}} \gamma'_{k,s}$, where $\gamma'_{k,s}$ is the optimal value of the following LP

$$\min \quad \sum_{i \in I_{1}} \widetilde{a}_{i}^{\mathsf{T}} u$$

$$\text{s.t.} \quad \widetilde{a}_{i}^{\mathsf{T}} u = 0 \qquad \forall i \in I_{0}$$

$$\widetilde{a}_{i}^{\mathsf{T}} u \geq 0 \qquad \forall i \in I_{1}$$

$$-1 \leq u_{j} \leq 1 \quad \forall j \in [n]$$

$$u_{k} = s$$

$$u = \sum_{i \in I} \lambda_{i} \widetilde{a}_{i}$$

$$\lambda_{i} \geq 0 \qquad \forall i \in I$$

if it is feasible, and $\gamma'_{k,s} := +\infty$ otherwise.

Next, we show that $\gamma'_{k,s} > 0$. This clearly holds if the LP is infeasible. Assume towards a contradiction that the LP is feasible, but $\gamma'_{k,s} \leq 0$. Then, $\gamma'_{k,s} = 0$ and it is achieved by a feasible u with $\widetilde{a}_i^\mathsf{T} u = 0$ for all $i \in I$. Since u is feasible, there exists $\lambda \geq 0$ with $u = \sum_{i \in I} \lambda_i \widetilde{a}_i$. But then $\|u\|_2^2 = u^\mathsf{T} u = \sum_{i \in I} \lambda_i \widetilde{a}_i^\mathsf{T} u = 0$, which contradicts $|u_k| = 1$.

Finally, note that the bit-complexity of the optimal value of any LP of this form is polynomially bounded by the bit-representation of matrix A. Indeed, the number of bits needed to write down such an LP for any choice of $k \in I$, $s \in \{+1, -1\}$, and for any I, I_0 , and I_1 is bounded by some polynomial quantity in size(A). In particular, given A, we can compute in polynomial time a rational value $\gamma^* > 0$ such that $\gamma^* < \gamma'_{k,s} / \sqrt{n}$ for all $k \in I$, $s \in \{+1, -1\}$, and for all possible I, I_0 , and I_1 . By the arguments above, it then follows that $\gamma^* < \gamma$.

3.3.3 Analysis: Fixed point constraints

In this section, we state and prove some simple properties that follow from the fixed point constraints $(\alpha, \beta) = (\overline{\alpha}, \overline{\beta})$. Recall that the system " $\widetilde{A}x \leq \widetilde{b}$ " corresponds to the system " $Ax \leq b$ " with the additional constraints " $x_i \leq R$ " and " $-x_i \leq R$ " for i = 1, ..., n. In particular, the first m rows of \widetilde{A} correspond to A and the first m entries in \widetilde{b} correspond to b.

The circuit *F* has been constructed in order to ensure that the following properties hold.

Claim 3.1. *If* $(\alpha, \beta) = (\overline{\alpha}, \overline{\beta})$ *, then:*

- 1. $\mu_i \in H(a_i^T x b_i)$ for i = 1, ..., m.
- 2. $||v||_2 \le K$ and, if $Ax \le b$, then $v \in \partial f(x; c)$.
- 3. $x = \overline{x}$ and

$$\varepsilon \cdot v + \widetilde{A}^{\mathsf{T}} \widetilde{\mu} = 0 \tag{8}$$

where $\widetilde{\mu} \in \mathbb{R}^{m+2n}$ satisfies $\widetilde{\mu} \geq 0$, as well as

$$\widetilde{\mu}_i > 0 \implies \widetilde{a}_i^{\mathsf{T}} x \geq \widetilde{b}_i$$

for all i = 1, ..., m + 2n. Furthermore, for $i \in [m]$ we also have $\widetilde{\mu}_i = \mu_i$.

Proof. The first statement follows from the construction of the Heaviside PL pseudo-circuit. The fact that $||v||_2 \le K$ follows from the definition of K (see (5)) and the assumption that $||c||_{\infty} \le C$. The fact that $v \in \partial f(x;c)$ when $Ax \le b$ follows from the assumption that $G_{\partial f}$ is a PL pseudo-circuit computing $\partial f(x;c)$, whenever x lies in feasible domain.

Since $\alpha = \overline{\alpha}$, it follows that $x = \overline{x}$, i.e., $x = \Pi_R(x - \varepsilon \cdot v - A^T \mu)$. This implies that

$$x - \varepsilon \cdot v - A^{\mathsf{T}} \mu - x = I_n \lambda^+ - I_n \lambda^-$$

where $\lambda^+, \lambda^- \in \mathbb{R}^n$ are nonnegative, and additionally satisfy $\lambda_j^+ > 0 \implies x_j = R$, and $\lambda_j^- > 0 \implies x_j = -R$. Here $I_n \in \mathbb{R}^{n \times n}$ denotes the identity matrix. Finally, noting that

$$\widetilde{A} = \begin{bmatrix} A \\ I_n \\ -I_n \end{bmatrix}$$

we can rewrite the equation as

$$\varepsilon \cdot v + \widetilde{A}^{\mathsf{T}} \widetilde{\mu} = 0$$

where we let

$$\widetilde{\mu} = \begin{bmatrix} \mu \\ \lambda^+ \\ \lambda^- \end{bmatrix}$$

Note that we indeed have $\widetilde{\mu} \ge 0$ and $\widetilde{\mu}_i > 0 \implies \widetilde{a}_i^\mathsf{T} x \ge \widetilde{b}_i$. In particular, for $i \in [m]$ this follows from $\widetilde{\mu}_i = \mu_i \in \mathsf{H}(a_i^\mathsf{T} x - b_i)$.

In the remainder of the proof, we show that x must necessarily be an optimal solution to the optimization problem. We first show that if x is feasible, it is necessarily optimal. We complete the proof by proving that x must necessarily be feasible.

3.3.4 Analysis: Feasibility implies optimality

Consider the case where x is feasible, i.e., $Ax \le b$. We will show that x must be an optimal solution. Since $x \in [-R, R]^n$ and $Ax \le b$, it follows that $\widetilde{A}x \le \widetilde{b}$. Thus, the third statement in Claim 3.1 yields that

$$\varepsilon \cdot v + \widetilde{A}^{\mathsf{T}} \widetilde{\mu} = 0 \tag{9}$$

where $\widetilde{\mu} \geq 0$ and where

$$\widetilde{\mu}_i > 0 \implies \widetilde{a}_i^\mathsf{T} x \ge \widetilde{b}_i \implies \widetilde{a}_i^\mathsf{T} x = \widetilde{b}_i.$$

In other words, $\widetilde{\mu}_i$ can only be strictly positive if the ith constraint is tight. Furthermore, since x is feasible, the second statement in Claim 3.1 yields that $v \in \partial f(x;c)$. As a result, given that $\varepsilon > 0$, the equality (9) can be interpreted as saying that the Karush-Kuhn-Tucker (KKT) conditions hold at point x for the following constrained minimization problem

min
$$f(x;c)$$

s.t. $\widetilde{A}x < \widetilde{b}$

which is the same as our optimization problem C in (1). Since, by assumption, f is convex on the feasible domain, the KKT conditions are also sufficient for optimality, and thus x is an optimal solution.

Formally, consider any feasible point z. We will show that $f(z) \ge f(x)$. Taking the inner product of (9) with z - x yields

$$\varepsilon \cdot v^{\mathsf{T}}(z-x) = (\widetilde{A}^{\mathsf{T}}\widetilde{\mu})^{\mathsf{T}}(x-z) = \widetilde{\mu}^{\mathsf{T}}\widetilde{A}(x-z) \ge 0$$

since $\widetilde{\mu} \ge 0$ and $\widetilde{\mu}_i > 0 \implies \widetilde{a}_i^\mathsf{T} x = \widetilde{b}_i \ge \widetilde{a}_i^\mathsf{T} z$. Given that $\varepsilon > 0$, it follows that $v^\mathsf{T}(z - x) \ge 0$. Finally, using the fact that $v \in \partial f(x;c)$, i.e., v is a subgradient of $f(\cdot;c)$ at x, together with the definition of subgradients, we obtain

$$f(z) \ge f(x) + v^{\mathsf{T}}(z - x) \ge f(x)$$

i.e., *x* is a global minimum of *f* on the feasible domain.

3.3.5 Analysis: Feasibility

We show that x must necessarily be feasible, i.e., $Ax \le b$. Assume towards a contradiction that x is not feasible.

Let $J := \{i \in [m+2n] : \widetilde{a}_i^\mathsf{T} x \ge \widetilde{b}_i\}$ and note that J is nonempty, since x is infeasible. By Claim 3.1 we have $\widetilde{\mu}_i > 0 \implies \widetilde{a}_i^\mathsf{T} x \ge \widetilde{b}_i \implies i \in J$, and thus $\widetilde{\mu}_i = 0$ for all $i \in [m+2n] \setminus J$. In particular, $\widetilde{A}^\mathsf{T} \widetilde{\mu} = \widetilde{A}_J^\mathsf{T} \widetilde{\mu}_J$, where \widetilde{A}_J denotes the restriction of \widetilde{A} to the subset of rows J, and similarly for $\widetilde{\mu}_J$. As a result, we can rewrite (8) from Claim 3.1 as

$$\varepsilon \cdot v + \widetilde{A}_J^{\mathsf{T}} \widetilde{\mu}_J = 0. \tag{10}$$

Let z^* be the projection of x onto the convex set $D_J := \{y \in \mathbb{R}^n : \widetilde{A}_J y \leq \widetilde{b}_J\}$, which is nonempty, since the feasible region $D := \{y \in \mathbb{R}^n : \widetilde{A}y \leq \widetilde{b}\}$ is nonempty by assumption. Note that $z^* \neq x$, because if $x \in D_J$, then we would also have $x \in D$ by the definition of J.

Let $I := \{i \in J : \widetilde{a}_i^\mathsf{T} z^* = \widetilde{b}_i\}$, i.e., the set of all constraints in J that are tight at z^* . Note that z^* , which is the projection of x onto D_J , is also the projection of x onto $D_I := \{y \in \mathbb{R}^n : \widetilde{A}_I y \leq \widetilde{b}_I\}$. Indeed, assume that z^* is not the projection of x onto D_I . Then there exists $y \in D_I$ with $(x - z^*)^\mathsf{T}(y - z^*) > 0$. But since all constraints in $J \setminus I$ are strictly satisfied at z^* , there exists a point y^* on the segment $[z^*, y]$ that satisfies $y^* \in D_J$ and $(x - z^*)^\mathsf{T}(y^* - z^*) > 0$. This is a contradiction to the fact that z^* is the projection of x onto D_J . Thus, z^* is indeed the projection of x onto D_I .

Next, we partition I into two sets: $I_0 := \{i \in I : \widetilde{a}_i^\mathsf{T} x = \widetilde{b}_i\}$ and $I_1 := \{i \in I : \widetilde{a}_i^\mathsf{T} x > \widetilde{b}_i\}$. Note that by construction $I_0 \cap I_1 = \emptyset$ and $I = I_0 \cup I_1$, since $I \subseteq J$. Furthermore, $I_1 \subseteq [m]$, because we have $x \in [-R, R]^n$, which means that $\widetilde{a}_i^\mathsf{T} x \leq \widetilde{b}_i$ for all $i \in [m+2n] \setminus [m]$. Finally, $I_1 \neq \emptyset$. Indeed, if $I_1 = \emptyset$, then $x \in D_I$, which contradicts $z^* \neq x$.

Taking the inner product of (10) with $z^* - x$ yields

$$\varepsilon \cdot v^{\mathsf{T}}(z^* - x) = (\widetilde{A}_{I}^{\mathsf{T}} \widetilde{\mu}_{I})^{\mathsf{T}}(x - z^*) = \widetilde{\mu}_{I}^{\mathsf{T}} \widetilde{A}_{I}(x - z^*) \ge \widetilde{\mu}_{I_{1}}^{\mathsf{T}} \widetilde{A}_{I_{1}}(x - z^*) = \mu_{I_{1}}^{\mathsf{T}} A_{I_{1}}(x - z^*) = \sum_{i \in I_{1}} a_{i}^{\mathsf{T}}(x - z^*)$$

where we used $\mu_i \geq 0$ and $\widetilde{a}_i^\mathsf{T} x \geq \widetilde{b}_i \geq \widetilde{a}_i^\mathsf{T} z^*$ for all $i \in J$, $\widetilde{\mu}_{I_1} = \mu_{I_1}$, $\widetilde{A}_{I_1} = A_{I_1}$, and $\mu_i = 1$ for all $i \in I_1 \subseteq [m]$, since $\mu_i \in H(a_i^\mathsf{T} x - b_i)$. As a result

$$\sum_{i \in I} a_i^{\mathsf{T}}(x - z^*) \le \varepsilon \cdot v^{\mathsf{T}}(z^* - x) \le \varepsilon \|v\|_2 \|x - z^*\|_2 \le \varepsilon K \|x - z^*\|_2 \le \gamma^* \|x - z^*\|_2$$
 (11)

where we used $||v||_2 \le K$ (Claim 3.1) and $\varepsilon = \gamma^*/K$.

We now show that this is a contradiction to our choice of γ^* . Let $u := (x - z^*)/\|x - z^*\|_2$ and note that u is well defined, since $z^* \neq x$. Assume, for now, that u is feasible for optimization problem (6) from Lemma 3.3, which we repeat here for convenience:

$$\begin{aligned} & \min \quad \sum_{i \in I_1} \widetilde{a}_i^\mathsf{T} u \\ & \text{s.t.} \quad \widetilde{a}_i^\mathsf{T} u = 0 \quad \forall i \in I_0 \\ & \quad \widetilde{a}_i^\mathsf{T} u \geq 0 \quad \forall i \in I_1 \\ & \quad \|u\|_2 = 1 \\ & \quad u = \sum_{i \in I} \lambda_i \widetilde{a}_i \\ & \quad \lambda_i > 0 \quad \forall i \in I \end{aligned}$$

In particular, the optimization problem is feasible, and thus by Lemma 3.3 its optimal value is (strictly) lower bounded by γ^* . As a result, $\sum_{i \in I_1} \widetilde{a}_i^\mathsf{T}(x-z^*) / \|x-z^*\|_2 > \gamma^*$, which yields $\sum_{i \in I_1} \widetilde{a}_i^\mathsf{T}(x-z^*) > \gamma^* \|x-z^*\|_2$. But this is a contradiction to (11).

It remains to show that $u=(x-z^*)/\|x-z^*\|_2$ is indeed feasible for optimization problem (6). The first two constraints are satisfied, because $\widetilde{a}_i^\mathsf{T} x = \widetilde{b}_i = \widetilde{a}_i^\mathsf{T} z^*$ for all $i \in I_0$, and $\widetilde{a}_i^\mathsf{T} x > \widetilde{b}_i = \widetilde{a}_i^\mathsf{T} z^*$ for all $i \in I_1$. Clearly, $\|u\|_2 = 1$. Thus, it remains to prove that there exists $\lambda \geq 0$ such that $x - z^* = \widetilde{A}_I^\mathsf{T} \lambda$. Since z^* is the projection of x onto $D_I = \{y \in \mathbb{R}^n : \widetilde{A}_I y \leq \widetilde{b}_I\}$, it follows that $(x - z^*)^\mathsf{T} (y - z^*) \leq 0$ for all $y \in D_I$. Given that $\widetilde{A}_I z^* = \widetilde{b}_I$, we thus obtain that $(x - z^*)^\mathsf{T} y \leq 0$ for all y with $\widetilde{A}_I y \leq 0$. However, by Farkas' lemma [Boyd and Vandenberghe, 2004], we know that exactly one of the following two statements holds:

1.
$$\exists y \in \mathbb{R}^n$$
: $\widetilde{A}_I y \leq 0$ and $(x - z^*)^\mathsf{T} y > 0$

2.
$$\exists \lambda \in \mathbb{R}^{|I|}: \quad \widetilde{A}_I^\mathsf{T} \lambda = x - z^* \text{ and } \lambda \geq 0$$

Since we have shown that the first statement does not hold, we deduce that there exists $\lambda \ge 0$ such that $\widetilde{A}_I^T \lambda = x - z^*$, as desired. As a result, $u = (x - z^*)/\|x - z^*\|_2$ is indeed feasible for optimization problem (6), and the proof is complete.

3.4 Implicit Functions and Correspondences

In this section we will construct PL arithmetic circuits computing univariate piecewise-linear functions and piecewise-constant correspondences, based on a succinct representation given by a Boolean circuit, built from Boolean AND, OR and NOT gates. We remark that conceptually, the ideas around the use of the triangle-wave function (see Definition 3.7) and the bit multiplication (see Definition 3.8 and the subsequent discussion) are attributed to Fearnley et al. [2023], who developed them, albeit for a different setting.

By interpreting non-negative integers by their binary representation, a Boolean function may be viewed as computing an integer-valued function. For $b=(b_{n-1},\ldots,b_0)\in\{0,1\}^n$, we denote by $\mathrm{bitVal}_n(b)$ the number $\mathrm{bitVal}_n(b)=(b_{n-1},\ldots,b_0)_2=\sum_{i=0}^{n-1}b_i2^i$ encoded by b in binary.

Definition 3.4. Let C be a Boolean circuit with n inputs and n outputs, thereby computing a function $C: \{0,1\}^n \to \{0,1\}^n$, and let $1 \le N < 2^n$ be an integer. We index the inputs and outputs of C with $\{0,\ldots,n-1\}$ and we consider $b \in \{0,1\}^n$ to represent the integer bitVal $_n(b)$. The integer-valued function represented by (C,N) is the function $f: \{0,1,\ldots,N\} \to \{0,1,\ldots,N\}$ given by

$$f(\mathsf{bitVal}_n(b)) = \min\{N, \mathsf{bitVal}_n(C(b))\} = \min\{N, (C(b)_{n-1}, \dots, C(x)_0)_2\} ,$$

for $b \in \{0,1\}^n$ such that bitVal_n $(b) \le N$.

A natural piecewise-linear function is then given by linear interpolation between function values.

Definition 3.5. Let *C* be a Boolean circuit with *n* inputs and *n* outputs, and let $1 \le N < 2^n$ be an integer. The piecewise-linear function represented by (C, N) is the function $g: \mathbb{R} \to \mathbb{R}$ given by

$$g(x) = \begin{cases} f(0) & \text{for } x < 0\\ (1 - (x - \lfloor x \rfloor))f(\lfloor x \rfloor) + (x - \lfloor x \rfloor)f(\lfloor x \rfloor + 1) & \text{for } 0 \le x < N\\ f(N) & \text{for } N \le x \end{cases},$$

where f is the integer-valued function represented by (C, N).

In a similar way, a natural piecewise-constant correspondence is given by extending function values to the right, until the next function value is defined.

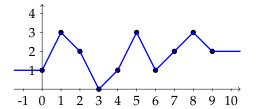
Definition 3.6. Let *C* be a Boolean circuit with *n* inputs and *n* outputs, and let $1 \le N < 2^n$ be an integer. The piecewise-constant correspondence represented by (C, N) is the correspondence $g: \mathbb{R} \rightrightarrows \mathbb{R}$ given by

$$g(x) = \begin{cases} f(0) & \text{for } x \le 0 \\ [\min(f(x-1), f(x)), \max(f(x-1), f(x))] & \text{for } x \in \{1, 2, \dots, N\} \\ f(\lfloor x \rfloor) & \text{for } x \in [0, N] \setminus \{0, 1, \dots, N\} \end{cases},$$

where f is the integer-valued function represented by (C, N).

An example illustrating Definition 3.5 and Definition 3.6 is given in Figure 2.

A function that will be used in both constructions is the familiar square-wave function, restricted to a bounded number of periods, as illustrated in Figure 3.



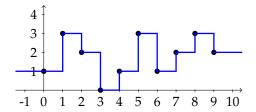
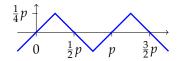


Figure 2: The piecewise-linear function (left) and piecewise-constant correspondence (right) defined implicitly by a Boolean circuit C and N=9 computing an integer valued function f as shown with dots.



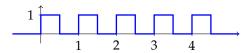


Figure 3: The function $T_0(x, p)$ used to define $S_0(x, p)$ (left) and the function $S_2(x, 1)$ (right) giving a square wave with 4.5 periods of length 1 in the interval [0, 4.5].

Definition 3.7. For a non-negative integer n, the correspondence $S_n : \mathbb{R}^2 \rightrightarrows [0,1]$, consisting of $2^n + 1/2$ periods of length p, in the interval $[0, (2^n + 1/2)p]$, is defined as $H(T_n(x, p))$, where T_n is the triangle-wave function defined inductively as follows. For n = 0, we define

$$T_0(x, p) = \max(\min(x, p/2 - x), \min(x - p, 3p/2 - x))$$
,

and for n > 0 we define $T_n(x, p) = T_{n-1}(\min(x, (2^n + 1/2)p - x), p)$.

Boolean operations may be viewed as special cases of PL functions, namely the Boolean functions AND(a,b), OR(a,b), and NOT(a) correspond to the PL functions min(a,b), max(a,b), and 1-a, for $a,b \in \{0,1\}$. The following is thus immediate.

Lemma 3.4. Let C be a Boolean circuit with n inputs and m outputs. Then there is a PL arithmetic circuit with n inputs and m outputs C' with the same number of gates as C such that C(b) = C'(b) for all $b \in \{0,1\}^n$.

In fact, AND(a, b) and thus min(a, b) may also be viewed as the multiplication operation for a, $b \in \{0,1\}$. Even more generally, min(a, b) may be viewed as a multiplication when just $b \in \{0,1\}$ and $a \in [0,1]$. As long as we ultimately only use this tool to prove PPAD-membership, we can make use of the linear-OPT-gate to construct a stronger multiplication gadget that does not require an a priori bound on the operand a.

Definition 3.8. Define the PL pseudo-circuit bitMult with two inputs and one output by bitMult(a, b) = $H(b-\frac{1}{2})a$.

Note that bitMult can indeed be computed by a PL pseudo-circuit by Lemma 3.1 (which uses the linear-OPT-gate internally). From the definition of the Heaviside function H the following is then immediate.

Lemma 3.5. We have bitMult(a, b) = ab, for any $a \in \mathbb{R}$ and any $b \in \{0, 1\}$.

We will show below that also the piecewise-linear functions and piecewise-constant correspondences represented by Boolean circuits may be computed by PL pseudo-circuits. In fact, for the case of piecewise-constant correspondences we will show a considerably stronger statement allowing the scaling of the input and output by a variable.

An important component of both constructions is *bit extraction*. Informally, bit extraction of a non-negative $x \ge 0$ refers to computing the binary expansion of |x|. This is clearly an inherently discontinuous

operation, but can nevertheless be circumvented in our applications. The bit extraction procedure we define below will actually take 2 inputs x and y, and the intention is to perform bit extraction on the number x/y, when $0 < y \le 1$.

Definition 3.9. For a positive integer n, we inductively define a PL pseudo-circuit bitExt $_n$ with two inputs and n outputs as follows. For n=1, we define bitExt $_1(x,y)=H(x-y)$, and for n>1 we define bitExt $_n(x,y)=(b_{n-1},\ldots,b_0)$ where $b_{n-1}=H(x-2^{n-1}y)$ and $(b_{n-2},\ldots,b_0)=\text{bitExt}_{n-1}(x-\text{bitMult}(2^{n-1}y,b_{n-1}),y)$.

Lemma 3.6. Let $n \ge 1$, assume $0 < y \le 1$, and let $\text{bitExt}_n(x,y) = b = (b_{n-1}, \ldots, b_0)$. Then, whenever $x/y \in [0,2^n] \setminus \{1,2,3,\ldots,2^n\}$ we have $b_i \in \{0,1\}$, for all i and $\text{bitVal}_n(b) = \sum_{i=0}^{n-1} b_i 2^i = \lfloor x/y \rfloor$.

Proof. Assume $x/y \in [0,2^n] \setminus \{1,2,3,\ldots,2^n\}$. Since in particular $x/y \neq 2^{n-1}$, by definition of the Heaviside function H we have that $b_{n-1} = 0$ when $x/y < 2^{n-1}$ and $b_{n-1} = 1$ when $x/y > 2^{n-1}$. This shows the case of n = 1. For n > 1 we have $x/y - b_{n-1}2^{n-1} \in [0,2^{n-1}] \setminus \{1,2,\ldots,2^{n-1}\}$, and by induction we have $\lfloor x/y - b_{n-1}2^{n-1} \rfloor = \sum_{i=0}^{n-2} b_i 2^i$. The result now follows since $\lfloor x/y - b_{n-1}2^{n-1} \rfloor = \lfloor x/y \rfloor - b_{n-1}2^{n-1}$, for $b_{n-1} \in \{0,1\}$.

Remark 3.5. The output of bitExt_n is of course well-defined for any value of x and y. In particular we can observe that $(\text{bitExt}_n(x,0))_i = H(x)$, for all i.

As an extension of Definition 3.8 and Lemma 3.5 we may in a PL arithmetic circuit *multiply* a number given in binary encoding by variables with another variable. This means that we may consider the output of a PL circuit, obtained from a Boolean circuit by Lemma 3.4, as representing a number given in binary encoding and then multiply that by a variable.

Definition 3.10. For an integer n > 0, define the PL pseudo-circuit bitMult_n having n + 1 inputs and one output by bitMult_n $(a, b_{n-1}, \ldots, b_0) = \sum_{i=0}^{n-1} \text{bitMult}(a, b_i) 2^i$.

Using Lemma 3.5 we then have the following.

Lemma 3.7. For any positive integer n, $b \in \{0,1\}^n$, and any $a \in \mathbb{R}$ we have $\text{bitMult}_n(a,b_{n-1},\ldots,b_0) = a \cdot \text{bitVal}_n(b)$.

For $b \in \{0,1\}^n$ we shall for brevity also simply write bitMult $(a,b) = \text{bitMult}_n(a,b_{n-1},\ldots,b_0)$.

3.4.1 Piecewise-linear functions

We will now develop a construction of a PL pseudo-circuit computing the piecewise-linear function g represented by (C, N), as defined in Definition 3.5, for a circuit C with n inputs and n outputs and an integer N satisfying $1 \le N < 2^n$. The general idea is to perform bit extraction, evaluate the integer-valued function f represented by (C, N) on two inputs, and then perform linear interpolation. The immediate obstacle is that bit extraction applied to $x \in [0, 2^n]$ fails, i.e., does not guarantee a Boolean output, when $x \in \{1, \ldots, 2^n\}$. To overcome this obstacle we perform this procedure on two different translations of the input, and use the square wave function to switch between the two, thereby masking out erroneous values. Let in the following $G = [0, 2^n] \setminus \{1, \ldots, 2^n\}$ be the set of points where bit extraction succeeds.

For simplicity, we first give a description of the computation without referring to Boolean circuits. Suppose $x \in [0, N]$. We consider the following two cases, each with a special case in an endpoint of the interval [0, N].

1. Assume $x \in [\lfloor x \rfloor, \lfloor x \rfloor + \frac{1}{2}]$ and assume first also $x \leq N - \frac{1}{2}$. Let $w_1 = x + \frac{1}{4}$ and $w_2 = x + \frac{5}{4}$. Note that $w_1, w_2 \in G$ and that $\lfloor w_1 \rfloor = \lfloor x \rfloor$ and $\lfloor w_2 \rfloor = \lfloor x \rfloor + 1$. It thus follows that $g(x) = g_1(x)$, where

$$g_1(x) = (\lfloor w_2 \rfloor - x) f(\lfloor w_1 \rfloor) + (x - \lfloor w_1 \rfloor) f(\lfloor w_2 \rfloor)$$
.

To also handle the special case of x=N, we proceed as follows. First, let $\delta_1=\mathrm{H}(x-(N-\frac{1}{4}))$ and define $w_1'=x+\frac{1}{4}-\delta_1$ and $w_2'=x+\frac{5}{4}-\delta_1$. In case x=N we have $\delta_1=1$ and thus $w_1'=x-\frac{3}{4}$, $w_2'=w_1=x+\frac{1}{4}$, and $w_1',w_2'\in G$. We then have $g(x)=g_1'(x)$, where

$$g_1'(x) = (\lfloor w_2' \rfloor - x) f(\lfloor w_1 \rfloor) + (x - \lfloor w_1' \rfloor) f(\lfloor w_2' \rfloor) .$$

Note that f is evaluated on $\lfloor w_1 \rfloor$ and $\lfloor w_2' \rfloor$ (and not $\lfloor w_1' \rfloor$ and $\lfloor w_2' \rfloor$), which in the case of x = N both equal N, thereby satisfying Definition 3.5.

Finally, note that in the general case, when $x \leq N - \frac{1}{2}$, we have $\delta_1 = 0$, and thus $w_1' = w_1$ and $w_2' = w_2$ which means that $g_1'(x) = g_1(x)$.

2. Assume $x \in [\lceil x \rceil - \frac{1}{2}, \lceil x \rceil]$ and assume first also $x \ge \frac{1}{2}$. Let $w_3 = x - \frac{1}{4}$ and $w_4 = x + \frac{3}{4}$. Note that $w_3, w_4 \in G$. We now have two sub-cases. First, assume that $x < \lceil x \rceil$. Then $\lfloor w_3 \rfloor = \lfloor x \rfloor$ and $\lfloor w_4 \rfloor = \lfloor x \rfloor + 1$. Next assume that $x = \lceil x \rceil$. Then $\lfloor w_3 \rfloor = \lfloor x \rfloor - 1$ and $\lfloor w_4 \rfloor = \lfloor x \rfloor$. In both sub-cases it thus follows that $g(x) = g_2(x)$, where

$$g_2(x) = (|w_4| - x)f(|w_3|) + (x - |w_3|)f(|w_4|)$$
.

To also handle the special case of x=0, we proceed in a similar way as the case of x=N above. First, let $\delta_2=\mathrm{H}(\frac{1}{4}-x)$ and define $w_3'=x-\frac{1}{4}+\delta_2$ and $w_4'=x+\frac{3}{4}+\delta_2$. In case x=0 we have $\delta_2=1$ and thus $w_3'=w_4=x+\frac{3}{4}$, $w_4'=x+\frac{7}{4}$ and $w_3',w_4'\in G$. We then have $g(x)=g_2'(x)$, where

$$g_2'(x) = (\lfloor w_4' \rfloor - x) f(\lfloor w_3' \rfloor) + (x - \lfloor w_3' \rfloor) f(\lfloor w_4 \rfloor)$$
.

Note that f is evaluated on $\lfloor w_3' \rfloor$ and $\lfloor w_4 \rfloor$ (and not $\lfloor w_3' \rfloor$ and $\lfloor w_4' \rfloor$), which in the case of x = 0 both equals 0, thereby satisfying Definition 3.5.

Finally, note that in the general case, when $x \ge \frac{1}{2}$, we have $\delta_2 = 0$, and thus $w_3' = w_3$ and $w_4' = w_4$ which means that $g_2'(x) = g_2(x)$.

We will now simply use the square wave function to select between the two cases. Namely, from the definition of S_n we have that $g(x) = g_2'(x) + S_n(x,1)(g_1'(x) - g_2'(x))$. Since $S_n(x,1)$ is defined as the composition $H(T_n(x,1))$ of the Heaviside function with the PL function $T_n(x,1)$, we can compute the term $S_n(x,1)(g_1'(x)-g_2'(x))$ using the linear-OPT-gate by Lemma 3.1. We now present the general construction of the PL pseudo-circuit.

Proposition 3.2. Let C be a Boolean circuit with n inputs and n outputs, and let $N < 2^n$. Then, for the purpose of proving PPAD-membership, we can construct a PL pseudo-circuit computing the piecewise-linear function g represented by (C, N).

Proof. Let C' be the PL arithmetic circuit obtained from C by Lemma 3.4. We construct a PL pseudo-circuit computing g(x), operating as follows.

- 1. Compute $x' = \max(0, \min(N, x))$. Thus $x' \in [0, N]$ and g(x) = g(x').
- 2. Compute $\delta_1 = H(x' (N \frac{1}{4}))$ and $\delta_2 = H(\frac{1}{4} x')$ using Lemma 3.1.
- 3. Compute $w_1 = x' + \frac{1}{4}$, $w'_1 = x' + \frac{1}{4} \delta_1$, and $w'_2 = x' + \frac{5}{4} \delta_1$.
- 4. Compute $\mathbf{b}_1 = \mathrm{bitExt}_n(w_1, 1)$, $\mathbf{b}_1' = \mathrm{bitExt}_n(w_1', 1)$, and $\mathbf{b}_2' = \mathrm{bitExt}_n(w_2', 1)$.
- 5. Compute $g'_1 = \text{bitMult}(\text{bitVal}(\mathbf{b}'_2) x', C'(\mathbf{b}_1)) + \text{bitMult}(x' \text{bitVal}(\mathbf{b}'_1), C'(\mathbf{b}'_2)).$
- 6. Compute $w_4 = x' + \frac{3}{4}$, $w_3' = x' \frac{1}{4} + \delta_2$, and $w_4' = x' + \frac{3}{4} + \delta_2$.
- 7. Compute $\mathbf{b}_4 = \mathrm{bitExt}_n(w_4, 1)$, $\mathbf{b}_3' = \mathrm{bitExt}_n(w_3', 1)$, and $\mathbf{b}_4' = \mathrm{bitExt}_n(w_4', 1)$.
- 8. Compute $g_2' = \text{bitMult}(\text{bitVal}(\mathbf{b}_4') x', C'(\mathbf{b}_3')) + \text{bitMult}(x' \text{bitVal}(\mathbf{b}_3'), C'(\mathbf{b}_4)).$
- 9. Output $g'_2 + S_n(x', 1)(g'_1 g'_2)$, where the last term is computed using Lemma 3.1.

The correctness of the circuit follows from the preceding case analysis.

3.4.2 Piecewise-linear correspondences

We will now, in a similar way, develop a construction of a PL pseudocircuit computing the piecewise-constant correspondence g represented by (C, N), as defined in Definition 3.6, for a circuit C with n inputs and n outputs and an integer N satisfying $1 \le N < 2^n$. Again the general idea is to perform bit extraction and evaluate the integer-valued function f represented by (C, N) on two inputs. Since the function we compute is piecewise-constant, we are now able to also scale the output by a variable. Let again $G = [0, 2^n] \setminus \{1, \dots, 2^n\}$ be the set of points where bit extraction succeeds.

We will be able to compute the correspondence $z \cdot g(x/y)$, but for simplicity we first describe the computation of g(x), also without referring to Boolean circuits. Thus, suppose now $x \in [0, N]$. We consider the same cases as above.

- 1. Assume $x \in [\lfloor x \rfloor, \lfloor x \rfloor + \frac{1}{2}]$. Let $w_1 = x + \frac{1}{4}$, and note that $w_1 \in G$ and that $\lfloor w_1 \rfloor = \lfloor x \rfloor$. Let $g_1(x) = f(\lfloor w_1 \rfloor)$. It follows that $g_1(x) = f(\lfloor x \rfloor)$.
- 2. Assume $x \in [\lceil x \rceil \frac{1}{2}, \lceil x \rceil]$ and assume first also $x \ge \frac{1}{2}$. Let $w_2 = x \frac{1}{4}$, and note that $w_2 \in G$. Let $g_2(x) = f(\lfloor w_2 \rfloor)$. We now have two subcases. First, assume that $x < \lceil x \rceil$. Then $\lfloor w_2 \rfloor = \lfloor x \rfloor$ and thus $g_2(x) = f(\lfloor x \rfloor)$. Next assume that $x = \lceil x \rceil$. Then $\lfloor w_2 \rfloor = \lfloor x \rfloor 1$ and thus $g_2(x) = f(\lfloor x \rfloor 1)$. To also handle the special case of x = 0, we proceed as follows. First, let $\delta = H(\frac{1}{4} x)$ and define $w_2' = x \frac{1}{4} + \delta$. Let $g_2'(x) = f(\lfloor w_2' \rfloor)$. In case x = 0 we have $\delta = 1$ and thus $w_2' = x + \frac{3}{4}$ and $w_2' \in G$. We thus have $\lfloor w_2' \rfloor = \lfloor x \rfloor$ and thus $g_2'(x) = f(\lfloor x \rfloor)$.

Finally, note that in the general case, when $x \ge \frac{1}{2}$, we have $\delta = 0$, and thus $w_2' = w_2$ which means that $g_2'(x) = g_2(x)$.

We will now simply use the square wave function to select between the two cases, letting $g(x) = g'_2(x) + S_n(x, 1)(g_1(x) - g'_2(x))$.

This will also be what gives the correct result in case $x \in \{1, ..., N\}$ where we have $g_1(x) = f(x)$ and $g'_2(x) = f(x-1)$ and selecting using the square-wave function yields gives any convex combination of these values. For the special case of x = 0 we also obtain the correct result, since $g_1(0) = g'_2(0) = f(0)$.

Since $S_n(x,1)$ is defined as the composition $H(T_n(x,1))$ of the Heaviside function with the PL function $T_n(x,1)$, we can compute the term $S_n(x,1)(g_1(x)-g_2'(x))$ using Lemma 3.1. We now present the general construction of the PL pseudo-circuit.

Proposition 3.3. Let C be a Boolean circuit with n inputs and n outputs, and let $N < 2^n$. Then, for the purpose of proving PPAD-membership, we can construct a PL pseudo-circuit computing a function h(x,y,z) such that when $0 < y \le 1$ we have $h(x,y,z) = z \cdot g(x/y)$, where g is the piecewise-constant function represented by (C,N).

Proof. Let C' be the PL arithmetic circuit obtained from C by Lemma 3.4. We construct a PL pseudo-circuit computing h(x,y,z), operating as follows.

- 1. Compute $x' = \max(0, \min(yN, x))$. Thus when y > 0 we have $\frac{x'}{y} \in [0, N]$ and $g(\frac{x}{y}) = g(\frac{x'}{y})$.
- 2. Compute $\delta = H(\frac{1}{4}y x')y$ using Lemma 3.1.
- 3. Compute $w_1 = x' + \frac{1}{4}y$ and $\mathbf{b}_1 = \mathrm{bitExt}_n(w_1, y)$.
- 4. Compute $g_1 = \text{bitMult}(z, C'(\mathbf{b}_1))$.
- 5. Compute $w_2' = x' \frac{1}{4}y + \delta$ and $\mathbf{b}_2' = \mathrm{bitExt}_n(w_2', y)$.
- 6. Compute $g'_2 = \text{bitMult}(z, C'(\mathbf{b}'_2))$.
- 7. Output $g'_2 + S_n(x', y)(g_1 g'_2)$, where the last term is computed using Lemma 3.1.

The correctness of the circuit follows from the preceding case analysis, using $\frac{x}{y}$ in place of x in the analysis.

4 Concave Games, Nash Equilibria and Other Equilibrium Notions

The first application of our linear-OPT-gate will be to the domain of strategic games for computing Nash equilibria, as well as other related equilibrium notions. To demonstrate the use of our technique, we first "warm up" with two basic applications, namely (a) computing Nash equilibria in *bimatrix games* (Section 4.1) and (b) computing fixed point solutions of *digraph threshold games* (Section 4.2). These two applications will be used as examples to demonstrate the use of our linear-OPT-gate via linear programs, similar to linear program \mathcal{P} (see Program (2) in Section 3) or via feasibility programs, similar to feasibility program \mathcal{Q} (see Program (3) in Section 3) respectively.

Then in Section 4.3 we define a very large class of games, which we refer to as *games with PL best response oracles (PLBRO games)*. This class has a technical definition, and essentially captures every game in which the best response of the agent can be computed by a PL arithmetic circuit (or, to be more precise, a PL pseudo-circuit). The PPAD-membership of PLBRO games is then immediate from our results in Section 3 via the use of our linear-OPT-gate. This provides us with a very strong unified tool to show the PPAD-membership of a very large class of games, by merely showing that these games are PLBRO games.

More precisely, in Section 4.3.1 we consider *generalized equilibria* in *concave games*. Our PPAD-membership proof will capture a large class of concave games that subsumes bimatrix games, polymatrix games and more generally a class of games that we call PL succinct games (recall Remark 2.1 for the interpretation of the term PL), as well as bilinear games, general threshold games, and essentially all concave games for which Nash equilibria in rational numbers exist. In Section 4.3.2, we consider an alternative equilibrium notion, due to Kintali et al. [2013], called *personalized equilibrium*, and in Section 4.3.3 we consider ε -proper equilibria in PL succinct games (including polymatrix games), a well-known equilibrium notion due to Myerson [1978].

With the linear-OPT-gate at hand, the proofs that we will develop in this section will be conceptually very simple, basically mimicking the simplest proofs of existence for these settings. More precisely, all of our PPAD-membership results for this section are obtained as special cases of our main theorem for PLBRO games (see Theorem 4.3). In fact, we also employ the same theorem to obtain PPAD-membership results for congestion games in Section 5.

4.1 Warm-up: Nash Equilibria in Bimatrix Games via Linear Programs

We define bimatrix games below.

Definition 4.1 (Bimatrix Game). A *bimatrix game* is a strategic (normal form) game played between two players. Each player $i \in \{1,2\}$ has a finite set of pure strategies $S_i = \{1,\ldots,m_i\}$, and a payoff function $u_i = S_1 \times S_2 \to \mathbb{R}$, mapping pairs of pure strategies to payoffs. The players can also randomize over their pure strategies. A *mixed strategy* x_i of player i is a probability distribution over pure strategies in S_i . The domain Σ_i of mixed strategies for player i is the $(m_i - 1)$ -dimensional unit simplex, i.e., $\Sigma_i := \{y \in \mathbb{R}_{\geq 0}^{m_i} \colon \sum_{j=1}^{m_i} y_j = 1\}$. Given a mixed strategy x_i for player i, the jth coordinate $x_{i,j}$ denotes the probability that player i is playing pure strategy j. The expected payoff $\tilde{u}_i(x_1, x_2)$ of player i given a pair of mixed strategies (x_1, x_2) is defined as:

$$\tilde{u}_i(\mathbf{x}_1, \mathbf{x}_2) = \sum_{j_1=1}^{m_1} \sum_{j_2=1}^{m_2} x_{1,j_1} \cdot x_{2,j_2} \cdot u_i(j_1, j_2)$$
(12)

The name "bimatrix games" comes from the fact that for 2 players, the payoffs can be expressed via two $m_1 \times m_2$ matrices, one for each player.

Definition 4.2 (Nash equilibrium). A pair of mixed strategies $(\mathbf{x}_1, \mathbf{x}_2) \in \Sigma_1 \times \Sigma_2$ is a (mixed) *Nash equilibrium* of a bimatrix game if for any player $i \in \{1, 2\}$ and any mixed strategy $\mathbf{x}_i' \in \Sigma_i$ of that player, it holds that $\tilde{u}_i(\mathbf{x}_i, \mathbf{x}_{3-i}) \geq \tilde{u}_i(\mathbf{x}_i', \mathbf{x}_{3-i})$, i.e., no player can improve their expected payoff by unilaterally deviating to any other mixed strategy.

Finding Nash equilibria. We are interested in finding a Nash equilibrium of a bimatrix game, when the payoff functions u_i for each player $i \in \{1,2\}$ are given explicitly as rational numbers. This is a total search problem, since a solution is always guaranteed to exist by Nash's Theorem [Nash, 1950], the proof of which employs the Kakutani fixed point theorem [Kakutani, 1941]. Papadimitriou [1994] showed that the problem lies in PPAD, which also implies the existence of rational solutions, i.e., Nash equilibria in which the mixed strategies of the players are rational numbers. Note that Papadimitriou's proof essentially appeals to an alternative proof of Nash equilibrium existence due to Cottle and Dantzig [1968] that formulates the problem as an LCP (see also [Lemke and Howson, 1964]); we discussed the general challenges of using this approach in Section 1.2.1. Indeed, even in the simple case of bimatrix games, an argument is required against ray termination.⁷ In contrast, our linear-OPT-gate allows us to organically obtain PPAD-membership from the standard, textbook existence proof of Nash [1950], without any further arguments.

In a celebrated paper, Chen et al. [2009] showed that the problem is in fact PPAD-complete (and the hardness holds even when one aims to find approximate equilibria). Of course, Nash's theorem applies to strategic games beyond bimatrix games, i.e., games with more than 2 players. In that case however, it was known already from Nash's original work that with 3 or more players, there are games for which all Nash equilibria are irrational. For those games, it has been shown that the problem of computing exact Nash equilibria is FIXP-complete [Etessami and Yannakakis, 2010].

We will demonstrate how our linear-OPT-gate, in particular through its capability of solving linear programs of the form \mathcal{P} (see Program (2) in Section 3), can be used to rather straightforwardly show that computing Nash equilibria in bimatrix games is in PPAD. For ease of notation, let -i = 3 - i. We start by observing that given a mixed strategy \mathbf{x}_{-i} for the other player, player i can find an optimal (i.e., an expected payoff-maximizing) mixed strategy \mathbf{y}_i via the solution to the following linear program:

Linear Program \mathcal{P}

maximize
$$ilde{u}_i(\mathbf{y}_i, \mathbf{x}_{-i})$$
 subject to $\sum_{j=1}^{m_i} y_{ij} = 1$ $y_{ij} \geq 0$ $j = 1, \ldots, m_i$

"Spelling out" the objective function by substituting Equation (12) into it, it becomes:

$$\tilde{u}_i(\mathbf{y}_i, \mathbf{x}_{-i}) = \sum_{j_i=1}^{m_i} \sum_{j_{-i}=1}^{m_{-i}} y_{i,j_i} \cdot x_{-i,j_{-i}} \cdot u_i(j_i, j_{-i})$$
(13)

Notice that Equation (13) is linear in y_i , and hence P is indeed a linear program.

Now, to prove membership of finding Nash equilibria in bimatrix games in PPAD, all we need to do is to construct a function whose fixed points will be the Nash equilibria of the bimatrix game, and argue that this function can be computed by a PL arithmetic circuit containing linear-OPT-gates. This is essentially the straightforward translation of the proof via the Kakutani fixed point theorem to a PPAD-membership proof.

Theorem 4.1. Computing a Nash equilibrium of a bimatrix game is in PPAD.

Proof. We construct a PL arithmetic circuit with linear-OPT-gates $F: \Sigma_1 \times \Sigma_2 \to \Sigma_1 \times \Sigma_2$, in which $F_i(\mathbf{x}_1, \mathbf{x}_2)$ is an optimal solution of linear program \mathcal{P} for player i. Since each linear program computes a best-response mixed strategy for the corresponding player, the resulting pair of mixed strategies at a fixed point of F is necessarily a Nash equilibrium. It remains to show that the linear program \mathcal{P} can be computed by our linear-OPT-gate. Indeed, all the constraints are linear functions of the variables \mathbf{y}_i , and the gate inputs \mathbf{x}_{-i} do not appear in the constraints. The feasible domain is non-empty and bounded, and the gradient

⁷Such an argument can be found in [Cottle et al., 2009] and also implicitly in [Cottle and Dantzig, 1968] and [Lemke, 1965].

of the objective function (with respect to the variables y_i) is linear in the variables y_i and gate inputs x_{-i} and hence can be computed by a PL arithmetic circuit. Thus, by Theorem 3.1 there is an equivalent PL arithmetic circuit that does not use linear-OPT-gates and the problem lies in PPAD by Remark 3.1.

We remark that in the proofs of our results from now on, we will not explicitly reference Theorem 3.1 and Remark 3.1, as their application will be straightforward.

Before we conclude the section, we remark that for (general) games of 3 or more players, the objective function would be a polynomial of degree at least 3 in the variables and the gate inputs, and its gradient could not be given by a PL pseudo-circuit. This is of course not a coincidence, as, as we said earlier, games with 3 or more players may only have irrational Nash equilibria.

4.2 Warm-up: Equilibria of Digraph Threshold Games via Feasibility Programs

Our next "warm-up" application is that of finding equilibria in *digraph threshold games*. A digraph threshold game is played on a directed graph between n players that choose a real number in [0,1], and their payoffs depend on the chosen numbers of their incoming neighbors and a threshold. Digraph Threshold games were defined by Papadimitriou and Peng $[2021]^8$ as a tool for proving PPAD-hardness of *public goods games* on directed graphs, and were later on used to establish the PPAD-hardness of other problems, e.g., see $[Chen et al., 2021b, 2022]^9$.

Definition 4.3 (Digraph Threshold Game). A digraph threshold game $\mathcal{G}(V, E, t)$ is defined on a directed graph G = (V, E) with |V| = n, and $t \in (0, 1)$ is a threshold. The nodes of G correspond to players, and each player $i \in V$ chooses a strategy $x_i \in [0, 1]$. Let $\mathbf{x} = (x_1, \dots, x_n)$ be the vector of (pure) strategies that we will refer to as a (pure) strategy profile. Let N(i) denote the (in-)*neighborhood* of player $i \in V$, i.e., the set consisting of all the players that have arcs to player i, i.e., $N(i) = \{j : (j, i) \in E\}$. A strategy profile \mathbf{x} is an equilibrium if it satisfies

$$x_{i} = \begin{cases} 0 & \text{if } \sum_{j \in N(i)} x_{j} > t \\ 1 & \text{if } \sum_{j \in N(i)} x_{j} < t \\ [0,1] & \text{if } \sum_{j \in N(i)} x_{j} = t \end{cases}$$
(14)

where in the above expression we have abused notation, using $x_i = [0, 1]$ to denote $x_i \in [0, 1]$.

While we did use the terminology of games, digraph threshold games can alternatively be viewed as finding fixed point solutions to a set of constraints. This is why we chose this application to demonstrate the use of our feasibility programs Q (see Program (3) in Section 3) which we can obtain from our linear-OPT-gate. Indeed, Equation (14) can straightforwardly be written as a feasibility program Q as follows:¹⁰

Feasibility Program
$$Q$$

$$\sum_{j \in N(i)} x_j > t \Rightarrow y_i = 0$$

$$\sum_{j \in N(i)} x_j < t \Rightarrow y_i = 1$$

$$0 \le y_i \le 1$$

⁸Papadimitriou and Peng [2021] used the term "threshold games" to refer to these games. However, this term is ambiguous, as it has also been used to refer to a specific class of congestion games, see [Ackermann et al., 2008]. In fact, both variants have been used in relation to the general class of public goods games [Papadimitriou and Peng, 2021; Klimm and Stahlberg, 2023], which adds to the potential for confusion. For this reason, we have added the term "digraph" in front of the name for distinction.

⁹To be precise, what is used for PPAD-hardness is their approximate version, which was shown to be PPAD-complete in [Papadimitriou and Peng, 2021]. Here we show the membership in PPAD for the exact version, strengthening the membership result. ¹⁰The perceptive reader might observe that digraph threshold games look a lot like the Heaviside function introduced in Example 2.1.

 $^{^{10}}$ The perceptive reader might observe that digraph threshold games look a lot like the Heaviside function introduced in Example 2.1. Indeed, one could obtain the PPAD-membership of digraph threshold games as fixed points of functions containing only Heaviside PL pseudo-circuits rather than linear-OPT-gates. We elected to use the feasibility program formulation instead, to introduce the reader to its use in light of further applications to come later. Also note that the feasibility program $\mathcal Q$ contains equalities; these can easily be modified to inequalities by splitting the constraints into two.

Note that y_i is the only variable of this feasibility program, while the x_i , $j \in N(i)$, are gate inputs.

In turn, membership in PPAD follows rather easily by constructing an appropriate function whose fixed point coordinates are the outcomes of these feasibility programs (one for each player), and arguing that it can be computed by a PL arithmetic circuit containing linear-OPT-gates. We have the following theorem.

Theorem 4.2. Computing an equilibrium of a digraph threshold game is in PPAD.

Proof. We construct a function $F : [0,1]^n \to [0,1]^n$ in which for a vector $\mathbf{x} = (x_1, \dots, x_n)$, $F_i(\mathbf{x})$ is the outcome of the feasibility program \mathcal{Q} for player $i \in N$. By definition, $\mathbf{x} = F(\mathbf{x})$ satisfies the constraints of Equation (14) for every player, and is thus an equilibrium of the digraph threshold game. Clearly, the gate inputs only appear on the left-hand side of the conditional constraints, and \mathcal{Q} is trivially always feasible.

4.3 PPAD-membership via PL Best Response Oracles

In the applications that we presented in Sections 4.1 and 4.2, what we essentially did was construct a function $F: \times_{i \in N} D_i \to \times_{i \in N} D_i$, where D_i is the strategy space of player i, such that coordinate $F_i(\mathbf{x})$ is a best response of player i to \mathbf{x}_{-i} . It is then clear that any fixed point of F must be an equilibrium. This is in fact the application of the Kakutani fixed point theorem to show equilibrium existence. PPAD-membership then followed from the fact that the best responses could be computed via linear-OPT-gates.

In this section we will formulate the aforementioned principle as a general theorem, which will have Theorems 4.1 and 4.2, as well as several other theorems that we will prove later in this section and in Section 5, as corollaries. Below we define the very general notion of a game with PL best response oracles, when the strategy spaces D_i are convex polytopes.

Definition 4.4 (Game with PL Best Response Oracles (PLBRO Game)). A *game with PL best response oracles* (*PLBRO*) is a game in which a best response of each player $i \in N$ is outputted by an oracle $C_i : D_1 ... \times ... D_{i-1} \times D_{i+1} \times ... \times D_n \to D_i$, which is given by a PL pseudo-circuit.

An equilibrium of a PLBRO game is a vector of strategies $(x_1, ..., x_n)$, one for each player, such that each player is choosing a best response. As we will see in the applications later in the section, the notion of "best response" as well as the notion of "equilibrium" can vary, depending on the application at hand. Using the linear-OPT-gate the following theorem becomes almost trivial.

Theorem 4.3. Computing an equilibrium of a PLBRO Game is in PPAD.

Proof. We construct a function $F: \times_{i \in N} D_i \to \times_{i \in N} D_i$ in which for a vector $\mathbf{x} = (x_1, \dots, x_n)$, $F_i(\mathbf{x})$ is the outcome of the oracle $C_i(\mathbf{x}_{-i})$ for player $i \in N$. By definition, when $\mathbf{x} = F(\mathbf{x})$, \mathbf{x} is an equilibrium.

In Theorem 4.1, C_i was given by a linear-OPT-gate (which is, by definition, a PL pseudo-circuit) computing solutions of linear program \mathcal{P} , whereas in Theorem 4.2, C_i was given by a PL pseudo-circuit computing solutions of feasibility program \mathcal{Q} . In the next subsections, we present several other applications of well-known games that we prove to be PLBRO games, thus establishing their PPAD-membership.

Remark 4.1. We remark that even if one is not necessarily interested in PPAD-membership, our technique also provides an easy approach for establishing the rationality of equilibria, as this is also implied by Theorem 4.3.

Remark 4.2. In Definition 4.4, we defined the input domain of the oracle C_i to not include D_i , i.e., the domain of strategies of player i, whose best-response the oracle is calculating. This is natural, since a best-response oracle can intuitively be best understood as a device which computes the best response of a player against the chosen strategies of her opponents only. Still, in our applications in Section 5, it will be useful to extend the definition of the oracle to be a function $C_i : \times_j D_j \to D_i$, in order to be able to apply Theorem 4.3 above, which still applies for this more general setting.

4.3.1 Concave Games and Generalized Equilibria

In this section we will generalize the PPAD-membership of Nash equilibria in the previous two sections to a class of more general games, and to a more general equilibrium notion. These will be special cases of *concave games* [Rosen, 1965] and *generalized equilibria* [Debreu, 1952] that admit equilibria in rational numbers.

Concave Games. Concave games are generalizations of strategic games in which there is a set N of n players, each of which has a strategy space $\Sigma_i \in \mathbb{R}^{m_i}$ which is compact and convex. Let $\Sigma = \Sigma_1 \times \ldots \times \Sigma_n$ be the space of strategy vectors; we will refer to an element $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ of Σ as a *strategy profile*. Each player i also has a payoff function $u_i : \Sigma \to \mathbb{R}$, which is continuous in \mathbf{x} and concave in \mathbf{x}_i , assuming that the rest of the strategy profile \mathbf{x}_{-i} is fixed.

Generalized Equilibrium. An *equilibrium* of a concave game is a strategy profile \mathbf{x} in which every player chooses a payoff-maximizing element in her strategy space, i.e., $u_i(\mathbf{x}) = \max_{\mathbf{y}_i \in \Sigma_i} u_i(\mathbf{y}_i, \mathbf{x}_{-i})$ for every player $i \in N$. Debreu [1952] defined the notion of a *generalized equilibrium*, in which a strategy profile \mathbf{x} also restricts the choices in the strategy space of the players. In particular, letting $\Sigma_{-i} = \Sigma_1 \times \dots \Sigma_{i-1} \times \Sigma_{i+1} \dots \Sigma_n$, there is a correspondence $\gamma_i : \Sigma_{-i} \rightrightarrows \Sigma_i$, which specifies the set $\gamma_i(\mathbf{x}_{-i}) \subseteq \Sigma_i$ that player i is "allowed" to use. A generalized equilibrium of the game is a strategy profile \mathbf{x} in which every player chooses a payoff-maximizing element in $\gamma_i(\mathbf{x}_{-i})$, i.e., $u_i(\mathbf{x}) = \max_{\mathbf{y}_i \in \gamma_i(\mathbf{x}_{-i})} u_i(\mathbf{y}_i, \mathbf{x}_{-i})$. The existence of a generalized equilibrium was established by Debreu [1952] in concave games, when γ_i is upper and lower hemicontinuous, convex-valued and non-empty valued. Similarly to Section 4.1, given a preference profile \mathbf{x}_{-i} of the other players, the set of payoff-maximizing (allowable) strategies for player i is given by the optimal solutions to the following convex program:

Convex Program
$$C_{gen}$$
maximize $u_i(\mathbf{y}_i, \mathbf{x}_{-i})$
subject to $\mathbf{y}_i \in \gamma_i(\mathbf{x}_{-i})$

Remark 4.3 (Concave Games and Generalized Equilibria). Concave games were studied by Debreu [1952] (in fact, "quasi-concave games", where the payoff functions can be quasi-concave) in the context of his generalized equilibrium result (coined a "Social Equilibrium" there, see also [Dasgupta and Maskin, 2015]). Interestingly, Rosen [1965] also studied concave games independently, without referencing Debreu's work. Seemingly the only difference between Rosen's and Debreu's setting is that the former does not require the strategy profile space Σ to necessarily be the product space of the players' strategy spaces Σ_i , for $i \in N$.

For the standard notion of equilibrium (rather than generalized equilibrium) Debreu's theorem was concurrently and independently proven by Fan [1952] and Glicksberg [1952], and for this reason it is often referred to as *the Debreu-Fan-Glicksberg theorem for continuous games*. In fact, we will be referring to this theorem again throughout the paper, as it has been used in previous existence proofs in the applications that we consider.

It is not hard to see that concave games generalize n-player strategic games: the set of mixed strategies (i.e., randomizations over the finite set of pure strategies) is compact and convex and the payoff function is continuous in \mathbf{x} and linear in \mathbf{x}_i . Thus a mixed Nash equilibrium in a strategic game is a special case of an equilibrium (and hence of a generalized equilibrium) in a concave game.

From the above discussion, it should be obvious that (generalized) equilibria of concave games are not guaranteed to be rational (indeed, the 3-player example of irrationality of Nash [1950] still applies, for example). We will consider subclasses of these games (based on the structure of the strategy spaces and the form of the payoff functions) and these equilibria (based on the structure of the sets induced by the functions γ_i) for which rational equilibria always exist; in fact, our proof of PPAD membership will also provide the certificate of rationality. We remark that for general concave games (under the necessary

assumptions for the OPT gate for FIXP to work), Filos-Ratsikas et al. [2023] proved membership in FIXP; their theorems only apply to equilibria rather than generalized equilibria, but the extension in their setting is almost immediate.

In our setting, we need to consider special cases of convex program C_{gen} which are amenable to the use of the linear-OPT-gate. Specifically, the objective function will be such that its supergradient¹¹ with respect to \mathbf{y}_i can be given by a PL pseudo-circuit, and the constraints will be linear inequalities with the gate inputs appearing only on the right-hand side of the constraints. We will also impose a bound on the domain of the variables \mathbf{y}_i . With this we have the following convex program:

Convex Program $\mathcal C$

maximize
$$u_i(\mathbf{y}_i, \mathbf{x}_{-i})$$

subject to $A_i \cdot \mathbf{y}_i \leq b_i(\mathbf{x}_{-i})$
 $\mathbf{y}_i \in [-R_i, R_i]^{m_i}$

For the payoff function u_i , we do not require access to the function itself but rather to its supergradient with respect to \mathbf{y}_i , which is provided in the input as a PL pseudo-circuit. Here, the strategy spaces of the players are given by linear inequalities which can depend on the strategies of the other players, and generally lie within a bounded domain $[-R_i, R_i]^{m_i}$, for some $R_i > 0$. The functions $b_i(\cdot)$ are also given as PL pseudo-circuits. The following theorem is a corollary of Theorem 4.3.

Theorem 4.4. Computing a generalized equilibrium of a concave game, in which the strategy spaces are given as in the constraints of convex program C and the supergradients of the payoff functions of the players are given by PL pseudo-circuits is in PPAD.

Proof. By the discussion above, it follows directly that convex program C can be computed by a linear-OPT-gate, which can be used as the oracle C_i for each player $i \in N$ in the corresponding PLBRO game.

The class of concave games is very general, and hence captures several games of interest. We provide some examples of the type of equilibrium results that are captured by Theorem 4.4 below. We will state the theorems for the computation of generalized equilibria, noting that they hold under the constraints for $b_i(\mathbf{x}_{-i})$ required for the linear-OPT-gate to work.

Bimatrix and Polymatrix Games. The theorem establishes that *generalized* Nash equilibria in bimatrix games exist, are rational, and are in PPAD, thus generalizing Theorem 4.1. It also establishes the same properties for a natural generalization of bimatrix games, called *polymatrix games* [Janovskaja, 1968; Howson, 1972]. These are *n*-player games in which the players' payoffs are additive over several 2-player games.

Definition 4.5 (Polymatrix game). A *polymatrix game* consists of a set N of n players, each with a set S_i of pure strategies $S_i = \{1, \ldots, m_i\}$ and a domain of mixed strategies $\Sigma_i := \{y \in \mathbb{R}_{\geq 0}^{m_i} \colon \sum_{j=1}^{m_i} y_j = 1\}$. Given a mixed strategy \mathbf{x}_i for player i, the jth coordinate $x_{i,j}$ denotes the probability that player i is playing the pure strategy j. For every pair of players i and i' with $i \neq i'$, there is an associated $(m_i \times m_{i'})$ -dimensional payoff matrix $\mathbf{A}_{i,i'} \in \mathbb{R}^{m_i \times m_{i'}}$, which determines the payoffs of player i in the bimatrix game against player i'. Given a mixed strategy profile $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$, the expected payoff of player i is defined as $\tilde{u}_i(\mathbf{x}) = \sum_{i' \in N, i' \neq i} (\mathbf{x}_i)^\mathsf{T} \cdot \mathbf{A}_{i,i'} \cdot \mathbf{x}_{i'}$.

Since the gradients of the payoff functions in polymatrix games are linear functions, Theorem 4.4 has the following corollary.

Corollary 4.1. Computing a generalized mixed Nash equilibrium of a polymatrix game is in PPAD.

¹¹We consider the supergradient here rather than the subgradient, since u_i is concave rather than convex, see also Remark 3.3.

PL Succinct Games. We observe that the reason which allowed us to use the linear-OPT-gate to prove PPAD-membership for bimatrix and polymatrix games above is that we were always able to construct a PL pseudo-circuit computing the following quantity:

$$\tilde{u}_i(j, \mathbf{x}_{-i}) = \mathbb{E}_{\mathbf{s}_{-i} \sim \mathbf{x}_{-i}} u_i(j, \mathbf{s}_{-i}), \tag{15}$$

where \mathbf{s}_{-i} denotes a vector of pure strategies (i.e., a pure strategy profile) of all players besides i. The quantity above is the expected payoff of player i when using pure strategy $j \in S_i$ against the mixed strategy \mathbf{x}_{-i} of the other players. As long as we have a PL pseudo-circuit for computing $\tilde{u}_i(j,\mathbf{x}_{-i})$, Theorem 4.4 goes through. We will use the term PL succinct games to refer to those games.

Definition 4.6 (PL Succinct Game). A game is a PL succinct game if for any player i and for any pure strategy $j \in S_i$, there exists a PL pseudo-circuit computing the expected utility $\tilde{u}_i(j, \mathbf{x}_{-i})$, for any profile of mixed strategies \mathbf{x}_{-i} of the other players.

We have the following corollary of Theorem 4.4.

Corollary 4.2. Computing a generalized mixed Nash equilibrium of a PL succinct game is in PPAD.

We draw parallels between our application and that of Daskalakis et al. [2006] and Papadimitriou and Roughgarden [2008]. Those works define classes of succinct games for which there is an oracle for computing the expected utility of the player. In [Papadimitriou and Roughgarden, 2008], this oracle is referred to as the *polynomial expectation property* and is used to show that correlated equilibria [Aumann, 1974] of games with this property can be computed in polynomial time. In [Daskalakis et al., 2006], it is shown that if the oracle is given by a *bounded division free straight-line program of polynomial length*, then these games are in PPAD. Crucially, this latter result concerns *approximate equilibria*. One could view our result as a complement to those two results, one which concerns *exact* equilibria in *rational* numbers.

General Threshold Games. Theorem 4.4 also establishes the PPAD-membership of *generalized* equilibria in digraph threshold games (see Section 4.2). To see this, we may redefine a player's payoff in a digraph threshold game (Equation (14)) as:

$$u_i(x_i, \mathbf{x}_{-i}) = x_i \cdot (t - \sum_{j \in N_i} x_j), \tag{16}$$

where $\mathbf{x}_{-i} = (x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n)$. Indeed, when $\sum_{j \in N_i} x_j > t$, the payoff is maximized when $x_i = 1$, when $\sum_{j \in N_i} x_j < t$, the payoff is maximized when $x_i = 0$, and when $\sum_{j \in N_i} x_j = t$, any choice of $x_i \in [0,1]$ maximizes the payoff. Therefore the equilibria of the game under this utility function are the same as those under the original definition. Additionally, the derivative of the utility function in Equation (16) w.r.t. x_i is linear and $x_i \in [0,1]$ for every player $i \in V$. The digraph threshold game is then a concave game satisfying the conditions of the statement of Theorem 4.4. One may also add constraints on the allowable values of x_i chosen by player i, given the chosen strategies \mathbf{x}_{-i} of the other players, as those of the constraints in convex program C. This gives us the following corollary, which generalizes Theorem 4.2.

Corollary 4.3. Computing a generalized equilibrium of a digraph threshold game is in PPAD.

Our approach can actually capture the complexity of a larger class of threshold games, which we call general threshold games.

Definition 4.7 (General Threshold Game). A general threshold game $\mathcal{G}(g_1, \dots, g_n)$ is given by functions $g_i : [0,1]^{n-1} \to \mathbb{R}$ for $i \in \mathbb{N}$. A strategy profile $\mathbf{x} \in [0,1]^n$ is an equilibrium if for every agent i it satisfies

$$x_{i} = \begin{cases} 0 & \text{if } g_{i}(\mathbf{x}_{-i}) > 0\\ 1 & \text{if } g_{i}(\mathbf{x}_{-i}) < 0\\ [0,1] & \text{if } g_{i}(\mathbf{x}_{-i}) = 0 \end{cases}$$

$$(17)$$

where in the above expression we have abused notation using $x_i = [0, 1]$ to denote $x_i \in [0, 1]$, and where $\mathbf{x}_{-i} = (x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n)$.

It is not hard to see that general threshold games are a generalization of digraph threshold games. General threshold games are also concave games, as, similarly to Equation (16), we can write the payoff functions as

$$u_i(x_i, \mathbf{x}_{-i}) = x_i \cdot (-g_i(\mathbf{x}_{-i})).$$
 (18)

If the functions $g_i(\mathbf{x}_{-i})$ can be given by PL pseudo-circuits, then Theorem 4.4 applies.

Theorem 4.5. Computing a generalized equilibrium of a general threshold game is in PPAD, as long as the functions $g_i(\mathbf{x}_{-i})$ are given by PL pseudo-circuits.

Some natural general threshold games for which Theorem 4.5 applies can be defined as follows:

- digraph threshold games (Definition 4.3), for which $g_i(\mathbf{x}_{-i}) = (\sum_{i \in N_i} x_i) t$.
- max digraph threshold games, for which $g_i(\mathbf{x}_{-i}) = (\max_{j \in N_i} x_j) t$. In these games, player i chooses $x_i = 0$ if the maximum of the strategies of her incoming neighbors is above a threshold t, $x_i = 1$ if it is below, and any value in [0,1] otherwise.
- min digraph threshold games, for which $g_i(\mathbf{x}_{-i}) = (\min_{j \in N_i} x_j) t$. In these games, player i chooses $x_i = 0$ if the minimum of the strategies of her incoming neighbors is above a threshold t, $x_i = 1$ if it is below, and any value in [0,1] otherwise.

Bilinear Games. Theorem 4.4 also establishes the rationality and PPAD membership of Nash equilibria in *bilinear games*, introduced by Garg et al. [2011]. Intuitively, these generalize bimatrix games in the sense that the strategy space does not have to be the unit simplex $\Sigma_i = \{y \in \mathbb{R}^{m_i}_{\geq 0} \colon \sum_{j=1}^{m_i} y_j = 1\}$, as in Definition 4.1, but it can be any compact polytope.

Definition 4.8 (Bilinear Games [Garg et al., 2011]). A *bilinear game* is a two-player game represented by two $m_1 \times m_2$ payoff matrices \mathbf{A}_1 and \mathbf{A}_2 , one for each player, and two compact polytopal strategy sets X_1 and X_2 . Let $\mathbf{E}_1 \in \mathbb{R}^{k_1 \times m_1}$ and $\mathbf{E}_2 \in \mathbb{R}^{k_2 \times m_2}$ and let $\mathbf{e}_1 \in \mathbb{R}^{k_1}$ and $\mathbf{e}_2 \in \mathbb{R}^{k_2}$ be two vectors. The strategy spaces of players 1 and 2 respectively are defined as

$$X_1 = \{ \mathbf{x} \in \mathbb{R}^{m_1} \colon \mathbf{E}_1 \cdot \mathbf{x} = \mathbf{e}_1, \mathbf{x} \ge 0 \}, \text{ and } X_2 = \{ \mathbf{x} \in \mathbb{R}^{m_2} \colon \mathbf{E}_2 \cdot \mathbf{x} = \mathbf{e}_2, \mathbf{x} \ge 0 \}.$$

Given a pair of strategies $(\mathbf{x}_1, \mathbf{x}_2) \in X_1 \times X_2$, the payoff of player 1 is $\mathbf{x}_1^\mathsf{T} \cdot \mathbf{A}_1 \cdot \mathbf{x}_2$ and the payoff of player 2 is $\mathbf{x}_1^\mathsf{T} \cdot \mathbf{A}_2 \cdot \mathbf{x}_2$.

A Nash equilibrium of the game is defined analogously to Definition 4.2. From Definition 4.8, it follows that bilinear games are concave games in which the gradients of the payoffs of the players are linear functions. Thus Theorem 4.4 has the following corollary.

Corollary 4.4. Computing a generalized Nash equilibrium of a bilinear game is in PPAD.

We remark that the PPAD-membership of finding equilibria in bilinear games had not explicitly been proven before our work, but it can be recovered rather implicitly via observing that these games are very much related to the *sequence form* of Koller et al. [1996], for which the authors devise an LCP.

The Debreu-Fan-Glicksberg theorem [1952] and Rosen's theorem [1965]. As we mentioned earlier, the Debreu-Fan-Glicksberg theorem for continuous games (or equivalently, Rosen's theorem) is often used in the literature to prove existence of equilibria for various games. It is in a sense stronger than Brouwer's fixed point theorem [Brouwer, 1911], as it can be used to prove the existence of equilibria in more general games. Our PPAD-membership of this section also provides a tool for proving PPAD-membership of these other problems where the Debreu-Fan-Glicksberg theorem applies, as long as the games that they consider satisfy the conditions required for Theorem 4.4. Generally speaking, if the problem in question can be

reduced to that of finding equilibria of a concave game in which the supergradients of the payoff functions can be computed by PL arithmetic circuits, then Theorem 4.4 implies its membership in PPAD.

The existence proofs of several of our applications in this section and in Section 5 are in fact established via the Debreu-Fan-Glicksberg theorem. One could transform those proofs into computational reductions (and then Theorem 4.4 would apply immediately); instead, we obtain the PPAD-membership of those problems directly from the more general Theorem 4.3.

4.3.2 Personalized Equilibria

Our next application of Theorem 4.3 is to finding *personalized equilibria* in graphical games, a notion introduced by Kintali et al. [2013]. Intuitively speaking, these equilibria allow players to "match" their strategies with those of their opponents, without obeying a product distribution.

Definition 4.9 (Hypergraph Game [Kintali et al., 2013]). Consider a game \mathcal{G} played on a hypergraph G = (V, E) among n different players of a set N. Each player $i \in N$ has a finite strategy set S_i , and $|S_i \cap S_j| = \emptyset$ for all $i, j \in N$ such that $i \neq j$. Let $V = \bigcup_{i \in N} S_i$. For each player $i \in N$, we have a set $E_i \subseteq E$ of hyperedges, which satisfy the following two conditions:

- if $e \in E_i$ then $|e \cap S_i| \le 1$ for any $j \in N$,
- if $e, e' \in E_i$ are distinct, then $e \not\subset e'$.

Lastly, each player $i \in N$ has a utility function $u_i : E_i \to \mathbb{R}$. A mixed strategy \mathbf{x}_i for player $i \in N$ is a probability distribution over S_i , and a weight assignment \mathbf{w}_i is a probability distribution over E_i .

One way to interpret Definition 4.9 above in relation to the "standard" games that we have seen so far, is that the hyperedges e correspond to pure strategy profiles, interpreted as sets of pure strategies, at most one for each player, and each player can choose how much weight $w_i(e)$ to assign to each such profile. A personalized equilibrium of \mathcal{G} is defined as follows:

Definition 4.10 (Personalized Equilibrium [Kintali et al., 2013]). A personalized equilibrium of \mathcal{G} consists of a vector of mixed strategies $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ and a vector of weight assignments $\{\mathbf{w}_1, \dots, \mathbf{w}_n\}$, such that for every player $i \in N$, $(\mathbf{x}_i, \mathbf{w}_i)$ is a solution to the following linear program (with variables \mathbf{x}_i and \mathbf{w}_i):

maximize
$$\sum_{e \in E_i} w_i(e)u_i(e)$$
subject to
$$\sum_{e:s \in e} w_i(e) \le x_j(s) \quad \forall s \in S_j, \forall j \neq i$$

$$\sum_{e:s \in e} w_i(e) = x_i(s) \quad \forall s \in S_i$$

$$\sum_{e \in E_i} w_i(e) = 1$$

$$\sum_{e \in E_i} x_i(s) = 1$$

$$w_i(e) \ge 0 \qquad \forall e \in E_i$$

$$x_i(s) \ge 0 \qquad \forall s \in S_i$$

$$(19)$$

We say that G is *well-behaved* if this LP is feasible for any player i and for any choice of strategies \mathbf{x}_j for the players $j \neq i$.

Features of our proof and Kintali et al.'s PPAD-membership result. To establish existence of a personalized equilibrium in every well-behaved hypergraph game \mathcal{G} as defined above, Kintali et al. [2013] essentially reduce the game to a concave game, and then invoke the Debreu-Fan-Glicksberg theorem (1952).

This already hints at the fact that we could obtain PPAD-membership of the problem as a corollary of Theorem 4.4. We will instead obtain it as corollary of the more general Theorem 4.3 rather straightforwardly; we do that in Theorem 4.6 below. Additionally, to obtain PPAD-membership (and as a result, rationality of equilibria), Kintali et al. [2013] first define an approximate version of the problem (the ε -personalized equilibrium), and reduce that problem to END-OF-LINE (see Definition 2.1), via a relatively involved construction. To obtain PPAD-membership for the exact problem (i.e., when $\varepsilon=0$) Kintali et al. [2013] construct an elaborate argument that appeals to linear programming compactness, by first showing that for sufficiently small ε , ε -personalized equilibria "almost satisfy" the constraints of the linear programs, and then carefully rounding the solution to obtain an exact equilibrium. Our technique allows us to reduce this whole argument to a few lines.

Theorem 4.6. Computing a personalized equilibrium of a well-behaved hypergraph game (Definition 4.9) is in PPAD.

Proof. We observe that the linear program (19) computes a best response \mathbf{x}_i for player i, given the other players' strategies \mathbf{x}_j . The PLBRO game oracle \mathcal{C}_i for player $i \in N$ will be given by a linear-OPT-gate which takes as input the vector \mathbf{x}_{-i} of strategies of the other players and outputs a best-response strategy \mathbf{x}_i of player i by solving the LP (19). The LP will always be feasible since the game is well-behaved. By Theorem 4.3, the theorem follows.

4.3.3 ε -proper Equilibria in PL Succinct Games

In this section we will consider an alternative equilibrium notion, that of ε -proper equilibrium [Myerson, 1978], which refines another well-known notion, that of ε -perfect equilibrium [Selten, 1975]. Both of these notions allow the players to make small mistakes ("trembles") when choosing their optimal mixed strategies, but ensure that these mistakes happen with small probability (related to the parameter ε). Recall the definition of PL succinct games (Definition 4.6); we will show that finding ε -proper equilibria of those games is in PPAD.

First, to demonstrate the main ideas, we will consider bimatrix games, which were studied in the past in the context of TFNP and ε -proper equilibria by Sørensen [2012]. Recall that bimatrix games can be expressed by two $m_1 \times m_2$ matrices, which we will henceforth denote by \mathbf{A}_1 and \mathbf{A}_2 . We provide the definition of an ε -proper equilibrium below.

Definition 4.11 (ε -proper equilibrium in a bimatrix game [Myerson, 1978]). Let $\varepsilon > 0$. A pair of mixed strategies ($\mathbf{x}_1, \mathbf{x}_2$) is an ε -proper equilibrium of a bimatrix game if \mathbf{x}_1 and \mathbf{x}_2 are *fully mixed* (i.e., $x_{i,j} > 0$ for any pure strategy $j \in S_i$, for $i \in \{1,2\}$), and

$$(\mathbf{A}_1 \cdot \mathbf{x}_2)_j < (\mathbf{A}_1 \cdot \mathbf{x}_2)_{j'} \Rightarrow x_{1,j} \leq \varepsilon \cdot x_{1,j'} \text{ for all } j, j' \in S_1,$$

 $(\mathbf{x}_1^\mathsf{T} \cdot \mathbf{A}_2)_j < (\mathbf{x}_1^\mathsf{T} \cdot \mathbf{A}_2)_{j'} \Rightarrow x_{2,j} \leq \varepsilon \cdot x_{2,j'} \text{ for all } j, j' \in S_2.$

We remark that the ε -perfect equilibrium mentioned earlier is defined analogously, with the only difference being that ε is not multiplied by the mixed strategy $x_{i,j'}$ on the right-hand side of the constraints.

We will obtain that computing ε -proper equilibria of bimatrix games is in PPAD, essentially as a corollary of Theorem 4.3. Before we present the theorem and the proof, we discuss the idea and its advantages over the previous PPAD-membership proof due to Sørensen [2012].

Features of our proof and the previous PPAD-membership result. Sørensen [2012] first provided a PPAD-membership result for computing ε-proper equilibria. His proof proceeds by showing that an ε-proper equilibrium can be recovered as a solution to an LCP, and thus can be found by Lemke's algorithm [Lemke, 1965]. As we highlighted in Section 1.2.1, this approach already introduces complications, mainly arguing against ray termination, which is also explicitly done in [Sørensen, 2012]. Besides that, to make sure that the constructed LCP has polynomial size, Sørensen employs an *extended formulation of the generalized permutahedron* due to Goemans [2015].

Figure 4: The feasibility programs Q_1 and Q_2 used in the proof of Theorem 4.7.

Our proof is conceptually much simpler: it suffices to embed the conditions of Definition 4.11 in a set of feasibility programs Q (see Program (3) in Section 3), one for each player $i \in \{1,2\}$, see Figure 4. Each of those feasibility programs will compute the best response of the corresponding player, where a "best response" here is a mixed strategy x_i that satisfies the conditions of Definition 4.11, i.e., when a pure strategy yields smaller expected payoff, then it is played with probability which is smaller by an ε multiplicative factor. Using these feasibility programs as the oracles \mathcal{C}_1 and \mathcal{C}_2 of an PLBRO game, we obtain the proof of Theorem 4.7 as a corollary of Theorem 4.3. We state the theorem next.

Theorem 4.7. Computing an ε -proper equilibrium of a bimatrix game is in PPAD.

Proof. Consider the feasibility programs Q_1 and Q_2 of Figure 4, for players 1 and 2 respectively. By Definition 4.11, a solution to Q_i , for $i \in \{1,2\}$ is a best response of player i in the bimatrix game. Note that the antecedents (namely, the constraints on the left-hand side of the implications) only contain gate inputs. Hence, these can be computed by linear-OPT-gates, so they can be used as oracles C_1 and C_2 in the corresponding PLBRO game (Definition 4.4) as long as they are solvable. Then, the theorem follows from Theorem 4.3. Solvability of Q_i for $i \in \{1,2\}$ is easy to see, by observing that it is of the form Q_{app} , presented in Section 3.2. By Lemma 3.2, it suffices to argue that the feasibility graph is acyclic. The feasibility graph G_{Q_i} consists of vertices corresponding to the different pure strategies of player i, and an edge (j,j') is only present if the expected utility of player i from j is strictly lower than the expected utility for j'. It is straightforward to verify that G_{Q_i} is acyclic.

Theorem 4.7 establishes the PPAD-membership of computing ε -proper equilibria in bimatrix games. In the following we will show that the proof can straightforwardly be extended to larger classes of games. We first offer the following remark.

Remark 4.4 (Proper and Trembling hand perfect equilibria). Myerson [1978] defined a *proper equilibrium* to be a limit point of ε-proper equilibria for $\varepsilon \to^+ 0$. Similarly, limits points of ε-perfect equilibria are trembling hand perfect equilibria [Selten, 1975]. With this definition, proper equilibria are *refinements* of trembling hand perfect equilibria, which, in turn, are refinements of Nash equilibria. Intuitively, a refinement is a special class of equilibria characterized by a set of principles that make them more "plausible". Note that ε-proper equilibria are not refinements of (exact) Nash equilibria, and this is why we refer to them as "alternative equilibrium notions". We believe they are still natural, as they can be interpreted as a model of *limited rationality*¹², with the players being "imperfectly rational" in their decisions.

In this context, Sørensen [2012] showed a stronger result, namely that finding *symbolic* ε -proper equilibria is in PPAD. Interestingly, the problem of finding a proper equilibrium is unlikely to be in TFNP [Hansen and Lund, 2018], and hence Sørensen's result does not provide a PPAD-membership result for all proper equilibria. We refer the reader to [Sørensen, 2012] for the appropriate definitions and the details.

¹²Selten [1975] in fact introduces the notion of trembling hand perfect equilibria using a similar narrative.

PL succinct games. We now explain how Theorem 4.7 extends rather straightforwardly to PL succinct games (Definition 4.6). The definition of ε -proper equilibria in such games generalizes that of Definition 4.11 straightforwardly. Recall the definition of the expected utility of player i when playing the pure strategy j against the mixed strategy profile \mathbf{x}_{-i} of the other players (Equation (15)).

Definition 4.12 (ε-proper equilibrium (general games) [Myerson, 1978]). Let $\varepsilon > 0$. A mixed strategy profile $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_n)$ is an ε-proper equilibrium of a strategic game if for any player $i \in N$, we have that \mathbf{x}_i is fully mixed (i.e., $x_{i,j} > 0$ for any pure strategy $j \in S_i$), and for any player $i \in N$

$$\tilde{u}_i(j, \mathbf{x}_{-i}) < \tilde{u}_i(j', \mathbf{x}_{-i}) \Rightarrow x_{i,j} \le \varepsilon \cdot x_{i,j'} \text{ for all } j, j' \in S_i.$$
 (20)

To extend the feasibility programs Q_i , for $i \in N$, to general games, one only has to substitute the corresponding constraints for each player $i \in N$ with those in Definition 4.12 above. In PL succinct games, the quantities $\tilde{u}_i(j,\mathbf{x}_{-i})$ for any player i and pure strategy j can be computed by a PL pseudo-circuit. The proof of the following theorem is then very similar to that of Theorem 4.7.

Theorem 4.8. Computing an ε -proper equilibrium of a PL succinct game is in PPAD.

Proof. Again, feasibility program Q_i , for $i \in N$, captures the best response of player $i \in N$. Each Q_i can be computed by a linear-OPT-gate, and is solvable by the same very simple argument used in the proof of Theorem 4.7. Thus Q_i can be used as the oracle C_i in the corresponding PLBRO game (Definition 4.4), and the theorem follows from Theorem 4.3.

Since polymatrix games (Definition 4.5) are PL succinct games, we obtain the following corollary.

Corollary 4.5. Computing an ε -proper equilibrium of a polymatrix game is in PPAD.

A proof of Corollary 4.5 was first provided by Hansen and Lund [2018]. Their proof essentially redoes all of the steps of the proof of Sørensen [2012] for bimatrix games, extending them to the more general case, stating and proving corresponding lemmas etc. Clearly, our linear-OPT-gate allows us to avoid all this labor and extend the PPAD-membership from bimatrix games to polymatrix games, and even beyond, rather straightforwardly. We conclude our discussion on ε -proper equilibria with the following remark.

Remark 4.5 (ε -proper equilibria can be irrational). For general multiplayer games (not necessarily PL succinct), Filos-Ratsikas et al. [2023] showed a FIXP-membership result via a system of conditional convex constraints, which generalizes the feasibility program Q_i . FIXP is the right class for these games, because it has been shown (see Footnote 1 in [Etessami et al., 2014]) that there are games in which all ε -perfect equilibria (and hence all ε -proper equilibria) are irrational.

5 Congestion Games

In this section we consider models of congestion games [Wardrop, 1952; Rosenthal, 1973], where players compete for resources. More precisely, we focus our study on *non-atomic* and *atomic splittable* congestion games, and for both we consider in particular the important subclasses of network congestion games. In particular, via the employment of our linear-OPT-gate we will obtain PPAD-membership results for three domains of congestion games, namely for

- finding equilibria in congestion games with rational and malicious players, studied by Babaioff et al. [2009], see Section 5.1.3,
- finding Wardrop equilibria in multi-class non-atomic network congestion games, studied by Meunier and Pradeau [2019], see Section 5.2.1,
- finding Nash equilibria multi-class atomic splittable network congestion games, studied by [Klimm and Warode, 2020], see Section 5.2.2.

All of our PPAD-membership results hold when the latency functions (an in some cases possibly their subgradients) are given by PL pseudo-circuits (recall Definition 2.7) or simply, by PL arithmetic circuits. In particular, they apply for example to all *piecewise-linear* latency functions that are given explicitly as part of the input. Via the machinery that we develop in Section 3.4, we can in fact obtain the PPAD-membership for the more general case where the latency functions and their subgradients are provided in the input *implicitly*, via PL pseudo-circuits. As such, our results are simultaneously *significiant simplifications* over the existing results in the literature, as well as *generalizations*, since the only known PPAD-membership results so far were for *linear* latency functions.¹³ In fact, for games with malicious players, complexity results had not been proven before our work.

We remark that all the games mentioned above fall into one of two main categories, either *non-atomic* congestion games or atomic splittable congestion games. As such, finding their equilibria does not simply fall in the class PLS [Johnson et al., 1988], which is known to capture the complexity of atomic (non-splittable) congestion games. In fact, finding equilibria for some of the games studied in this section has been proven to be PPAD-complete [Klimm and Warode, 2020].

Before we proceed with the technical parts of the section, we present the previous works on these domains and compare the previous approaches to our proofs which use the linear-OPT-gate.

Congestion games with malicious players. The study of congestion games with malicious players was initiated by Karakostas and Viglas [2007]. Here, we consider the model studied by Babaioff et al. [2009], in which there is a continuum of rational players, who are trying to minimize their total load, and a single malicious player, who is trying to maximize the load experienced by the rational players. Babaioff et al. proved the existence of equilibria in such games with non-decreasing latency functions. They also prove that when the latency functions are concave, these games have pure equilibria, by appealing to the Debreu-Fan-Glicksberg theorem [1952]. This already hints that PPAD-membership can be recovered as a corollary of Theorem 4.4; we will instead obtain it as a corollary of the more general Theorem 4.3. We remark that for these games, a PPAD-membership result was not previously known.¹⁴

Multi-class non-atomic network congestion games. In multi-class non-atomic network congestion games, there is a continuum of players divided into different classes. Existence of Wardrop equilibria in this setting was first established by Schmeidler [1973], via an application of the Debreu-Fan-Glicksberg theorem [1952]. A different proof of existence was provided by Milchtaich [2000], who also studied the equilibrium uniqueness. The first computational complexity results on the problem were provided by Meunier and Pradeau [2019], who proved that the problem of finding a Wardrop equilibrium lies in PPAD, when the latency functions on the resources of the game are linear. Their proof goes via the "LCP approach" (see Section 1.2.1). It turns out that their LCP formulation cannot be solved by the vanilla version of Lemke's algorithm, and so they devise a similar pivoting algorithm, tailored to their problem. As in the case of Lemke's algorithm, they argue explicitly against ray termination.

Multi-class atomic splittable network congestion games. The existence of Nash equilibria in atomic splittable congestion games follows from the Debreu-Fan-Glicksberg theorem [1952], see [Klimm and Warode, 2020]. The computational complexity of the problem with player-specific linear latency functions was studied by Klimm and Warode [2020], who proved its PPAD-completeness. What is important for us is the most challenging part of that result, which is the PPAD-membership. Klimm and Warode's proof is rather involved, and goes via the development of a homotopy method for tracing an equilibrium given the demand rates of the players. This gives rise to a new pivoting algorithm, similar in spirit to Lemke's

¹³In the context of congestion games, the term "affine" is often used instead of "linear". Boyd and Vandenberghe [2004] define a linear function as of the form $f(x) = c^T x$, and an affine function as the sum of a linear function and a constant. We use the term "linear" to refer to all affine functions.

¹⁴Babaioff et al. [2009] mention in the conclusion of their work that their existence proofs establish membership of the problem in PPAD. This claim is however not straightforward, and there is no proof to support it. In any case, such a claim would most certainly refer only to *approximate* equilibria, as these games do not always have rational equilibria for any concave latency function.

algorithm, or more precisely, the Lemke-Howson algorithm (see Section 1.2.1). The method solves the problem of finding a Nash equilibria as a system of linear equations involving *excess flows*, *vertex potentials* and *block Laplacians*. At a very high level, the authors use the excess and potentials to define an undirected version of the End-of-Line graph (see Definition 2.1), and the determinant of the block Laplacians to define a unique orientiation of the edges, effectively reducing the problem to End-of-Line. It is interesting to mention that Klimm and Warode do mention that the Nash equilibrium problem can be formulated as an LCP, but it is unclear whether Lemke's algorithm can solve it, motivating the development of their new algorithm.

Our results and proofs. The linear-OPT-gate allows us to avoid any of the technical complications of the proofs of Meunier and Pradeau [2019] and Klimm and Warode [2020] (which are rather involved, especially the latter), and essentially obtain the PPAD-membership for all of these problems as simple corollaries of Theorem 4.3 (or even Theorem 4.4, as the games in this section are PL concave). In fact, as we mentioned earlier, we obtain generalizations of those results, from linear latency functions to any latency function that is given (in some cases together with its subgradient) by a PL arithmetic circuit. In particular, they capture all piecewise-linear latency functions that are given explicitly as part of the input, as well as those where the latencies and their subgradients are given implicitly by the aforementioned circuits (see Section 3.4). For congestion games with malicious players, our PPAD-membership result for this same class of latency functions is the first such complexity result for any version of the problem.

5.1 Multi-class Congestion Games

In the models we consider, the players are divided into a finite number k of classes, and we refer to these as multi-class congestion games. The main difference between the different models we consider arise from how the classes of players are interpreted. Apart from this, the models have many commonalities, and we start by describing these.

Remark 5.1 (Notation). In this section we shall for convenience make use of the (common) shorthand notation of writing a function argument as a subscript. That is, we may write f_a in place of f(a), for a given function $f: A \to B$.

Definition 5.1 (Multi-class Congestion Games). A multi-class congestion game with a finite number of k classes is given by a finite set E of resources and, for each class $i \in [k]$, a set of pure strategies $\Sigma_i \subseteq 2^E$, consisting of subsets of resources. Each resource is equipped with class-dependent continuous latency functions, denoted by $\ell_e^i \colon \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ for $e \in E$ and $i \in [k]$. Note that we do not assume that the latency functions are non-decreasing. A load on resources is a function $x \colon E \to \mathbb{R}_{\geq 0}$. The latency functions extend additively to latency functions for pure strategies that are functions of loads by letting $\ell_S^i(x) = \sum_{e \in S} \ell_e^i(x_e)$.

We associate to each class $i \in [k]$ a positive weight demand d_i and let $d = d_1 + \cdots + d_k$ be the total weight demand. A load allocation for class $i \in [k]$ is a function $f^i : \Sigma_i \to \mathbb{R}_{\geq 0}$ such that $\sum_{S \in \Sigma_i} f_S^i = d_i$. The load allocation f^i induces a load $x^i : E \to \mathbb{R}_{\geq 0}$ for class $i \in [k]$ and a resulting total load $\overline{x} : E \to \mathbb{R}_{\geq 0}$ given as follows.

$$x_e^i = \sum_{\substack{S \in \Sigma_i \\ e \in S}} f_S^i , \quad e \in E, i = 1, \dots, k .$$

$$\overline{x}_e = \sum_{i=1}^k x_e^i , \quad e \in E .$$

$$(21)$$

5.1.1 Non-atomic congestion games

In the *non-atomic* model, each class of players represents a continuum of players. Formally we consider the set of all players to be given by a bounded real interval I of length d, partitioned into k measurable sets I_1, \ldots, I_k with respect to the Lebesgue measure μ , such that $\mu(I_i) = d_i$, for $i \in [k]$, thereby defining the

k classes of players. A strategy profile is a measurable function $\rho: I \to 2^E$ such that $\rho(I_i) \subseteq \Sigma_i$ for every $i \in [k]$. The restriction $\rho_i = \rho_{\restriction I_i}$ of ρ to the subset I_i yields the strategy profile $\rho_i \colon I_i \to \Sigma_i$ of the players of class i. The strategy profile induces a load allocation profile $f = (f^1, \ldots, f^k)$ by letting $f_S^i = \mu(\rho^{-1}(S))$ for $S \in \Sigma_i$ and $i \in [k]$, which in turn defines a load profile $x = (x^1, \ldots, x^k)$ and the total load x on resources.

The central equilibrium notion for non-atomic congestion games is the *Wardrop equilibrium* [Wardrop, 1952], here defined for multi-class congestion games. The notion is very similar to the Nash equilibrium that we used in the previous sections. In fact, the Wardrop equilibrium coincides with the notion of Nash equilibrium for this formulation in terms of a continuity of players. We will use the term Wardrop equilibrium to refer to equilibria in non-atomic games, and the term *Nash equilibrium* or simply *equilibrium* to refer to atomic splittable games, see Section 5.1.2.

Definition 5.2 (Wardrop equilibrium). A strategy profile ρ is a Wardrop equilibrium if for all $i \in [k]$ it holds that $\rho(I_i) \subseteq \arg\min_{S \in \Sigma_i} \ell_S^i(\overline{x})$, where \overline{x} is the total load induced by ρ .

We shall also say that the load allocation profile induced by a Wardrop equilibrium ρ *itself* is a Wardrop equilibrium. Note that a different strategy profile ρ' may induce the same load allocation as profile ρ without being a Wardrop equilibrium. In that case, however, there would exist another Wardrop equilibrium ρ'' such that ρ' and ρ'' differ only on a null set with respect to μ .

Definition 5.3 (Wardrop equilibrium — load allocation formulation). A load allocation profile $f = (f^1, \ldots, f^k)$ is a Wardrop equilibrium if $\ell_S^i(\overline{x}) = \min_{S' \in \Sigma_i} \ell_{S'}^i(\overline{x})$, whenever $f_S^i > 0$, for all i, where \overline{x} is the total load defined by (f^1, \ldots, f^k) .

Before presenting a fixed point formulation of Wardrop equilibria, we introduce some further notation, which will be used again in the setting of atomic splittable congestion games in Section 5.1.2 below.

Given a load allocation $f=(f^1,\ldots,f^k)$ we let $f^{-i}=(f^1,\ldots,f^{i-1},f^{i+1},f^k)$ denote the load allocation profile for classes different from i. For $i\in[k]$ and $S\in\Sigma_i$, we define the function L_S^i as follows. For a load allocation profile f and a load allocation g^i for class i, we let

$$L_S^i(g^i, f^{-i}) = \sum_{e \in S} \ell_e^i \left(\sum_{\substack{T \in \Sigma_i \\ e \in T}} g_T^i + \sum_{\substack{j \neq i}} \sum_{\substack{T \in \Sigma_j \\ e \in T}} f_T^j \right)$$
(22)

Note that if $x^{-i} = (x^1, \dots, x^{i-1}, x^{i+1}, x^k)$ are the induced loads by f^{-i} , and y^i is the induced load by g, then $L^i_S(g^i, f^{-i}) = \sum_{e \in S} \ell^i_e(y^i_e + \sum_{j \neq i} x^j_e)$. In particular, if \overline{x} is the total load induced by f, then $L^i_S(f) = \ell^i_S(\overline{x})$.

Let $i \in [k]$ and consider the following linear program in variables g_S^i .

Linear Program \mathcal{P}_{WE}

$$\begin{array}{ll} \text{minimize} & \sum\limits_{S \in \Sigma_i} L_S^i(f) g_S^i \\ \\ \text{subject to} & \sum\limits_{S \in \Sigma_i} g_S^i = d_i \\ \\ & g_S^i \geq 0, \text{ for all } S \in \Sigma_i \end{array}$$

An optimal solution $g^i=(g^i_S)_{S\in\Sigma_i}$ must satisfy that $L^i_S(f)=\min_{S'\in\Sigma_i}L^i_{S'}(f)$ whenever $g^i_S>0$. This gives rise to a fixed point characterization of Wardrop equilibrium.

Proposition 5.1. A load allocation profile $f = (f^1, ..., f^k)$ is a Wardrop equilibrium if and only if f^i is an optimal solution to the linear program \mathcal{P}_{WE} for all $i \in [k]$.

Looking ahead, the characterization of Proposition 5.1 via linear program \mathcal{P}_{WE} will be employed to prove PPAD-membership. In particular, each \mathcal{P}_{WE} will capture the best response of each player, and a linear-OPT-gate will serve as the oracle for computing this best response. This effectively will render the game a PLBRO game (Definition 4.4), and PPAD-membership will follow from Theorem 4.3. We present a concrete application of this in Section 5.1.3 below. First, we define the setting of atomic splittable congestion games.

5.1.2 Atomic splittable congestion games

In the atomic splittable model each class is representing a single player controlling the entire weight demand of the class. We shall thus refer to class i simply as player i, and the strategy space of player i is the set of load allocations of class i. Given a load allocation profile f, player i experiences latency $\sum_{S \in \Sigma_i} L_S^i(f) f_S^i$ which becomes the cost function C^i the player wishes to minimize. We thus define $C^i(g^i, f^{-i}) = \sum_{S \in \Sigma_i} L_S^i(g^i, f^{-i}) g_S^i$.

The central equilibrium notion for atomic splittable congestion games we consider is the *Nash equilibrium* with respect to these cost functions.

Definition 5.4 (Nash Equilibrium — load allocation formulation). A load allocation profile $f = (f^1, ..., f^k)$ is a *Nash equilibrium* if and only if $C^i(f) \leq C^i(g^i, f^{-i})$ for all $i \in [k]$.

If one imposes an additional convexity assumption involving the latency functions, then the game essentially becomes a concave game, and thus possesses a Nash equilibrium by the Debreu-Fan-Glicksberg theorem [1952]. The following lemma follows immediately from definitions.

Lemma 5.1 (Convexity assumption for latency functions). Suppose that for all $i \in [k]$ and $e \in E$ the latency function ℓ_e^i satisfies that the function $x \mapsto \ell_e^i(x+c)x$ is convex for all $c \ge 0$. Then, for all $i \in [k]$ and all load allocation profiles f the cost function $g^i \mapsto C^i(g^i, f^{-i})$ is convex.

Remark 5.2. In the literature on congestion games, a common assumption of a latency function ℓ is that the function $x \mapsto \ell(x)x$ is convex. This is in general not sufficient to guarantee that the function $x \mapsto \ell(x+c)x$ for $c \ge 0$ is convex, as required in the model we consider. It is however sufficient if one additionally assumes that ℓ is non-decreasing.

With the assumptions of Lemma 5.1 we consider the following convex program in variables g_S^t .

Convex Program \mathcal{C}_{NE}

minimize
$$\sum_{S \in \Sigma_i} L_S^i(g^i, f^{-i}) g_S^i$$
 subject to $\sum_{S \in \Sigma_i} g_S^i = d_i$ $g_S^i \geq 0$, for all $S \in \Sigma_i$

Directly from definitions we then have the following.

Proposition 5.2. In a congestion game with latency functions satisfying the assumptions of Lemma 5.1, a load allocation profile $f = (f^1, ..., f^k)$ is a Nash equilibrium if and only if f^i is an optimal solution to the convex program C_{NE} for all $i \in [k]$.

Again, convex program \mathcal{C}_{NE} essentially calculates the best response for each player, and hence will be computed by an oracle given by a linear-OPT-gate. We will apply this concretely in Section 5.2.1 in the context of network congestion games, but after we will have reformulated \mathcal{C}_{NE} via a more concise description of the strategies in terms of flows on resources rather than load allocations.

Linear Program $\mathcal{P}^R_{ ext{WE}}$	Convex Program $\mathcal{C}_{ ext{NE}}^{M}$
minimize $\sum_{S \in \Sigma_R} L_S^R(f^R, f^M) g_S^R$	maximize $\sum_{S \in \Sigma_R} L_S^R(f^R, g^M) f_S^R$
subject to $\sum_{S \in \Sigma_R} g_S^R = d_R$	subject to $\sum_{S \in \Sigma_M} g_S^M = d_M$
$g_S^R \geq 0$, for all $S \in \Sigma_i$	$g_S^M \ge 0$, for all $S \in \Sigma_i$

Figure 5: The linear program \mathcal{P}_{WE}^R for the rational players (left) and the convex program \mathcal{C}_{NE}^M for the malicious player (right), in the fixed point formulation of the equilibrium.

5.1.3 PPAD-membership of congestion games with malicious players

A model of non-atomic congestion games with a *malicious* player was introduced by [Babaioff et al., 2009]. In this model there are two classes of players, the *rational* players and the *malicious* player. We shall use the index set $\{R, M\}$ for these classes rather than $\{1, 2\}$. The class of rational players represents a continuum of players. As in the basic case of non-atomic congestion games as given in Section 5.1.1, these are given by a bounded real interval I of length d_R and a strategy of the rational players is a measureable function $\rho_R \colon I \to \Sigma_R$. The strategy ρ induces a load allocation f^R , and we will use that to re-express the equilibrium condition in terms of f^R .

The class of the malicious player represents a single player that controls a weight demand d_M , as in the case of atomic splittable games as given in Section 5.1.2. However, unlike the basic model of atomic splittable congestion games, the malicious player can use a *mixed* strategy, i.e., a probability distribution ρ_M over load allocations f^M . Since the malicious player is using a mixed strategy, what the rational players aim to minimize is the expected load of their chosen strategy. The malicious player on the other hand aims to maximize the expected total load experienced by the rational players. We define an equilibrium of this game explicitly below.

Definition 5.5. The strategy profile (ρ_R, ρ_M) is an *equilibrium* if the following conditions hold, where x^R is the load induced by ρ^R and x^M is the load induced by f^M .

- 1. $\rho_R \subseteq \arg\min_{S \in \Sigma_R} \mathbb{E}_{f^M \sim \rho_M} [\ell_S^R(x^R + x^M)].$
- 2. $\operatorname{supp}(\rho_M) \subseteq \operatorname{arg\,max}_{f^M} \sum_{e \in E} \ell_S^R(x_e^R + x_e^M) x_e^R$.

We reformulate Definition 5.5 in terms of load allocations for the rational players, similarly to the case of Wardrop equilibria, and thus view the pair (f^R, ρ_M) as a strategy profile. For convenience we use the function L_S^i defined in Section 5.1.1.

Definition 5.6. The strategy profile (f^R, ρ_M) is an equilibrium if

- 1. $\mathbb{E}_{f^M \sim \rho_M} L_S^R(f^R, f^M) = \min_{S' \in \Sigma_R} \mathbb{E}_{f^M \sim \rho_M} L_{S'}^R(f^R, f^M)$ whenever $f_S^R > 0$.
- 2. $\operatorname{supp}(\rho_M) \subseteq \operatorname{arg\,max}_{f^M} \sum_{S \in \Sigma_R} L_S^R(f^R, f^M) f_S^R$.

Babaioff et al. [2009] proved that every congestion game with a malicious player with non-decreasing latency functions has an equilibrium. They also proved, employing the Debreu-Fan-Glicksberg theorem, that when the latency functions are concave, such a game has a pure equilibrium, i.e. an equilibrium of the form (f^R, f^M) . We will obtain a PPAD-membership result for this case. The result will be based on the following fixed formulation of the problem, very similar to that of linear program \mathcal{P}_{WE} presented in Section 5.1.1, and convex program \mathcal{C}_{NE} presented in Section 5.1.2, refined for the problem at hand.

For the rational players, best responses are expressed by linear program \mathcal{P}_{WE}^R in variables g^R of the left-hand side of Figure 5.. For the malicious player, the best response is expressed by convex program \mathcal{C}_{NE}^M in variables g^M of the right-hand side of Figure 5.

From the fixed point formulation in terms of the programs of Figure 5, PPAD-membership is almost immediate via arguing that those best response can be computed by linear-OPT-gates, as long as the subgradients of the objective functions can be computed by a PL pseudo-circuit. Hence we have the following theorem.

Theorem 5.1. Computing an equilibrium of a congestion game with a malicious player in which the latency functions are given by PL pseudo-circuits and their subgradients are given by PL pseudo-circuits computing piecewise-constant functions is in PPAD.

Proof. Linear program \mathcal{P}_{WE}^R of Figure 5 computes the best response of the class of rational players, and \mathcal{C}_{NE}^M computes the best response of the malicious players. As long as those optimization programs can be computed by linear-OPT-gates, then the game is a PLBRO game (Definition 4.4, noting Remark 4.2 as the oracles also input a players' own strategy) and PPAD-membership follows from Theorem 4.3. Indeed, for both programs the feasible domain is non-empty and bounded with no gate inputs appearing in the left-hand side of the constraints. By the statement of the theorem, the latency functions can be computed by PL pseudo-circuits. In the case of linear program \mathcal{P}_{WE}^R , the subgradient is a sum of latency functions (one for each $S \in \Sigma_R$), which by extension can also be computed by a PL pseudo-circuit. For convex program \mathcal{C}_{NE}^M , by the definition of the latency function $L_S^R(f^R, g^M)$ in Equation (22), we can use the PL pseudo-circuits that compute the latency functions, as well as those that compute their subgradients. Those subgradients are piecewise-constant functions, which we then can multiply by the variables g^M by the machinery developed in Section 3.4. In the end, we obtain a function which can be computed by a PL pseudo-circuit, and the whole linear program can be computed by a linear-OPT-gate. □

A direct corollary of Theorem 5.1 is to the case when the latency functions are piecewise-linear functions given explicitly as part of the input. Indeed, in that case the objective functions are piecewise-quadratic and their subgradients are piecewise-linear functions, hence computable by a PL pseudo-circuit.

We complement Theorem 5.1 above with an example showing the tightness of the class of latency functions considered in the statement of theorem. Indeed, if one moves to more general concave latency functions, then the only equilibria of the game may be irrational.

Example 5.1 (Only irrational equilibria). Consider a game with only rational players (the game with a malicious player is a generalization). Define the function $f: \mathbb{R}_{>0} \to \mathbb{R}_{>0}$ by

$$f(x) = \begin{cases} 3x - x^2 & \text{for } x \le 1\\ 1 + x & \text{for } x > 1 \end{cases}.$$

The function f passes through the origin and is continuous, differentiable, increasing, and concave. Consider the non-atomic congestion game with two resources 1 and 2, and latency functions $l_1(x) = f(x)$ and $l_2(x) = 2f(x)$. If (x, 1 - x) is an equilibrium, then it must hold that $3x - x^2 = 2(3(1 - x) - (1 - x)^2)$ which as $x \in [0, 1]$ implies that $x = (\sqrt{41} - 5)/2$.

We remark that for games with general concave latency functions, a FIXP-membership result is implied by (although not explicitly stated in) [Filos-Ratsikas et al., 2023], as these games are concave games.

5.2 Network Congestion Games

We now define an interesting subclass of congestion games, called *network congestion games*. Network congestion games allow the players' strategies to be succinctly described by a flow network.

Multi-commodity flow networks To be able to define multi-class network congestion games we consider a *multi-commodity flow model*. More precisely, consider networks given by a directed graph G = (V, E) on which we consider routing of k commodities. For each commodity $i \in [k]$ we are given a source node $s_i \in N$, a target node $t_i \in N$, and a flow demand $d_i > 0$. A cost function c is in general a function $c \in E \to \mathbb{R}$, but we will however only consider non-negative cost functions. We assume separate cost functions c^i for each commodity $i \in [k]$.

A flow x is a function $x : E \to \mathbb{R}$. To be feasible, the flow x is required to be non-negative. The balance b(x) of x is the function given by

$$b(x)_u = \sum_{uv \in E} x_{uv} - \sum_{vu \in E} x_{vu} \text{ for } u \in N.$$

For every $i \in [k]$, define the vector b^i by

$$b_{s_i}^i = d_i, \ b_{t_i}^i = -d_i, \ \text{and} \ b_u^i = 0, \ \text{for every} \ u \in N \setminus \{s_i, t_i\}.$$

The cost of x with respect to commodity i is $\sum_{uv \in E} c^i_{uv} x_{uv}$.

A multi-commodity flow in G consists of a *flow profile* $x=(x^1,\ldots,x^k)$ of flows for each commodity $i\in [k]$. The individual flow x^i is *feasible* if x^i is nonnegative and satisfies the balance constraint $b(x^i)=b^i$. We will say that the flow profile x is feasible if x^i is feasible for all i. The total flow \overline{x} induced by such a multi-commodity flow $x=(x^1,\ldots,x^k)$ is the sum $\overline{x}=\sum_{i=1}^k x^i$.

For a directed path P in G, we denote by γ^P the unit path flow given by

$$\gamma_{uv}^P = 1$$
 for $uv \in P$ and $\gamma_{uv}^P = 0$ for $uv \notin P$.

Similarly, for a directed cycle C in G, we denote by γ^{C} the unit cycle flow given by

$$\gamma_{uv}^{C} = 1$$
 for $uv \in C$ and $\gamma_{uv}^{C} = 0$ for $uv \notin C$.

The costs $c^{i}(P)$ and $c^{i}(C)$ of P and C with respect to commodity i are given as

$$c^i(P) = \sum_{uv \in P} c^i_{uv}$$
 and $c^i(C) = \sum_{uv \in C} c^i_{uv}$.

Note that $c^i(P) = c^i(\gamma^P)$ and $c^i(C) = c^i(\gamma^C)$. Let \mathcal{P}_i denote the set of all directed (s_i, t_i) paths in G and let $\mathcal{P} = \bigcup_{i=1}^k \mathcal{P}_i$. Let \mathcal{C} denote the set of all directed cycles in G.

A routing of commodity i in G is a flow x^i of the form $x^i = \sum_{P \in \mathcal{P}_i} \alpha_P \gamma^P$, where $\sum_{P \in \mathcal{P}_i} \alpha_P = d_i$ and $\alpha_P \geq 0$ for every $P \in \mathcal{P}_i$. Note that x^i is feasible by definition. A multi-commodity routing is a multi-commodity flow (x^1, \ldots, x^k) such that x^i is a routing of commodity i for every $i \in [k]$. The network model we consider is uncapacitated, i.e., it does not place any upper limits on the flow on arcs. This means that a minimum cost routing must place non-zero flow only on minimum cost paths.

Lemma 5.2. Suppose that $x^i = \sum_{P \in \mathcal{P}_i} \alpha_P \gamma^P$ is a routing of commodity i minimizing the cost $c^i(x^i)$. Then $c^i(P) = \min_{P' \in \mathcal{P}_i} c^i(P')$ whenever $\alpha_P > 0$.

It is well known that any flow may be decomposed path and cycle flows. We shall make use of the following special case.

Lemma 5.3. Let x^i be a feasible flow for commodity i. Then there are $\alpha_P \geq 0$ for $P \in \mathcal{P}_i$ and $\beta_C \geq 0$ for $C \in \mathcal{C}$ such that $x^i = \sum_{P \in \mathcal{P}_i} \alpha_P \gamma^P + \sum_{C \in \mathcal{C}} \beta_C \gamma^C$.

If $b(x^i) = b^i$ the cost of x^i with respect to commodity i is thus given as $c(x^i) = \sum_{P \in \mathcal{P}_i} \alpha_P c^i(P) + \sum_{C \in \mathcal{C}} \beta_C c^i(C)$. We have the following direct consequence.

Corollary 5.1. Let x^i be a feasible flow for commodity i minimizing the cost $c^i(x^i)$. If $c^i(C) > 0$ for any cycle C for which there exist $uv \in E$ such that $x^i_{uv} > 0$, then x^i is a routing of commodity i.

Multi-class network congestion games. From a multi-commodity flow network G = (V, E) we naturally obtain a multi-class congestion game. The set of resources is simply the set E of arcs in the network and the set Σ_i of pure strategies of class i is equal to the set \mathcal{P}_i of directed (s_i, t_i) paths in G. This gives a very succinct description of the strategies that allows the relevant optimization problems to be formulated with variables corresponding to resources rather than strategies. Note that a load allocation of class i in the congestion game corresponds exactly to a routing of commodity i in G.

For the case of Wardrop equilibrium we consider the following linear program in variables y_{uv}^i , where $y^i = (y_{uv}^i)_{uv \in E}$, x is a given non-negative multi-commodity flow and \overline{x} the resulting total flow.

Linear Program
$$\mathcal{P}_{WE}^{G}$$
minimize $\sum_{uv \in E} \ell_{uv}^{i}(\overline{x}_{uv})y_{uv}^{i}$
subject to $b(y^{i}) = b^{i}$
 $y^{i} \geq 0$

This linear program is simply expressing minimum cost feasible flows for commodity i with costs given by the latency functions and the total flow \overline{x} . In order to identify these minimum cost flows with minimum cost routing , using Corollary 5.1, we need to impose a very mild restriction on the latency functions.

Definition 5.7. We say that the latency functions l_{uv}^i are without *free cycles* if for every $i \in [k]$, every $C \in \mathcal{C}$ and every feasible flow x, for which $x_{uv} > 0$ for some $uv \in C$, we have $\sum_{uv \in C} \ell^i(x_{uv}) > 0$.

We have the following fixed point characterization of Wardrop equilibrium in network congestion games, using Lemma 5.2 and Corollary 5.1.

Proposition 5.3. In a network congestion games with latency functions without free cycles, a flow $x = (x^1, ..., x^k)$ is a Wardrop equilibrium if and only x^i is an optimal solution to the linear program \mathcal{P}_{WE}^G for all $i \in [k]$.

For the case of atomic splittable congestion games and with the assumptions of Lemma 5.1 we consider the following convex program in variables y_{uv}^i , where $y^i = (y_{uv}^i)_{uv \in E}$.

$$\begin{array}{ll} & \underline{\text{Convex Program } \mathcal{C}_{\text{NE}}^G} \\ \text{minimize} & \sum_{uv \in E} \ell_{uv}^i (y_{uv}^i + \sum_{j \neq i} x_{uv}^j) y_{uv}^i \\ \text{subject to} & b(y^i) = b^i \\ & v^i > 0 \end{array}$$

Analogously to the case of the linear program \mathcal{P}_{WE}^G we would like to ensure that optimal solutions to \mathcal{C}_{NE}^G are routings of commodity i. Here however, the cost functions of the flow depend on the flow variables, and we thus require a stronger assumption on the latency functions.

Proposition 5.4. In a network congestion game with non-decreasing latency functions without free cycles and satisfying the assumptions of Lemma 5.1, a flow $x = (x^1, ..., x^k)$ is a Nash equilibrium if and only if x^i is an optimal solution to the convex program C_{NE}^G for all $i \in [k]$.

Proof. We need to prove that each x^i that is an optimal solution to $\mathcal{C}_{\mathrm{NE}}^G$ is a routing of commodity i. By Lemma 5.3 we have $\alpha_P \geq 0$ for $P \in \mathcal{P}_i$ and $\beta_C \geq 0$ for $C \in \mathcal{C}$ such that $x^i = \sum_{P \in \mathcal{P}_i} \alpha_P \gamma^P + \sum_{C \in \mathcal{C}} \beta_C \gamma^C$. Suppose for contradiction that there exists $C \in \mathcal{C}$ such that $\gamma_C > 0$ and define the new flow $\hat{y}^i = y^i - \beta_C \gamma^C$. Since γ^C is a cycle flow and $\beta_C \gamma^C \leq y^i$ it follows that \hat{y}^i is feasible. Since the latency functions are non-decreasing it follows that $\ell^i_{uv}(\hat{y}^i_{uv} + \sum_{j \neq i} x^j_{uv}) \leq \ell^i_{uv}(\hat{y}^i_{uv} + \sum_{j \neq i} x^j_{uv})$ for all $uv \in E$, and thus also $\ell^i_{uv}(\hat{y}^i_{uv} + \sum_{j \neq i} x^j_{uv})\hat{y}^i_{uv} \leq \ell^i_{uv}(\hat{y}^i_{uv} + \sum_{j \neq i} x^j_{uv})y^i_{uv}$ for all $uv \in E$. Now, since the latency functions are without free cycles there exist $uv \in C$ such that $\ell^i(y^i_{uv} + \sum_{j \neq i} x^j_{uv}) > 0$. It follows that $\ell^i(\hat{y}^i_{uv} + \sum_{j \neq i} x^j_{uv})\hat{y}^i_{uv} \leq \ell^i(y^i_{uv} + \sum_{j \neq i} x^j_{uv})\hat{y}^i_{uv}$. Taken together this contradicts the optimality of y^i .

5.2.1 PPAD-membership for non-atomic network congestion games

The PPAD-membership result for finding Wardrop equilibria in multi-class network non-atomic congestion games follows almost immediately from the fixed point characterization of equilibria in Proposition 5.3. The following result generalizes that of Meunier and Pradeau [2019].

Theorem 5.2. Finding a Wardrop equilibrium of a multi-class network non-atomic congestion game when the latency functions are without free cycles and they are given by PL pseudo-circuits is in PPAD.

Proof. Linear program \mathcal{P}_{WE}^G can be interpreted as computing the best response of a player corresponding to class i, for the equilibrium concept of Wardrop equilibrium. \mathcal{P}_{WE}^G obviously has a non-empty and bounded feasible domain. Since the latency functions ℓ^i_{uv} can be computed by PL pseudo-circuits, the subgradient of the objective function can be computed by a PL pseudo-circuit, and hence the whole linear program can be computed by a linear-OPT-gate. By using this linear-OPT-gate as the oracle, the game becomes a PLBRO game (Definition 4.4), and the theorem follows as a corollary of Theorem 4.3.

Again, a direct corollary of Theorem 5.2 is to the case when the latency functions are piecewise-linear functions given explicitly as part of the input. Indeed, in that case the objective functions are piecewise-quadratic and their subgradients are piecewise-linear functions, hence computable by a PL pseudo-circuit.

Remark 5.3 (No free cycles). As we mentioned earlier, the no free cycles condition (see Definition 5.7 is rather mild one. It is in fact a milder condition that the usual condition in this literature, which requires that if $l_e(x) = 0$, then x = 0, i.e., the latencies are 0 only on 0 points. The works on linear latency functions implicitly make this assumption, and we obtain strict generalizations of those via considering more general latency functions.

For completeness, we also provide a simple example in which, if one goes beyond the latency functions captured by Theorem 5.2, e.g., to quadratic latency functions, then the game may possess only irrational Wardrop equilibria.

Example 5.2 (Only irrational Wardrop equilibria). Consider a very simple nonatomic network congestion game where there is only one class given by the interval [0,1]. The node set V of the network G consists of two nodes s and t. For the arc set E, there are two arcs e_1 and e_2 from s to t for the class, with latency functions $l_{e_1}(x) = x^2$ and $l_{e_2}(x) = 2x^2$. Now if (x, 1 - x) is an equilibrium, then it must hold that $x^2 = 2 \cdot (1 - x)^2$ which as $x \in [0, 1]$ implies that $x = 2 - \sqrt{2}$.

Example 5.2 establishes that the class captured by our Theorem 5.2 is tight, as if one moves to larger classes of latency functions, Wardrop equilibria may be irrational. We remark that for these cases, a FIXP-membership result follows from the results of [Filos-Ratsikas et al., 2023] (althought not explicitly stated there), since multi-class non-atomic network congestion games are concave games.

5.2.2 PPAD-membership for atomic splittable network congestion games

In the same fashion as in the previous section, the PPAD-membership result for finding Nash equilibria in multi-class network atomic congestion games follows almost immediately from the fixed point characterization of equilibria in Proposition 5.4. The following result generalizes that of Klimm and Warode [2020]. We also again note the remarkable simplification of the proof that the linear-OPT-gate allows us to achieve, compared to that of Klimm and Warode [2020].

Theorem 5.3. Finding a Nash equilibrium of a multi-class network atomic splittable congestion game when the latency functions are without free cycles, they are given by PL pseudo-circuits and their subgradients are given by PL pseudo-circuits computing piecewise-constant functions is in PPAD.

Proof. The proof is very similar to that of Theorem 5.2 above. Convex program C_{NE}^G can be interpreted as computing the best response of a player corresponding to class i. C_{NE}^G obviously has a non-empty

and bounded feasible domain. To compute the subgradient of the objective function, we can use the PL pseudo-circuits that compute the latency functions, as well as those that compute their subgradients. Those subgradients are piecewise-constant functions, which we then can multiply by the variables y_{uv} by the machinery developed in Section 3.4. In the end, we obtain a function which can be computed by a PL pseudo-circuit, and the whole linear program can be computed by a linear-OPT-gate. By using this linear-OPT-gate as the oracle, the game becomes a PLBRO game (Definition 4.4 and the theorem follows as a corollary of Theorem 4.3.

Once gain, we obtain the PPAD-membership of multi-class network atomic splittable congestion game when the latency functions are piecewise-linear functions given explicitly as part of the input as a direct corollary of Theorem 5.3. The result is however more general, and also captures cases where the latency functions and their subgradients are accessed implicitly via their corresponding PL pseudo-circuits. Additionally, similarly to the statement of Theorem 5.2, we remark that the no free cycles condition is implicitly made in the case of linear latencies, and hence we obtain a strict generalization of the previous results.

Again, for completeness, we provide a simple example of a game that only has irrational equilibria, if the latency functions go beyond those captured by Theorem 5.3.

Example 5.3 (Only irrational Nash equilibria). The example is the same as the one in Example 5.2. We consider a network with a single player and two arcs e_1 and e_2 from s to t, with the same latency functions as before. Here, since we have an atomic splittable game, the single player has to minimize the latency $x \cdot l_1(x) + (1-x) \cdot l_2(1-x) = x^3 + 2 \cdot (1-x)^3$ which for $x \in [0,1]$ is minimized in $x = 2 - \sqrt{2}$.

Again, Example 5.3 establishes that the class captured by our Theorem 5.3 is tight, as if one moves to larger classes of latency functions, Nash equilibria may be irrational. Again, for similar reasons as before, a FIXP-membership for general concave latency functions follows from the results of [Filos-Ratsikas et al., 2023].

6 Competitive Equilibria in Arrow-Debreu Markets

In this section, we show how our linear-OPT-gate can be used to show the PPAD-membership of finding competitive equilibria in Arrow-Debreu markets. We will consider such markets in which the utility and production functions have a certain form which allows equilibrium points to be rational numbers. For a gentle introduction, we first apply the technique to the simpler cases of exchange markets (i.e., no production) with linear utilities (Section 6.1) and of markets with linear utilities and productions (Section 6.2), before we move to our most advanced applications.

The most general known class of utilities and productions for which rational competitive equilibria are possible are those of *Leontief-free utilities and productions*, introduced by Garg et al. [2018], who also proved the PPAD-membership of finding equilibria in those markets. We provide an alternative proof via the employment of the linear-OPT-gate, which is *conceptually* and *technically* much simpler in Section 6.3.

The Leontief-free class generalizes the well-known class of *separable piecewise linear concave (SPLC)* utilities, studied by several works, e.g., see [Vazirani and Yannakakis, 2011; Garg et al., 2015; Garg and Vazirani, 2014]. SPLC utilities is in turn a generalization of linear utilities in which every agent has a piecewise linear concave utility for the amount of a good *j* that she receives, and her total utility for her bundle is additive over goods.

SSPLC, a new class of utilities. In Section 6.4, we define a new class of utility functions, coined *succinct* separable piecewise linear concave (SSPLC) utilities, which generalize the SPLC class in a different way than the Leontief-free class. Intuitively, these are functions which can have exponentially many pieces, but those are accessed *implicitly*, via access to boolean circuits computing their slopes. In this sense, the class differs from SPLC utilities not in the definition of the utility function, but in the way that it is inputted in the computational problem. As such, the existence of rational solutions for this class already implicitly

follows from previous work [Garg and Vazirani, 2014] for SSPLC utilities. As we explain below however, the "LCP-approach" is inherently limited in its ability to establish computational results for this class, and in particular PPAD-membership. In contrast, by taking advantage of the machinery that we developed in Section 3.4, we show that our linear-OPT-gate can be used in very much the same fashion as in all the other proofs of the section, to prove that computing competitive equilibria for SSPLC utilities is in PPAD. Our proof also incorporates (explicit) SPLC production functions; we provide a discussion on the challenges of extending our technique to the case of SSPLC production functions as well in Remark 6.15.

Features of our Proof vs Previous Approaches

Before we proceed, we first discuss the previous results that appeared in a succession of important works in the area, and highlight the advantages of our approach over those.

Previous work and previous proofs. The origins of the literature that deals with the PPAD-membership of finding *exact* and *rational* competitive equilibria for markets for which this is possible can be traced back to the work of Eaves [1976]. Eaves studied the exchange market model with linear utilities (the same one we present first in Section 6.1) and proved that its competitive equilibria can be found by Lemke's algorithm [Lemke, 1965]. Referencing our discussion in Section 1.2.1, Eaves' approach, similarly to most subsequent approaches, formulates the problem as an LCP. Clearly, the class PPAD had not been defined then, but as we explained in Section 1.2.1, the PPAD-membership of the problem is implied by his proof.

Vazirani and Yannakakis [2011] considered the case of SPLC utilities. Their proof does not employ the "LCP approach", but rather the "approximation and rounding approach", see Section 1.2.1. An issue with this method is that very small changes in the prices may result in drastic changes in the optimal bundles of the consumers, and hence Vazirani and Yannakakis [2011] devise a set of technical lemmas that allow them to "force" certain allocations over others. Subsequently, Garg et al. [2015] proved the PPAD-membership of equilibria for the same class of utility functions, via instead using the "LCP approach". Garg et al. point out that the LCP that they derive does not fall into any of the classes known to avoid secondary rays, and hence they necessarily devise an argument against ray termination (see Section 1.2.1). Besides that, the formulation of the LCP itself "turns out to be quite complex" [Garg et al., 2015]. An integral part of the proof, which is also useful in our regime, is the definition of the "desire" for a good *j*, to be the total amount of the good represented by its non-zero utility segments in the SPLC function.

For markets with production, the aforementioned work of Vazirani and Yannakakis [2011] also provides a PPAD-membership result for markets with SPLC utilities and SPLC productions. The same PPAD-membership result was later recovered by [Garg and Vazirani, 2014] via the "LCP approach"; this is in fact the paper from which the quote of Section 1.2.1 is taken. The quote highlights the increasing challenge of developing these LCPs and establishing their successful termination. Indeed, the LCP of [Garg and Vazirani, 2014] naturally differs from that of [Garg et al., 2015], in that it needs to account for production. This is done via devising a set of linear programs, and then using complementary slackness and their feasibility conditions to develop the LCP needed for production. The non-homogeneity of the resulting LCP for the equilibrium problem is dealt with in a manner which is different from previous works [Eaves, 1976; Garg et al., 2015] and, naturally, since the developed LCP is different, Garg and Vazirani again need to argue against ray termination.

The last of the works in this sequence regarding competitive markets for goods is by Garg et al. [2018]. Garg et al. define a new class of utilities/productions, coined "Leontief-free utilities/productions", which generalize the SPLC functions mentioned earlier (see Section 6.3 for a precise definition). What is interesting about this class is that it contains functions which are not separable, but yet they admit rational solutions, hence dispelling a potential suspicion that non-separability is what essentially leads to irrationality. Garg et al. [2018] obtain the PPAD-membership of finding competitive equilibria in those markets via, again, employing the "LCP approach". Their LCP formulation turns out to be even more complex than those of previous works, as it has to differentiate between "normal" and "abnormal" variables, the latter preventing the employment of Lemke's algorithm. To circumvent this, they exploit some additional structure of their

nonstandard LCP, and then they also modify Lemke's algorithm, to account for the possibility of abnormal variables becoming zero. Finally, as they devise a new LCP, they also have to argue once again against ray termination.

Our proofs. We employ the linear-OPT-gate to provide proofs which are conceptually and technically simpler than the ones discussed above. In particular, we formulate the optimal consumption (i.e., the consumers' utility maximization) and the optimal production (i.e, the firms' profit maximization) as linear programs (e.g., see Figure 7), which can be effectively substituted by linear-OPT-gates in a PL arithmetic circuit. Intuitively, one could view these as the "parallel" of best responses that we used in previous sections, in the domain of markets. What differentiates the proofs of this section is that these consumptions/productions need to be supported at a set of market-clearing prices.

To make sure that we can work with linear constraints (in the inputs to PL arithmetic circuit), we apply a standard variable change which was first used by Eaves [1976], and then subsequently on all of the works that apply the "LCP approach": instead of sets of goods for consumption and production, we use the *expenditures* of the consumers to buy goods in their bundles, and of the firms to produce goods using other goods as raw material. Interestingly, Eaves [1976] attributes this idea to Gale, and hence we refer to it as *Gale's substitution*, see Remark 6.2. For the prices, we develop a feasibility program (e.g., see Figure 8), which at a high level establishes that if certain goods (or segments) have smaller expenditures, then they are sufficiently cheaper. This is the main property that we use to argue that the prices computed by the feasibility program at a fixed point are market-clearing.

For our results in Section 6.4, we take advantage of the capability of the linear-OPT-gate to compute implicit functions and correspondences, as detailed in Section 3.4. In contrast, the "LCP approach" would need to have explicit variables for each segment, resulting in an LCP of exponential size in the size of the input (which is the size of the given input circuits). This is sufficient to prove rationality of solutions, but not to construct the computational reduction needed for a PPAD-membership.

Compared to the applications in other sections of our paper, the proofs that we provide in this section are probably the most technically involved. The main complication lies in arguing the market clearing of the outputted prices, which still requires a relatively short proof. We remark that most of the space in the following sections is used for introducing the corresponding settings and putting our technique in context, rather than the proofs themselves.

Finally, we emphasize that, as we mentioned also in Section 1.2.1, the "LCP approach" satisfies some other desirable properties, e.g., see [Garg et al., 2018] for a discussion. As our focus here is the PPAD-membership of the problems, we do not discuss them further.

Roadmap. We provide a brief roadmap of the section. In Section 6.1 we introduce our technique in the simple setting of exchange markets with linear utilities. To gently incorporate production, in Section 6.2 we consider markets with linear utilities and linear productions. In Section 6.3 we consider the case of Leontief-free utilities and productions. Finally in Section 6.4 we prove the PPAD-membership of finding competitive equilibria in markets with SSPLC utilities and SPLC productions.

6.1 Exchange Markets with Linear Utilities

First, we explain how our linear-OPT-gate can be used to show the PPAD-membership of finding competitive equilibria in a simple but fundamental variant of the Arrow-Debreu market model, that of *exchange markets* with *linear utilities*. An exchange market is one in which there is no production, and each consumer brings an endowment to the market. The PPAD-membership for these markets demonstrates the strength of the technique for a simple model, before moving on to apply it to more general market settings.

Our main theorem in this section is the following:

Theorem 6.1. Computing a competitive equilibrium in an exchange market with linear utilities is in PPAD.

Exchange Markets. In an exchange market \mathcal{M} , we have a set N of consumers and a set G of infinitely divisible goods. Let n = |N| and m = |G|. We will typically use index i to refer to consumers and j or g to refer to goods. Each consumer brings an endowment $w_i = (w_{i1}, \ldots, w_{im})$ to the market, with $w_{ij} \geq 0$ for all $i \in N$ and $j \in G$. We may assume without loss of generality that for every good j, we have $\sum_{i \in N} w_{ij} = 1$, i.e., that the total endowment of each good is 1. We will use $\mathbf{x}_i = (x_{i1}, \ldots, x_{im})$ to denote the vector of quantities of goods allocated to consumer $i \in N$ in \mathcal{M} , and we will call it the *bundle* of consumer i. Let $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ be the vector of such bundles. We will use $\mathbf{p} = (p_1, \ldots, p_m)$ to denote the vector of prices in \mathcal{M} , one for each good $j \in G$. Prices are non-negative, so $p_j \geq 0$ for all $j \in G$. Given a vector of prices \mathbf{p} the *budget* of consumer $i \in N$ is defined as $\sum_{j \in G} w_{ij} p_j$; intuitively, this is the amount of money that the consumer acquires by selling her endowment at prices \mathbf{p} .

Utility Functions. Every consumer has a utility function $u_i : \mathbb{R}^m_{\geq 0} \to \mathbb{R}_{\geq 0}$ mapping a bundle \mathbf{x}_i to a non-negative real number. In this section, these utilities are *linear*, i.e., every consumer $i \in N$ has a utility u_{ij} for every good $j \in G$, and her utility for the bundle \mathbf{x}_i is $u_i(\mathbf{x}_i) = \sum_{j \in G} u_{ij} x_{ij}$, where u_{ij} is the value of consumer i for good j.

Competitive Equilibrium. Next, we provide the definition of a competitive equilibrium (often referred to as a *market* or a *Walrasian* equilibrium).

Definition 6.1 (Competitive equilibrium). A competitive equilibrium of an exchange market \mathcal{M} is a pair $(\mathbf{p}^*, \mathbf{x}^*)$ consisting of a prices \mathbf{p}^* and and bundles $\mathbf{x}^* = (\mathbf{x}_1^*, \dots, \mathbf{x}_n^*)$ such that

```
1. u_i(x_i^*) = \sum_{j \in G} x_{ij}^* u_{ij} is maximized, subject to \sum_{j \in G} x_{ij}^* p_j^* \le \sum_{j \in G} w_{ij} p_j^* and x_{ij}^* \ge 0, for any consumer i \in N. (bundle optimality)
```

2. $\sum_{i \in N} x_{ij}^* = 1$, for any good $j \in G$. (market clearing)

Condition 1 above guarantees that at the equilibrium prices \mathbf{p}^* , every consumer receives the best possible bundle that they can buy given their budget and that each good is allocated in a non-negative amount. Condition 2 guarantees that the market clears, i.e., that the total quantity of goods sold is equal to the total endowment $\sum_{i \in N} w_{ij}$, which, recall, we have assumed without loss of generality that is equal to 1.

Sufficiency conditions. A competitive equilibrium as defined above exists for every exchange market \mathcal{M} , under some *sufficiency conditions*. Following Vazirani and Yannakakis [2011], we will use the sufficiency conditions used by Eaves [1976], namely that:

- 1. For every good $j \in G$, there exists some consumer $i \in N$ that values the good positively, i.e., $u_{ij} > 0$.
- 2. For every consumer $i \in N$, there exists some good $j \in G$ that the consumer endows in a positive amount, i.e., $w_{ij} > 0$.
- 3. The market is *not reducible*. A market is reducible where there exists a proper subset $N' \subset N$ of the consumers and a subset $G' \subseteq G$ of the goods such that
 - (a) all of the goods in G' are entirely endowed by consumers in N', i.e., $\sum_{i \in N'} w_{ij} = 1$ for all $j \in G$,
 - (b) all of the consumers in N' only value positively goods in G', i.e., for all $i \in N'$ and $g \in G \setminus G'$, it holds that $u_{ig} = 0$.

Reducibility intuitively captures the fact that if a strict subset of consumers completely endow a subset of the goods and only value those goods positively, then they could form their own market \mathcal{M}' and exchange between them. Looking ahead, the non-reducibility of the market is a special case of a property called *strong connectivity of the economy graph* [Maxfield, 1997], which we will make use of for the more general market settings that we consider later on.

Optimality and bang-per-buck (BPB). The bundle optimality condition of the competitive equilibrium stipulates that each consumer buys the best possible bundle at the given prices. For linear markets, these optimal bundles have a crisp characterization, in terms of the *bang-per-buck (BPB)*. Given consumer $i \in N$ and prices \mathbf{p} , the BPB of a good $j \in G$ is defined as $\mathrm{BPB}_i(j) = \frac{u_{ij}}{p_j}$. An optimal bundle only contains non-zero quantities of goods for which the BPB is maximum, i.e., $j \in \arg\max_{j \in G} \mathrm{BPB}_i(j)$.

Bounds on the prices. We may without loss of generality assume that all the prices are strictly positive, i.e., $p_j > 0$ for all $j \in G$. Indeed, if $p_j = 0$, then there would be infinite demand for good j, contradicting market clearing. As such, in any equilibrium, the prices are strictly positive. Note that by this assumption, the quantity $BPB_i(j)$ is well-defined for every $j \in G$.

Remark 6.1 (Normalized Prices). Given that $p_j > 0$ for all $j \in G$, we can normalize the prices to sum to 1 without loss of generality, i.e., we may assume that $\sum_{j \in G} p_j = 1$.

We will use a parameter ε to capture the fact that if the price p_j for a good $j \in G$ is sufficiently smaller than the price $p_{j'}$ for a good $j' \in G$, then BPB(j) > BPB(j'). This will allow us to "control" which goods certain consumers buy in their optimal bundles. Specifically, we can compute $\varepsilon > 0$ such that

If
$$p_j \le \varepsilon \cdot p_{j'}$$
 and $u_{ij} > 0$ then $BPB_i(j) > BPB_i(j')$.

Additionally, we can pick ε to be sufficiently small such that $\varepsilon < \frac{w_{ij}}{m}$ for all $i \in N$ and all $j \in G$ with $w_{ij} > 0$. Given ε , we will impose a stricter lower bound on the prices, which will be useful later on: in particular, we will assume that for all $j \in G$, $p_j \geq \frac{\varepsilon^m}{m}$. Our PPAD-membership proof will also establish that a competitive equilibrium still always exists even under this additional restriction.

6.1.1 Preprocessing

The high-level overview of the proof will be the following. We will define a function F from a convex compact domain D to itself, in a way that ensures that a competitive equilibrium of \mathcal{M} can be recovered from a fixed point of F. The input and output variables of the function will be outputs of a linear program \mathcal{P} (see Program (2) in Section 3) and a feasibility program \mathcal{Q} (see Program (3) in Section 3). Intuitively, \mathcal{P} will be used to compute optimal bundles \mathbf{x}_i for the consumers, and \mathcal{Q} will be used to compute equilibrium prices \mathbf{p} . These programs will be of a particular form amenable to the use of our linear-OPT-gate. As such, we will be be able to encode F via a PL arithmetic circuit containing linear-OPT-gate.

Straightforward choice: x and p. A straightforward choice would be to have the pair (x, p) as the input/output to the function F. In that case, the linear program P would be the following:

maximize
$$\sum_{i \in N} \sum_{j \in M} u_{ij} x_{ij}$$
subject to
$$\sum_{j \in G} x_{ij} p_j \leq \sum_{j \in G} w_{ij} p_j, \ \forall i \in N$$
$$x_{ij} \geq 0, \ \forall i \in N, j \in G$$
 (23)

While the variables of this linear program are the variables x_{ij} , the prices p_j are gate inputs (see Definition 3.2), and the budget constraints of the form $\sum_{j \in G} x_{ij} p_j \leq \sum_{j \in G} w_{ij} p_j$ cannot be present in a linear program that we would like to compute by the linear-OPT-gate.

Change of variables: Expenditure. To circumvent this obstacle, we follow an idea used by Eaves [1976] that "linearizes" the constraints via an appropriate change of variables. Eaves [1976] attributes this idea to Gale, via private communication. Thus, we refer to this change of variables as *Gale's substitution*.

Remark 6.2 (Gale's substitution). Let $q_{ij} = x_{ij}p_j$ be the *expenditure* of consumer i on good j, i.e., how much money the consumer spends on the good given an allocation x_{ij} and a price p_j . This change of variables is by now very much standard in the literature, used among others by Eaves [1976]; Garg and Vazirani [2014]; Garg et al. [2018]; Chaudhury et al. [2022].

With this at hand, the budget constraints now become

$$\sum_{j\in G}q_{ij}\leq \sum_{j\in G}w_{ij}p_j.$$

These are clearly linear in the variables q_{ij} with no gate inputs appearing in the expression. In turn, the objective function of \mathcal{P} becomes

$$\sum_{i \in N} \sum_{j \in M} \frac{u_{ij}}{p_j} q_{ij}.$$

Interestingly, the coefficient of each variable q_{ij} is the bang-per-buck $BPB_i(j)$ of good j for consumer i. In an optimal solution to \mathcal{P} , the consumer will only spend money on goods with maximum BPB. That being said, the format of the objective function (with p_j appearing in the denominator) is not one that can be handled by the linear-OPT-gate, as its subgradient cannot be computed by a PL pseudo-circuit. For this reason, we transform the objective function to

$$\sum_{i \in N} \sum_{j \in M} \frac{p_j}{u_{ij}} q_{ij}.$$

and the maximization in linear program (23) to minimization. Clearly, in an optimal solution, the consumer is again only spending on goods with minimum ratio $\frac{p_j}{u_{ij}}$, i.e., with maximum BPB, therefore purchasing an optimal bundle. This objective function form is of a form amenable to the use of the linear-OPT-gate, as its subgradient is a linear function.

However, there is a potential issue that arises when converting the maximization objective to minimization. If we keep the budget constraint $\sum_{j\in G}q_{ij}\leq \sum_{j\in G}w_{ij}p_j$ intact, then setting $q_{ij}=0$ for all $i\in N$ and $j\in G$ is an optimal solution to the linear program, while it is clearly not an expenditure consistent with a competitive equilibrium. To handle this, we observe that in an optimal solution to linear program (23), where the objective is maximization, we may assume that all the budget constraints have zero slack, i.e., that $\sum_{j\in G}x_{ij}^*p_j^*=\sum_{j\in G}w_{ij}p_j^*$. Indeed, if one could increase some variable x_{ij} without violating the budget constraint for some consumer $i\in N$, then the consumer would certainly not receive a lower utility in the objective function. Correspondingly, the constraints $\sum_{j\in G}q_{ij}\leq \sum_{j\in G}w_{ij}p_j$ can be converted to equality constraints, i.e., $\sum_{j\in G}q_{ij}=\sum_{j\in G}w_{ij}p_j$. In turn, these constrain the expenditure variables to receive positive values in an optimal solution, ensuring that each consumer spends their entire budget.

Summarizing, the linear program \mathcal{P} can be seen on the left-hand side of Figure 6.

Excess expenditure and the program \mathcal{Q} . While linear program \mathcal{P} will be used to obtain the optimal expenditures q_{ij} (and as consequence the optimal bundles x_{ij}), as we mentioned earlier, the equilibrium prices will be obtained via a feasibility program \mathcal{Q} . Before we define the program, we introduce the notion of excess expenditure, which will be useful later on. The *excess expenditure* e_j of a good $j \in G$ is defined as the difference between the total expenditure of all consumers $i \in N$ for that good and the price of the good, i.e.,

$$e_j = \sum_{i \in N} q_{ij} - p_j.$$

Then, at equilibrium prices \mathbf{p}^* we will have market clearing, i.e., $e_i^* = 0$ for all $j \in G$.

We are now ready to define our feasibility program Q; see the right-hand side of Figure 6. The equilibrium prices will be obtained as the output \mathbf{p}^* of Q.

$$\begin{array}{c|c} \underline{\text{Linear Program }\mathcal{P}} & \underline{\text{Feasibility Program }\mathcal{Q}} \\ \\ \text{minimize } \sum_{i \in N} \sum_{j \in G} \frac{p_j}{u_{ij}} q_{ij} & e_j < e_{j'} \Rightarrow p_j \leq \varepsilon \cdot p_{j'}, \ \forall j, j' \in G \\ \\ \text{subject to } \sum_{j \in G} q_{ij} = \sum_{j \in G} w_{ij} p_j & p_j \leq \varepsilon \cdot p_{j'}, \ \forall j \in G \\ \\ q_{ij} \geq 0, \ \forall j \in G & \sum_{j \in G} p_j = 1 \end{array}$$

Figure 6: The linear program \mathcal{P} used to recover optimal bundles (left) and the feasibility program \mathcal{Q} used to find market-clearing prices (right).

6.1.2 Membership in PPAD: The proof of Theorem 6.1.

We will develop the proof in three steps, namely (a) construction of the function *F* and arguing that it can be represented by a PL arithmetic circuit containing linear-OPT-gates, (b) showing that the linear-OPT-gate can compute all the necessary components, and (c) arguing that a competitive equilibrium can be recovered from a fixed point of *F*.

The function *F***.** Given the above, we will define $F: D \rightarrow D$ with domain

$$D = \left\{ p \in \Delta^{m-1} : \forall j \in G, p_j \ge \frac{\varepsilon^m}{m} \right\} \times [0, 1]^{nm}.$$

An input to F is a pair $(\bar{\mathbf{p}}, \bar{\mathbf{q}})$ of prices and expenditures, where $\bar{\mathbf{q}} = (\bar{q}_{ij})_{i \in N, j \in G}$ and the output is another such pair $(\mathbf{p}^*, \mathbf{q}^*)$. The domain D is the one above since $\sum_{j \in G} \bar{p}_j = 1$ (see Remark 6.1), and by the fact that in any feasible set of expenditures, $\sum_{j \in G} \bar{q}_{ij} \leq \sum_{j \in G} w_{ij} \bar{p}_j \leq 1$.

Remark 6.3. Note that while the linear program \mathcal{P} and the feasibility program \mathcal{Q} were written in a general form in Figure 6, when using $(\bar{\mathbf{p}}, \bar{\mathbf{q}})$ as the input, we would have to substitute p_j with \bar{p}_j in \mathcal{P} (with the variables being q_{ij}) and q_{ij} and p_{ij} with \bar{q}_{ij} and \bar{p}_j respectively in the expression for e_j in \mathcal{Q} (with the variables being p_j). While we do not write this out explicitly in all of the applications in our paper, here we thought it would be important to mention, in order to avoid any confusion with p_j potentially being on the left-hand side of the conditional constraints in \mathcal{Q} . It is \bar{p}_j that appears on the left-hand side, which is a gate input rather than a variable, and thus is in accordance with the definition of feasibility programs in (3) of Section 3.

Computation by the linear-OPT-gate. In the next lemma, we argue formally what we have alluded to earlier, namely that the solutions to the linear program \mathcal{P} and the feasibility program \mathcal{Q} of Figure 6 can be computed by our linear-OPT-gate.

Lemma 6.1. Consider the linear program \mathcal{P} and the feasibility program \mathcal{Q} of Figure 6. An optimal solution to \mathcal{P} can be computed by the linear-OPT-gate. \mathcal{Q} is solvable, and a solution can be computed by the linear-OPT-gate.

Proof. For the linear program \mathcal{P} , we observe that the feasible domain $[0,1]^n$ is non-empty and bounded. The gate inputs p_j appear only on the right-hand side of the constraints, and the subgradient of the objective function is linear, and hence can be given by a PL pseudo-circuit. For the feasibility program \mathcal{Q} , there are no variables appearing in the left-hand side of the conditional constraints (see also Remark 6.3), and no gate inputs appearing on the right-hand side. This means that the linear-OPT-gate correctly computes the outcome of the program, assuming that the program is solvable. To argue solvability, notice that \mathcal{Q} is of the form \mathcal{Q}_{app} as defined in Section 3.2, and hence by Lemma 3.2 it suffices to argue that its feasibility graph $G_{\mathcal{Q}}$ is acyclic. In our case here, the vertices $V_{\mathcal{Q}}$ of $G_{\mathcal{Q}}$ are the goods, and an edge (j,j') exists if and only if $e_j < e_{j'}$. Clearly, $G_{\mathcal{Q}}$ is acyclic.

Arguing optimality and market clearing. To conclude the proof, what is left to show is that a fixed point $(\mathbf{p}^*, \mathbf{q}^*)$ of F indeed corresponds to a competitive equilibrium. We argue that in the following lemma.

Lemma 6.2. Let $(\mathbf{p}^*, \mathbf{q}^*)$ be a fixed point of F. Then the pair $(\mathbf{p}^*, \mathbf{x}^*)$, where $x_{ij}^* = q_{ij}^*/p_j^*$ is a competitive equilibrium of \mathcal{M} .

Proof. By the form of the objective function of the linear program \mathcal{P} of Figure 6, each consumer spends money only on goods with maximum BPB and therefore the allocation that she receives is only one of goods for which she has maximum BPB. Therefore, the consumer receives an optimal bundle which satisfies Condition 1 of the competitive equilibrium definition in Definition 6.1. The allocation quantities x_{ij}^* can straightforwardly be recovered from the values of q_{ij}^* .

What remains to show is that \mathbf{p}^* is a vector of market-clearing prices. By the definition of e_j , this is equivalent to arguing that for all $j \in G$, we have that $e_j^* = 0$. We first argue that $\sum_{j \in G} e_j^* = 0$. Indeed:

$$\sum_{j \in G} e_j^* = \sum_{j \in G} \left(\sum_{i \in N} q_{ij}^* - p_j^* \right) = \sum_{j \in G} \sum_{i \in N} q_{ij}^* - \sum_{j \in G} p_j^* = \sum_{j \in G} \sum_{i \in N} w_{ij} p_j^* - 1$$

$$= \sum_{j \in G} p_j^* \cdot \sum_{i \in N} w_{ij} - 1 = \sum_{j \in G} p_j^* - 1 = 0,$$

where in the calculation above we used:

- in Equations 3 and 6, that $\sum_{j \in G} p_j^* = 1$, which is without loss of generality (see Remark 6.1),
- in Equation 5, that $\sum_{i \in N} w_{ij} = 1$, which is without loss of generality, and
- in Equation 3, that $\sum_{j \in G} \sum_{i \in N} q_{ij}^* = \sum_{j \in G} \sum_{i \in N} w_{ij} p_j^*$, which follows from the constraints of linear program \mathcal{P} of Figure 6.

Define the set *J* to be the set of goods with minimum excess expenditure, i.e.,

$$J = \{j \in G : e_j^* \le e_{j'}^*, \text{ for all } j' \in G\}.$$

It now suffices to show that $e_j^* \ge 0$ for all $j \in J$. Assume by contradiction that there exists some $j_1 \in J$ such that $e_{j_1}^* < 0$. By definition of the set J, we have that

for all
$$j \in J$$
, we have $e_j^* < 0$. (24)

Also, by the fact that $\sum_{j\in G} e_j^* = 0$ which we established above, it must be the case that there also exists $j_2 \in G \setminus J$ such that $e_{j_2}^* > 0$. In particular $G \setminus J \neq \emptyset$. Define

$$N' = \{i \in N : \text{ there exists } j \in G \setminus J \text{ such that } w_{ij} > 0\}.$$

In words, N' is the set of consumers with strictly positive endowment for some good which is not in the set J. We will argue that

there exists
$$i_0 \in N'$$
 and $j \in J$ such that $u_{i_0 j} > 0$.

To see this, assume by contradiction that for all $i \in N'$ and $j \in J$, it was the case that $u_{ij} = 0$. Note also that for any good $j' \in G \setminus J$, it holds that $w_{ij'} > 0$ only for consumers $i \in N'$. That would imply that N' and $G \setminus J$ would constitute a reduced market, contradicting Feasibility Condition 3.

Consider any good $j \in J$ and any good $j' \in G \setminus J$ such that $w_{i_0j'>0}$. By the definition of J, we have that $e_j^* < e_{j'}^*$. By the fact that \mathbf{p}^* is a solution to the feasibility program $\mathcal Q$ of Figure 6, we have that $p_j^* \le \varepsilon \cdot p_{j'}^*$. We have the following inequalities:

$$\sum_{g \in G} w_{i_0 g} \cdot p_g^* \ge w_{i_0 j'} \cdot p_{j'}^* \quad \text{and} \quad p_{j'}^* \ge \frac{m \cdot p_j^*}{w_{i_0 j'}}.$$
 (25)

The first inequality above is trivial, since $j' \in G$. The second inequality follows since by the choice of ε ,

$$p_{j'}^* \ge \frac{p_j^*}{\varepsilon} \ge \frac{p_j^*}{w_{i_0j'}/m} = \frac{m \cdot p_j^*}{w_{i_0j'}}.$$

Since j was chosen to be any good in J, the inequalities in (25) hold for j such that p_j^* is the maximum price amongst all goods in J. Therefore in (25) we can substitute p_j^* by $\max_{g \in J} p_g^*$ and we have

$$\sum_{g \in G} w_{i_0 g} \cdot p_g^* \ge w_{i_0 j'} \cdot p_{j'}^* \ge m \cdot \max_{g \in J} p_g^* \ge \sum_{g \in J} p_g^*. \tag{26}$$

Inequality 26 implies that consumer i_0 has enough budget to buy all the goods in J. At the same time, by the choice of ε , consumer i_0 prefers to buy quantities of good j rather than j', i.e., $BPB_{i_0}(j) > BPB_{i_0}(j')$. Since consumer i_0 only spends money on goods with maximum BPB, we have that

$$\sum_{g \in G} q_{i_0 g}^* = \sum_{g \in I} q_{i_0 g}^*. \tag{27}$$

From Inequality 26, Equation 27, and since \mathbf{q}^* is a feasible solution to linear program \mathcal{P} of Figure 6 (and hence satisfies the constraints $\sum_{g \in G} q_{i_0g}^* = \sum_{g \in G} w_{i_0g} \cdot p_g^*$), we have that

$$\sum_{g \in J} q_{i_0 g}^* \ge \sum_{g \in J} p_g^*.$$

This last inequality however implies that there must exist $j_0 \in J$ for which $q_{i_0j_0}^* \geq p_{j_0}^*$ and therefore $e_{j_0}^* = \sum_{i \in N} q_{ij_0}^* - p_{j_0}^* \geq 0$, contradicting (24). This completes the proof.

6.2 Arrow-Debreu Markets with Linear Utilities and Productions

We now move on to the next fundamental variant of the Arrow-Debreu market model, that of markets with *linear utilities* as well as *linear productions*. Our results about exchange markets in Section 6.1 were used to demonstrate the application of our technique to one of the simplest variants of the main market model. The results of this section can be seen as extending this exposition to the case where the markets also have linear production functions. This is rather informative for the reader, as it illustrates the approach that will be used for the most general market setting that we prove PPAD-membership for, which we present in Section 6.3.

Our main theorem in this section is the following:

Theorem 6.2. Computing a competitive equilibrium in an Arrow-Debreu market with linear utilities and linear productions is in PPAD.

We provide the main definitions for Arrow-Debreu markets with linear utilities and linear productions below. To ensure that the section is self-contained, we have elected to fully define the setting, rather than to only highlight the changes to the definitions of exchange markets of Section 6.1. Still, to avoid unnecessary repetition, in the preprocessing and setup steps for the proof, we do refer to the appropriate arguments presented in Section 6.1.

Markets with Production. In an Arrow-Debreu market with production \mathcal{M} , we have a set N of consumers, a set G of infinitely divisible goods, and a set F of firms. Let n=|N|, m=|G|, and $\ell=|F|$. We will typically use index i to refer to consumers, j or g to refer to goods and f to refer to firms. Each consumer brings an endowment $w_i=(w_{i1},\ldots,w_{im})$ to the market, with $w_{ij}\geq 0$ for all $i\in N$ and $j\in G$. We may assume without loss of generality that for every good j, we have $\sum_{i\in N}w_{ij}=1$, i.e., that the total endowment of each good is 1. We will use $\mathbf{x}_i=(x_{i1},\ldots,x_{im})$ to denote the vector of quantities of goods allocated to

consumer $i \in N$ in \mathcal{M} , and we will call it the *bundle* of consumer i. Let $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_n)$ be the vector of such bundles. We will use $\mathbf{p} = (p_1, \dots, p_m)$ to denote the vector of *prices* in \mathcal{M} , one for each good $j \in G$. Prices are non-negative, so $p_j \geq 0$ for all $j \in G$. Given a vector of prices \mathbf{p} , the *budget* of consumer $i \in N$ is defined as $\sum_{j \in G} w_{ij} p_j$; intuitively, this is the amount of money that the consumer acquires by selling her endowment at prices \mathbf{p} .

Utility Functions. Every consumer has a utility function $u_i : \mathbb{R}^m_{\geq 0} \to \mathbb{R}_{\geq 0}$ mapping a bundle \mathbf{x}_i to a non-negative real number. In this section, these utilities are *linear*, i.e., every consumer $i \in N$ has a utility u_{ij} for every good $j \in G$, and her utility for the bundle \mathbf{x}_i is $u_i(\mathbf{x}_i) = \sum_{j \in G} u_{ij} x_{ij}$, where u_{ij} is the value of consumer i for good j.

Firm Shares. Each consumer $i \in N$ has a share $\theta_{if} \in [0,1]$ of the profit of each firm $f \in F$. We assume that the profits are entirely shared among the consumers, i.e., for every firm $f \in F$, we have that $\sum_{i \in N} \theta_{if} = 1$.

Production Functions and Conversion Rates. Each firm $f \in F$ produces a set of goods using a set of goods as raw material. Following Garg and Vazirani [2014], for simplicity we will assume without loss of generality that each firm produces a single good. We may also assume without loss of generality that each firm uses a single good as raw material for the production. This because the production functions that we will consider are separable, and hence a firm using/producing multiple goods can be split into multiple firms using/producing single goods, with the shares of the agents' being duplicated (see also [Garg and Vazirani, 2014]). Given the above we will let:

- g_f^{out} be the good produced by firm $f \in F$; we will refer to this as the *output good*,
- g_f^{in} be the good used as raw material by firm $f \in F$; we will refer to this as the *input good*.

For every firm $f \in F$ there is a function P_f which determines the firm's ability to produce units of the output good g_f^{out} as a function of quantities of the input good g_f^{in} . In this section, we will assume that these functions are linear, of the form $P_f(y) = c_f \cdot y$, where $c_f \geq 0$ is a fixed *conversion rate* for firm f, specifying that the firm can use g units of good g_f^{in} to produce g_f^{in} to produce g_f^{out} and g_f^{out} for the goods g_f^{out} and g_f^{in} respectively, the *profit* of g_f^{out} is defined as $g_f^{\text{out}} \cdot g_f^{\text{out}} \cdot g_f^{\text{out$

Competitive Equilibrium. We are now ready to define the notion of a competitive equilibrium in markets with production.

Definition 6.2 (Competitive Equilibrium - Markets with Production). A competitive equilibrium of an Arrow-Debreu market with linear utilities and linear production functions is a triple $(\mathbf{p}^*, \mathbf{x}^*, \mathbf{y}^*)$ consisting of non-negative prices \mathbf{p}^* , non-negative bundles $\mathbf{x}^* = (\mathbf{x}_1^*, \dots, \mathbf{x}_n^*)$ and non-negative amounts of input goods $\mathbf{y}^* = (y_1^*, \dots, y_\ell^*)$, such that

- 1. $p_{g_f^{\text{out}}}^* \cdot c_f \cdot y_f^* p_{g_f^{\text{in}}}^* \cdot y_f^*$ is maximized, for any firm $f \in F$. (firm profit maximization)
- 2. $u_i(\mathbf{x}_i)$ is maximized for every consumer $i \in N$, subject to $\sum_{j \in G} p_j^* \cdot x_{ij} \leq \sum_{j \in G} p_j^* \cdot w_{ij} + \sum_{f \in F} \theta_{if} \cdot (p_{g_f^{\text{out}}}^* \cdot c_f \cdot y_f^* p_{g_f^{\text{in}}}^* \cdot y_f^*).$ (bundle optimality)
- 3. $z_{j}^{*} \leq 0$, and $z_{j}^{*}p_{j}^{*} = 0$, where $z_{j}^{*} = \sum_{i \in N} x_{ij}^{*} + \sum_{f \in F: g_{f}^{\text{in}} = j} y_{f}^{*} \sum_{f \in F: g_{f}^{\text{out}} = j} c_{f} \cdot y_{f}^{*} 1$, for every good $j \in G$. (market clearing)

Condition 1 requires that at the chosen set of prices \mathbf{p}^* , each firm maximizes its profit, given its production functions. Condition 2 requires that at the chosen set of prices \mathbf{p}^* , each consumer maximizes her utility subject to their budget constraints, where the budget consists of the amount earned from selling all the consumer's endowments $\sum_{j \in G} p_j^* w_{ij}$ and the profit share $\sum_{f \in F} \theta_{if} \cdot (p_{g_f^{\text{out}}}^* \cdot c_f \cdot y_f^* - p_{g_f^{\text{in}}}^* \cdot y_f^*)$ of the consumer from the production of the firms. Finally, Condition 3 is the market clearing condition, which requires that the total consumption of each good is at most the total production plus the total endowment of the consumers, and supply equals demand for all goods which are not priced at 0. As we detail later on in the section, we may in fact assume without loss of generality that in any competitive equilibrium *all the prices are positive*, and hence Condition 3 reduces to $z_j^* = 0$ for all $j \in G$. Note that in Condition 3 we have used that $\sum_{\in N} w_{ij} = 1$ for each good $j \in G$.

Sufficiency Conditions. A competitive equilibrium as defined above exists for every market \mathcal{M} , under some sufficiency conditions. The weakest known such conditions are the ones provided by Maxfield [1997] (see also [Garg and Vazirani, 2014]), which generalize the sufficiency conditions of Eaves [1976] that we used in Section 6.1. We use the following conditions, which are very close to that of Maxfield [1997].

- 1. For every good $j \in G$, there exists some consumer $i \in N$ that values the good positively, i.e., $u_{ij} > 0$.
- 2. For every consumer $i \in N$, there exists some good $j \in G$ that the consumer endows in a positive amount, i.e., $w_{ij} > 0$.
- 3. Consider a graph $\mathcal{G}_F(\mathcal{M})$ in which the nodes are the goods, and an edge (j,j') has weight

$$\alpha_{jj'} = \max_{f \in F: \ g_f^{\text{out}} = j', g_f^{\text{in}} = j} c_f,$$

i.e., j' can be produced from j at conversion rate $\alpha_{jj'}$ by some firm $f \in F$. Then, for any cycle $C = (g_0, g_1), (g_1, g_2), \dots, (g_{k-2}, g_{k-1})$ of $G_F(\mathcal{M})$ the product of the weights of the edges is less than 1, i.e., $\prod_{e \in C} \alpha_e < 1$.

This condition is known as the *no production out of nothing and no vacuous production* condition. Indeed, if $\prod_{e \in C} \alpha_e > 1$, then it would be possible to increase the quantity of some good, without decreasing the quantity of any other good. The case of $\prod_{e \in C} \alpha_e = 1$ refers to the case of vacuous production, which is also disallowed in our model.¹⁵

- 4. Consider the *economy graph* $\mathcal{G}_E(\mathcal{M})$ of the market \mathcal{M} in which the nodes are the consumers and the firms, and
 - there is an edge (i, i') between consumer/firm-node i and consumer-node i' if i endows/produces a good j for which $u_{i'j} > 0$,
 - there is an edge (i, f) between consumer/firm-node i and firm-node f, if i endows/produces the raw good g_f^{in} that f uses for production.

Then, $G_E(\mathcal{M})$ contains a strongly connected component containing all the consumer-nodes. This condition generalizes that of the market non-reducibility condition that we used in Section 6.1.

Remark 6.4 (Bounds on Production). Condition 3 above imposes a bound on the total amount of a production of any firm $f \in F$ in a competitive equilibrium. Note that the production starts from finite endowments $\sum_{i \in N} w_{ij} = 1$ for all $j \in G$. Since no firm operates at a loss at an equilibrium, any cycle of production would violate Condition Item 3. Since there are no such cycles, production can take place along chains. The longest such chain is obviously bounded by m, and the maximum production of any

¹⁵Maxfield [1997] technically allows for vacuous production. Vacuous production is disallowed in the conditions imposed in the original setting of Arrow and Debreu [1954] and is also disallowed by Garg and Vazirani [2014].

firm can be bounded by some sufficiently large global constant $L^c = (\text{e.g.})$, some constant such that $L^c \ge m^m (\max_f c_f + 1)^n$. See also [Garg and Vazirani, 2014] for a very similar argument. Looking ahead, this will allow us to impose "loose" upper bounds on the production and consumption in the linear programs that we will devise, without compromising the existence of a competitive equilibrium.

Optimality and bang-per-buck (BPB). Similarly to Section 6.1, the optimal bundles of Condition 2 in Definition 6.2 are characterized by their *bang-per-buck (BPB)*. Given consumer $i \in N$ and prices \mathbf{p} , the BPB of a good $j \in G$ is defined as $BPB_i(j) = \frac{u_{ij}}{p_j}$. An optimal bundle only contains non-zero quantities of goods for which the BPB is maximum, i.e., $j \in \arg\max_{i \in G} BPB_i(j)$.

Bounds on the prices. Again, similarly to Section 6.1, we may assume without loss of generality that all the prices are strictly positive, i.e., $p_j > 0$ for all goods $j \in G$. Indeed, not that for any good j, there is some consumer i with $u_{ij} > 0$. Then its price p_j cannot be 0 in any equilibrium. If j is not being produced, then the demand of consumer i cannot be satisfied. If its being produced, it has to be produced using an input good of 0 price as well, for the production to be profitable. This in fact implies that it has to be produced along a chain of goods with 0 price, contradicting the market clearing condition. Very similar arguments have been made in the related literature, e.g., see [Garg and Vazirani, 2014; Garg et al., 2018]. Note that by these assumptions, the quantity BPB $_i(j)$ is well defined for every $j \in G$.

Remark 6.5 (Normalized Prices). Given that $p_j > 0$ for all $j \in G$, we can normalize the prices to sum to 1 without loss of generality, i.e., we may assume that for every good $j \in G$, we have that $\sum_{j \in G} p_j = 1$.

Again, in a similar manner to Section 6.1, we will use a parameter ε to capture the fact that if the price p_j for a good $j \in G$ is sufficiently smaller than the price $p_{j'}$ for a good $j' \in G$, then BPB(j) > BPB(j'). Specifically, we can compute $\varepsilon > 0$ such that

If
$$p_j \le \varepsilon \cdot p_{j'}$$
 and $u_{ij} > 0$ then $BPB_i(j) > BPB_i(j')$.

Additionally, we can pick ε to be sufficiently small such that $\varepsilon < \frac{w_{ij}}{m}$ for all $i \in N$ and all $j \in G$. Given ε , we will impose a stricter lower bound on the prices, which will be useful later on: in particular, we will assume that for all $j \in G$, $p_j \ge \frac{\varepsilon^m}{m}$.

6.2.1 Preprocessing

The high-level approach will again be to define a function F from a convex compact domain D to itself, and show that a competitive equilibrium of M can be recovered from a fixed point of F. We will need to show that F can be computed by a PL arithmetic circuit and hence, similarly to Section 6.1, we cannot work directly with the allocations \mathbf{x} and the quantities of input goods \mathbf{y} as in Definition 6.2, as those would need to be multiplied with the the prices \mathbf{p} , which will also be inputs to the circuit. To circumvent this, we will again apply a change of variables very similar to Gale's substitution (see Remark 6.2) and work with the expenditure variables as we did in Section 6.1. In more detail, we will need such variables for the expenditure of the consumers as before, but we will also need variables for the expenditure and the revenue of the firms in production. The same variable change was used by Garg and Vazirani [2014].

We will then make use of our linear-OPT-gate to obtain the equilibrium quantities as outcomes of linear programs and a feasibility program. In particular, linear programs \mathcal{P}_1 and \mathcal{P}_1 will be used to compute the optimal consumer and firm expenditures respectively, and from those the optimal bundles \mathbf{x} and optimal quantities of input goods \mathbf{y} will be recovered. The prices will be the outcome of a feasibility program \mathcal{Q} .

Consumer expenditure. For the consumers, we will use the standard change of variables that we used in Section 6.1 for exchange markets. Namely, we will let $q_{ij} = x_{ij}p_j$ be the *expenditure* of consumer i on good j, i.e., how much money the consumer spends on the good given an allocation x_{ij} and a price p_i .

Figure 7: The linear programs \mathcal{P}_1 and \mathcal{P}_2 used to recover optimal bundles (left), and optimal production (right) respectively.

Firm expenditure. For the firms, we will use a similar substitution for the case of production. Namely, we will let

- $s_f = p_{g_f^{\text{in}}} \cdot y_f$ be the *expenditure* of firm f on its input good g_f^{in} , i.e., the amount of money spend on y_f units of the input good.
- $r_f = p_{g_f^{\text{out}}} \cdot c_f \cdot y_f$ be the *revenue* of firm f from its output good g_f^{out} , i.e., the amount of money earned from $c_f \cdot y_f$ units of the output good.¹⁶

By definition, the profit of firm $f \in F$ is then exactly $r_f - s_f$.

The program \mathcal{P}_1 for optimal consumer expenditure. We can now write the linear program \mathcal{P}_1 , the solution of which will give us the optimal expenditures q_{ij}^* for the consumers; see the left-hand side of Figure 7. It now becomes evident why we need to apply the aforementioned substitution: without it the quantity $\sum_{f \in F} \theta_{if} \cdot (p_{g_f^{\text{out}}} \cdot c_f \cdot y_f - p_{g_f^{\text{in}}} \cdot y_f)$ would appear in the constraints of the linear program \mathcal{P}_1 . The constants $p_{g_f^{\text{in}}}$, $p_{g_f^{\text{out}}}$ and y_f are gate inputs for the linear program, and such a constraint cannot be handled by our linear-OPT-gate. With the gate inputs r_f and s_f instead, the constraint becomes linear and thus respects the conditions of the linear-OPT-gate. Note that the linear program \mathcal{P}_1 is parameterized by two constants L and C. Intuitively, $C \cdot L + 1$ is an upper bound on the amount of money that consumer i can spend on good j. Recall that there is a global upper bound L^c on the total possible production, and as a result there are global bounds on the total possible expenditure for both production and consumption. We will set L and C to be sufficiently large to not constrain these global upper bound; the precise bounds are established in the proofs of Claim 6.1 and Claim 6.2.

Similarly to Section 6.1, in an optimal solution, the consumer is again only spending on goods with minimum ratio $\frac{p_j}{u_{ij}}$, i.e., with maximum BPB, therefore purchasing an optimal bundle. At the same time, the subgradient of this objective function can be computed by a PL pseudo-circuit.

The program \mathcal{P}_2 for optimal firm expenditure. We are now ready to define the linear program for each firm $f \in F$, the optimal solution of which will be the optimal expenditure s_f^* of firm f on the amount y_f^* of the input good that the firm uses for production; see the right-hand side of Figure 7. The linear program \mathcal{P}_2 is also parameterized by the constant L, which we mentioned above; intuitively, L imposes an upper bound on the quantity of the input good that firm f can use, and will be large enough to not constrain the global upper bound L^c on production.

¹⁶We remark that Garg and Vazirani [2014] used r_f to refer to the expenditure (spending) and s_f to refer to the revenue. We have switched those so that the variables names are more indicative of the quantities that they represent.

Remark 6.6 (\mathcal{P}_2 ensures optimal production). At first glance, the objective function of linear program \mathcal{P}_2 might seem a bit unintuitive: why are we multiplying the expenditure with the profit? How do we guarantee that in the competitive equilibrium, each firm maximizes its profit? To provide some intuition, we remark that linear program \mathcal{P}_2 could equivalently be substituted by the following feasibility program $\mathcal{Q}(\mathcal{P}_2)$:

$$\begin{split} & \frac{\text{Feasibility Program } \mathcal{Q}(\mathcal{P}_2)}{c_f \cdot p_{g_f^{\text{out}}} - p_{g_f^{\text{in}}} > 0 \Rightarrow s_f = L \cdot p_{g_f^{\text{in}}}}{c_f \cdot p_{g_f^{\text{out}}} - p_{g_f^{\text{in}}} < 0 \Rightarrow s_f = 0} \\ & 0 \leq s_f \leq L \cdot p_{g_f^{\text{in}}} \end{split}$$

Indeed, linear program \mathcal{P}_2 has a very simple form, and hence in an optimal solution

- the firm spends as much as possible on the input good (i.e., s_f is maximum), when it is profitable to produce, (i.e., when $c_f \cdot p_{g_f^{\text{out}}} p_{g_f^{\text{in}}} > 0$),
- the firm spends as little as possible on the input good (i.e., s_f is 0), when it is not profitable to produce, (i.e., when $c_f \cdot p_{g_f^{\text{out}}} p_{g_f^{\text{in}}} < 0$),
- the firm spends any feasible amount when producing or not producing yield the same profit (i.e., when $c_f \cdot p_{g_f^{\text{out}}} p_{g_f^{\text{in}}} = 0$).

We could have in fact used $Q(\mathcal{P}_2)$ rather than \mathcal{P}_2 , as it is a valid feasibility program that can be solved using the linear-OPT-gate. However, we elected to go the the linear program \mathcal{P}_2 instead, as it sets up the machinery that we will use in Section 6.3, where the corresponding linear program will be more involved and cannot simply be substituted by a feasibility program as above.

We remark that in a competitive equilibrium, we will establish that $c_f \cdot p_{g_f^{\text{out}}} - p_{g_f^{\text{in}}} \leq 0$, and hence the firm will never be required to use exactly L units of g_f^{in} for production.

Remark 6.7 (Calculating r_f from s_f). Looking ahead, the terms r_f which appear in the constraints of linear program \mathcal{P}_1 (and also in the feasibility program \mathcal{Q} , see below), will not be inputs to the function F that we will construct to compute a competitive equilibrium. These will need to be recovered from the values of s_f , or more precisely, the terms themselves will need to be substituted with expressions of the term s_f . This can be done as follows:

$$r_f = s_f + \max\{0, c_f \cdot p_{g_f^{\text{out}}} - p_{g_f^{\text{in}}}\} \cdot L$$
 (28)

As we will only be concerned with the fixed point behavior of our function F, this equation needs to correctly capture the revenue of firm f at a competitive equilibrium. Indeed, given our discussion above, we have that:

- When $c_f \cdot p_{g_f^{\text{out}}} p_{g_f^{\text{in}}} < 0$, the firm does not spend any amount on producing and as a result it does not produce anything. In that case $r_f^* = s_f^* = 0$.
- When $c_f \cdot p_{g_f^{\text{out}}} p_{g_f^{\text{in}}} > 0$, the firm spends an amount of $s_f^* = L \cdot p_{g_f^{\text{in}}}$ for L units of the input good and obtains a revenue of $r_f^* = L \cdot c_f \cdot p_{g_f^{\text{out}}}$.
- When $c_f \cdot p_{g_f^{\text{out}}} p_{g_f^{\text{in}}} = 0$, the firm can spend any amount s_f^* for production, and we have $r_f^* = s_f^*$ by definition.

In each case, Equation (28) correctly computes the value of r_f^* from s_f^* at a competitive equilibrium.

Feasibility Program
$$\mathcal{Q}$$

$$e_{j} < e_{j'} \Rightarrow p_{j} \leq \varepsilon \cdot p_{j'}, \ \forall j, j' \in G$$

$$p_{j} \geq \frac{\varepsilon^{m}}{m}, \ \forall j \in G$$

$$\sum_{j \in G} p_{j} = 1$$

Figure 8: The feasibility program Q used to find market-clearing prices.

Excess expenditure and the program \mathcal{Q} . Similarly to Section 6.1, the equilibrium prices \mathbf{p}^* will be obtained via a feasibility program \mathcal{Q} . Before we define the program, we introduce the notion of excess expenditure; the definition is very similar to the one we used in Section 6.1, except that it now takes into account the expenditure/revenue due to production. The *excess expenditure* e_j of a good $j \in G$ is defined as the difference between the total expenditure of all consumers $i \in N$ and firms $f \in F$ for that good and the price of the good plus the total revenue of firms $f \in F$ from that good, i.e.,

$$e_j = \sum_{i \in N} q_{ij} + \sum_{f \in F: \ g_f^{\text{in}} = j} s_f - p_j - \sum_{f \in F: \ g_f^{\text{out}} = j} r_f.$$

Then, at equilibrium prices p_i^* we will have market clearing, i.e., $e_i^* = 0$.

We are now ready to define our feasibility program Q; see Figure 8. The equilibrium prices will be obtained as the output \mathbf{p}^* of Q.

6.2.2 Membership in PPAD: The proof of Theorem 6.2.

We will again develop the proof in three steps, namely (a) construction of the function F and arguing that it can be represented by a PL arithmetic circuit containing linear-OPT-gates, (b) showing that the linear-OPT-gate can compute all the necessary components, and (c) arguing that a competitive equilibrium can be recovered from a fixed point of F.

The function *F***.** Given the above, we will define $F: D \to D$ with domain

$$D = \left\{ p \in \Delta^{m-1} : \forall j \in G, p_j \ge \frac{\varepsilon^m}{m} \right\} \times [0, C \cdot L + 1]^{nm} \times [0, L]^{\ell}.$$

An input to F is a triple $(\bar{\mathbf{p}}, \bar{\mathbf{q}}, \bar{\mathbf{s}})$ of prices, consumer expenditures and firm expenditures, where $\bar{\mathbf{q}} = (\bar{q}_{ij})_{i \in N, j \in G}$ and $\bar{\mathbf{s}} = (\bar{s}_1, \dots \bar{s}_\ell)$, and the output is another such triple $(\mathbf{p}^*, \mathbf{q}^*, \mathbf{s}^*)$. The domain D is the one above since $\sum_{j \in G} \bar{p}_j = 1$ (see Remark 6.5), and by the upper bounds of $C \cdot L + 1$ and L set on the consumer expenditures in linear program \mathcal{P}_1 and the firm expenditures in linear program \mathcal{P}_2 respectively.

Computation by the linear-OPT-gate. Next, we argue that the solutions to the linear programs \mathcal{P}_1 and \mathcal{P}_2 and the feasibility program \mathcal{Q} of Figure 6 can be computed by our linear-OPT-gate.

Lemma 6.3. Consider the linear programs \mathcal{P}_1 , \mathcal{P}_2 of Figure 7, and the feasibility program \mathcal{Q} of Figure 8. An optimal solution to \mathcal{P}_1 and \mathcal{P}_2 can be computed by the linear-OPT-gate. \mathcal{Q} is solvable, and a solution can be computed by the linear-OPT-gate.

Proof. For the linear program \mathcal{P}_1 , the feasible domain $[0,1]^n$ is non-empty and bounded. The gate inputs p_j and s_f appear only on the right-hand side of the constraints, and the subgradient of the objective function is linear, and hence can be given by a PL pseudo-circuit. For the linear program \mathcal{P}_2 , the feasible domain [0,L]

is non-empty and bounded, and the gate inputs $p_{g_f^{\text{in}}}$ only appear on the right-hand side of the constraints. The subgradient of the objective function is linear, and hence can be given by a PL pseudo-circuit. For the feasibility program \mathcal{Q} , the arguments that it is of the correct form and that it is solvable are identical to those of Lemma 6.1, noting the updated definition of the expenditure e_i .

Arguing optimality and market clearing. To conclude the proof, what is left to show is that a fixed point $(\mathbf{p}^*, \mathbf{q}^*, \mathbf{s}^*)$ of F indeed corresponds to a competitive equilibrium. We argue that in the following lemma.

Lemma 6.4. Let $(\mathbf{p}^*, \mathbf{q}^*, \mathbf{s}^*)$ be a fixed point of F. Then the triple $(\mathbf{p}^*, \mathbf{x}^*, \mathbf{y}^*)$, where $x_{ij}^* = q_{ij}^*/p_j^*$ and $y_f^* = s_f^*/p_{g_f^{in}}^*$ is a competitive equilibrium of \mathcal{M} .

Proof. By the form of the objective function of the linear program \mathcal{P}_1 of Figure 7, each consumer spends money only on goods with maximum BPB and therefore the allocation that she receives is only one of goods for which she has maximum BPB. Therefore, the consumer receives an optimal bundle which satisfies Condition 2 of definition in Definition 6.2. The allocation quantities x_{ij}^* can straightforwardly be recovered from the values of q_{ij}^* . As we discussed earlier in Remark 6.6, an optimal solution to linear program \mathcal{P}_2 also results in the optimal expenditure for the firms.

What remains to show is that \mathbf{p}^* is a vector of market-clearing prices. By the definition of e_j , this is equivalent to arguing that for all $j \in G$, we have that $e_i^* = 0$. We first argue that $\sum_{i \in G} e_i^* = 0$. Indeed:

$$\begin{split} \sum_{j \in G} e_j^* &= \sum_{j \in G} \left(\sum_{i \in N} q_{ij}^* - p_j^* + \sum_{f \in F: \ g_f^{\text{in}} = j} s_f^* - \sum_{f \in F: \ g_f^{\text{out}} = j} r_f^* \right) \\ &= \sum_{i \in N} \sum_{j \in G} q_{ij}^* - \sum_{j \in G} p_j^* + \sum_{f \in F} (s_f^* - r_f^*) \\ &= \sum_{j \in G} \sum_{i \in N} w_{ij} p_j^* - 1 + \sum_{i \in N} \sum_{f \in F} \theta_{if} \cdot (r_f^* - s_f^*) + \sum_{f \in F} (s_f^* - r_f^*) \\ &= \sum_{j \in G} p_j^* \cdot \sum_{i \in N} w_{ij} - 1 + \sum_{f \in F} (r_f^* - s_f^*) \cdot \sum_{i \in N} \theta_{if} \cdot \sum_{f \in F} (s_f^* - r_f^*) \\ &= \sum_{i \in N} w_{ij} - 1 + \sum_{f \in F} (r_f^* - s_f^*) + \sum_{f \in F} (s_f^* - r_f^*) \\ &= 0 \end{split}$$

where in the calculations above we used:

- in Equation 2, that $\sum_{j \in G} \sum_{f \in F: g_f^{\text{in}} = j} s_f^* = \sum_{f \in F} s_j^*$ and that $\sum_{j \in G} \sum_{f \in F: g_f^{\text{out}} = j} r_f^* = \sum_{f \in F} r_{f'}^*$
- in Equation 3, that $\sum_{i \in N} \sum_{j \in G} q_{ij}^* = \sum_{i \in N} \sum_{j \in G} w_{ij} p_j^* + \sum_{i \in N} \sum_{f \in F} \theta_{if} (r_f^* s_f^*)$, which follows from the constraints of linear program \mathcal{P}_2 of Figure 7,
- in Equations 3 and 5, that $\sum_{i \in G} p_i^* = 1$, which is without loss of generality, see Remark 6.5,
- in Equation 5, that $\sum_{i \in N} \theta_{if} = 1$ for every $f \in F$, by definition, and
- in Equation 6, that $\sum_{i \in N} w_{ij} = 1$, which is without loss of generality.

From the above, it suffices to prove that $e_j^* \ge 0$, for all $j \in G$. Assume by contradiction that there exists some $j_1 \in G$ such that $e_{j_1}^* < 0$; we will obtain a contradiction to the strong connectivity of the economy graph of the market. We define the following three sets:

- $J = \{j \in G : e_j^* \le e_{j'}^*$, for all $j' \in G\}$. In other words, J is the set of goods with minimum excess expenditure.

- $N_J = \{i \in N : \text{ there exists } j \in J \text{ such that } u_{ij} > 0\}$. In other words, N_J contains the set of consumers that value at least one item in J positively.
- $F_J = \{ f \in F : g_f^{in} \in J \}$. In other words, F_J contains the set of firms for which the input good is in J.

Since $e_{j_1}^* < 0$, by the definition of J we know that

for all
$$j \in J$$
, we have $e_j^* < 0$. (29)

Also, since $e_{j_1}^* < 0$, and $\sum_{j \in G} e_j^* = 0$, there must exist some other good j_2 such that $e_{j_2}^* > 0$. In particular, that implies that J is a strict subset of G, i.e., $J \subseteq G$. We state and prove the following claim.

Claim 6.1. For all $i \in N_I$ and all $j' \in G \setminus J$, it holds that $w_{ij'} = 0$.

Proof. Assume by contradiction that there exists some consumer $i_0 \in N_J$ and some good $j' \in G \setminus J$ such that $w_{i_0j'} > 0$. We will show that this implies that there exists some $j_\ell \in J$ with $e_{j_\ell}^* \ge 0$, contradicting Statement (29). Since \mathbf{p}^* is a solution to feasibility program \mathcal{Q} of Figure 8, we have that $p_j^* \le \varepsilon \cdot p_{j'}^*$, for every $j \in J$. Hence, by choosing ε to be sufficiently small, we may lower-bound the budget of consumer i by

$$w_{i_0j'} \cdot p_{j'}^* \ge \frac{w_{i_0j'}}{\varepsilon \cdot m} \cdot \sum_{j \in J} p_j^* \ge (C \cdot L + 1) \sum_{j \in J} p_j^*$$

Since $i_0 \in N_J$, by the choice of ε , there is at least one good $g \in J$ such that $BPB_{i_0}(g) > BPB_{i_0}(g')$, for all $g' \in G \setminus J$, i.e., the consumer prefers to buy quantities of good g rather than any good which is not in the set J. Let $j \in J$ be a good with maximum BPB for consumer i_0 . Since $q_{i_0j}^*$ is an optimal solution to linear program \mathcal{P}_1 of Figure 7, it should be that $q_{i_0j}^* = C \cdot L + 1$, which implies that $q_{i_0j}^* \geq (C \cdot L + 1)p_j^*$. By substituting that into the definition of the excess expenditure e_i^* , we obtain:

$$\begin{split} e_{j}^{*} &= \sum_{i \in N} q_{ij}^{*} + \sum_{f \in F: \ g_{f}^{\text{in}} = j} s_{f}^{*} - p_{j}^{*} - \sum_{f \in F: \ g_{f}^{\text{out}} = j} r_{f}^{*} \\ &\geq q_{i_{0}j}^{*} + \sum_{f \in F: \ g_{f}^{\text{in}} = j} s_{f}^{*} - p_{j}^{*} - \sum_{f \in F: \ g_{f}^{\text{out}} = j} r_{f}^{*} \\ &\geq (C \cdot L) p_{j}^{*} - \sum_{f \in F: \ g_{f}^{\text{out}} = j} r_{f}^{*} \end{split}$$

Therefore, it suffices to show that $\sum_{f \in F: g_f^{\text{out}} = j} r_f^* \leq (C \cdot L) p_j^*$; in that case we will have $e_j^* \geq 0$ and will obtain a contradiction, by setting $j = j_\ell$. Now consider any firm $f \in F$. Referencing also Remark 6.6, note that

- if $c_f \cdot p_{g_f^{\text{out}}}^* < p_{g_f^{\text{in}}}^*$, then the firm does not produce anything, and we have $r_f^* = 0$,
- if $c_f \cdot p_{g_f^{\text{out}}}^* > p_{g_f^{\text{in}}}^*$, the firm produces L units of g_f^{out} and we have $r_f^* = L \cdot c_f \cdot p_{g_f^{\text{out}}}$,
- if $c_f \cdot p_{g_f^{\text{out}}}^* > p_{g_f^{\text{in}}}^*$, the firm may produce any amount and we have $r_f^* = s_f^* \leq L \cdot c_f \cdot p_{g_f^{\text{out}}}$, since s_f^* is a feasible solution to linear program \mathcal{P}_2 of Figure 7.

By letting $C \ge |F| \cdot \max_{f \in F} c_f$, it follows that $\sum_{f \in F: \ g_f^{\text{out}} = j} r_f^* \le (C \cdot L) p_j^*$ and we obtain a contradiction. \square

Claim 6.1 establishes that a consumer in the set N_J cannot endow any good that is not in the set J in a positive quantity. Since $J \subsetneq G$, this implies that $N_J \subsetneq N$, since every good is positively endowed by some consumer $i \in N$. Now consider any consumer $i \in N_J$ and any consumer $i' \in N \setminus N_J$ or any firm $f \in F \setminus F_J$.

Claim 6.1 implies that there cannot be an edge (i,i') or an edge (i,f) in the economy graph of the market. Indeed, for (i,i'), notice that by the claim consumer i only endows positively goods that are in J, but consumer i' has value 0 for them by virtue of being in $N \setminus N_J$. Similarly, for (i,f), f uses good $g_f^{\text{in}} \in G \setminus J$ by definition, but consumer i only positively endows goods in J.

Next, we state and prove the following claim.

Claim 6.2. For any $f \in F_J$, it holds that $g_f^{out} \in J$.

Proof. Assume by contradiction that there exists some firm $f_0 \in F_J$ with $g_{f_0}^{\text{out}} \notin J$. For convenience, let $j = g_{f_0}^{\text{out}}$ be that good, and let $j_0 = g_{f_0}^{\text{in}}$ be the firm's input good. Note that by definition of F_J , we have that $j_0 \in J$. By Statement (29), we have that $e_{j_0}^* < 0$. Since \mathbf{p}^* is a solution to feasibility program Q of Figure 8, we have that $p_{j_0}^* \le \varepsilon \cdot p_j^*$. The value of ε can be chosen sufficiently small for this to imply that $c_{f_0} \cdot p_j^* - p_{j_0}^* > 0$. Since this is the multiplier of the variable s_{f_0} in the objective function of linear program \mathcal{P}_2 , the optimality of $s_{f_0}^*$ as a solution to the linear program implies that $s_{f_0}^* = L \cdot p_{j_0}^*$. Using this we may bound the expenditure $e_{j_0}^*$ as follows:

$$\begin{aligned} e_{j_0}^* & \geq L \cdot p_{j_0}^* - \sum_{f \in F: \ g_f^{\text{out}} = j_0} r_f^* - p_{j_0}^* \\ & = (L - 1) p_{j_0}^* - \sum_{f \in F: \ g_f^{\text{out}} = j_0} r_f^* \end{aligned}$$

where in the calculations above we used the obvious fact that $\sum_{f \in F: g_f^{\text{in}} = j} s_f^* \ge s_{f_0}$.

Obviously, if good j_0 is not being produced by any firm $f \in F$, we can simply set $L \ge 1$ and obtain that $e_{j_0}^* \ge 0$, a contradiction, since $r_f^* = 0$ for all firms that have j_0 as their output good. Now consider the sequence of goods j_0, j_1, \ldots, j_k , where good $j_{\ell-1}$ is being produced (in positive quantity) by some firm f_ℓ using j_ℓ as the input good. Note that by the discussion above, we may assume that $k \ge 1$. Additionally, by the "no production out of nothing and no vacuous production" feasibility condition of the market \mathcal{M} , it also holds that $k \le m$, as otherwise the graph $G_F(\mathcal{M})$ would have a cycle in which the product of the weight of the edges would be at least 1.

Finally, we remark that for each $\ell \in \{0, 1, ..., k\}$, we have that $j_{\ell} \in J$. To see this, observe that from the conditional constraints of feasibility program \mathcal{Q} of Figure 8, the price for any good in J is at least ε times smaller than the price of any good in $G \setminus J$. By taking ε to be small enough, we can guarantee that if a firm f is producing a good $g_f^{\text{out}} \in J$, then it must use a good $g_f^{\text{in}} \in J$ as input, otherwise s_f^* would not be an optimal expenditure. Since $j_0 \in J$ by the definition of F_J , it must hold that $j_\ell \in J$ for all $\ell \in \{0, 1, ..., k\}$.

We will argue by induction that for $1 \le \ell' \le k$, we have:

$$r_{f_{\ell}}^* \ge \left(\frac{L-1}{|F|^{\ell} \cdot \prod_{i=1}^{\ell-1} c_{f_i}} - \sum_{i=1}^{\ell-1} \frac{1}{|F|^{i} \cdot \prod_{j=\ell-i+1}^{\ell-1} c_{f_j}}\right) \cdot p_{j_{\ell-1}}^* \tag{30}$$

With this at hand, at step k we will have

$$e_{j_k}^* \geq \left(\frac{L-1}{c_{f_k} \cdot |F|^k \cdot \prod_{i=1}^{k-1} c_{f_i}} - \sum_{i=1}^{k-1} \frac{1}{c_{f_k} \cdot |F|^i \cdot \prod_{j=k-i+1}^{k-1} c_{f_j}} - 1\right) p_{j_k}^* - \sum_{f \in F: \ g_f^{\text{out}} = j_k} r_f^*$$

Since k is the last index in the sequence, i.e, good j_{k-1} is the last to be produced, we know that $r_f^* = 0$ for any firm f that has j_k as its output good. From the above, we have that

$$e_{j_k}^* \ge \left(\frac{L-1}{c_{f_k} \cdot |F|^k \cdot \prod_{i=1}^{k-1} c_{f_i}} - \sum_{i=1}^{k-1} \frac{1}{c_{f_k} \cdot |F|^i \cdot \prod_{j=k-i+1}^{k-1} c_{f_j}} - 1\right) p_{j_k}^*$$

By picking L to be sufficiently large, e.g., large enough for the following to hold

$$\frac{L}{(|F| \cdot \max_f c_f)^m} - \frac{m}{(\min_f c_f)^m} \ge 0,$$

we obtain a contradiction. We compute the proof below by proving Inequality (30).

Base Case: Let $\ell=1$, and consider firm f_1 that produces good j_0 from good j_1 . It follows that in the objective function of linear program \mathcal{P}_2 of Figure 7 corresponding to firm f_1 , the multiplier of the expenditure s_{f_1} is non-negative, i.e., $c_{f_1} \cdot p_{j_0}^* - p_{j_1^*} \geq 0$. There are two cases:

- If
$$c_{f_1} \cdot p_{j_0}^* - p_{j_1^*} > 0$$
, then $s_{f_1}^* = L \cdot p_{j_1}^*$.

- If
$$c_{f_1} \cdot p_{j_0}^* - p_{j_1^*} = 0$$
, then $s_{f_1}^* = r_{f_1}^* \geq \frac{L-1}{|F|} \cdot p_{j_0}^*$

where the last inequality follows from the fact that $e_{j_0} < 0$, by Statement (29). In either case, in the second case using the fact that $p_{j_0}^* \ge p_{j_1}^*/c_{f_1}$, we can bound the expenditure $e_{j_1}^*$ as follows:

$$0 > e_{j_1}^* \ge \left(\frac{L-1}{|F| \cdot c_{f_1}} - 1\right) \cdot p_{j_1}^* - \sum_{f \in F: \ g_f^{\text{out}} = j_1} r_f^*$$

From this, we get that

$$r_{f_2}^* \ge \left(\frac{L-1}{|F|^2 \cdot c_{f_1}} - \frac{1}{|F|}\right) \cdot p_{j_1}^*$$

Induction Step: Consider some step ℓ and consider firm f_ℓ that produces good $j_{\ell-1}$ from good j_ℓ . It follows that in the objective function of linear program \mathcal{P}_2 of Figure 7 corresponding to firm f_ℓ , the multiplier of the expenditure s_{f_ℓ} is non-negative, i.e., $c_{f_\ell} \cdot p_{j_{\ell-1}}^* - p_{j_\ell^*} \geq 0$. There are two cases:

- If
$$c_{f_\ell}\cdot p_{j_{\ell-1}}^*-p_{j_\ell^*}>0$$
, then $s_{f_\ell}^*=L\cdot p_{j_\ell}^*$.

- If
$$c_{f_{\ell}} \cdot p_{j_{\ell-1}}^* - p_{j_{\ell}^*} = 0$$
, then

$$s_{f_{\ell}}^* = r_{f_{\ell}}^* \ge \left(\frac{L-1}{|F|^{\ell} \cdot \prod_{i=1}^{\ell-1} c_{f_i}} - \sum_{i=1}^{\ell-1} \frac{1}{|F|^{i} \cdot \prod_{j=\ell-i+1}^{\ell-1} c_{f_j}}\right) \cdot p_{j_{\ell-1}}^*$$

where the inequality in the second case follows from the induction hypothesis. In either case, in the second case using the fact that $p_{j_{\ell-1}}^* \geq p_{j_{\ell}}^*/c_{f_{\ell}}$, we can bound the expenditure $e_{j_{\ell}}^*$ as follows:

$$0 > e_{j_{\ell}}^* \ge \left(\frac{L-1}{|F|^{\ell} \cdot \prod_{i=1}^{\ell} c_{f_i}} - \sum_{i=1}^{\ell-1} \frac{1}{|F|^{i} \cdot \prod_{j=\ell-i+1}^{\ell} c_{f_j}} - 1\right) \cdot p_{j_{\ell}}^* - \sum_{f \in F: \ g_f^{\text{out}} = j_1} r_f^*$$

From this, we obtain that

$$r_{f_{\ell+1}}^* \geq \left(\frac{L-1}{|F|^{\ell+1} \cdot \prod_{i=1}^{\ell} c_{f_i}} - \sum_{i=1}^{\ell} \frac{1}{|F|^i \cdot \prod_{j=\ell-i+1}^{\ell} c_{f_i}}\right) \cdot p_{j_{\ell}}^*$$

Claim 6.2 implies that for any firm $f \in F_J$, there cannot be an edge to any consumer $i \in N \setminus N_J$ or any firm $f' \in F \setminus F_J$ in the economy graph. Indeed, for an edge (f,i), the claim asserts that the production good g_f^{out} of f will be in the set J, whereas consumer i only values goods in the set $G \setminus J$ positively, by virtue of being in $N \setminus N_J$. Similarly, for an edge (f, f'), firm f produces the good $g_f^{\text{out}} \in J$, whereas the input good g_f^{in} of firm f' is in $G \setminus J$, by virtue of f' being in $F \setminus F_J$.

From the two paragraphs succeeding Claim 6.1 and Claim 6.2, N_J and $N \setminus N_J$ are two strongly connected components in the economy graph of the market, contradicting the sufficiency condition requiring that in the economy graph there is a strongly connected component containing all the consumer-nodes. This completes the proof.

6.3 Markets with Leontief-free Utilities and Productions

In this section we show how our linear-OPT-gate can be used to obtain the PPAD-membership of finding competitive equilibria in Arrow-Debreu markets with Leontief-free utilities and productions. Recall that, as we mentioned in the beginning of the section, this is the largest class of functions for which membership in PPAD (and hence rationality of solutions) has been proven [Garg et al., 2018]. In the following section we define another class of functions which generalizes all previous ones, but its generality is incomparable to the Leontief-free case. Our main theorem of the section is the following.

Theorem 6.3. Computing a competitive equilibrium of an Arrow-Debreu market with Leontief-free utilities and productions is in PPAD.

Our proof in this section provides a significant simplification over that of Garg et al. [2018]. To keep the section as self-contained as possible, we define these markets in detail here, rather than explain how they generalize the markets of Section 6.2, but we make appropriate references to that section.

Markets with Production. In an Arrow-Debreu market with production \mathcal{M} , we have a set N of consumers, a set G of infinitely divisible goods, and a set F of firms. Let n = |N|, m = |G|, and $\ell = |F|$. We will typically use index i to refer to consumers, j or g to refer to goods and f to refer to firms. Each consumer brings an endowment $w_i = (w_{i1}, \ldots, w_{im})$ to the market, with $w_{ij} \geq 0$ for all $i \in N$ and $j \in G$. We may assume without loss of generality that for every good j, we have $\sum_{i \in N} w_{ij} = 1$, i.e., that the total endowment of each good is 1. We will use $\mathbf{x}^i = (x_{i1}, \ldots, x_{im})$ to denote the vector of quantities of goods allocated to consumer $i \in N$ in \mathcal{M} , and we will call it the *bundle* of consumer i. Let $\mathbf{x} = (\mathbf{x}^1, \ldots, \mathbf{x}^n)$ be the vector of such bundles. We will use $\mathbf{p} = (p_1, \ldots, p_m)$ to denote the vector of *prices* in \mathcal{M} , one for each good $j \in G$. Prices are non-negative, so $p_j \geq 0$ for all $j \in G$. Given a vector of prices \mathbf{p} , the *budget* of consumer $i \in N$ is defined as $\sum_{j \in G} w_{ij} p_j$; intuitively, this is the amount of money that the consumer acquires by selling her endowment at prices \mathbf{p} .

Utility Functions. Every consumer has a utility function u_i : $\mathbb{R}^m_{\geq 0} \to \mathbb{R}_{\geq 0}$ mapping a bundle \mathbf{x}^i to a nonnegative real number. In this section, we consider *Leontief-free utilities*. Such utility functions are specified by a finite list K_i of *tranches* $\{s_k^i\}_{k\in K_i}$. Associated with every tranche s_k^i is a number $L_k^i\in\mathbb{R}_{\geq 0}\cup\{\infty\}$, which is an upper bound for the total utility that can be accrued from this tranche, and for every $j\in G$ there is a number $u_{ik}^i\geq 0$, which is the rate at which good j provides utility for consumer i on tranche k. That is, the

¹⁷Garg et al. [2018] us the term "segments" to refer to these "portions" of the utility function. We use the term "segments" for the parts of the SSPLC functions in Section 6.4 which has a different meaning, so we adopt the term "tranche" here instead.

utility consumer i receives from the bundle x^i is calculated by solving the following linear program.

$$\begin{aligned} \text{maximize } & \sum_{k \in K_i} \sum_{j \in G} u^i_{jk} x^i_{jk} \\ \text{subject to } & \sum_{j \in G} u^i_{jk} x^i_{jk} \leq L^i_k, \quad \text{for all } k \in K_i \\ & \sum_{k \in K_i} x^i_{jk} \leq x^i_j, \quad \text{for all } j \in \mathcal{G} \end{aligned}$$

Note that the class of linear utilities that we considered in Sections 6.1 and 6.2 is a special case of this construction where every tranche is unbounded and can accrue utility from only a single good. In the case of SPLC utilities which we mentioned in the beginning of Section 6, the restriction that the tranches must be unbounded is dropped.

Firm Shares. Each consumer $i \in N$ has a share $\theta_{if} \in [0,1]$ of the profit of each firm $f \in F$. We assume that the profits are entirely shared among the consumers, i.e., for every firm $f \in F$, we have that $\sum_{i \in N} \theta_{if} = 1$.

Production functions. Every firm f has a *Leontief-free production function*. Here we will use slightly different terminology from Section 6.2, following the one used by Garg et al. [2018]. We will say that each firm f has a *partitioning* of the set of goods G into *raw goods* and *produced goods*, i.e., $G = \mathcal{R}_f \sqcup \mathcal{P}_f$. The firm then first converts the raw goods into *raw units*, and these in turn are converted into produced goods.

More precisely, the production function of firm f is specified by a list of tranches $\{s_k^f\}_{k \in K_f}$. To each of these tranches we have the following associated quantities:

- a number $L_k^f \in \mathbb{R}_{\geq 0} \cup \{\infty\}$, which is a bound on how many raw units can be produced on this tranche, for every raw good $j \in \mathcal{R}_{f,k} \subseteq \mathcal{R}_f$,
- a number $\alpha_{jk}^f > 0$, which denotes the rate at which good j can be converted into raw units on this tranche (i.e., the raw-good-to-raw-unit conversion rate), and,
- for every produced good $j' \in \mathcal{P}_{f,k} \subseteq \mathcal{P}_f$, a number $\beta_{j'k}^f > 0$, which denotes the rate at which raw units can be converted into good j' on this tranche (i.e,, the raw-unit-to-produced-good conversion rate).

Let $\mathbf{y}^f = (y_j)_{j \in G}$ be the *production vector* of firm f, i.e., the vector of amounts of raw goods and produced goods involved in its production function.

Remark 6.8 (One tranche for each firm). Note that whether a firm produces on a tranche k is independent of whether it produces on another tranche k'. Therefore, to simplify the exposition, we may assume that the production function of any firm has only one tranche. If a firm's production function has multiple tranche, then we may simply "split" the firm into one firm for every tranche of the function, and allocate to the consumers shares in the new firms that are identical to the shares they had in the original firm. This change will not impact the production of the firms or the budget constraints of the consumers (see Definition 6.4 below). Given this, we can write α_j^f and $\beta_{j'}^f$ for the conversion rate of firm f from quantities of the raw good f to raw units, and for the conversion rate of the firm from raw units to quantities of the produced good f', respectively. Similarly, we can also write f for the bound on the number of raw units that the firm can produce on its tranche.

Next, we define the optimal production for firms and the optimal consumption for consumers. We start from the former.

Optimal Production. Consider firm f, and let p_j and p'_j be the prices of goods j and j' in the market respectively. Using some raw good $j \in \mathcal{R}_f$, the firm can produce 1 raw unit on its tranche at a *cost* of p_j/α_j^f . For any produced good $j' \in \mathcal{P}_f$, the firm can use 1 raw unit to produce $\beta_{j'}^f$ units of good j', resulting in a *revenue* of $p_{j'}\beta_{j'k}^f$. With this in mind, we define the *cost-per-unit (cpu)* and *revenue-per-unit (rpu)* of firm f as follows:

$$cpu^{f}(\mathbf{p}) = \min_{j \in \mathcal{R}_f} \frac{p_j}{\alpha_j^f} \quad \text{and} \quad rpu^{f}(\mathbf{p}) = \max_{j' \in \mathcal{P}_f} p_{j'} \beta_{j'}^f$$

Intuitively, the cpu of a firm is the minimum cost that it incurs from converting any of its raw goods into raw units, and the rpu is the maximum revenue that it obtains by converting raw units into any of its produced goods.

The *profit-per-unit* (ppu) of firm f is then defined as

$$ppu^f(\mathbf{p}) = rpu^f(\mathbf{p}) - cpu^f(\mathbf{p})$$

We can now define a firm's optimal production.

Definition 6.3 (Optimal Production). Given the production vector \mathbf{y}^f of firm f, a production for the firm is *optimal* if the following conditions are satisfied:

1.
$$\sum_{j' \in \mathcal{P}_f} y_{i'}^f / \beta_{j'}^f \le \sum_{j \in \mathcal{R}_f} \alpha_j^f y_j^f \le L^f$$
. (feasibility)

- 2. If $ppu^f(\mathbf{p}) < 0$, then $\mathbf{y}^f = \mathbf{0}$. (zero unprofitable production)
- 3. If $ppu^f(\mathbf{p}) > 0$, then $\sum_{j \in \mathcal{R}_f} \alpha_j^f y_j^f = L^f$. (maximum profitable production of raw units)
- 4. If $\operatorname{rpu}^f(\mathbf{p}) > 0$, then $\sum_{j' \in \mathcal{P}_f} y_{j'}^f / \beta_{j'}^f = \sum_{j \in \mathcal{R}_f} \alpha_j^f y_j^f$. (maximum usage of raw units)
- 5. For every $j \in \mathcal{R}_f$, if $y_i^f > 0$, then $p_j / \alpha_i^f = \text{cpu}^f(\mathbf{p})$. (only use cost-optimal raw goods)
- 6. For every $j' \in \mathcal{P}_f$, if $y_{i'}^f > 0$, then $p_{j'}\beta_{i'}^f = \operatorname{rpu}^f(\mathbf{p})$. (only produce revenue-optimal produced goods)

Before we proceed, with offer the following remark with respect to Condition Item 3 above.

Remark 6.9 (Maximum Profitable Production). In Definition 6.3, when $L^f = \infty$, Condition 3 stipulates that the firm should use an infite amount of raw goods for production. As we explained in the previous section (see Remark 6.4) and as we we will reiterate in the context of the markets of this section in Remark 6.10 below, in a competitive equilibrium (see Definition 6.4) there is a finite, global upper bound to how much a firm can produce. In turn, this also imposes a global upper bound on how much a consumer can consume. Looking ahead, in the linear programs that we will construct, we will use some upper bounds on production and consumption, which will be sufficiently large to not constrain the aforementioned global upper bounds. For the case of production, that bound will be represented by L, and then, Item 3 should be interpreted as

If
$$ppu^f(\mathbf{p}) > 0$$
, then $\sum_{j \in \mathcal{R}_f} \alpha_j^f y_j^f = \min\{L^f, L\}$,

where L will be guaranteed to be large enough such that $\min\{L^f, L\} = L$ only when $L^f = \infty$.

Optimal Consumption and bang-per-buck (BPB). For the consumers, the optimal consumption amounts to them utilizing tranches in order of decreasing bang-per-buck. Given consumer $i \in N$ and prices \mathbf{p} , the BPB of a pair (j,k) consisting of a good $j \in G$ and a tranche $k \in K_i$ is defined as $BPB_i(j,k) = u^i_{jk}/p_j$. In particular, $BPB_i(j,k)$ has a maximum BPB over pairs $(j,k) \in G \times K_i$, then consumer i will buy quantities of good j, and "allocate" the good to tranche k until the upper bound L^i_k on the utility that the consumer gain from this tranche has been met. This process is repeated until the budget of the consumer is exhausted.

Competitive Equilibrium. We are now ready to define the notion of a competitive equilibrium in markets with Leontief-free utilities and productions.

Definition 6.4 (Competitive Equilibrium - Markets with Leontief-free Utilites and Productions). A competitive equilibrium of an Arrow-Debreu market with Leontief-free utilities and Leontief-free productions is a tuple $(\bar{\mathbf{p}}, \bar{\mathbf{x}}, \bar{\mathbf{y}})$ consisting of non-negative prices $\bar{\mathbf{p}}$, non-negative bundles $\bar{\mathbf{x}}^i$ for $i \in \mathcal{N}$, and non-negative amounts of goods $\bar{y}^f \in \mathbb{R}_{>0}^{|\mathcal{R}_f \cup \mathcal{P}_f|}$ for $f \in F$ satisfying the following conditions:

- 1. For every firm $f \in F$, \bar{y}^f is an optimal production vector for f. (firm profit maximization)
- 2. For every consumer $i \in \mathcal{N}$, $\bar{\mathbf{x}}^i$ is an optimal consumption vector for i under the budget constraint $\sum_{j \in G} \sum_{k \in K_i} \bar{p}_j \cdot \bar{x}^i_{jk} \leq \sum_{j \in G} w^i_j \bar{p}_j + \sum_{f \in F} \theta^i_f \left(\sum_{j \in \mathcal{P}_f} \bar{p}_j \cdot \bar{y}^f_j \sum_{j \in \mathcal{R}_f} \bar{p}_j \cdot \bar{y}^f_j \right)$, (bundle optimality)
- 3. $\bar{z}_j \leq 0$, and $\bar{z}_j \cdot \bar{p}_j$, where for every good $j \in \mathcal{G}$, $\bar{z}_j = \sum_{i \in N} \sum_{k \in K_i} \bar{x}^i_{jk} + \sum_{f \in F: j \in \mathcal{R}_f} \bar{y}^f_j \sum_{f \in F: j \in \mathcal{P}_f} \bar{y}^f_j 1$ (market clearing)

Condition 1 requires that at the chosen set of prices $\bar{\mathbf{p}}$, each firm maximizes its profit, given its production functions. Condition 2 requires that at the chosen set of prices $\bar{\mathbf{p}}$, each consumer maximizes her utility subject to their budget constraints, where the budget consists of the amount earned from selling all the consumer's endowments and the profit share of the consumer from the production of the firms on the different tranches. Finally, Condition 3 is the market clearing condition, which requires that the total consumption of each good is at most the total production plus the total endowment of the consumers, and supply equals demand for all goods which are not priced at 0. Similarly to Section 6.2, and as we explain later, we may in fact assume without loss of generality that in any competitive equilibrium *all the prices are positive*, and hence Condition 3 reduces to $\bar{z}_j = 0$ for all $j \in G$. Note that in Condition 3 we have used that $\sum_{i \in N} w_{ij} = 1$ for each good $i \in G$.

Sufficiency Conditions. A competitive equilibrium as defined above exists for every market \mathcal{M} , under some sufficiency conditions. Similarly to Section 6.2, we will use the set of conditions used by Maxfield [1997], also used in the series of papers on market equilibria that we mentioned in the beginning of Section 6.

- 1. For every consumer $i \in N$, there exists some good $j \in G$ that the consumer endows in a positive amount, i.e., $w_{ij} > 0$.
- 2. We will say that a consumer $i \in N$ is *nonsatiated* for $good j \in G$ if there exists some tranche k such that $u^i_{jk} > 0$ and $L^i_k = \infty$. Following Garg et al. [2018], we will assume that for any good $j \in G$, there exists some consumer $i \in N$ that is nonsatiated with respect to j. Similarly, a firm is *nonsatiated* for $good j \in G$ if $j \in \mathcal{R}_f$ and $L^f = \infty$.

We remark that this condition naturally generalizes the condition that "for any good there exists some consumer with positive utility for that good", which we used in Sections 6.1 and 6.2.

3. Consider a graph $\mathcal{G}_F(\mathcal{M})$ in which the nodes are the goods, and an edge (j,j') has weight

$$w_{jj'} = \max_{f \in F: j \in \mathcal{R}_f \land j' \in \mathcal{P}_f} \alpha_j^f \cdot b_j'^f$$
,

If the set is empty, the weight is defined to be 0.

The above weight captures the fact that j' can be produced from j, via the intermediate raw units, at combined conversion rate $w_{jj'}$ by some firm $f \in F$. Then, for any cycle $C = (g_0, g_1), (g_1, g_2), \dots, (g_{k-2}, g_{k-1})$ of $G_F(\mathcal{M})$ the product of the weights of the edges is less than 1, i.e., $\prod_{e \in C} \alpha_e < 1$.

This condition is known as the *no production out of nothing and no vacuous production* condition. Indeed, if $\prod_{e \in C} \alpha_e > 1$, then it would be possible to increase the quantity of some good, without decreasing the quantity of any other good. The case of $\prod_{e \in C} \alpha_e = 1$ refers to the case of vacuous production, which is also disallowed in our model, similarly to the related works. e.g., see Garg and Vazirani [2014].

- 4. Consider the *economy graph* $\mathcal{G}_E(\mathcal{M})$ of the market \mathcal{M} in which the nodes are the consumers and the firms, and
 - there is an edge (i, i') between consumer-node i and consumer/firm-node i' if i endows a good j for which i' is nonsatiated.
 - there is an edge (f,i) between firm-node f and consumer/firm-node i, if $L^f = \infty$ and there exists some good $j \in \mathcal{P}_f$ for which i is nonsatiated.

Then, $G_E(\mathcal{M})$ contains a strongly connected component containing all the consumer-nodes. This condition generalizes the corresponding strong connectivity condition that we used in Section 6.2.

Remark 6.10 (Bounds on Production). Very similarly to Section 6.2, we remark that there are some inherent bounds on how much a firm can produce on a tranche, even if for that segment we have $L^f = \infty$. The idea here is very similar to that of Remark 6.4; given that we start from a finite set of endowed goods, the absence of cycles of profitable production in an equilibrium restricts the production to take place in chains of length bounded by m. This implies a global upper bound L^c on how many raw goods can be used or produced. Again, looking ahead, this will allow us to impose "loose" upper bounds on the production and consumption in the linear programs that we will devise, without compromising the existence of a competitive equilibrium.

Bounds on the prices. Again, similarly to Section 6.2, we may assume without loss of generality that all the prices are strictly positive, i.e., $p_j > 0$ for all goods $j \in G$. The argument to achieve this is very similar to the one used in Section 6.2, and has also been established in [Garg et al., 2018]. Note that by this assumption, the quantity BPB_i(j, k) is well-defined for every $j \in G$ and $k \in K_i$.

Remark 6.11 (Normalized Prices). Given that $p_j > 0$ for all $j \in G$, we can normalize the prices to sum to 1 without loss of generality, i.e., we may assume that for every good $j \in G$, we have that $\sum_{i \in G} p_i = 1$.

Again, in a similar manner to Sections 6.1 and 6.2, we will use a parameter ε to capture the fact that if the price p_j for a good $j \in G$ is sufficiently smaller than the price $p_{j'}$ for a good $j' \in G$, then BPB(j,k) > BPB(j',k'), for any $k,k' \in K_i$. Specifically, we can compute $\varepsilon > 0$ such that

If
$$p_j \le \varepsilon \cdot p_{j'}$$
 and $u^i_{jk} > 0$ then $BPB_i(j,k) > BPB_i(j',k')$.

Additionally, we can pick ε to be sufficiently small such that $\varepsilon < \frac{w_{ij}}{m}$ for all $i \in N$ and all $j \in G$. Given ε , we will impose a stricter lower bound on the prices, which will be useful later on: in particular, we will assume that for all $j \in G$, $p_j \ge \frac{\varepsilon^m}{m}$.

$$\begin{array}{ll} & \underline{\text{Linear Program }\mathcal{P}_{\text{con}}} \\ \\ \text{minimize} & \sum_{j \in \mathcal{G}} \sum_{k \in K_i} \frac{p_j}{u^i_{jk}} \cdot q^i_{jk} \\ \\ \text{subject to} & \sum_{j \in \mathcal{G}} q^i_{jk} \leq \min\{L^i_k, CL+1\} \cdot \min_j \frac{p_j}{u^i_{jk}}, \ \forall k \in K_i \\ \\ & \sum_{j \in \mathcal{G}} \sum_{k \in K_i} q^i_{jk} = \sum_{j \in \mathcal{G}} w^i_j p_j + \sum_{f \in \mathcal{F}} \theta^i_f \cdot \left(\sum_{j \in \mathcal{P}_f} r^f_j - \sum_{j \in \mathcal{R}_f} s^f_j\right) \\ & q^i_{jk} \geq 0, \ \forall j \in \mathcal{G}, \ \forall k \in K_i \end{array}$$

Figure 9: The linear program \mathcal{P}_{con} for the optimal expenditures in consumption. Note that CL + 1 will be chosen to be large enough such that $\min\{L_k^i, CL + 1\} = CL + 1$ only when $L_k^i = \infty$.

6.3.1 Preprocessing

The approach that we will take for the proof is very similar to the one that we used in Section 6.2. The main difference comes from the fact that we now need to consider pairs of (goods, tranches) rather than just goods. This mainly complicates notation, but the type of arguments that we make are very much along the same lines as in the previous section. Again, we employ a standard variable change (see *Gale's Substitution* in Remark 6.2) to work with expenditures for the consumers and the firms, rather than with quantities of goods. This is to avoid multiplications of parameters which will be input to the circuit that we will construct, in particular the prices \mathbf{p} and the quantities x_{jk}^i , and y_{jk}^i in the bundle optimality condition of the competitive equilibrium in Definition 6.4.

Consumer expenditure. For the consumers, we will use the following standard change of variables. We will let $q_{jk}^i = x_{jk}^i \cdot p_j$ be the *expenditure* of consumer i on good j and tranche k, i.e., how much money the consumer spends on the good at a given tranche given an allocation x_{jk}^i and a price p_j .

Firm expenditure. For the firms, we will use a similar substitution for the case of production. Namely, we will let

- For $j \in \mathcal{R}_f$, let $s_j^f = p_j \cdot y_j^f$ be the *expenditure* of firm f on its raw good j on all tranches, i.e., the amount of money spend on y_f units of the raw good j, at given set of prices \mathbf{p} .

Let
$$\mathbf{s}^f = \left(s_j^f\right)_{j \in \mathcal{R}_f}$$
 and let $\mathbf{s} = \left(\mathbf{s}^f\right)_{f \in F}$.

- For $j \in \mathcal{P}_f$, let $r_j^f = p_j \cdot y_j^f$ be the *revenue* of firm f from its produced good j on all tranches, i.e., the amount of money earned from y_f units of the produced good j, at a given set of prices \mathbf{p} .

Let
$$\mathbf{r}^f = \left(r_j^f\right)_{j \in \mathcal{P}_f}$$
 and let $\mathbf{r} = \left(\mathbf{r}^f\right)_{f \in F}$.

The program \mathcal{P}_{con} for optimal consumer expenditure. We are now ready to write the linear program \mathcal{P}_1 , the solution of which will give us the optimal expenditures for the consumers, see Figure 9. In the linear program, notice the quantity $\min\{L_k^i, CL+1\}$. CL+1 will be chosen to be large enough, so that $\min\{L_k^i, CL+1\} = CL+1$ only when $L_k^i = \infty$. This is to capture the cases in which the consumer should be allowed to spend an infinite amount of money on goods on a tranche, since there is upper bound on

the utility the she can accrue from this tranche. Still, we remark again (see Remark 6.10) that there are inherent bounds on how much a consumer can spend, which do not come from L_k^i but rather from the inherent global upper bounds on production. In particular, it holds that $\sum_{j \in G} u_{jk}^i x_{jk}^i \leq L^p$, where L^p is a global upper bound on the consumption resulting from the global upper bound L^c on the production.

In previous sections, it was straightforward to see that the from the optimal solution \mathbf{q}^i of the corresponding linear program, one could recover the optimal bundle \mathbf{x}^i of consumer i. Here, we prove that in a simple lemma below. Note that we only need to guarantee that the linear program computes the optimal expenditures correctly at a competitive equilibrium, and so we can use the fact that $\sum_{j \in G} u^i_{ik} x^i_{jk} \leq L^p$.

Lemma 6.5. Suppose that \mathbf{q}^i is an optimal solution to linear program \mathcal{P}_{con} of Figure 9, and let Let $x^i_{jk} = q^i_{jk}/p_j$ for all $j \in G$ and $k \in K_i$. Then, the resulting bundle \mathbf{x}^i is an optimal consumption vector for consumer i at a competitive equilibrium of the market.

Proof. First note that the linear program is feasible, because by the strong connectivity of the economy graph $G_E(\mathcal{M})$, i is non-satiated with respect to some good, which means that she can spend its entire budget on this good. By construction, consumer i will spend money in order of increasing BPB $_i(j,k)$, establishing the optimality of the resulting bundle \mathbf{x}^i . It remains to show that the feasibility constraints are satisfied. Note that if $x^i_{jk} > 0$, then $q^i_{jk} > 0$, establishing that $p_j/u^i_{jk} = \min_{j'} p_{j'}/u^i_{j'k}$. Hence, for any $i \in G$ an $k \in K_i$, we have that

$$\sum_{j \in G} u^{i}_{jk} x^{i}_{jk} = \sum_{j \in G} \left(\frac{u^{i}_{jk}}{p_{j}} \right) q^{i}_{jk} = \frac{1}{\min\limits_{j' \in G}} \sum_{p_{j'} / u^{i}_{j'k}} \sum_{j \in G} q^{i}_{jk}$$

$$\leq \frac{1}{\min\limits_{j' \in G}} \sum_{p_{j'} / u^{i}_{j'k}} \cdot \min\{L^{i}_{k}, CL + 1\} \cdot \min_{j' \in G} \frac{p_{j'}}{u^{i}_{j'k}} = \min\{L^{i}_{k}, CL + 1\}$$

This completes the proof of the lemma, since CL+1 will be chosen to be larger than any finite L_k^i and larger than L^p .

The programs \mathcal{P}_{prod1} and \mathcal{P}_{prod2} for optimal firm expenditure. In Section 6.2 we devised a single linear program for the optimal production of the firms. Here, since we have variables for both the expenditures on raw goods and the expenditures on produced goods, we will devise two linear programs, one for each case. Those can be seen in Figure 10. Again, in Section 6.2 the corresponding linear program almost straightforwardly captured the optimal production of the firm which it represented; here we argue that this is the case with another simple lemma.

Lemma 6.6. Suppose that \mathbf{s}^f and \mathbf{r}^f are solutions to the linear programs \mathcal{P}_{prod1} and \mathcal{P}_{prod2} of Figure 10 respectively. Let $y_j^f = s_j^f/p_j$ for $j \in \mathcal{R}_f$ and $y_{j'}^f = r_{j'}^f/p_{j'}$ for $j' \in \mathcal{P}_f$. Then \mathbf{y}^f is an optimal production vector for firm f at a competitive equilibrium of the market.

Proof. We will argue that \mathbf{y}^f satisfies all the conditions of Definition 6.3. Since \mathbf{y}^f consists of optimal solutions to the two linear programs, Condition 5 and Condition 6 are clearly satisfied. Now, suppose that $\operatorname{ppu}^f(p) < 0$. This implies that all the coefficients of linear program $\mathcal{P}_{\operatorname{prod}1}$ are negative, meaning that $s_j^f = 0$ for all $j \in \mathcal{R}_f$. By the constraints of linear program $\mathcal{P}_{\operatorname{prod}1}$, it follows that also $r_j^f = 0$ for all $j \in \mathcal{P}_f$. Hence $y^f = \mathbf{0}$, and Condition 2 also holds. If $\operatorname{ppu}^f(p) > 0$, then at least one coefficient of linear program $\mathcal{P}_{\operatorname{prod}1}$ is positive, implying that $\sum_{j \in \mathcal{R}_f} s_j^f = \min\{L^f, L\} \cdot \operatorname{cpu}^f(\mathbf{p})$. Also, if $s_j^f > 0$, then $p_j/\alpha_j^f = \operatorname{cpu}^f(\mathbf{p})$. Hence, we have that

$$\sum_{j \in \mathcal{R}_f} \alpha_j^f y_j^f = \sum_{j \in \mathcal{R}_f} \frac{\alpha_j^f}{p_j} \cdot s_j^f = \frac{1}{\text{cpu}^f(\mathbf{p})} \sum_{j \in \mathcal{R}_f} s_j^f = \frac{1}{\text{cpu}^f(\mathbf{p})} \cdot \min\{L^f, L\} \cdot \text{cpu}^f(\mathbf{p}) = \min\{L^f, L\}$$

$$\begin{array}{|c|c|c|} \underline{\text{Linear Program }\mathcal{P}_{\text{prod1}}} & \underline{\text{Linear Program }\mathcal{P}_{\text{prod2}}} \\ \text{maximize} & \sum_{j \in \mathcal{R}_f} \left(\text{rpu}^f(\mathbf{p}) - \frac{p_j}{\alpha_j^f} \right) s_j^f \\ \text{subject to} & \sum_{j \in \mathcal{R}_f} s_j^f \leq \min\{L^f, L\} \cdot \text{cpu}^f(\mathbf{p}) \\ & s_j^f \geq 0, \ \forall j \in \mathcal{R}_f \end{array} \qquad \begin{array}{|c|c|c|} \underline{\text{maximize}} & \underline{\sum_{j' \in \mathcal{P}_f} \left(p_{j'} \beta_{j'}^f \right) \cdot r_{j'}^f} \\ \text{subject to} & \sum_{j' \in \mathcal{P}_f} r_{j'}^f = \sum_{j \in \mathcal{R}_f} s_j^f + B \\ & B = \max\{0, \text{rpu}^f(\mathbf{p}) - \text{cpu}^f(\mathbf{p})\} \cdot \min\{L^f, L\} \\ & r_{j'}^f \geq 0, \ \forall j' \in \mathcal{P}_f \end{array}$$

Figure 10: The linear programs $\mathcal{P}_{\text{prod1}}$ and $\mathcal{P}_{\text{prod2}}$ for the optimal expenditures in production, for the raw goods (left) and for the produced goods (right). Note that L will be sufficiently large such that $\min\{L^f, L\} = L$ will only hold when $L^f = \infty$, in both programs. Also note that the variable B is only used for notational convenience.

and hence Condition 3 is satisfied (see also Remark 6.9). What remains is to argue is that \mathbf{y}^f satisfied Condition Item 1 of Definition 6.3, i.e. that it is a feasible production vector. If $\mathbf{ppu}^f(\mathbf{p}) < 0$, then $\mathbf{y}^f = \mathbf{0}$ is feasible. If $\mathbf{ppu}^f(\mathbf{p}) > 0$, then it can be verified that $\sum_{j \in \mathcal{R}_f} \alpha_j^f y_j^f = \sum_{j \in \mathcal{P}_f} y_j^f / \beta_j^f$. Finally, if $\mathbf{ppu}^f(\mathbf{p}) = 0$, then

$$\sum_{j' \in \mathcal{P}_f} \frac{y_{j'}^f}{\beta_{i'}^f} = \sum_{j' \in \mathcal{P}_f} \frac{r_{j'}^f}{p_{j'} \cdot \beta_{i'}^f} = \sum_{j' \in \mathcal{P}_f} \frac{r_{j'}^f}{\mathrm{rpu}^f(\mathbf{p})} = \sum_{j \in \mathcal{R}_f} \frac{s_j^f}{\mathrm{cpu}^f(\mathbf{p})} = \sum_{j \in \mathcal{R}_f} \frac{\alpha_j^f s_j^f}{p_j} = \sum_{j \in \mathcal{R}_f} \alpha_j^f y_j^f.$$

This establishes all of the properties of Definition 6.3. Finally, let $\hat{G} = \{j \in R_f : \operatorname{rpu}(\mathbf{p}) - p_j/\alpha_j^f > 0\}$. The corresponding constraint of linear program $\mathcal{P}_{\operatorname{prod1}}$ stipulates that $\sum_{j \in \hat{G}} s_j^f = \min\{L^f, L\} \cdot \operatorname{cpu}^f(\mathbf{p})$. Our proof will establish that in a competitive equilibrium, if $L^f > L^c$ (the global upper bound on production), then \hat{G} does not contain any goods. This is to establish that the optimality of s_j^f does not impose any artificial constraints on the expenditure on raw goods due to the constraints of the linear program.

Excess expenditure and the program \mathcal{Q} . Similarly to Sections 6.1 and 6.2, the equilibrium prices will be obtained via a feasibility program \mathcal{Q} . Before we define the program, we introduce the notion of excess expenditure; the definition is very similar to the one we used in previous sections. The *excess expenditure* e_j of a good $j \in G$ is defined as the difference between the total expenditure of all consumers $i \in N$ and firms $f \in F$ for that good and the price of the good plus the total revenue of firms $f \in F$ from that good, i.e.,

$$e_j = \sum_{i \in N} \sum_{k \in K_i} q_{jk}^i + \sum_{f \in F: j \in \mathcal{R}_f} s_j^f - p_j - \sum_{f \in F: j \in \mathcal{P}_f} r_j^f.$$

Then, at equilibrium prices $\bar{\mathbf{p}}$ we will have market clearing, i.e., $\bar{e}_i = 0$, for all $i \in G$.

We are now ready to define our feasibility program Q; see Figure 11. The program looks in fact identical with that which we used in the previous sections; the only difference being the updated definition of the expenditures e_j . The equilibrium prices will be obtained as the output $\bar{\mathbf{p}}$ of Q.

6.3.2 Membership in PPAD: The proof of Theorem 6.3

We will again develop the proof in three steps, namely (a) construction of the function *F* and arguing that it can be represented by a PL arithmetic circuit containing linear-OPT-gates, (b) showing that the

Feasibility Program
$$Q$$

$$e_{j} < e_{j'} \Rightarrow p_{j} \leq \varepsilon \cdot p_{j'}, \ \forall j, j' \in G$$

$$p_{j} \geq \frac{\varepsilon^{m}}{m}, \ \forall j \in G$$

$$\sum_{j \in G} p_{j} = 1$$

Figure 11: The feasibility program Q used to find market-clearing prices.

linear-OPT-gate can compute all the necessary components, and (c) arguing that a competitive equilibrium can be recovered from a fixed point of *F*.

The function *F***.** Given the above, we will define $F: D \to D$ with domain

$$D = \left\{ p \in \Delta^{m-1} : \forall j \in G, p_j \ge \frac{\varepsilon^m}{m} \right\} \times_{i \in N} \left([0, C \cdot L + 1]^{m|K_i|} \right) \times [0, L]^{\ell m}.$$

An input to F is a tuple $(\hat{\mathbf{p}}, \hat{\mathbf{q}}, \hat{\mathbf{s}}, \hat{\mathbf{r}})$ of prices, consumer expenditures and firm expenditures for raw goods and produced goods, and the output is another such tuple $(\bar{\mathbf{p}}, \bar{\mathbf{q}}, \bar{\mathbf{s}}, \bar{\mathbf{r}})$. The domain D is the one above since $\sum_{j \in G} \hat{p}_j = 1$ (see Remark 6.11), and by the upper bounds of $C \cdot L + 1$ and L imposed on the maximum expenditures for consumption and production respectively.

Computation by the linear-OPT-gate. Next, we argue that the solutions to the linear programs \mathcal{P}_{con} of Figure 9, \mathcal{P}_{prod1} and \mathcal{P}_{prod2} of Figure 10, and the feasibility program \mathcal{Q} of Figure 11 can be computed by our linear-OPT-gate.

Lemma 6.7. Consider the linear program \mathcal{P}_{con} of Figure 9, the linear programs \mathcal{P}_{prod1} and \mathcal{P}_{prod2} of Figure 10, and the feasibility program \mathcal{Q} of Figure 11. An optimal solution to \mathcal{P}_{con} , to \mathcal{P}_{prod1} and to \mathcal{P}_{prod2} can be computed by the linear-OPT-gate. \mathcal{Q} is solvable, and a solution can be computed by the linear-OPT-gate.

Proof. The proof is very similar to that of Lemma 6.3. For all of the linear programs, the feasibly domains non-empty and bounded. The gate inputs appear only on the right-hand side of the constraints: these are the prices \hat{p}_j , and the production expenditures \hat{s}_j^f and \hat{r}_j^f for linear program \mathcal{P}_{con} , the prices \hat{p}_j (as part of cpu $^f(\hat{\mathbf{p}})$) for linear program \mathcal{P}_{prod1} , the prices \hat{p}_j (as part of cpu $^f(\hat{\mathbf{p}})$) and the expenditures s_j^f for linear program \mathcal{P}_{prod2} . The subgradients of the objective functions for all three linear programs are linear, and hence can be computed by a PL pseudo-circuit. For the feasibility program \mathcal{Q} , the arguments that it is of the correct form and that it is solvable are identical to those of Lemma 6.3 (and hence also of Lemma 6.1), noting the updated definition of the expenditure e_j .

Arguing optimality and market clearing. To conclude the proof, what is left to show is that a fixed point $(\bar{\mathbf{p}}, \bar{\mathbf{q}}, \bar{\mathbf{s}}, \bar{\mathbf{r}})$ of F indeed corresponds to a competitive equilibrium. We argue that in the following lemma.

Lemma 6.8. Let $(\bar{\mathbf{p}}, \bar{\mathbf{q}}, \bar{\mathbf{s}}, \bar{\mathbf{r}})$ be a fixed point of F. Then the triple $(\bar{\mathbf{p}}, \bar{\mathbf{x}}, \bar{\mathbf{y}}, \bar{\mathbf{r}})$, where $\bar{x}^i_{jk} = \bar{q}^i_{jk}/\bar{p}_j$, $\bar{y}^f_j = \bar{s}^f_j/\bar{p}_j$ for $j \in \mathcal{R}_f$ and $\bar{y}^f_{j'} = \bar{r}^f_{j'}/\bar{p}_{j'}$ for $j' \in \mathcal{P}_f$ is a competitive equilibrium of \mathcal{M} .

Proof. In Lemma 6.5 and Lemma 6.6 we argued that the linear programs \mathcal{P}_{con} , \mathcal{P}_{prod1} and \mathcal{P}_{prod2} correctly calculate the optimal consumption vectors of the consumers and the optimal production vectors of the firms. What is left to argue is that $\mathbf{\bar{p}}$ is a vector of market-clearing prices. The proof will follow along very

much the same lines as the one of Lemma 6.4, using updated the updated terminology and notation of this section.

By the definition of e_j , arguing about market-clearing is equivalent to arguing that for all $j \in G$, we have that $\bar{e}_j = 0$. Using a very similar calculation to the one that we used in Lemma 6.4, this is equivalent to arguing that $\bar{e}_j \geq 0$ for all $j \in G$. We provide the calculation below.

$$\begin{split} \sum_{j \in G} \bar{e}_j &= \sum_{j \in G} \left(\sum_{i \in N} \sum_{k \in K_i} \bar{q}^i_{jk} - \bar{p}_j + \sum_{f \in F: j \in \mathcal{R}_f} \bar{s}_f - \sum_{f \in F: j \in \mathcal{P}_f} \bar{r}_f \right) \\ &= \sum_{i \in N} \sum_{j \in G} \sum_{k \in K_i} \bar{q}^i_{jk} - \sum_{j \in G} \bar{p}_j + \sum_{f \in F} (\bar{s}_f - \bar{r}_f) \\ &= \sum_{j \in G} \sum_{i \in N} w_{ij} \bar{p}_j - 1 + \sum_{i \in N} \sum_{f \in F} \theta_{if} \cdot (\bar{r}_f - \bar{s}_f) + \sum_{f \in F} (\bar{s}_f - \bar{r}_f) \\ &= \sum_{j \in G} \bar{p}_j \cdot \sum_{i \in N} w_{ij} - 1 + \sum_{f \in F} (\bar{r}_f - \bar{s}_f) \cdot \sum_{i \in N} \theta_{if} \cdot \sum_{f \in F} (\bar{s}_f - \bar{r}_f) \\ &= \sum_{i \in N} w_{ij} - 1 + \sum_{f \in F} (\bar{r}_f - \bar{s}_f) + \sum_{f \in F} (\bar{s}_f - \bar{r}_f) \\ &= 0, \end{split}$$

where in the calculations above we used:

- in Equation 2, that $\sum_{j \in G} \sum_{f \in F: j \in \mathcal{R}_f} \bar{s}_f = \sum_{f \in F} \bar{s}_j$ and that $\sum_{j \in G} \sum_{f \in F: j \in \mathcal{P}_f} \bar{r}_f = \sum_{f \in F} \bar{r}_f$,
- in Equation 3, that $\sum_{i \in N} \sum_{j \in G} \sum_{k \in K_i} \bar{q}_{ij} = \sum_{i \in N} \sum_{j \in G} w_{ij} \bar{p}_j + \sum_{i \in N} \sum_{f \in F} \theta_{if}(\bar{r}_f \bar{s}_f)$, which follows from the bundle optimality of the market equilibrium in Definition 6.4, which at an optimal solution is sastisfied with equality,
- in Equations 3 and 5, that $\sum_{j \in G} \bar{p}_j = 1$, which is without loss of generality, see Remark 6.11,
- in Equation 5, that $\sum_{i \in N} \theta_{if} = 1$ for every $f \in F$, by definition, and
- in Equation 6, that $\sum_{i \in N} w_{ij} = 1$, which is without loss of generality.

Assume by contradiction that there exists some $j_1 \in G$ such that $\bar{e}_{j_1} < 0$; we will obtain a contradiction to the strong connectivity of the economy graph of the market. We define the following three sets:

- $J = \{j \in G : \bar{e}_j \leq \bar{e}_{j'}$, for all $j' \in G\}$. In other words, J is the set of goods with minimum excess expenditure.
- $N_I = \{i \in N : \text{ there exists } j \in I \text{ such that consumer } i \text{ is nonsatiated for good } j\}$.
- $F_I = \{ f \in F : \text{ there exists } j \in J \text{ such that firm } f \text{ is nonsatiated for good } j \}.$

Since $e_{j_1}^* < 0$, by the definition of J we know that

for all
$$j \in J$$
, we have $e_j^* < 0$. (31)

Also, since $e_{j_1}^* < 0$, and $\sum_{j \in G} e_j^* = 0$, there must exist some other good j_2 such that $e_{j_2}^* > 0$. In particular, that implies that J is a strict subset of G, i.e., $J \subsetneq G$. We state and prove the following claim.

Claim 6.3. For all $i \in N_I$ and all $j' \in G \setminus J$, it holds that $w_{ii'} = 0$.

Proof. The proof of the claim is very similar to that of Claim 6.1 which we presented in Section 6.2. Assume by contradiction that there exists some consumer $i_0 \in N_J$ and some good $j' \in G \setminus J$ such that $w_{i_0j'} > 0$. We will show that this implies that there exists some $j_\ell \in J$ with $\bar{e}_{j_\ell} \ge 0$, contradicting Statement (31). Since $\bar{\mathbf{p}}$ is a solution to feasibility program \mathcal{Q} of Figure 8, we have that $\bar{p}_j \le \varepsilon \cdot \bar{p}_{j'}$, for every $j \in J$. Hence, by choosing ε to be sufficiently small, we may lower-bound the budget of consumer i by

$$w_{i_0j'} \cdot p_{j'}^* \ge \frac{w_{i_0j'}}{\varepsilon \cdot m} \cdot \sum_{j \in J} \bar{p}_j \ge (C \cdot L + 1) \sum_{j \in J} p_j^*$$

Since $i_0 \in N_J$, by the choice of ε , there is at least one good $g \in J$ such that $\operatorname{BPB}_{i_0}(g,k) > \operatorname{BPB}_{i_0}(g',k)$, for all $g' \in G \setminus J$, and for any two tranches $k, k \in K_i$, i.e., the consumer prefers to buy quantities of good g rather than any good which is not in the set J, on any tranche. Since consumer i_0 is nonsatiated for some good $j \in J$ (by virtue of being in N_J), by the fact that \bar{q}_{i_0j} is an optimal solution to linear program $\mathcal{P}_{\operatorname{con}}$ of Figure 9, it should be that $\bar{q}_{i_0j} = C \cdot L + 1$, which implies that $\bar{q}_{i_0j} \geq (C \cdot L + 1)p_j^*$. By substituting that into the definition of the excess expenditure \bar{e}_j , we obtain the following inequality:

$$0 > \bar{e}_j^* \ge (CL+1)\bar{p}_j - \sum_{f \in F: j \in \mathcal{P}_f} \bar{r}_j^f - \bar{p}_j = CL \cdot \bar{p}_j - \sum_{f \in F: j \in \mathcal{P}_f} \bar{r}_j^f.$$

Therefore, it suffices to show that

$$\sum_{f \in F: j \in \mathcal{P}_f} \bar{r}_j^f \leq (C \cdot L) \cdot \bar{p}_j;$$

in that case we will have $\bar{e}_j \geq 0$ and we will obtain a contradiction. Now consider any firm $f \in F$. Note that if $\bar{r}^f \leq 0$, then the inequality above clearly follows. Otherwise, if $\bar{r}^j_f > 0$, then $\operatorname{rpu}(\bar{\mathbf{p}}) = \bar{p}_j \cdot \beta_j^f$. In this case, using the constraints of linear program $\mathcal{P}_{\operatorname{prod2}}$ of Figure 10, we may bound the production of good j by firm f as follows:

$$\begin{split} &\bar{r}_j^f \leq \sum_{j \in \mathcal{R}_f} \bar{s}_j^f + \max\{0, \operatorname{rpu}^f(\bar{\mathbf{p}}) - \operatorname{cpu}^f(\bar{\mathbf{p}})\} \cdot \min\{L^f, L\} \\ &\leq \operatorname{rpu}^f(\bar{\mathbf{p}}) \cdot \min\{L^f, L\} \\ &= (\beta_j^f \min\{L^f, L\}) \cdot \bar{p}_j \\ &\leq (\beta_j^f L) \bar{p}_j. \end{split}$$

This clearly bounds \bar{r}_j^f as long as the $L^f \neq \infty$, as in that case $L^f \leq L$. $L^f = \infty$, then $\bar{r}j^f$ is upper bounded by the global upper bound L^c on the production. Again, $L^c \leq L$, and the above inequality follows. By letting $C \geq |\mathcal{F}| \cdot \max\{\beta_j^f \colon f \in \mathcal{F} \land j \in \mathcal{P}_f\}$, we obtain the contradiction $0 > \bar{e}_j \geq 0$. This concludes the proof of the claim.

Claim 6.3 establishes that a consumer in the set N_J cannot endow any good that is not in the set J in a positive quantity. Since $J \subseteq G$, this implies that $N_J \subseteq N$, since every good is positively endowed by some consumer $i \in N$. Now consider any consumer $i \in N_J$ and any consumer $i' \in N \setminus N_J$ or any firm $f \in F \setminus F_J$. Claim 6.3 implies that there cannot be an edge (i,i') or an edge (i,f) in the economy graph of the market. Indeed, for (i,i'), notice that by the claim consumer i only endows positively goods that are in J, but consumer i' has value 0 for them by virtue of being in $N \setminus N_J$, and hence she is not nonsatiated for any good endowed positively by consumer i. Similarly, for (i,f), we have that $\mathcal{R}_f \subseteq G \setminus J$ by definition, but consumer i only positively endows goods in J, so clearly f cannot be nonsatiated for any of the goods endowed by consumer i.

Next, we state and prove the following claim.

Claim 6.4. Consider and $f \in F_I$, with $L^f = \infty$. For any good $j' \in \mathcal{P}_f$, it holds that $j' \in J$.

Proof. The claim also follows a very similar idea to that of Claim 6.4 which we presented in Section 6.2. Let $\alpha = \max_{f',j} \alpha_j^{f'}$ and $\beta = \max_{f',j} \beta_j^{f'}$. Assume by contradiction that for firm f, there is some good $j' \in \mathcal{P}_f$ such that $j' \in G \setminus J$. Note that f is nonsatiated with respect to some good in J (by virtue of being in F_J and $L^f = \infty$). Additionally, since $\bar{\mathbf{p}}$ is an optimal solution to feasibility program \mathcal{Q} of Figure 11, we have that $\bar{p}_j \cdot \varepsilon \bar{p}_{j'}$ for all goods $j \in J$. By the fact that $\bar{\mathbf{s}}^f$ is an optimal solution to linear program $\mathcal{P}_{\text{prod1}}$, it follows that firm f will utilize only goods from J as raw goods for its production. Hence, there exists some good $j_1 \in J$ such that

$$\bar{s}_{j_1}^f \geq \frac{L \cdot \operatorname{cpu}^{f_1}(\mathbf{p})}{m} \geq \frac{L}{m\alpha} \cdot \bar{p}_{j_1}$$

Using this, we conclude that

$$0 > \bar{e}_{j_1} \ge \left(\frac{L}{m\alpha} - 1\right) \bar{p}_{j_1} - \sum_{f': j_1 \in \mathcal{P}_{f_1}} \bar{r}_{j_1}^{f'} \tag{32}$$

If j_1 were not being produced, we would reach the contradiction that $0 > \bar{e}_{j_1} \ge 0$ by picking L sufficiently large. If j_1 is being produced, then it must be produced using some good j_2 from J. This is because by the choice of ε , it is only profitable for any firm to produce units of goods $j \in J$ (that have very low prices) only by using units of goods $j \in J$ as raw goods. Again, if j_2 is not being produced, we reach a contradiction by choosing L sufficiently large, and otherwise we may repeat the argument. Since there cannot be a cycle of profitable production in the graph $G_{\mathcal{F}}(\mathcal{M})$, by the no production out of nothing and no vacuous production sufficiency condition, the argument will need to be repeated at most m times, one for each good in the possibly longest production chain of the graph. In the end, we can pick L to be large enough to ensure that we get a contradiction.

We remark that we made the very same argument in the proof of Claim 6.2 via establishing appropriate bounds inductively for every step of the chain. In the case of markets with Leontief-free productions however, calculating the resulting quantities becomes very tedious, so we elected to present the proof a bit differently.

As it stands, Inequality 32 implies that there exists a firm f_1 such that $r_{j_1}^{f_1} \ge \left(\frac{L}{\ell m \alpha} - \frac{1}{\ell}\right) \bar{p}_{j_1}$. As f_1 produces a good in J, it holds that

- 1. $rpu^{f_1}(\mathbf{p}) \ge cpu^{f_1}(\mathbf{p})$ and
- 2. f_1 produces only using goods from J.

We consider two cases, depending on whether the inequality in Inequality 1 is strict or not.

Case 1: $\operatorname{rpu}^{f_1}(\bar{\mathbf{p}}) = \operatorname{cpu}^{f_1}(\bar{\mathbf{p}})$. In this case, we find that

$$\left(\frac{L}{\ell m\alpha} - \frac{1}{\ell}\right)\bar{p}_{j_1} \leq \bar{r}_{j_1}^{f_1} \leq \sum_{j \in \mathcal{P}_{f_1}} \bar{r}_j^{f_1} = \sum_{j \in \mathcal{R}_{f_1}} \bar{s}_j^{f_1}$$

From this it follows that there exist a good j_2 such that $\bar{s}_{j_2}^{f_1} \geq \left(\frac{L}{\ell m^2 \alpha} - \frac{1}{\ell m}\right) \bar{p}_{j_1}$. Using that $\bar{p}_{j_2}/\alpha_{j_2}^{f_1} = \text{cpu}^{f_1}(\bar{\mathbf{p}}) = \text{rpu}^{f_1}(\bar{\mathbf{p}}) = \beta_{j_1}^{f_1} \bar{p}_{j_1}$, this implies that $\bar{s}_{j_2}^{f_1} \geq \left(\frac{L}{\ell m^2 \alpha^2 \beta} - \frac{1}{\ell m \alpha \beta}\right) \bar{p}_{j_2}$. Now we may bound \bar{e}_{j_2} as before and repeat the argument.

Case 2: $\operatorname{rpu}^{f_1}(\bar{\mathbf{p}}) > \operatorname{cpu}^{f_1}(\bar{\mathbf{p}})$. First, we show that $L^{f_1} = \infty$. Assume that it is not. Then we have:

$$\left(\frac{L}{\ell m \alpha} - \frac{1}{\ell}\right) \bar{p}_{j_1} \leq \bar{r}_{j_1}^{f_1} \leq \sum_{j \in \mathcal{P}_{f_1}} \bar{r}_{j}^{f_1} \leq \min\{L^{f_1}, L\} \cdot \operatorname{rpu}^{f_1}(\bar{\mathbf{p}}) \leq (\min\{L^{f_1}, L\}\beta) \bar{p}_{j_1} \leq (L^{f_1}\beta) \bar{p}_{j_1}, \quad (33)$$

where the last inequality follows because L is chosen sufficiently large for $L_1^f \leq L$ to hold, since $L_1^f \neq \infty$ by assumption. For L sufficiently large though, this leads to a contradiction. Hence we may assume henceforth that $L^{f_1} = \infty$.

Given the above, as $\operatorname{rpu}^{f_1}(\bar{\mathbf{p}}) > \operatorname{cpu}^{f_1}(\bar{\mathbf{p}})$, we have that $\sum_{j \in \mathcal{R}_{f_1}} \bar{s}_j^{f_1} = L \cdot \operatorname{cpu}^{f_1}(\bar{\mathbf{p}})$. As f_1 produces a good from J, f_1 can only produce using goods from J as raw goods. This is because, as we discussed earlier, the prices of goods in J are sufficiently smaller that those for goods in $G \setminus J$, and hence any profitable production that uses goods in J as raw goods should also involve only goods in J as produced goods. Therefore, we establish that there exists a $j_2 \in J$ such that $\bar{s}_{j_2}^{f_1} \geq \frac{L}{m} \operatorname{cpu}^{f_1}(\bar{\mathbf{p}}) \geq \frac{L}{m\alpha} \bar{p}_{j_2}$. Using this, we may again bound \bar{e}_{j_2} and repeat the arguments above.

Finding $j_k \in J$ with $0 > \bar{e}_{j_k} \ge 0$. As we mentioned earlier, by the *no production out of nothing and no vacuous production* sufficiency condition, the above argument has to be repeated at most m times, as that is the maximum possible length of any profitable production chain. In both of the two cases consider above, L becomes smaller by a certain factor as a function of the input parameters. This means that L can be picked large enough to guarantee that at the end of the chain, when the last good is not being produced, we obtain that $0 > \bar{e}_{j_k} \ge 0$, a contradiction.

This concludes the proof of the claim.

Claim 6.4 implies that for any firm $f \in F_J$, with $L^f = \infty$, there cannot be an edge to any consumer $i \in N \setminus N_J$ or any firm $f' \in F \setminus F_J$ in the economy graph. Indeed, for an edge (f,i), the claim asserts that the any produced good j' of f will be in the set J, whereas consumer i only values goods in the set $G \setminus J$ positively, by virtue of being in $N \setminus N_J$. Similarly, for an edge (f,f'), firm f produces only goods in J, whereas the for firm f' we have $\mathcal{R}_{f'} \subseteq G \setminus J$, by virtue of f' being in $F \setminus F_J$.

From the two paragraphs succeeding Claim 6.3 and Claim 6.4, N_J and $N \setminus N_J$ are two strongly connected components in the economy graph of the market, contradicting the sufficiency condition requiring that in the economy graph there is a strongly connected component containing all the consumer-nodes. This completes the proof.

6.4 Arrow-Debreu Markets with Succinct SPLC Utilities and Production

In this section we provide our PPAD-membership result for *succinct* separable piecewise-linear utilities (SSPLC). These are SPLC utilities, i.e., each consumer has a piecewise-lineaer utility function over different pieces (or segments) of each good, and her total utility for her bundle is additive over those utility functions. The "succinct" part comes from the fact that, in contrast with the SPLC utilities that have been studied before in the literature (e.g., see [Garg et al., 2015; Garg and Vazirani, 2014; Vazirani and Yannakakis, 2011]), these functions can be accessed implicitly, given access to a boolean circuit that evaluates the function at different points. Effectively, these allows us to model functions with exponentially many pieces, while still developing a reduction which is polynomial in the size of the input, i.e., the size of those circuits. Our main theorem of the section is the following.

Theorem 6.4. Computing a competitive equilibrium in an Arrow-Debreu market with SSPLC utilities and SPLC production functions is in PPAD.

Note that we are also incorporating (explicit) SPLC productions in our result. Whether we can generalize those to SSPLC production functions is an interesting technical open problem; we discuss the challenges and why our technique falls short in this regard in Remark 6.15.

Markets with Production. In an Arrow-Debreu market with production \mathcal{M} , we have a set N of consumers, a set G of infinitely divisible goods, and a set F of firms. Let n = |N|, m = |G|, and $\ell = |F|$. We will typically use index i to refer to consumers, j or g to refer to goods and f to refer to firms. Each consumer brings an endowment $w_i = (w_{i1}, \ldots, w_{im})$ to the market, with $w_{ij} \geq 0$ for all $i \in N$ and $j \in G$. We may assume without loss of generality that for every good j, we have $\sum_{i \in N} w_{ij} = 1$, i.e., that the total endowment of each good is 1. We will use $\mathbf{x}_i = (x_{i1}, \ldots, x_{im})$ to denote the vector of quantities of goods allocated to consumer $i \in N$ in \mathcal{M} , and we will call it the *bundle* of consumer i. Let $\mathbf{x} = (x_1, \ldots, x_n)$ be the vector of such bundles. We will use $\mathbf{p} = (p_1, \ldots, p_m)$ to denote the vector of *prices* in \mathcal{M} , one for each good $j \in G$. Prices are non-negative, so $p_j \geq 0$ for all $j \in G$. Given a vector of prices \mathbf{p} , the *budget* of consumer $i \in N$ is defined as $\sum_{j \in G} w_{ij} p_j$; intuitively, this is the amount of money that the consumer acquires by selling her endowment at prices \mathbf{p} .

Utility functions. Every consumer has a utility function $u_i : \mathbb{R}^m_{\geq 0} \to \mathbb{R}_{\geq 0}$ mapping a bundle \mathbf{x}_i to a non-negative real number. In this section, these utilities are *succinct separable piecewise-linear concave* (*SSLPC*). A utility u_i being SPLC means that it is separable over the goods and that the utility accrued from each good is given by a piecewise-linear concave function u_{ij} , that is, $u_i(\mathbf{x}_i) = \sum_{j=1}^m u_{ij}(x_{ij})$. The word *succint* refers to the piecewise-linear utilities concave u_{ij} being given by boolean circuits computing their slopes, thus allowing for an exponential number of changes to the value of the slope. Concretely, we assume that the utility functions are given by the following data:

- 1. A natural number $A \in \mathbb{N}$ such that the piecewise linear functions u_{ij} have constant slope in the interval (A, ∞) .
- 2. A natural number $B \in \mathbb{N}$ such that the slopes of the utility functions can change only in the points $\frac{k}{B} \cdot A$ for $k \in \{0, 1, \dots, B\}$.
- 3. For every $i \in N$ and $j \in G$, a boolean circuit C_{ij} with $\lfloor \log_2(B) \rfloor + 1$ input bits and output bits which on input $k \in \{0,1,\ldots,B\}$ computes $1/u_{ijk}$, where u_{ijk} is the slope of u_{ij} in the interval ((k/B)A,((k+1)/B)A). Note that this makes sense only if the slope is strictly positive. Also, the utility functions are concave, meaning that $u_{ij0} \ge u_{ij1} \ge \cdots \ge u_{ijB}$.
- 4. For every $i \in N$ and $j \in J$, if the slope of u_{ij} eventually becomes zero, then we are given a natural number D_{ij} such that the slope becomes zero in the point $(D_{ij}/B)A$.

Remark 6.12. The reason that we assume that the boolean circuits C_{ij} compute the inverse of the slopes is that this allows us to compute bang-per-buck ratios which is required in our construction.

Firm Shares. Each consumer $i \in N$ has a share $\theta_{if} \in [0,1]$ of the profit of each firm $f \in F$. We assume that the profits are entirely shared among the consumers, i.e., for every firm $f \in F$, we have that $\sum_{i \in N} \theta_{if} = 1$.

Production functions. In this section we assume that the firms have *separable piecewise-linear concave* production functions given explicitly. Every firm f has a single *output good* g_f^{out} which it can produce, and for every $j \in G$, it has a piecewise-linear concave function $P_j^f : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ defining the ability of f to produce g_f^{out} as a function of the good f. The overall production of firm f from bundle $\mathbf{y} = (y_1, \dots, y_m)$ is then given by f (\mathbf{y}) = \mathbf{y}). Because of separability, we may assume that every firm has an *input good* g_f^{in} , which is the only good with which it can produce a strictly positive amount of g_f^{out} . As such the production function f is simply a piecewise-linear concave function. We assume that this function is given to us explicitly as follows:

1. The nonnegative real line $\mathbb{R}_{\geq 0}$ is divided into a finite number K_f of subsequent segments of length L_{fk} for $1 \leq k \leq K_f$. The length of the final segment is infinite.

2. To every segment is associated a *conversion rate* $c_{fk} \ge 0$ which is the slope of P_f in this segment, that is, c_{fk} is the rate at which f can convert good g_f^{in} to good g_f^{out} on this segment. The production function P^f being concave means that $c_{f1} \ge c_{f2} \ge \cdots \ge c_{fK_f}$.

One may wonder why we do not assume that the slopes of the production functions are given succinctly by a boolean circuit as we did for the utility functions. We discuss the problems arising from this representation later in the section.

Competitive Equilibrium. We are now ready to define the notion of a competitive equilibrium in markets with production.

Definition 6.5 (Competitive Equilibrium - Markets with SSPLC Utilities and SPLC Production). A competitive equilibrium of an Arrow-Debreu market with SSPLC utilities and PLC production functions is a triple $(\mathbf{p}^*, \mathbf{x}^*, \mathbf{y}^*)$ consisting of non-negative prices \mathbf{p}^* , non-negative bundles $\mathbf{x}^* = (\mathbf{x}_1^*, \dots, \mathbf{x}_n^*)$ and non-negative amounts of input goods $\mathbf{y}^* = (y_1^*, \dots, y_\ell^*)$, such that

- 1. $p_{g_f^{\text{out}}}^* \cdot P_f(y_f^*) p_{g_f^{\text{in}}}^* \cdot y_f^*$ is maximized, for any firm $f \in F$. (firm profit maximization)
- 2. $u_i(\mathbf{x}_i)$ is maximized for every consumer $i \in N$, subject to $\sum_{j \in G} p_j^* \cdot x_{ij} \leq \sum_{j \in G} p_j^* \cdot w_{ij} + \sum_{f \in F} \theta_{if} \cdot (p_{g_f^{\text{out}}}^* \cdot P_f(y_f^*) p_{g_f^{\text{in}}}^* \cdot y_f^*).$ (bundle optimality)
- 3. $z_{j}^{*} \leq 0$, and $z_{j}^{*} p_{j}^{*} = 0$, where $z_{j}^{*} = \sum_{i \in N} x_{ij}^{*} + \sum_{f \in F: g_{f}^{in} = j} y_{f}^{*} \sum_{f \in F: g_{f}^{out} = j} P_{f}(y_{f}^{*}) 1$, for every good $j \in G$. (market clearing)

Condition 1 requires that at the chosen set of prices \mathbf{p}^* , each firm maximizes its profit, given its production functions. Condition 2 requires that at the chosen set of prices \mathbf{p}^* , each consumer maximizes her utility subject to their budget constraints, where the budget consists of the amount earned from selling all the consumer's endowments $\sum_{j \in G} p_j^* w_{ij}$ and the profit share $\sum_{f \in F} \theta_{if} \cdot (p_{g_f^{\text{out}}}^* \cdot P_f(y_f^*) - p_{g_f^{\text{in}}}^* \cdot y_f^*)$ of the consumer

from the production of the firms. Finally, Condition 3 is the market clearing condition, which requires that the total consumption of each good is at most the total production plus the total endowment of the consumers, and supply equals demand for all goods which are not priced at 0. As we detail later on in the section, we may in fact assume without loss of generality that in any competitive equilibrium *all the prices* are positive, and hence Condition 3 reduces to $z_j^* = 0$ for all $j \in G$. Note that in Condition 3 we have used that $\sum_{i \in N} w_{ij} = 1$ for each good $j \in G$.

Sufficiency Conditions. A competitive equilibrium as defined above exists for every market \mathcal{M} , under some sufficiency conditions. Similarly to Section 6.2, we will use the set of conditions used by Maxfield [1997], also used in the series of papers on market equilibria that we mentioned in the beginning of Section 6.

1. Consider a graph $\mathcal{G}_F(\mathcal{M})$ in which the nodes are the goods, and an edge (j, j') has weight

$$w_{jj'} = \max_{f \in F: j = g_f^{\text{in}} \land j' = g_f^{\text{out}}} c_{f1}$$

If the set is empty, the weight is defined to be 0.

The above weight captures the fact that j' can be produced from j at conversion rate $w_{jj'}$ by some firm $f \in F$. Then, for any cycle $C = (g_0, g_1), (g_1, g_2), \dots, (g_{k-2}, g_{k-1})$ of $G_F(\mathcal{M})$ the product of the weights of the edges is less than 1, i.e., $\prod_{e \in C} \alpha_e < 1$.

This condition is known as the *no production out of nothing and no vacuous production* condition. Indeed, if $\prod_{e \in C} \alpha_e > 1$, then it would be possible to increase the quantity of some good, without

decreasing the quantity of any other good. The case of $\prod_{e \in C} \alpha_e = 1$ refers to the case of vacuous production, which is also disallowed in our model, similarly to the related works, e.g., see [Garg and Vazirani, 2014].

- 2. We will say that a consumer $i \in N$ is *nonsatiated* for $good\ j \in G$ if $C_{ij}(B) > 0$. We will assume that for any good $j \in G$, there exists some consumer $i \in N$ that is nonsatiated with respect to j. Similarly, we say that a firm f *nonsatiated* for $good\ j \in G$ if $j = g_f^{in}$ and $c_{fK_f} > 0$.
- 3. Consider the *economy graph* $\mathcal{G}_E(\mathcal{M})$ of the market \mathcal{M} in which the nodes are the consumers and the firms, and there is an edge from node a to node b in node a owns/produces a good for which node b is nonsatiated. Then, $\mathcal{G}_E(\mathcal{M})$ contains a strongly connected component containing all the consumer-nodes. This condition generalizes the corresponding strong connectivity condition that we used in Section 6.2.

Remark 6.13 (Bounds on Production). Condition 1 above imposes a bound on the total amount of a production of any firm $f \in F$ in a competitive equilibrium. Note that the production starts from finite endowments $\sum_{i \in N} w_{ij} = 1$ for all $j \in G$. Since no firm operates at a loss at an equilibrium, any cycle of production would violate Condition Item 1. Since there are no such cycles, production can take place along chains. The longest such chain is obviously bounded by m, and the maximum production of any firm can be bounded by some sufficiently large global constant L^c (e.g., some constant such that $L^c \geq m^m (\max_f C_f(0) + 1)^n$). See also [Garg and Vazirani, 2014] for a very similar argument. Looking ahead, this will allow us to impose "loose" upper bounds on the production and consumption in the linear programs that we will devise, without compromising the existence of a competitive equilibrium.

Optimality and bang-per-buck (BPB). Similarly to Section 6.1, the optimal bundles of Condition 2 in Definition 6.2 are characterized by their *bang-per-buck (BPB)*. Given consumer $i \in N$ and prices \mathbf{p} , the bang-per-buck of good good j on segment k is defined as $BPB_i(j,k) = \frac{u_{ijk}}{p_i}$.

Bounds on the prices. Recall that we have assumed that for every good *j*, there is some agent *i* that is nonsatiated with respect to *j*. Hence, using very similar arguments to Section 6.2 and Section 6.3, we may assume that all the prices are strictly positive.

Remark 6.14 (Normalized Prices). Given that $p_j > 0$ for all $j \in G$, we can normalize the prices to sum to 1 without loss of generality, i.e., we may assume that for every good $j \in G$, we have that $\sum_{i \in G} p_i = 1$.

Again, in a similar manner to Section 6.1, we will use a parameter ε to capture the fact that if the price p_j for a good $j \in G$ is sufficiently smaller than the price $p_{j'}$ for a good $j' \in G$, then BPB(j) > BPB(j'). Specifically, we can compute $\varepsilon > 0$ such that

If
$$p_j \le \varepsilon \cdot p_{j'}$$
 and $u_{ijk} > 0$ then $BPB_i(j,k) > BPB_i(j',k')$ for all k' .

Because the inverses of the utilities are in the set $\{1,2,\ldots,B\}$ by assumption, the actual utilities are in the set $\{1/B,1/(B-1),\ldots,1/2,1\}$. For the above to hold, it is thus sufficient that if $p_j \leq \varepsilon p_{j'}$, then $(1/B)/p_j \geq 1/p_{j'}$. For this implication to hold, it suffices to choose $\varepsilon \leq 1/B$. Additionally, we can pick ε to be sufficiently small such that $\varepsilon < \frac{w_{ij}}{m}$ for all $i \in N$ and all $j \in G$ such that $w_{ij} > 0$. Given ε , we will impose a stricter lower bound on the prices, which will be useful later on: in particular, we will assume that for all $j \in G$, $p_j \geq \frac{\varepsilon^m}{m}$.

Expenditures. As in all of the previous sections, we do not work directly with the quantities \mathbf{x} , \mathbf{y} . Instead we work with the consumer and firm *expenditures* respectively, which are defined similarly to those of Section 6.2. Specifically, for every for agent i we will have a variable q_{ij} for every $j \in G$ which denotes the spending of agent i on good j, and for every f and every $1 \le k \le K_f$, we will have a variable s_{fk} which denotes the spending of firm f on its input good g_f^{in} on the kth segment.

6.4.1 Membership in PPAD: The proof of Theorem 6.4

We will again develop the proof in three steps, namely (a) construction of the function *F* and arguing that it can be represented by a PL arithmetic circuit containing linear-OPT-gates, (b) showing that the linear-OPT-gate can compute all the necessary components, and (c) arguing that a competitive equilibrium can be recovered from a fixed point of *F*.

The function *F*. We define a function $F: D \to D$ where

$$D = \{ p \in \Delta^{m-1} \colon \forall j \in G, p_j \ge \frac{\varepsilon^m}{m} \} \times [0, CL + 1]^{nm} \times_f [0, L]^{K_f}$$
(34)

An input to F is a tuple $(\mathbf{p}, \mathbf{q}, \mathbf{s})$ of prices, consumer expenditures and firm expenditures for raw goods, and the output is another such tuple $(\bar{\mathbf{p}}, \bar{\mathbf{q}}, \bar{\mathbf{s}})$.

For any firm f, the output spending variable $\bar{\mathbf{s}}_f = (s_{fk})_{1 \leq k \leq K_f}$ is set to be the output of the linear-OPT-gate for the following linear program.

Linear Program \mathcal{P}_{prod}

maximize
$$\sum_{k=1}^{K_f} (c_{fk} p_{g_f^{\text{in}}} - p_{g_f^{\text{out}}}) \cdot v_{fk}$$
 (35) subject to
$$0 \le v_{fk} \le \min\{L_{fk}, L\} \cdot p_{g_f^{\text{in}}}$$

As in the case considered in Section 6.2, we define

$$r_{fk} = s_{fk} + \max\{0, c_{fk} \cdot p_{g_f^{\text{out}}} - p_{g_f^{\text{in}}}\} \cdot \min\{L_{fk}, L\}.$$

Furthermore, we let $s_f = \sum_{k=1}^{K_f} s_{fk}$ and $r_f = \sum_{k=1}^{K_f} r_{fk}$. Then, for every agent i, we let the output expenditure \mathbf{q}_i be the output of the linear-OPT-gate of the following linear program.

Linear Program \mathcal{P}_{con}

minimize
$$\sum_{j \in G} \left(p_j \cdot C_{ij} \left(\frac{q_{ij}}{p_j} \right) + (B+1) \cdot H \left(q_{ij} - \frac{D_{ij}}{B} A p_j \right) \right) \cdot v_{ij}$$
subject to
$$\sum_{j \in G} v_{ij} = \sum_{j \in G} w_{ij} p_j + \sum_{f \in F} \theta_{if} \cdot (r_f - s_f)$$

$$0 \le v_{ij} \le (C \cdot L + 1), \ \forall j \in G$$

$$(36)$$

The first term $p_j \cdot C_{ij}(q_{ij}/p_j)$ of the coefficient in front of v_{ij} is the PL pseudo-circuits from Proposition 3.3 computing the product of p_j and the piecewise constant correspondence represented by (C_{ij}, B) . This term should represent the inverse bang-per-buck ratio of agent i for good j when her spending on the good is q_{ij} . The second term in the coefficient in front of v_{ij} in the objective function only appears if the slope of the PLC function that agent i has for good j eventually becomes zero. This term is meant to ensure that in an optimal solution, an agent does not spend money on a segment where the slope is zero.

Excess Expenditure and the Feasibility Program for the prices. Finally, for every $j \in G$, similarly to Section 6.2 and Section 6.3, we define the *excess expenditure* of good j as

$$e_j = \sum_i q_{ij} + \sum_{f: g_f^{\text{in}} = j} s_f - \sum_{f: g_f^{\text{out}} = j} r_f - p_j.$$

As in previous sections, we let the output price vector \bar{p} be the output of the following feasibility program.

Feasibility Program Q

$$e_{j} < e_{j'} \Rightarrow p_{j} \leq \varepsilon \cdot p_{j'}, \ \forall j, j' \in G$$

$$p_{j} \geq \frac{\varepsilon^{m}}{m}, \ \forall j \in G$$

$$\sum_{j \in G} p_{j} = 1$$
(37)

Computation by the linear-OPT-gate. Next, we argue that the solutions to the linear programs and the feasibility program can be computed by our linear-OPT-gate. The subgradients of the LPs for production and consumption can be computed by PL circuits. The main part of the construction is the PL pseudocircuit from Proposition 3.3, computing the correspondence $p_j \cdot C_{ij}(q_{ij}/p_j)$. For the feasibility program, the arguments that it is of the correct form and that it is solvable are identical to those of Lemma 6.3 (and hence also of Lemma 6.1), noting the updated definition of the expenditure e_j .

Fixed points. Suppose that $(\mathbf{p}, \mathbf{q}, \mathbf{s})$ is a fixed point of F. Let $f \in F$ be some firm. As \mathbf{s}_f is a solution to LP (35), we have that $s_{fk} > 0$ only if $c_{fk} \cdot p_{g_f^{\text{out}}} \ge p_{g_f^{\text{in}}}$. Furthermore, if the inequality is strict, then the firm produces using all of that segment. We conclude that \mathbf{s} is an optimal production vector at prices \mathbf{p} . Next, we prove the following:

Claim 6.5. Let $i \in N$ and let $x_{ij} = q_{ij}/p_j$ for every $j \in G$. Then \mathbf{x}_i is an optimal bundle for agent i at prices \mathbf{p} .

Proof. First we show that if the slope of u_{ij} becomes zero at some point, then $q_{ij} \leq \frac{D_{ij}}{B} \cdot A \cdot p_j$. If this were not the case, then the coefficient in front of v_{ij} would be at least B+1. However, by assumption there is a good j' for which agent i is nonsatiated, and the coefficient in front of $v_{ij'}$ is at most

$$p_{j'} \cdot C_{ij'} \left(\frac{q_{ij'}}{p_{j'}} \right) \le p_{j'} \cdot B \le B < B + 1.$$

Because \mathbf{q}_i is an optimal solution to LP (35), this would then imply that $q_{ij} = 0 \le \frac{D_{ij}}{B} \cdot A \cdot p_j$. This means that any agent i can only spend money on a good j up until the point where the inverse of the bang-per-buck ratio is no longer well-defined. Now the result follows, because all the goods $j \in G$ such that $q_{ij} > 0$ must have the minimal coefficient. From this it follows that agent i has bought the goods in a greedy manner according to the bang-per-buck ratio, meaning that \mathbf{q}_i represents an optimal spending for agent i.

What remains to show is that the market clears, that is, $e_j = 0$ for all $j \in G$. As in previous sections, we can use a standard calculation to show that $\sum_{j \in G} e_j = 0$. Hence, it suffices to argue that $e_j \ge 0$ for all $j \in G$. We define:

- $J = \{j \in G : e_j \le e_{j'} \text{ for all } j' \in G\}$
- $N_I = \{i \in \mathbb{N} : \text{ there exists } j \in J \text{ such that } i \text{ is nonsatiated for good } j\}$

- $F_I = \{ f \in F : \text{ there exists } j \in J \text{ such that } f \text{ is nonsatiated for good } j \}$

Assume by contradiction that $e_j < 0$ for $j \in J$. As $\sum_{j \in G} e_j = 0$, this implies that there must exists a good j' such that $e_{j'} > 0$, meaning that $J \subseteq G$. The contradiction will be obtain via the following two claims, very similar to Claim 6.1 and Claim 6.2. Using an identical reasoning to that of Lemma 6.4, the two claims together will establish that the economy graph $\mathcal{G}_E(\mathcal{M})$ is not strongly connected, violating the corresponding sufficiency condition.

Claim 6.6. If $i \in N_I$ and $j' \in G \setminus J$, it holds that $w_{ij'} = 0$.

Proof. Assume for contradiction that $i \in N_J$ and $j' \in G \setminus J$ and that $w_{ij'} > 0$. As in previous sections, this implies that we can lower bound the budget of agent i by $(CL+1)\sum_{j\in J}p_j$ by picking $\varepsilon \leq \frac{w_{min}}{m\cdot(CL+1)}$. By assumption, i is nonsatiated with respect to some $j \in G$. As $p_j \leq \varepsilon p_{j''}$ for all $j'' \in G \setminus J$, we get by choice of ε that i will spend her entire budget on goods in J. Hence, there is some $j \in J$ such that $q_{ij} \geq (CL+1)p_j$. This implies that

$$e_j \ge (CL+1)p_j - \sum_{f: g_f^{\text{out}} = j} r_f - p_j \tag{38}$$

Now for any f with $g_f^{\text{in}} = j$, we have that $r_f = \sum_{k=1}^{K_f} r_{fk} \le \sum_{k=1}^{K_f} Lc_{fk} p_j \le (K_f \cdot \max_{f \in F} c_{f1}) Lp_j$. By picking $C \ge |F| \cdot \max_f K_f \cdot \max_f c_{f1}$, we obtain the contradiction that $e_j \ge 0$. We conclude that if $i \in N_J$ and $j' \in G \setminus J$, then $w_{ij'} = 0$.

Claim 6.7. *If* $f \in F_J$, then $g_f^{out} \in J$.

Proof. Let $f \in F_J$ and let $j = g_f^{\text{in}}$. Assume for contradiction that $g_f^{\text{out}} \notin J$. Then $p_j \le \varepsilon p_{g_f^{\text{out}}}$, meaning that f will produce fully if $\varepsilon > 0$ is chosen sufficiently small. Hence $s_f \ge s_{fK_f} = Lp_j$, and so

$$0 > e_j \ge (L - 1) \cdot p_j - \sum_{f': g_{f'}^{\text{out}} = j} r_{f'}$$
(39)

If no firm produces j, then we reach a contradiction by choosing $L \geq 1$. The above inequality implies that there must exist a firm f_1 such that $g_{f_1}^{\text{out}} = j$ and $r_{f_1} \geq \frac{L-1}{\ell} p_j$. Let $j_1 = g_{f_1}^{\text{in}}$ and note that $j_1 \in J$. As in previous sections, we may thus bound $\frac{L-1}{\ell} p_j \leq r_{f_1} \leq c_{f_1 1} s_{f_1}$. If any production occurs, then $c_{f_1 1} p_j \geq p_{j_1}$. These two inequalities combine to show that $s_{f_1} \geq \frac{L-1}{\ell \cdot c_{f_1}^2} p_{j_1}$. Using this, we may again bound

$$0 > e_{j_1} \ge \left(\frac{L-1}{\ell \cdot c_{f_1 1}^2} - 1\right) \cdot p_{j_1} - \sum_{f' : g_{f'}^{\text{out}} = j_1} r_{f'}$$

$$\tag{40}$$

Again, if j_1 is not produced, then we may pick L large enough to reach a contradiction. Otherwise we repeat the previous argument. This can happen at most m times, because of the *no assumption out of nothing* assumption.

Beyond SSPLC Utilities?

We conclude the section with two remarks. The first one briefly discusses the challenges of using the machinery of Section 3.4 for SSPLC productions also.

Remark 6.15. The reason for not allowing a succinct representation of the production functions is that this seemingly necessitates multiplying a piecewise-linear function by an input variable of the circuit. A natural

idea would be to have the cumulative sum of conversion rates given succinctly by a boolean circuit C'. However, in order to compute the revenue of firm f given spending s_f , one would then have to compute the piecewise-linear function $p_{g_f^{\text{out}}} \cdot C'(s_f/p_{g_f^{\text{in}}})$ which is something that is not captured by the results in Section 3.4.

The second remark regards the possibility of defining a class of "succinct Leontief-free functions", similarly to the case of SSPLC.

Remark 6.16. One might hope to define a class which captures the case of Leontief-free utility and production functions and the case of SSPLC utility functions and SPLC production functions simultaneously. For the utility functions, one approach would be to consider utility functions u^i given by a finite list K_i of tranches $\{s_k^i\}_{k\in K_i}$. Associated with a tranche s_k^i would again be a number $L_k^i\in\mathbb{R}_{\geq 0}\cup\{\infty\}$ which is an upper bound for the total amount of utility that can be accrued from this tranche. Also, for every good j, instead of having a single number u_{jk}^i which is the rate at which good j provides utility for agent i on tranche k, we would have that j would provide utility for agent i on segment k according to a piecewise-linear concave function U_{jk}^i given to us succinctly in the form of a boolean circuit. Such a class of utility functions would generalize both Leontief-free utility functions and SSPLC utility functions. However, because the rates at which goods provide utility is non-constant, it is unclear how to obtain constraints that mimic the constraint $\sum_{j\in G} U_{ik}^i(x_{jk}^i) \leq L_k^i$ after transforming the variables via Gale's substitution (Remark 6.2).

7 Pacing Equilibria in Auto-Bidding Auctions

In this section, we consider environments which we will broadly refer to as "auto-bidding auctions". In these, a set of items are sold separately to a set of buyers via parallel single-item auctions (e.g., first-price auctions or second-price auctions). To participate in these auctions, each buyer *i scales* her valuations for the items by the same multiplier α_i , which is known as a *pacing multiplier*, under some constraints on her expenditure, typically provided by budgets, or return-on-investment (ROI) thresholds. The objective is to find a *pacing equilibrium* (see Definitions 7.1 and 7.3), i.e., a vector of pacing multipliers and allocations which ensure that the auction is run correctly and that it satisfies the spending constraints of all the buyers simultaneously.

In Section 7.1 we consider the former case, i.e., that of single-item auctions with budget constraints. For the case of first-price auctions, Borgs et al. [2007] showed that a pacing equilibrium always exists and can be computed in polynomial time via a convex program. Later on Conitzer et al. [2022a] showed that in this case the pacing multipliers are unique in every pacing equilibrium, and presented an interesting connection between pacing equilibria and solutions of the well-known Eisenberg-Gale convex program for quasi-linear utilities [Cole et al., 2017]. For second-price auctions, the existence of a pacing equilibrium was established in [Conitzer et al., 2022b]. Chen et al. [2021a] later on proved that the problem of computing a pacing equilibrium in these auctions is in PPAD, which also implied that a solution described by rational numbers also exists. We provide an alternative PPAD-membership result for the same problem. Compared to the proof of Chen et al. [2021a], our proof makes uses of our linear-OPT-gate and is *significantly* simpler. We highlight the main ideas of our proof and how they differ from those of Conitzer et al. [2022b] and Chen et al. [2021a] in Section 7.1.1 below. We remark that Chen et al. [2021a] also showed the problem to be PPAD-hard, thus establishing its PPAD-completeness.

In Section 7.2, we also consider the case of (on average) ROI-constrained buyers, in the model introduced by Li and Tang [2023]. The existence of pacing equilibria for this case was proven by Li and Tang [2023], following a very similar approach to that of Conitzer et al. [2022b]. Our proof follows along very much the same lines as the one we present for the case of budget-constrained buyers in Section 7.1, and is again significantly simpler. Interestingly, it turns out that our linear-OPT-gate cannot be used to obtain PPAD-membership of the problem in this case. Still, using the OPT-gate for FIXP developed by Filos-Ratsikas et al. [2023], we prove membership of the problem in the class FIXP. This is the first membership result for this variant of the problem; we provide more details in Section 7.2.2. In Section 7.2.3 we show that membership

in FIXP (rather than PPAD) is necessary, because pacing equilibria in this case can be irrational. Whether the problem of computing pacing equilibria in second-price auctions for ROI-constrained buyers is actually FIXP-complete is an intriguing open problem. We note that for a notion of approximate pacing equilibria, a PPAD-hardness result is known from [Li and Tang, 2023].

7.1 Pacing Equilibria in Auctions with Budgets

Following Chen et al. [2021a], we will refer to the auction market environment as a *second-price pacing game* (SPPG) \mathcal{G} , consisting of

- a set *N* of buyers, with n = |N|,
- a set *G* of items, with m = |G|,
- a set of vectors of *valuations* $\mathbf{v}_i = (v_{i1}, \dots v_{im})$ for every buyer $i \in N$, where v_{ij} denotes the value of buyer $i \in N$ for item $j \in G$; without loss of generality we assume that for all $j \in G$, there exists some $i \in N$ such that $v_{ij} > 0$,
- a vector of *budgets* $\mathbf{B} = (B_1, \dots, B_n)$, such that $B_i > 0$ for every buyer $i \in N$.

For each buyer $i \in N$, there is a *pacing multiplier* $\alpha_i \in [0,1]$, and let $\alpha = (\alpha_1, \dots, \alpha_n)$ be the vector of those multipliers. Given α , the *bid* of buyer i on item $j \in G$ is given by $\alpha_i \cdot v_{ij}$, i.e., every buyer *scales* their value v_{ij} by the multiplier α_i . The items are sold in parallel second-price auctions, which implies that

- 1. the item is allocated to the buyer with the highest bid, breaking ties using some tie-breaking rule in case there are multiple such buyers. Let $h_i(\alpha) = \max_{i \in N} \alpha_i v_{ij}$ be the highest bid on item $j \in G$.
- 2. the price of item $j \in G$ is the second-highest bid. Let $p_i(\alpha) = \min_{i \in N} \max_{i' \in N, i \neq i'} \alpha_{i'} v_{i'j}$ be that price.

Another way to interpret Condition 1 above is that each one of the (potential) winners achieves a fraction of the item, which can be interpreted as the probability that that buyer is the final winner of the auction for that item. Formally, let $W_j(\alpha) = \{i \in N : \alpha_i v_{ij} = h_j(\alpha)\}$ be the set of possible winners (those with the highest bid) for item $j \in G$. Each buyer $i \in N$ receives an allocation x_{ij} of item $j \in N$, with the restriction that $x_{ij} = 0$ for all $i \in N \setminus W_j(\alpha)$.

Pacing Equilibria. We are now ready to define the notion of a pacing equilibrium. Informally, a pacing equilibrium is a set of pacing multipliers and allocations that satisfy some constraints that guarantee the validity of the auction and that the buyers are in a sense meeting their maximum buying capabilities. Following Conitzer et al. [2022a,b], we formally refer to those equilibria as *second-price pacing equilibria*.

Definition 7.1 (Second-price Pacing Equilibrium (SPPE)). A pair (α, \mathbf{x}) , with $\alpha \in [0, 1]^n$ and $\mathbf{x} \in [0, 1]^{nm}$, is a *second-price pacing equilibrium (SPPE)* of a SPPG \mathcal{G} , if the following conditions hold:

```
I.a. \sum_{i \in N} x_{ij} \le 1, for all j \in G, (feasible allocation)
```

I.b.
$$x_{ij} > 0 \Rightarrow \alpha_i v_{ij} = h_i(\alpha)$$
, for all $i \in N$ and $j \in G$, (only highest bids win)

I.c.
$$h_j(\alpha) > 0 \Rightarrow \sum_{i \in N} x_{ij} = 1$$
, for all $j \in G$, (full allocation of items with positive bids)

I.d.
$$\sum_{j \in G} x_{ij} p_j(\alpha) \leq B_i$$
, for all $i \in N$, (budget constraints)

I.e.
$$\sum_{j \in G} x_{ij} p_j(\alpha) < B_i \Rightarrow \alpha_i = 1$$
, for all $i \in N$, (maximum pacing)

We remark that Definition 7.1 requires the flexibility to choose the way the items are allocated to the possible winners; this is in fact necessary to guarantee the existence of a SPPE for every SPPG. As we mentioned earlier, a SPPE is always guaranteed to exist [Conitzer et al., 2022b; Chen et al., 2021a]. Our PPAD-membership proof also provides an alternative, easier proof of existence. We provide a high-level overview of the proof, as well as a comparison to the existing proofs in the next section below.

7.1.1 Features of our proof

Previous proofs of existence and PPAD-membership. The existence of SPPE in SPPGs was first established by Conitzer et al. [2022b]. Their proof proceeds by defining an (ε, H) -smoothed pacing game, for $\varepsilon > 0$ and H > 0, which is derived from the SPPG \mathcal{G} as follows:

- There is an artificial *reserve bid* of 2ε on all items.
- The set of possible winners $W_j(\alpha)$ (i.e., those buyers $i \in N$ with $x_{ij} > 0$ for item $j \in G$) contains all buyers with *almost* the highest bid (i.e., a bid of at least $h_j(\alpha) \varepsilon$). Each buyer in W_j receives a specific fraction of item j (i.e., x_{ij} is fixed for all $i \in N$ and $j \in N$, as a function of the α_i 's.) and pays a specific price s_{ij} (see [Conitzer et al., 2022b], Definition 3 for the exact allocations and prices).
- There is an artificial item j_a of unlimited supply. Buyer $i \in N$ receives a quantity α_i of that good and values that good at $v_{ij_a} = 2\varepsilon$.
- Every buyer $i \in N$ has a utility which is a step function depending on the budget B_i , the price s_{ij} she pays for quantities of item j, α_i , ε and the parameter H in the definition of the (ε, H) -smoothed pacing game (see [Conitzer et al., 2022b], Definition 3 for the exact utility function).

The reason for introducing all these seemingly convoluted conditions is that they ensure that the (ε, H) -smoothed pacing game satisfies a set of desired properties, namely (a) compactness and convexity of the strategy space, (b) continuity of the utility function in all strategies and (c) quasi-concavity of the utility function in a buyer's own strategy. With these at hand, one can then invoke the Debreu-Fan-Glicksberg theorem for continuous games [1952], which we presented in Section 4.3.1, to establish the existence of a Nash equilibrium.

The proof of Conitzer et al. [2022b] proceeds by proving Properties (a-c) above, and then obtains the existence of a Nash equilibrium for the (ε, H) -smoothed pacing game. To obtain the existence of a SPPE, they consider a sequence of Nash equilibria for (ε, H) -smoothed pacing games, and show that as ε converges to 0 and H converge to infinity, a SPPE is obtained as a limit point.

Besides the overhead of defining (ε, H) -smoothed pacing games and establishing their properties, the proof uses off-the-shelf existence theorems and limit arguments. Therefore, it is not surprising that it cannot be used to obtain PPAD-membership, or even to establish that rational solutions are always possible. Both of these properties were established via an alternative proof provided by Chen et al. [2021a], which we describe briefly next.

Similarly in spirit to Conitzer et al. [2022b], Chen et al. [2021a] define an *approximate* version of a SPPE, called a (δ, γ) -SPPE, with $\delta, \gamma > 0$, in which

- Condition I.b. in Definition 7.1 has been substituted by $x_{ij} > 0 \Rightarrow a_i v_{ij} = (1 \delta)h_j(\alpha)$, for all $i \in N$ and all $j \in G$.
- Condition I.e. in Definition 7.1 has been substituted by $\sum_{j \in G} x_{ij} p_j(\alpha) < (1 \gamma) B_i \Rightarrow \alpha_i \geq (1 \gamma)$, for all $i \in N$.

In other words, they consider an approximate SPPE in which the set of winners $W_j(\alpha)$ contains buyers that have approximately the highest bid, and in which a buyer that does not spend almost all of her budget uses almost a maximum pacing multiplier. On top of that, they define a variant of (δ, γ) -SPPE, which they refer to as smooth (δ, γ) -SPPE in which the allocation rule is specified, and is in fact similar to that used by Conitzer et al. [2022b] in their definition of a (ε, H) -smoothed pacing game defined above. By definition, a smooth- (δ, γ) -SPPE is (δ, γ) -SPPE. Their PPAD-membership proof then proceeds in three steps:

- They show the PPAD-membership of smooth (δ, γ) -SPPE via a reduction to the computational version of Sperner's lemma [Sperner, 1928]. This implies the existence of a (δ, γ) -SPPE.
- They devise an intricate rounding procedure, which converts (δ, γ) -SPPE into γ -SPPE, i.e., (δ, γ) -SPPE for which $\delta = 0$, and establish the correctness of this procedure via a series of arguments.

- They employ a general technique (introduced by Etessami and Yannakakis [2010]) of converting approximate solutions of inversely exponential precision to exact solutions via a carefully constructed linear program. They use this to convert a γ -SPPE into a SPPE.

Our proof. Our proof is markedly different from the aforementioned ones, and conceptually much simpler, while at the same time obtaining both existence and PPAD-membership, and as a result establishing the existence of SPPE described by rational numbers. Contrary to Conitzer et al. [2022b] and Chen et al. [2021a], we do not need to define relaxations of the SPPE. Instead, we apply Gale's substitution (see Remark 6.2), i.e., the standard change of variables from allocations x_{ij} to expenditures $q_{ij} = x_{ij}p_j(\alpha)$, see Section 7.1.2 below.¹⁸ This is in fact the very same change of variables that we used in Section 6 for competitive markets, and serves as a means to escape having to deal with quantities of the form $x_{ij}p_j(\alpha)$, as such multiplications cannot be handled by PL arithmetic circuits.

Membership in PPAD then follows from constructing a simple function F whose fixed points will be the SPPE. For each buyer $i \in N$, the expenditures q_{ij} will be obtained as the solutions to a linear program which can be handled by our linear-OPT-gate, and the pacing multipliers will be the solutions to a single equation involving the inputs and the outputs of the function. Arguing that a fixed point of F indeed corresponds to a SPPE reduces to establishing that Conditions I.a. to I.e. of Definition 7.1 are satisfied, which turns out to be rather simple.

7.1.2 Preprocessing

Before we present our proof, we will first modify and simplify the constraints of Definition 7.1, to make them amenable to the use of our linear-OPT-gate. En route to that, we will prove that without loss of generality, we may assume that in any SPPE (α, \mathbf{x}) , both the pacing multipliers α_i and the prices $p_j(\alpha)$ can be assumed to be bounded away from 0, which we will use later on.

Lower bounds on the pacing multipliers and prices. Below we prove simple lower bounds on the value of the pacing multipliers α_i^* and the prices $p_i(\alpha^*)$ in any SPPE (α^*, \mathbf{x}^*) .

Lemma 7.1. Let (α^*, \mathbf{x}^*) be any SPPE. Then, for any $i \in N$, we have that $\alpha_i^* \in [\ell_i, 1]$, where $\ell_i = \min\{1, B_i / \sum_{i \in G} v_{ij}\}$.

Proof. If $\alpha^* = 1$ then the lemma holds trivially. Otherwise, by Condition I.e., we have that

$$B_i = \sum_{j \in G} x_{ij}^* p_j(\alpha^*) \le \sum_{j \in G: x_{ij}^* > 0} h_j(\alpha^*) = \sum_{j \in G: x_{ij}^* > 0} \alpha_i^* v_{ij} \le \alpha_i^* \sum_{j \in G} v_{ij},$$

where the first inequality above holds by the fact that $x_{ij}^*p_j(\alpha^*) \le x_{ij}^*h_j(\alpha^*) \le h_j(\alpha^*)$, since $h_j(\alpha^*) \ge p_j(\alpha^*)$ by the definition of the auction, and the second equation holds by Condition I.b..

Lemma 7.2. Let (α^*, \mathbf{x}^*) be any SPPE. Then, for any $j \in G$, we may assume without loss of generality that $p_i(\alpha^*) > 0$.

Proof. Assume that there is some item j' for which $p'_j(\alpha^*) = 0$. For any choice of pacing multipliers α^* , i.e., in any PE, we may choose an arbitrary allocation of the good j' for which $x_{ij'} > 0$ only for buyers $i \in W_j(\alpha)$. This way, Conditions I.b. and I.c. of Definition 7.1 are satisfied. Since $p'_j(\alpha^*) = 0$, the term for j' contributes 0 to the sum $\sum_{j \in G} x^*_{ij} p_j(\alpha^*)$ for every buyer $i \in N$, and hence Conditions I.d. and I.e. are unaffected. Condition I.a. is trivially satisfied as well. Therefore, given any SPPE $(\alpha^*, \mathbf{x}^*_{-j})$ for items $j \in G \setminus \{j'\}$, we get the same SPPE, together with item j' allocated as described above.

¹⁸Interestingly, Conitzer et al. [2022a] also apply this change of variables in their Mixed-Integer Linear Programming (MILP) formulation (see Section 6.3 in their paper), but not in the existence proof.

Change of variables. Looking at the conditions of Definition 7.1, we observe that in Condition I.d., the allocation x_{ij} is multiplied by the price $p_j(\alpha)$. Looking ahead, this condition will appear as a constraint in a linear program \mathcal{P} that we will aim to solve via our linear-OPT-gate, as part of a PL arithmetic circuit. This linear program would have x_{ij} as the variables, but $p_j(\alpha)$ would be gate inputs in the context of the linear-OPT-gate. Hence, such a constraint could not be handled by the linear-OPT-gate. To remedy this, we work in a very similar fashion as we did in Section 6, via applying Gale's substitution (Remark 6.2). Namely, we will let $q_{ij} = x_{ij}p_j(\alpha)$ be the *expenditure* of buyer $i \in N$ on item $j \in G$. Additionally, Lemma 7.1 implies that $h_j(\alpha^*) > 0$ for all items $j \in G$. In turn, this implies that Conditions I.a. and I.c. can be combined into a single condition, namely

 $\sum_{i \in N} x_{ij} = 1, \text{ for all } j \in G.$

Putting everything together, we have the following equivalent definition of a *second-price pacing expenditure equilibrium* (SPPEE), from which a SPPE can be straightforwardly recovered.

Definition 7.2 (Second-price Pacing Expenditure Equilibrium (SPPEE)). A pair (α, \mathbf{q}) , with $\alpha \in [0, 1]^n$ and $\mathbf{q} \in [0, 1]^{nm}$, is a *second-price pacing expenditure equilibrium (SPPEE)* of an SPPG \mathcal{G} , if the following conditions hold:

II.a. $\sum_{i \in N} q_{ij} = p_j(\alpha)$, for all $j \in G$, (feasible expenditure)

II.b. $q_{ij} > 0 \Rightarrow \alpha_i v_{ij} = h_j(\alpha)$, for all $i \in N$ and $j \in G$, (only highest bids spend)

II.c. $\sum_{j \in G} q_{ij}(\alpha) \leq B_i$, for all $i \in N$, (budget constraints)

II.d. $\sum_{j \in G} q_{ij} < B_i \Rightarrow \alpha_i = 1$, for all $i \in N$. (maximum pacing)

7.1.3 Membership in PPAD

We are now ready to prove the main result of the section.

Theorem 7.1. Computing a SPPE of any SPPG G is in PPAD.

Following our standard methodology, we will develop the proof in three steps, namely (a) construction of the function *F* and arguing that it can be computed by a PL arithmetic circuit containing linear-OPT-gates, (b) showing that the linear-OPT-gate can compute all the necessary components, and (c) arguing that a SPPE can be recovered from a fixed point of *F*.

The function F. We define the function $F: D \to D$ with domain $D = \prod_{i \in N} [l_i, 1] \times \prod_{i \in N} \prod_{j \in G} [0, v_{ij}]$. Let (α, \mathbf{q}) and (α^*, \mathbf{q}^*) denote inputs and outputs to F respectively. For any item $j \in G$, the vector of expenditures $\mathbf{q}_i^* = (q_{1j}^*, \dots, q_{nj}^*)$ is obtained as the output of the following linear program.

Linear Program \mathcal{P}

maximize
$$\sum_{i \in N} (\alpha_i v_{ij}) q_{ij}$$

subject to $\sum_{i \in N} q_{ij} \le p_j(\alpha)$
 $q_{ij} \ge 0, \ \forall i \in N$

The pacing multipliers will be obtained via the following equation

$$\alpha_i^* = \max\left\{\frac{B_i}{\sum_{j \in G} v_{ij}}, \min\left\{1, \alpha_i + B_i - \sum_{j \in G} q_{ij}\right\}\right\}$$
(41)

Equation (41) uses max, min and addition operations on the variables, and hence can be computed by a PL arithmetic circuit. Similarly, recall that $p_j(\alpha) = \min_{i \in N} \max_{i' \in N, i \neq i'} \alpha_{i'} v_{i'j}$, and hence the linear program \mathcal{P} can also be computed by a PL arithmetic circuit.

Computation by the linear-OPT-gate. In the next lemma, we argue that the output \mathbf{q}_{j}^{*} of the linear program \mathcal{P} can be computed by the linear-OPT-gate.

Lemma 7.3. Consider the linear program \mathcal{P} . An optimal solution to \mathcal{P} can be computed by a linear-OPT-gate.

Proof. We will argue that \mathcal{P} satisfies all the properties required for the linear-OPT-gate, as highlighted in Section 3.2. The feasible domain $[0,1]^n$ of \mathcal{P} is non-empty and bounded, and the gate inputs $p_j(\alpha)$ appear only on the right-hand side of the constraints. The subgradient of the objective function is a linear function, and hence can be computed by a PL pseudo-circuit.

Fixed points lead to SPPE. We will show that a fixed point (α^*, \mathbf{q}^*) of F is a SPPEE (Definition 7.2); a SPPE can then be straightforwardly recovered by setting $x_{ij} = q_{ij}/p_j(\alpha^*)$ for all $i \in N$ and $j \in G$, since all the prices can be assumed to be non-zero by Lemma 7.2. In particular, we show that $(\alpha, \hat{\mathbf{q}}) = (\alpha^*, \mathbf{q}^*)$ satisfies Conditions II.a. to II.d. of Definition 7.2.

- 1. Let $j \in G$ be any item. Since \mathbf{q}_j is an optimal solution to linear program \mathcal{P} , and since $\alpha_i v_{ij}$ is positive for all $i \in N$, the "payment" constraint of \mathcal{P} must be tight, i.e., $\sum_{i \in N} q_{ij}^* = p_j(\alpha) = p_j(\alpha^*)$. Hence Condition II.a. is satisfied.
- 2. Consider any buyer $i \in N$ and item $j \in N$ such that $q_{ij}^* > 0$. By the optimality of \mathbf{q}_j^* , $\alpha_i v_{ij}$ must be maximum, i.e., $\alpha_i v_{ij} = \alpha_i^* \cdot v_{ij} = h_i(\alpha^*)$. Hence Condition II.b. is satisfied.
- 3. Let $i \in N$ be any buyer. Assume by contradiction that Condition II.c. is violated, i.e., that $B_i < \sum_{i \in N} q_{ij}^*$. Since $\alpha_i = \alpha_i^*$, Equation (41) implies that $\alpha_i^* = B_i / \sum_{j \in G} v_{ij}$. However, since \mathbf{q}_j^* is a feasible solution to the linear program \mathcal{P} , we have

$$\begin{split} \sum_{j \in G} q_{ij}^* &= \sum_{j \in G: q_{ij}^* > 0} q_{ij}^* = \sum_{j \in G: \alpha_i^* v_{ij} = h_j(\alpha^*)} q_{ij}^* \leq \sum_{j \in G: \alpha_i^* v_{ij} = h_j(\alpha^*)} p_j(\alpha^*) \\ &\leq \sum_{j \in G: \alpha_i^* v_{ij} = h_j(\alpha^*)} h_j(\alpha^*) = \sum_{j \in G: \alpha_i^* v_{ij} = h_j(\alpha^*)} \alpha_i^* v_{ij} \leq \sum_{j \in G} \alpha_i^* v_{ij} \\ &= \frac{B_i}{\sum_{j \in G} v_{ij}} \sum_{j \in G} v_{ij} = B_i, \end{split}$$

where in the calculations above:

- the second equation follows from Condition II.b., which we we established in Item 2 above,
- the first inequality follows from the "payment" constraints of the linear program \mathcal{P} , and
- the third inequality follows from the fact that $h_i(\alpha^*) \ge p_i(\alpha^*)$ by definition.

We have reached a contradiction.

4. Consider some buyer $i \in N$ such that $\sum_{j \in G} q_{ij}^* < B_i$. Since $\alpha_i = \alpha_i^*$, Equation (41) in that case implies that $\alpha_i^* = 1$, and hence Condition II.d. is satisfied.

This completes the proof.

7.2 Pacing Equilibria in ROI-constrained Auctions

We now turn our attention to market environments where the total expenditure of each bidder is not constrained by a fixed budget, but by a constraint on (average) return-on-investement (ROI), in the setting proposed by Li and Tang [2023]. For consistency with Section 7.1, we will refer to these environments as second-price pacing ROI games (SPPRG). A SPPRG \mathcal{G} consists of

- a set *N* of buyers, with n = |N|,
- a set *G* of items, with m = |G|,
- a set of vectors of *valuations* $\mathbf{v}_i = (v_{i1}, \dots v_{im})$ for every buyer $i \in N$, where v_{ij} denotes the value of buyer $i \in N$ for item $j \in G$; without loss of generality we assume that for all $j \in G$, there exists some $i \in N$ such that $v_{ii} > 0$.

Similarly to Section 7.1, for each buyer $i \in N$ there is a pacing multiplier α_i and the buyer's bid on item j is $\alpha_i \cdot v_{ij}$. For this setting to be meaningful, following [Li and Tang, 2023], we will not constrain the pacing multipliers to lie in [0,1], but in [1,A] for some constant $A \in \mathbb{R}$, $A \ge 1$. Again, the items are sold in parallel second-price auctions, which implies that

- 1. the item is allocated to the buyer with the highest bid, breaking ties using some tie-breaking rule in case there are multiple such buyers. Let $h_i(\alpha) = \max_{i \in N} \alpha_i v_{ij}$ be the highest bid on item $j \in G$.
- 2. the price of item $j \in G$ is the second-highest bid. Let $p_i(\alpha) = \min_{i \in N} \max_{i' \in N, i \neq i'} \alpha_{i'} v_{i'j}$ be that price.

Again, let $W_j(\alpha) = \{i \in N : \alpha_i v_{ij} = h_j(\alpha)\}$ be the set of possible winners (those with the highest bid) for item $j \in G$. Each buyer $i \in N$ receives an allocation x_{ij} of item $j \in N$, with the restriction that $x_{ij} = 0$ for all $i \in N \setminus W_j(\alpha)$.

ROI Pacing Equilibria. Next, we define the notion of a second-price ROI pacing equilibrium.¹⁹ Note that the only significant difference between Definition 7.3 and Definition 7.1 of Section 7.1 is that the budget constraints have now been replaced by ROI constraints.

Definition 7.3 (Second-price ROI Pacing Equilibrium (SPRPE)). A pair (α, \mathbf{x}) , with $\alpha \in [1, A]^n$ and $\mathbf{x} \in [0, 1]^{nm}$, is a *second-price ROI pacing equilibrium (SPRPE)* of an SPPRG \mathcal{G} , if the following conditions hold:

III.a. $\sum_{i \in N} x_{ij} \le 1$, for all $j \in G$,

(feasible allocation)

III.b. $x_{ij} > 0 \Rightarrow \alpha_i v_{ij} = h_j(\alpha)$, for all $i \in N$ and $j \in G$,

(only highest bids win)

III.c. $h_i(\alpha) > 0 \Rightarrow \sum_{i \in N} x_{ij} = 1$, for all $j \in G$,

(full allocation of items with positive bids)

III.d. $\sum_{i \in G} x_{ij} p_i(\alpha) \leq \sum_{i \in G} x_{ij} v_{ij}$, for all $i \in N$,

(ROI constraints)

III.e.
$$\sum_{i \in G} x_{ij} p_i(\alpha) < \sum_{i \in G} x_{ij} v_{ij} \Rightarrow \alpha_i = A$$
, for all $i \in N$.

(maximum pacing)

Li and Tang [2023] established that a SPRPE always exists. Their proof follows closely that of Conitzer et al. [2022b] for second-price pacing equilibria in auctions with budgets, which we referred to in Section 7.1.1. In particular, they also define a very similar (ε, H) -smoothed game that satisfies the properties required for the Debreu-Fan-Glicksberg theorem [1952], and recover a SPRPE as a limit point of the sequence of Nash equilibria of this game. The proof that we present in this section follows very much along the same lines of the proof that we presented in Section 7.1 and is thus conceptually much simpler than that of Li and Tang [2023]. In addition, since it does not exhibit the discontinuous arguments of the proof of Li and Tang [2023], it can also be used to show FIXP-membership of the problem, as we highlight in Section 7.2.2.

Since $\alpha_i^* \ge 1$ for any $i \in N$, and since for every $j \in G$, there exists some buyer $i \in N$ with $v_{ij} > 0$, we can again merge Conditions III.a. and III.c. above and obtain the following equivalent definition for a SPRPE.

Definition 7.4 (Second-price ROI Pacing Equilibrium (SPRPE)-simplified). A pair (α, \mathbf{x}) , with $\alpha \in [1, A]^n$ and $\mathbf{x} \in [0, 1]^{nm}$, is a *second-price ROI pacing equilibrium (SPRPE)* of an SPPRG \mathcal{G} , if the following conditions hold:

¹⁹Li and Tang [2023] refer to these equilibria as "auto-bidding equilibria"; we choose the term "second-price ROI pacing equilibria" for consistency with the results of Section 7.1.

IV.a. $\sum_{i \in N} x_{ij} = 1$, for all $j \in G$, (feasible allocation) IV.b. $x_{ij} > 0 \Rightarrow \alpha_i v_{ij} = h_j(\alpha)$, for all $i \in N$ and $j \in G$, (only highest bids win) IV.c. $\sum_{j \in G} x_{ij} p_j(\alpha) \leq \sum_{j \in G} x_{ij} v_{ij}$, for all $i \in N$, (ROI constraints) IV.d. $\sum_{j \in G} x_{ij} p_j(\alpha) < \sum_{j \in G} x_{ij} v_{ij} \Rightarrow \alpha_i = A$, for all $i \in N$. (maximum pacing)

7.2.1 A new proof of existence

We state the main theorem of this section.

Theorem 7.2. A SPRPE exists for any SPPRG G.

Proof. We define the function $F: D \to D$ with domain $D = \prod_{i \in N} [1, A] \times \prod_{i \in N} \prod_{j \in G} [0, 1]$. Let (α, \mathbf{x}) and (α^*, \mathbf{x}^*) denote inputs and outputs to F respectively. For any item $j \in G$, the vector of allocations $\mathbf{x}_i^* = (x_{1i}^*, \dots, x_{ni}^*)$ is obtained as the output of the following linear program.

Linear Program \mathcal{P}

$$\begin{aligned} & \text{maximize } \sum_{i \in N} (\alpha_i v_{ij}) x_{ij} \\ & \text{subject to } \sum_{i \in N} x_{ij} = 1, \ \forall i \in N \\ & x_{ij} \geq 0, \ \forall i \in N \end{aligned}$$

The pacing multipliers will be obtained via the following equation

$$\alpha_i^* = \max \left\{ 1, \min \left\{ A, \alpha_i + \sum_{i \in G} x_{ij} v_{ij} - \sum_{i \in G} x_{ij} p_j(\alpha_i) \right\} \right\}$$
(42)

We will show that a fixed point (α^*, \mathbf{x}^*) of F is a RPE. In particular, we show that $(\alpha, \mathbf{x}) = (\alpha^*, \mathbf{x}^*)$ satisfies Conditions IV.a. to IV.d. of Definition 7.4.

- 1. Let $j \in G$ be any item. Since \mathbf{x}_j is an optimal solution to linear program \mathcal{P} , and since $\alpha_i v_{ij}$ is positive for all $i \in N$, the allocation constraint of \mathcal{P} must be tight, i.e., $\sum_{i \in N} x_{ij}^* = 1$. Hence Condition IV.a. is satisfied.
- 2. Consider any buyer $i \in N$ and item $j \in N$ such that $x_{ij}^* > 0$. By the optimality of \mathbf{x}_j^* , $\alpha_i v_{ij}$ must be maximum, i.e., $\alpha_i v_{ij} = \alpha_i^* \cdot v_{ij} = h_i(\alpha^*)$. Hence Condition IV.b. is satisfied.
- 3. Let $i \in N$ be any buyer. Assume by contradiction that Condition IV.c. is violated, i.e., that $\sum_{j \in G} x_{ij}^* v_{ij} < \sum_{i \in N} x_{ij}^* p_j(\alpha^*)$. Since $\alpha_i = \alpha_i^*$, Equation (42) implies that $\alpha_i^* = 1$. In turn, this implies that for any item $j \in G$, we have $v_{ij} = \alpha_i^* v_{ij}$. Consider any such an item $j' \in G$ for which $x_{ij'}^* > 0$. By Condition IV.b. which was established in Item 2 above, we have that $\alpha_{ij'}^* v_{ij'} = h_j'(\alpha^*)$. Combining the above two expressions, and the fact that $h_j'(\alpha^*) \geq p_j'(\alpha^*)$, we obtain that $v_{ij'} \geq p_j'(\alpha^*)$. By summing over all items $j \in G$ with $x_{ij} > 0$, we obtain a contradiction.
- 4. Consider some buyer $i \in N$ such that $\sum_{j \in G} x_{ij}^* p_j(\alpha^*) < \sum_{j \in G} x_{ij}^* v_{ij}$. Since $\alpha_i = \alpha_i^*$, Equation (42) in that case implies that $\alpha_i^* = A$, and hence Condition III.d. is satisfied.

This completes the proof. \Box

7.2.2 Membership in the class FIXP

The main technical difference between our proof in Section 7.1 and that in Section 7.2 is that in the latter case, we did not perform a variable change. Indeed, the nature of the ROI constraints in Condition IV.c. prevents us from doing that, as the variables x_{ij} appear in both sides of the constraints, once multiplied by the constants v_{ij} and once with the gate inputs $p_j(\alpha)$. The problem with having expressions of the form $\sum_{j \in G} x_{ij} p_j(\alpha)$ is that we are not allowed to multiply two parameters which are both inputs to a PL arithmetic circuit. We can do that however with (general) arithmetic circuits, see Definition 2.2.

As we discussed in Section 2, computation of fixed points via (general) arithmetic circuits corresponds to the class FIXP [Etessami and Yannakakis, 2010], which allows for computation of solutions that are described by irrational numbers. Here we can make use of the OPT-gate for FIXP that was designed by Filos-Ratsikas et al. [2023], in order to show FIXP-membership of the problem. We have the following theorem.

Theorem 7.3. Computing a SPRPE of any SPPRG G is in FIXP.

Proof. To turn the existence proof of Theorem 7.2 to a FIXP-membership proof, we have to

- argue that the function *F* that we defined in Section 7.2.1 can be computed by an arithmetic circuit,
- argue that the OPT-gate for FIXP of [Filos-Ratsikas et al., 2023] can solve the linear program \mathcal{P} of Section 7.2.1.

The first part follows by observing that all the parameters of the linear program \mathcal{P} and those of Equation (41) can be computed using the standard operations of the circuit. Note that compared to the corresponding argument in Section 7.1, here we indeed need the capability to multiply input variables, in order to compute α_i^* in Equation (41). Linear program \mathcal{P} is very simple and can be easily seen to satisfy all the properties required for the OPT-gate for FIXP to work, stated in [Filos-Ratsikas et al., 2023].

7.2.3 Limits of rationality: second-price ROI pacing equilibria can be irrational

To round off the section, it remains to determine whether our FIXP-membership result for SPRPE, rather than a PPAD-membership result, is due to our proof technique falling short, or whether there is some inherent reason for this. As we show below, there are simple examples of SPRPGs $\mathcal G$ for which all SPRPE are irrational and hence a PPAD-membership result is not possible.

Example 7.1 (Example of a SPRPG \mathcal{G} that only has irrational SPRPE). Consider a SPRPG \mathcal{G} with 2 buyers and 3 items, with $\mathbf{v}_1 = (1,0,1)$ and $\mathbf{v}_2 = (0,1,2)$. Let A be sufficiently large, e.g., A > 4; the proof can easily be adapted for other values of A by appropriate rescaling of the parameters. Via a case analysis, we show that in any SPRPE, the pacing multipliers α_i for $i \in \{1,2\}$ are irrational.

Let (α, \mathbf{x}) be a SPRPE. Clearly,

- buyer 1 will be allocated the entire quantity of item 1, i.e., $x_{11} = 1$, at price $p_1(\alpha) = 0$,
- buyer 2 will be allocated the entire quantity of item 2, i.e., $x_{22} = 1$, at price $p_2(\alpha) = 0$.

Case 1: $\alpha_1 < 2\alpha_2$. In this case, buyer 2 wins the entirety of item 3 at price α_1 , thus we have $p_3(\alpha) = \alpha_1$, $\alpha_{13} = 0$ and $\alpha_{23} = 1$. By the ROI constraint (Condition III.d. of Definition 7.3) for buyer 2 it holds that

$$\alpha_1 = p_3(\alpha) \cdot x_{23} \le 1 + 2 \cdot x_{23} = 3$$

However, we have that

$$x_{11} \cdot p_1(\alpha) = 0 < 1 = v_{11} \cdot x_{11} \tag{43}$$

By the maximum pacing constraint (Condition III.e. of Definition 7.3) for buyer 1, it follows that $\alpha_1 = A$. Hence $A = \alpha_1 \le 3$, which is not possible by the choice of A. This implies that (α, \mathbf{x}) is not an SPRPE, a contradiction.

Case 2: $\alpha_1 > 2\alpha_2$. In this case, buyer 1 wins the entirety of item 3 at price $2\alpha_2$, thus we have $p_3(\alpha) = 2\alpha_2$, $x_{13} = 1$ and $x_{23} = 0$. By the ROI constraint (Condition III.d. of Definition 7.3) for buyer 1 it holds that

$$2\alpha_2 \cdot x_{13} = p_3(\alpha) \cdot x_{13} \le 1 + 1 \cdot x_{13} = 2 \tag{44}$$

and hence $\alpha_2 \le 1$, which implies that $\alpha_2 = 1$, since $\alpha_2 \in [1, A]$. Therefore, we have that $\alpha_2 \cdot v_{21} = 1$, whereas $\sum_{j \in \{1,2,3\}} x_{2j} \cdot p_j(\alpha) = 0$. By the maximum pacing constraint (Condition III.e. of Definition 7.3) for buyer 2, it follows that $\alpha_2 = A \le 1$, which is not possible since A > 1. This implies that (α, \mathbf{x}) is not an SPRPE, a contradiction.

Case 3: $\alpha_1 = 2\alpha_2$. In this case, both buyers 1 and 2 submit the same bid, and hence they are both eligible to receive positive quantities of item 3. Since the first- and second-highest bids coincide, the price of the item is $p_3(\alpha) = \alpha_1 = 2\alpha_2 \ge 2$, since $\alpha_i \in [1, A]$ for $i \in \{1, 2\}$. By the ROI constraint (Condition III.d. of Definition 7.3) for both buyers, we have

$$p_3(\alpha) \cdot x_{13} \le 1 + x_{13} \tag{45}$$

$$p_3(\alpha) \cdot x_{23} \le 1 + 2x_{23} \tag{46}$$

Next, we consider whether these inequalities can be strict or not. First, if the second inequality is strict, then by the maximum pacing constraint (Condition III.e. of Definition 7.3) for buyer 2, we have that $\alpha_2 = A$. However, then $\alpha_1 = 2\alpha_2 = 2A$, which contradicts the fact that $\alpha_1 \in [1, A]$. This implies that the second inequality is in fact an equality, as otherwise (α, \mathbf{x}) is not an SPRPE.

Now suppose that the first inequality is strict. By the maximum pacing constraint (Condition III.e. of Definition 7.3) for buyer 1, we have that $\alpha_1 = A$ and hence $p_3(\alpha) = A$. Now adding the two inequalities together we have

$$A = \alpha_1 = p_3(\alpha) \cdot (x_{13} + x_{23}) < 2 + x_{13} + 2x_{23} = 3 + x_{23} \le 4 \tag{47}$$

which is a contradiction since A > 4. This implies that the first inequality is also in fact an equality, as otherwise (α, \mathbf{x}) is not an SPRPE.

We are now left with two equations, namely that

$$x_{12} = \frac{1}{p_3(\alpha) - 1}$$
 and $x_{23} = \frac{1}{p_3(\alpha) - 2}$

Adding these together, we have that

$$1 = x_{13} + x_{23} = \frac{1}{p_3(\alpha) - 1} + \frac{1}{p_3(\alpha) - 2}$$

Multiplying through by the denominators, we obtain the quadratic equation $p_3(\alpha)^2 - 5p_3(\alpha) + 5 = 0$ which has only irrational solutions $(5 \pm \sqrt{5})/2$. Since $p_3(\alpha) \ge 2$, this means that $p_3(\alpha) = (5 + \sqrt{5})/2$ is the unique solution corresponding to a SPRPE. We conclude that $\alpha_1 = p_3(\alpha)$ and $\alpha_2 = p_3(\alpha)/2$ are irrational.

8 Fair Division

In this section we consider applications of our linear-OPT-gate to fair division. In particular, we will mainly consider two rather fundamental settings, namely *envy-free cake cutting* and *rental harmony*. The former is one of the prototypical problems in fair division [Gamow and Stern, 1958], the origins of which date back to the late 1940s and the work of Steinhaus [1949], and concerns the fair division of a single infinitely divisible resource among a set of agents with heterogeneous preferences over parts of the resource. The

latter was studied famously by Su [1999], who credits its origins to the chore division problem of Gardner [1978], and is concerned with the fair allocation or rooms to tenants, taking into account their preferences over rooms and the rent for each room.

The PPAD-membership for *approximate* envy-free cake cutting has been known since the work of Deng et al. [2012], and has been hinted at by the existence proof of Simmons [Su, 1999] that in fact goes via an approximate version and applies Sperner's Lemma [Sperner, 1928]. Filos-Ratsikas et al. [2023] recently showed that finding an exact envy-free division is in FIXP in general. Still, there are cases for which exact solutions are rational (e.g., when the preferences are captured by piecewise-constant densities), and for those FIXP is not the appropriate class. Our main theorem in Section 8.1 below extends the ideas of Filos-Ratsikas et al. to show the desired PPAD-membership result for these cases. We remark that a FIXP-hardness result is also known from [Filos-Ratsikas et al., 2023], as well as a PPAD-hardness result from [Deng et al., 2012] for approximate divisions, but those only concern very general versions of the problem and leave much room for improvement.

For rental harmony, there were no known complexity results before our work, to the best of our knowledge. It turns out that our linear-OPT-gate allows us to obtain the PPAD-membership of finding fair partitions using very much the same approach as for the case of envy-free cake cutting, again for these cases for which partitions in rational numbers exist. For more general settings (for which all partitions might be irrational), we complement our results with a FIXP-membership proof, based on the OPT-gate for FIXP of [Filos-Ratsikas et al., 2023].

We remark that very recently Caragiannis et al. [2023] already used the linear-OPT-gate (which we made them aware of via personal communications) to establish that computing envy-free and Pareto-optimal allocations of multiple divisible goods is in PPAD.

8.1 Envy-free cake cutting

We start from arguably the most fundamental fair division problem, that of *envy-free cake cutting*.

Definition 8.1 ((Contiguous) envy-free cake cutting). Let the interval [0,1] be called a *cake*, which is to be divided into n subintervals (pieces) using n-1 cuts. Let $\mathbf{x} \in \Delta^{n-1}$ denote a division of the cake, i.e., each division is a point in the simplex, and let x_j be the j'th coordinate (or "the j'th piece"). There is a set N of n agents, and for each piece j, each agent $i \in N$ has a *valuation function* $u_{ij} : \Delta^{n-1} \to \mathbb{R}_{\geq 0}$, assigning a real number to a division of the cake. Given a division \mathbf{x} , we will say that agent $i \in N$ *prefers* the j-th piece if $u_{ij}(\mathbf{x}) \geq u_{ij'}(\mathbf{x})$ for any piece j'. A division \mathbf{x} is envy-free if there exists a permutation π of $\{1, \ldots, n\}$ such that for every $i \in N$, agent i prefers piece $\pi(i)$.

We offer the following remarks regarding Definition 8.1:

- The definition considers the *contiguous* version of the problem, where the pieces are single intervals. In the more general version of the problem, each piece can be a collection of possibly disconnected intervals. Since a contiguous division is clearly a division, the PPAD-membership of computing contiguous divisions is a stronger result.
- The definition above is very general, in the sense that the valuations of the agents for the pieces do not only depend on the pieces themselves, but they could depend on the whole division **x**, e.g., on how the remaining pieces have been allocated. This captures for example valuations that exhibit externalities. In contrast, several textbooks on the problem (e.g., see [Robertson and Webb, 1998; Brams and Taylor, 1996; Procaccia, 2013]) often consider the problem where the valuation of an agent only depends on her own piece, and in fact these valuations are captured by additive measures over the cake (i.e., the value of an agent for the union of two intervals is the sum of her value for each interval). Again, with regard to PPAD-membership, the more general setting that we consider here makes the result stronger.

Sufficiency condition: Hungriness. We will say that an agent $i \in N$ is *hungry*, if she prefers a non-empty piece of cake to an empty piece. An instance of the envy-free cake cutting problem satisfies the hungriness condition if all of the agents are hungry.

Known results for existence and complexity. The existence of envy-free cake cutting divisions (in the sense of Definition 8.1) was proven independently by Woodall [1980] and Simmons (credited in [Su, 1999]), under the hungriness condition. Both of these proofs go via proving existence for an approximate version of the problem (via Sperner's Lemma [Sperner, 1928] or the related K-K-M Lemma [Knaster et al., 1929]) and then take limits to obtain the existence of exact divisions. An alternative proof by Woodall [1980] uses Brouwer's fixed point theorem [Brouwer, 1911]. The first continuous proof of existence was given by Filos-Ratsikas et al. [2023] in the context of proving that computing an envy-free division is in the class FIXP, via the employment of their OPT-gate for FIXP:

Theorem 8.1 ([Filos-Ratsikas et al., 2023]). Computing an envy-free division of the cake is in FIXP.

In general, FIXP is indeed the right class for the computation of exact envy-free divisions, as there are simple examples showing that even when the valuations are given by linear density functions, all envy-free divisions might be irrational (e.g., see [Bei et al., 2012]). We provide a simple one below for completeness:

Example 8.1 (Example where all envy-free cake cutting divisions are irrational). Consider an instance with two identical agents and let u be their common valuation function. For u, we will specify the densities over the cake [0,1], which are both given by f(t) = 2t. Let z be the point where the cake is cut at an envy-free division. It must hold that u([0,z]) = u([z,1]), i.e.,

$$\int_0^z 2t \ dt = \int_z^1 2t \ dt$$

The first integral evaluates to z^2 and the second integral evaluates to $1 - z^2$. It follows that the only envy-free division is obtained by cutting the cake at $z = 1/\sqrt{2}$, an irrational number.

Still, there are interesting valuation functions for which rational envy-free divisions exist, and for those cases a FIXP-membership result is unsatisfactory. For example, it is known that when agents have piecewise-constant density functions (i.e., step functions), an envy-free division in rational numbers exists (e.g., see [Goldberg et al., 2020]). For these cases for which rationality is possible, we would like to obtain a PPAD-membership result instead. This is achievable via the "approximation and rounding" approach which was discussed in Section 1.2.1, i.e., to start from an ε -approximately envy-free division of the cake, known to be computable in PPAD from [Deng et al., 2012], and then round it to an exact solution, as long as ε is small enough [Goldberg et al., 2020]. We provide an alternative, conceptually much simpler proof, via the employment of our linear-OPT-gate, which does not require approximations or rounding. We will obtain the PPAD-membership of the problem for any valuation function that is given by a PL pseudo-circuit. We state the main theorem of the section.

Theorem 8.2. Computing an envy-free division of the cake when the agents' valuation functions are given by PL pseudo-circuits is in PPAD.

The valuation functions are the integrals of the density functions, so Theorem 8.2 immediately implies a PPAD-membership result for envy-free divisions with piecewise-constant densities, which we mentioned above. Using the machinery developed in Section 3.4, we can also capture simple interesting cases where the inputs are given as the integrals of the density functions rather than the utilities themselves. We develop the proof in Section 8.1.1 below.

8.1.1 The proof of Theorem 8.2

The proof follows very much along the same lines of that of Theorem 8.1 presented in [Filos-Ratsikas et al., 2023], with a crucial modification to make it amenable to the use of PL arithmetic circuits. For that reason,

Figure 12: The linear program \mathcal{P}_1 used for capacities \mathbf{c}_i (left) and the linear program \mathcal{P}_2 used for the flow \mathbf{y} .

it is instructive to first present the proof of [Filos-Ratsikas et al., 2023], and then explain how to obtain the proof of Theorem 8.2 from there.

Envy-free divisions as matchings in bipartite graphs. Let $\mathbf{x} \in \Delta^{n-1}$ be a division of the cake. Let G = (N, P, E) be the bipartite graph in which N is the set of agents, P is the set of pieces and an edge (i, j), $i \in N, j \in P$ is in E if and only if agent i prefers piece j. Given this interpretation, \mathbf{x} is envy-free if and only if G admits a perfect matching. We will find that perfect matching via a maximum flow argument, therefore we define capacities c_e on the edges of the corresponding flow network.

Envy-free divisions as fixed points. Filos-Ratsikas et al. [2023] construct a function $F: D \to D$ and recover envy-free divisions from its fixed points. In particular the domain of F will be $D = \Delta^{n-1} \times (\Delta^{n-1})^n \times ([0,1]^n)^n$. In D:

- a point $\mathbf{x} \in \Delta^{n-1}$ will be the division of the cake,
- a point $\mathbf{c}_i \in \Delta^{n-1}$ represents the capacities of the edges from agent $i \in N$,
- a point \mathbf{y}_i in $[0,1]^n$ represents the flow along the edges from agent $i \in \mathbb{N}$.

In other words, the input of F will be a vector $(\mathbf{x}, \mathbf{c}_1, \dots, \mathbf{c}_n, \mathbf{y}_1, \dots, \mathbf{y}_n)$ and let $(\mathbf{x}^*, \mathbf{c}_1^*, \dots, \mathbf{c}_n^*, \mathbf{y}_1^*, \dots, \mathbf{y}_n^*)$ denote the output. Let $r_k = \max\{0, 1 - \sum_{i=1}^n y_{ik}\}$ denote the *flow excess* incoming into piece k. The division \mathbf{x}^* will be computed via the following equations:

$$x_j^* = \frac{x_j + r_j}{1 + \sum_{k=1}^n r_k}, \text{ for all pieces } j$$
 (48)

The capacities \mathbf{c}_i^* and the flows \mathbf{y}_i^* will be obtained as the outcomes of the linear programs of Figure 12. \mathcal{P}_1 is a set of linear programs (one for each agent $i \in N$), the optimal solutions of which will give the capacities c_{ii}^* , and \mathcal{P}_2 is a linear program that gives the values y_{ii}^* of the flow.

Remark 8.1. The $1/n^3$ term in the constraints of \mathcal{P}_2 ensures that there exists a *strictly feasible* point (i.e., a point $\hat{\mathbf{z}}$ such that the general constraint $\mathbf{C} \cdot \mathbf{z} \leq \mathbf{d}$ is satisfied with strict inequality, i.e., $\mathbf{C} \cdot \hat{\mathbf{z}} < \mathbf{d}$, and hence satisfies the Slater condition [Slater, 1950]). This is required for the OPT-gate for FIXP of Filos-Ratsikas et al. [2023] but is not required for our linear-OPT-gate. In particular, we could have the constraints be $0 \leq y_{ij} \leq c_{ij}$ instead, and in fact that would even slightly simplify the proofs. However, we elected to keep linear program \mathcal{P}_2 with the term $1/n^3$ in the constraints, for two reasons. First, it allows us to directly compare with the proof of [Filos-Ratsikas et al., 2023]. Secondly, it allows us to refer to linear program \mathcal{P}_2 when we prove the FIXP-membership for the rental harmony problem in Section 8.2.

Filos-Ratsikas et al. [2023] state and prove the following simple lemma.

Lemma 8.1 ([Filos-Ratsikas et al., 2023]). *If the total flow* $\sum_{i \in N} \sum_{j} y_{ij}^* = n$, then \mathbf{x}^* is an envy-free division.

Proof Sketch. We sketch the proof here, and refer the reader to [Filos-Ratsikas et al., 2023] for the (only slightly longer and more detailed) complete proof. By Hall's Theorem [Hall, 1935], it follows that unless x^* is an envy-free division indeed, there will be some set of agents $A \subseteq N$ such that |N(A)| < |A|, where N(A) is the set of pieces preferred by agents in A. Linear program \mathcal{P}_1 dictates that in an optimal solution, $c^*_{ij} > 0$ only for pieces that agent i prefers, i.e., $c^*_{ij} = 0$ for any $i \in A$ and $j \neq N(A)$. Using this in linear program \mathcal{P}_2 , we get that $y^*_{ij} \in [0,1/n^3]$, which allows the total flow out of A to be bounded by a quantity strictly smaller than |A| and consequently, the total flow in the network by a quantity strictly smaller than n, obtaining a contradiction.

Lemma 8.1 reduces the problem of finding an envy-free division to that of finding a flow of total value n. In turn, by the definition of r_j , this is equivalent to stating that $r_j = 0$ for all pieces j. Filos-Ratsikas et al. [2023] use the definition of x_j^* to establish that in a fixed point of F, this reduces to proving the statement of Lemma 8.2 below. Indeed, the definition of x_j^* establishes that in a fixed point (where $x_j^* = x_j$), it must be the case that

$$x_j \cdot \sum_{k=1}^n r_k = r_j$$
, for all pieces j ,

which implies that

$$x_j > 0$$
 and $r_j = 0 \Rightarrow \sum_{k=1}^n r_k = 0$,

as desired.

Lemma 8.2 ([Filos-Ratsikas et al., 2023]). There exists some piece j with $r_i^* = 0$ and $x_i^* > 0$.

Proof. Since ${\bf x}$ is a valid division, there exists some ℓ such that $x_\ell^*>0$. If $r_\ell^*=0$ we are done, so assume that $r_\ell^*>0$. By the definition of r_ℓ^* , that means that piece ℓ has positive residual capacity, which implies that the total flow in the network is less than n. Hence, there exists some agent $i\in N$, such that the total flow out of i is less than 1, i.e., $\sum_{k=1}^n y_{ik}^*<1$. The next step is to establish that there exists some piece $\ell'\in N(i)$, where N(i) is the set of preferred pieces for agent i, with $r_{\ell'}^*=0$, which will conclude the proof, since $x_{\ell'}>0$ follows from the hungriness condition. Assume by contradiction that for all $k\in N(i)$, it holds that $r_k^*>0$. Note that $\sum_{k=1}^N y_{ik}^*<0$ as established before, and that $\sum_{k\in N(i)} c_{ik}^*=1$, which means that it is possible to send more flow from i to N(i). Since each piece in N(i) is preferred by agent i, this contradicts the optimality of ${\bf y}^*$ as a solution to linear program \mathcal{P}_2 .

We now move on to the proof of Theorem 8.2. The proof mimics that presented above, except for one crucial part: we can no longer use Equation (48) to retrieve the envy-free division as a fixed point of the function F. This is because Filos-Ratsikas et al. [2023] only need to argue that F can be represented by an arithmetic circuit that uses OPT-gates for FIXP, whereas we need to argue that it is possible to represent F by a PL arithmetic circuit using linear-OPT-gates. In particular, we cannot divide by $1 + \sum_{k=1}^{n} r_k$ in a PL arithmetic circuit.

It turns out that there is an easy "fix" for this. Consider the following linear program \mathcal{P}_0 .

Linear Program
$$\mathcal{P}_0$$

maximize
$$\sum_{k=1}^{n} r_k \cdot x_k$$
 subject to
$$\sum_{k=1}^{n} x_k = 1$$

$$x_k \ge 0 \text{ for any piece } k$$

The division \mathbf{x}^* will be obtained as the optimal solution to a set of linear programs of the form \mathcal{P}_0 , one for each piece. We argue that a fixed point of the function F is an envy-free division below. The PPAD-membership of the problem then follows by restricting the functions $u_{ij}(\mathbf{x})$ to assume an appropriate form that allows F to be computable by a PL arithmetic circuit.

Proof of Theorem 8.2. Consider a fixed point of the function F and let \mathbf{x}^* be the corresponding division, which is an optimal solution of linear program \mathcal{P}_0 by design. Note that in such an optimal solution, positive weight is only placed on variables x_ℓ for which r_ℓ is maximum, i.e., $x_\ell^* > 0 \Rightarrow r_\ell^* = \max_k r_k^*$. Hence, if there exists some piece f such that f suc

 r_k^* can obviously be computed by a PL arithmetic circuit. Linear programs \mathcal{P}_0 , \mathcal{P}_1 and \mathcal{P}_2 use only linear constraints with the gate inputs appearing only on the right-hand side of the constraints. The gate inputs do not appear in the objective function of \mathcal{P}_2 , whereas the objective function of \mathcal{P}_0 has a linear subgradient. For \mathcal{P}_1 , the subgradient of the objective function is a linear function in $u_{ij}(\mathbf{x})$, and since $u_{ij}(\mathbf{x})$ is given by a PL pseudo-circuit, so is the subgradient of the objective function. This establishes that F can be represented by a PL arithmetic circuit, which concludes the proof.

8.2 Rental Harmony

We define the rental harmony problem, notably studied by Su [1999].

Definition 8.2 (Rental harmony). In the rental harmony problem, the total rent of a house has to be divided among the n rooms of the house, in a way that makes it possible to assign the rooms to n tenants in an envy-free manner, i.e., no tenant would prefer to have someone else's room (with the respective rent) to their own. Formally, we denote the rent by the unit interval [0,1]. Each tenant has a valuation function $u_{ij}: \Delta^{n-1} \to \mathbb{R}_{\geq 0}$, for each room j, assigning a real number to each division of the rent. Given a division \mathbf{x} , we will say that tenant i prefers the j-th room if $u_{ij}(\mathbf{x}) \geq u_{ij'}(\mathbf{x})$ for any room j'. A division \mathbf{x} is envy-free if there exists a permutation π of $\{1,\ldots,n\}$ such that for every $i \in \{1,\ldots,n\}$, tenant i prefers room $\pi(i)$.

Known results for existence. The existence of a solution to the rental harmony problem was proven by Su [1999], via an interesting adaptation of the idea of Simmons for proving the existence of envy-free cake cutting solutions, one which employs a "dual Sperner labelling". Similarly to the proofs for cake-cutting, the proof also appeals to limits of approximate solutions. To the best of our knowledge, computational complexity results about the general version of the problem did not exist before our work.

One can see the similarities between Definition 8.2 above and Definition 8.1 of Section 8.1. In fact, those are even more clear if we consider the following problem, which we refer to as *envy-free chore division*.

Definition 8.3 ((Contiguous) envy-free chore division). Let the interval [0,1] be called a *chore*, which is to be divided into n subintervals (pieces) using n-1 cuts. Let $\mathbf{x} \in \Delta^{n-1}$ denote a division of the chore, i.e., each division is a point in the simplex, and let x_j be the j'th coordinate (or "the j'th piece"). There is a set N of n agents, and for each piece j, each agent $i \in N$ has a *valuation function* $u_{ij}: \Delta^{n-1} \to \mathbb{R}_{\geq 0}$, assigning a real number to a division of the cake. Given a division \mathbf{x} , we will say that agent $i \in N$ *prefers* the j-th piece if $u_{ij}(\mathbf{x}) \geq u_{ij'}(\mathbf{x})$ for any piece j'. A division \mathbf{x} is envy-free if there exists a permutation π of $\{1,\ldots,n\}$ such that for every $i \in N$, agent i prefers piece $\pi(i)$.

Looking at Definition 8.3 above, one can observe two things:

- (a) It defines the same problem as the one in Definition 8.2, by simply substituting the term "room" with "piece" (and the term "tenant" with "agent").
- (b) It defines the same problem as the one in Definition 8.1 in Section 8.1, by simply substituting the term "chore" with "cake".

Observation (a) is because we are using a very general definition of envy-free chore-division (as we used a very general definition of envy-free cake-cutting in Section 8.1), one in which there is a different valuation function for each piece. In that generality, there is no difference between a piece and a room. For restricted valuation functions, the two problems could be different. We remark again that for proving membership results, considering these very general versions makes the results stronger.

Now looking at Observation (b), it seems as if the envy-free chore division problem and the envy-free cake cutting problem are the same. Is that really the case? The difference lies in the *sufficiency condition*. Recall that for cake-cutting, the sufficiency condition was hungriness, i.e., that agents always prefer a non-empty piece to any empty piece. For chore division/rental harmony, we will have the exact opposite.

Sufficiency condition: Miserly agents. We will say that an agent i is a *miser*, if she prefers an empty piece of the chore (respectively a room with zero rent) to a non-empty piece (respectively a room with non-zero rent). An instance of the chore division/rental harmony problem satisfies the miserly agents condition if all of the agents are misers.

Given the above, in our setting envy-free chore division (Definition 8.3) and rental harmony (Definition 8.2) are equivalent, so we will use the terminology of envy-free chore division in the proof that we will develop, mainly for consistency with Section 8.1. Our main theorem of the section is the following, which is reminiscent of Theorem 8.2 for envy-free cake cutting.

Theorem 8.3. Computing an envy-free division of the chore (i.e., a rental harmony partition) when the agents' valuation functions are given by PL arithmetic circuits in PPAD.

As in the case of Section 8.1, using the machinery developed in Section 3.4, we can also capture simple interesting cases where the inputs are given as the integrals of the density functions rather than the utilities themselves.

Before we proceed with the proof, we make the following remark.

Remark 8.2. One may be inclined to believe that there would be an easy reduction from envy-free chore division to envy-free cake cutting, e.g., by setting $u_{ij}^{\text{cake}}(\mathbf{x}) = -u_{ij}^{\text{chore}}(\mathbf{x})$ for all agents i and pieces j. Indeed, this would turn a hungry agent into a miser, thus establishing the correct sufficiency condition. However, letting $\mathbf{x}^{\text{cake}} = \mathbf{x}^{\text{chore}}$ would not result in an envy-free chore division, as shown by the simple following example with 2 agents. Let $x_1^{\text{cake}} = [0,1/2]$ and $x_2^{\text{cake}} = (1/2,1]$ and let $u_{11}^{\text{cake}}(x_1^{\text{cake}}, x_2^{\text{cake}}) = 1 - \epsilon$ and $u_{12}^{\text{cake}}(x_1^{\text{cake}}, x_2^{\text{cake}}) = \epsilon$ for some $\epsilon < 1$. Let agent 2 be indifferent between any allocation of the cake. We have that $u_{11}^{\text{chore}}(x_1^{\text{chore}}, x_2^{\text{chore}}) = -u_{11}^{\text{cake}}(x_1^{\text{cake}}, x_2^{\text{cake}}) = -1 + \epsilon$ and $u_{12}^{\text{chore}}(x_1^{\text{chore}}, x_2^{\text{chore}}) = -u_{12}^{\text{cake}}(x_1^{\text{cake}}, x_2^{\text{cake}}) = -\epsilon$. Agent 1 obviously prefers piece 2 rather than piece 1, and this is not an envy-free chore division.

In this example it is easy to remedy that, by labelling the pieces oppositely, i.e., by having $x_1^{\text{cake}} = x_2^{\text{chore}}$ and vice-versa. What happens however in more complex scenarios, with more agents and different valuations? Finding the appropriate division would essentially end up constructing the dual Sperner labelling used by Su [1999]. Our results in this section provide an alternative way of finding an envy-free chore division, which is very much in line with the approach that we used in Section 8.1. In other words, our technique based on the linear-OPT-gate provides a *unified proof of existence* of solutions for both envy-free cake cutting and rental harmony.

8.3 The proof of Theorem 8.3

Exactly in the same manner as in Section 8.1, we will consider a bipartite graph with agents i on one side and pieces j on the other side, with edges (i, j) indicating that agent i prefers pieces j. An envy-free chore division will correspond to a perfect matching in the graph, which we will again compute via a maximum flow argument in a fixed point.

Again, we construct a function $F: D \to D$ and recover envy-free chore divisions from its fixed points. In particular the domain of F will be $D = \Delta^{n-1} \times (\Delta^{n-1})^n \times ([0,1]^n)^n$. In D:

- a point $\mathbf{x} \in \Delta^{n-1}$ will be the division of the chore,
- a point $\mathbf{c}_i \in \Delta^{n-1}$ represents the capacities of the edges from agent $i \in N$,
- a point \mathbf{y}_i in $[0,1]^n$ represents the flow along the edges from agent $i \in N$.

In other words, the input of F will be a vector $(\mathbf{x}, \mathbf{c}_1, \dots, \mathbf{c}_n, \mathbf{y}_1, \dots, \mathbf{y}_n)$ and let $(\mathbf{x}^*, \mathbf{c}_1^*, \dots, \mathbf{c}_n^*, \mathbf{y}_1^*, \dots, \mathbf{y}_n^*)$ denote the output. As in Section 8.1, the capacities and the flow values will be obtained as optimal solutions to the linear programs \mathcal{P}_1 and \mathcal{P}_2 of Figure 12 respectively.

Again, we will use $r_k = \max\{0, 1 - \sum_{i=1}^n y_{ik}\}$ to denote the flow excess incoming into piece k. Lemma 8.1 holds verbatim for our setting here as well, which establishes that it suffices to argue that in our fixed point, $r_j^* = 0$ for all pieces j. Now, the chore division \mathbf{x}^* will be an optimal solution to the following linear program, which is a straightforward adaptation of linear program \mathcal{P}_0 presented in Section 8.1.

Linear Program
$$\mathcal{P}_0^c$$

minimize $\sum_{k=1}^n r_k \cdot x_k$

subject to $\sum_{k=1}^n x_k = 1$
 $x_k \ge 0$ for any piece k

The following lemma is analogous to Lemma 8.2, although the proof is somewhat different.

Lemma 8.3. In a fixed point of F, $r_i^* = 0$ for all pieces j.

Proof. Consider a fixed point $(\mathbf{x}^*, \mathbf{c}_1^*, \dots, \mathbf{c}_n^*, \mathbf{y}_1^*, \dots, \mathbf{y}_n^*)$ of F. For any piece j, $x_j^* > 0 \Rightarrow r_j^* = \min_k r_k^*$, which follows from the fact that x_j^* is an optimal solution to linear program \mathcal{P}_0^c , and hence only places positive weight to pieces with minimum r_k^* .

We will first argue that $x_j^* > 0$ for all pieces j. Assume first that $x_{j0}^* = 0$ for some piece j_0 and consider any piece j such that $x_j^* > 0$; note that at least one such j must exist by the fact that \mathbf{x}^* is a valid division of the chore. By the miserly agents sufficiency condition, we have that any agent $i \in N$ prefers j_0 to j, i.e., $u_{ij_0}(\mathbf{x}^*) > u_{ij}(\mathbf{x}^*)$. Since c_{ij}^* is an optimal solution to linear program \mathcal{P}_1 , it follows that $c_{ij}^* = 0$. In turn, by the corresponding constraint of linear program \mathcal{P}_2 , it follows that $y_{ij}^* \leq 1/n^3$, and that $\sum_{i=1}^n y_{ij}^* \leq 1/n^2$. From this, we conclude that $r_i^* \geq 1 - 1/n^2$.

Since $x_j^* > 0$, by the discussion above this implies that $\min_k r_k^* \ge 1 - 1/n^2$, which of course implies that $r_\ell^* \ge 1 - 1/n^2$ for all pieces ℓ . It follows that $\sum_{i=1}^n y_{i\ell}^* \le 1/n^2$ for all pieces ℓ , and by summing over ℓ we obtain that $\sum_{\ell=1}^n \sum_{i=1}^n y_{ij}^* < 1/n$. Since \mathbf{c}_i is a feasible solution to linear program \mathcal{P}_1 , there exists a set of pieces A_ℓ with $\sum_{i \in A_\ell} c_{ij}^* = 1$ for at least one agent $i \in N$ (and those are precisely the pieces with $x_\ell^* = 0$). That implies that there is a solution to linear program \mathcal{P}_2 for which $\sum_{\ell=1}^n \sum_{i=1}^n y_{ij} \ge 1$. This contradicts the fact that \mathbf{y}^* is an optimal solution to linear program \mathcal{P}_2 .

Now since $x_j^* > 0$ for all pieces j, by the discussion in the first paragraph of the proof it follows that $r_j^* = r_{j'}^*$ for all pieces j and j'. Let r^* denote the value of r_j for any j. We will argue that $r^* = 0$, which will conclude the proof. Assume by contradiction that $r^* > 0$, or, equivalently, that $\sum_{i=1}^n y_{ij}^* < 1$ for piece j. This implies that in linear program \mathcal{P}_2 the corresponding constraint for piece j is not be tight. In turn, that means that there is a feasible flow \mathbf{y} of value larger than that of \mathbf{y}^* , contradicting the fact that \mathbf{y}^* is an optimal solution to linear program \mathcal{P}_2 .

We can now conclude the proof of Theorem 8.3.

Proof of Theorem 8.3. Lemma 8.3 establishes that a fixed point of F corresponds to an envy-free chore division. It remains to show that F can be computed by a PL arithmetic circuit containing linear-OPT-gates. This is virtually identical to the corresponding argument in the proof of Theorem 8.2.

8.3.1 FIXP-membership for (unrestricted) rental harmony

In the envy-free cake cutting application, Theorem 8.2 provides a crisper complexity result than that of Theorem 8.1, for the case when the valuation functions are given by PL pseudo-circuits, which admits envy-free divisions in rational numbers. For rental harmony/chore division, the situation is similar, except, as we mentioned earlier, a FIXP-membership result (or any other kind of complexity result for that matter) was not known before our work. It is almost straightforward to generalize our proof developed in Section 8.3 to obtain a theorem analogous to Theorem 8.1, for the case of fairly dividing a chore as well, when the valuation functions are represented by general arithmetic circuits (not necessarily PL).

Theorem 8.4. Computing an envy-free division of the chore is in FIXP.

Proof. Construct a function F exactly as in Section 8.3 and consider a fixed point $(\mathbf{x}^*, \mathbf{c}_1^*, \dots, \mathbf{c}_n^*, \mathbf{y}_1^*, \dots, \mathbf{y}_n^*)$ of F. The arguments on why \mathbf{x}^* constitutes an envy-free division are identical to those developed in the previous section. What we need to establish is that F can be computed by an arithmetic circuit that contains instances of the OPT-gate for FIXP developed by Filos-Ratsikas et al. [2023]. In reality, this has practically already been established in [Filos-Ratsikas et al., 2023]: linear programs \mathcal{P}_1 and \mathcal{P}_2 are identical to those used in [Filos-Ratsikas et al., 2023], and linear program \mathcal{P}_0 is also amenable to the use of their OPT-gate for FIXP. We note that it is possible to multiply two inputs in arithmetic circuits, and hence the valuations $u_{ij}(\mathbf{x})$ in the objective function of linear program \mathcal{P}_1 are not restricted as in Theorem 8.3.

We conclude the section with an example showing that if one goes beyond linear valuations, there are cases where all envy-free divisions are irrational, and hence the FIXP-membership result of Theorem 8.4 (rather than a PPAD-membership result) is justified.

Example 8.2 (Example where all envy-free chore divisions are irrational). Consider an instance of the rental harmony/envy-free chore division problem with two agents. A division of the chore is represented by a point $(x, 1 - x) \in \Delta^1$. The common utility functions of the two agents are given by $u_1(x, 1 - x) = 2(1 - x)^2$ and $u_2(x, 1 - x) = x^2$. Note that if x = 0 the agents will prefer piece 1, and if x = 1 the agents will prefer piece 2. Therefore, the agents are indeed misers. For (x, 1 - x) to be an envy-free chore division, then it must hold that $2(1 - x)^2 = x^2$. As $x \in [0, 1]$, this implies that $x = 2 - \sqrt{2}$, so the only solution is irrational.

9 Conclusion and Future Work

In this work, we developed the linear-OPT-gate, a powerful general-purpose tool for showing the PPAD-membership of problems that have *exact rational solutions*. We demonstrated its strength by applying it to a plethora of domains related to game theory, competitive markets, auto-bidding auctions and fair division. For those applications, we obtain new results and generalizations of the state-of-the-art complexity results, as well as *significant* simplifications in terms of the proof techniques.

There are some interesting open directions related to our work, mainly in the domain of competitive markets. First, it will be very interesting to see whether one could extend our machinery in Section 3.4 to also capture markets with SSPLC production sets; we discussed the challenges of this task in Remark 6.15. Similarly, it would be interesting to try to design a class of succinct utility and production functions that subsumes all the known classes for which rational solutions are known to exist, i.e., one that would generalize the Leontief-free class of functions. Finally, one application that we did not study in our work is that of *competitive markets for mixed manna*, where there are goods but also bads to be allocated to the consumers. Chaudhury et al. [2022] studied these markets for SPLC utility functions and provided a PPAD-membership result. There does not seem to be any technical obstacle to applying our technique on those markets as well (and also possibly incorporating production functions as well); the details are still to be worked out.

Looking at the big picture, our linear-OPT-gate complements and refines the OPT-gate for FIXP of Filos-Ratsikas et al. [2023] as a tool to proving computational membership of *exact* problems in the appropriate complexity classes. One interesting question is whether one could hope to develop a similar gate for

approximate problems, i.e., an optimization gate that could be used in a very similar manner to those other two gates to establish PPAD-membership of more general problems (with irrational solutions), for their approximate versions. This certainly introduces new challenges and intriguing questions. One would have to work with approximate rather than exact fixed points. How should the gate be constructed to be useful in this regard? Should the gate work approximately as well? Applications domains like competitive markets, where the approximation in the competitive equilibrium notion comes from relaxing the clearing condition rather than the bundle optimality of the consumers, seem to suggest otherwise.

Acknowledgments

Kristoffer Arnsfelt Hansen and Kasper Høgh were supported by the Independent Research Fund Denmark under grant no. 9040-00433B. Alexandros Hollender was supported by the Swiss State Secretariat for Education, Research and Innovation (SERI) under contract number MB22.00026.

References

- Heiner Ackermann, Heiko Röglin, and Berthold Vöcking. On the impact of combinatorial structure on congestion games. *Journal of the ACM (JACM)*, 55(6):1–22, 2008.
- Kenneth J. Arrow and Gerard Debreu. Existence of an equilibrium for a competitive economy. *Econometrica: Journal of the Econometric Society*, pages 265–290, 1954.
- Robert J Aumann. Subjectivity and correlation in randomized strategies. *Journal of mathematical Economics*, 1 (1):67–96, 1974.
- Moshe Babaioff, Robert Kleinberg, and Christos H Papadimitriou. Congestion games with malicious players. *Games and Economic Behavior*, 1(67):22–35, 2009.
- Santiago Balseiro, Yuan Deng, Jieming Mao, Vahab Mirrokni, and Song Zuo. Robust auction design in the auto-bidding world. *Advances in Neural Information Processing Systems*, 34:17777–17788, 2021a.
- Santiago Balseiro, Anthony Kim, Mohammad Mahdian, and Vahab Mirrokni. Budget-management strategies in repeated auctions. *Operations research*, 69(3):859–876, 2021b.
- Santiago R. Balseiro and Yonatan Gur. Learning in repeated auctions with budgets: Regret minimization and equilibrium. *Management Science*, 65(9):3952–3968, 2019.
- Santiago R. Balseiro, Yuan Deng, Jieming Mao, Vahab S. Mirrokni, and Song Zuo. The landscape of auto-bidding auctions: Value versus utility maximization. In *Proceedings of the 22nd ACM Conference on Economics and Computation (EC)*, pages 132–133, 2021c.
- Xiaohui Bei, Ning Chen, Xia Hua, Biaoshuai Tao, and Endong Yang. Optimal proportional cake cutting with connected pieces. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI)*, pages 1263–1269, 2012.
- Christian Borgs, Jennifer Chayes, Nicole Immorlica, Kamal Jain, Omid Etesami, and Mohammad Mahdian. Dynamics of bid optimization in online advertisement auctions. In *Proceedings of the 16th international conference on World Wide Web (WWW)*, pages 531–540, 2007.
- Stephen P. Boyd and Lieven Vandenberghe. Convex Optimization. Cambridge University Press, 2004.
- Steven J. Brams and Alan D. Taylor. Fair Division: From cake-cutting to dispute resolution. Cambridge University Press, 1996.
- L. E. J. Brouwer. Über Abbildung von Mannigfaltigkeiten. Mathematische Annalen, 71:97–115, 1911.

- Ioannis Caragiannis, Kristoffer Arnsfelt Hansen, and Nidhi Rathi. On the complexity of pareto-optimal and envy-free lotteries. *arXiv preprint arXiv*:2307.12605, 2023.
- Bhaskar Ray Chaudhury, Jugal Garg, Peter McGlaughlin, and Ruta Mehta. A complementary pivot algorithm for competitive allocation of a mixed manna. *Mathematics of Operations Research*, 48(3):1630–1656, 2022.
- Thomas Chen, Xi Chen, Binghui Peng, and Mihalis Yannakakis. Computational hardness of the Hylland-Zeckhauser scheme. In *Proceedings of the 33rd ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2253–2268, 2022.
- Xi Chen, Xiaotie Deng, and Shang-Hua Teng. Settling the complexity of computing two-player Nash equilibria. *Journal of the ACM (JACM)*, 56(3):1–57, 2009.
- Xi Chen, Dimitris Paparas, and Mihalis Yannakakis. The complexity of non-monotone markets. *Journal of the ACM*, 64(3):20:1–20:56, 2017.
- Xi Chen, Christian Kroer, and Rachitesh Kumar. The complexity of pacing for second-price auctions. In *Proceedings of the 22nd ACM Conference on Economics and Computation (EC)*, page 318, 2021a.
- Xi Chen, Christian Kroer, and Rachitesh Kumar. Throttling equilibria in auction markets. In *Proceedings of the 17th International Conference on Web and Internet Economics (WINE)*, page 551, 2021b.
- Richard Cole, Nikhil Devanur, Vasilis Gkatzelis, Kamal Jain, Tung Mai, Vijay V. Vazirani, and Sadra Yazdanbod. Convex program duality, Fisher markets, and Nash social welfare. In *Proceedings of the 18th ACM Conference on Economics and Computation (EC)*, pages 459–460, 2017.
- Vincent Conitzer, Christian Kroer, Debmalya Panigrahi, Okke Schrijvers, Nicolas E. Stier-Moses, Eric Sodomka, and Christopher A. Wilkens. Pacing equilibrium in first price auction markets. *Management Science*, 68(12):8515–8535, 2022a.
- Vincent Conitzer, Christian Kroer, Eric Sodomka, and Nicolas E. Stier-Moses. Multiplicative pacing equilibria in auction markets. *Operations Research*, 70(2):963–989, 2022b.
- Richard W. Cottle and George B. Dantzig. Complementary pivot theory of mathematical programming. *Linear Algebra and its Applications*, 1(1):103–125, 1968.
- Richard W. Cottle, Jong-Shi Pang, and Richard E. Stone. The linear complementarity problem. SIAM, 2009.
- Partha Sarathi Dasgupta and Eric S. Maskin. Debreu's social equilibrium existence theorem. *Proceedings of the National Academy of Sciences*, 112(52):15769–15770, 2015.
- Constantinos Daskalakis, Alex Fabrikant, and Christos H. Papadimitriou. The game world is flat: The complexity of Nash equilibria in succinct games. In *Proceedings of the 33rd International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 513–524, 2006.
- Constantinos Daskalakis, Paul W. Goldberg, and Christos H. Papadimitriou. The complexity of computing a Nash equilibrium. *SIAM Journal on Computing*, 39(1):195–259, 2009.
- Gerard Debreu. A social equilibrium existence theorem. *Proceedings of the National Academy of Sciences*, 38 (10):886–893, 1952.
- Argyrios Deligkas, John Fearnley, Themistoklis Melissourgos, and Paul G. Spirakis. Computing exact solutions of consensus halving and the Borsuk-Ulam theorem. *Journal of Computer and System Sciences*, 117:75–98, 2021.
- Xiaotie Deng, Qi Qi, and Amin Saberi. Algorithmic solutions for envy-free cake cutting. *Operations Research*, 60(6):1461–1476, 2012.

- B Curtis Eaves. A finite algorithm for the linear exchange model. *Journal of Mathematical Economics*, 3(2): 197–203, 1976.
- Kousha Etessami and Mihalis Yannakakis. On the complexity of Nash equilibria and other fixed points. *SIAM Journal on Computing*, 39(6):2531–2597, 2010.
- Kousha Etessami, Kristoffer Arnsfelt Hansen, Peter Bro Miltersen, and Troels Bjerre Sørensen. The complexity of approximating a trembling hand perfect equilibrium of a multi-player game in strategic form. In *Proceedings of the 7th International Symposium on Algorithmic Game Theory (SAGT)*, pages 231–243, 2014.
- Ky Fan. Fixed-point and minimax theorems in locally convex topological linear spaces. *Proceedings of the National Academy of Sciences*, 38(2):121–126, 1952.
- John Fearnley, Paul Goldberg, Alexandros Hollender, and Rahul Savani. The complexity of gradient descent: $CLS = PPAD \cap PLS$. *Journal of the ACM*, 70(1):1–74, 2022.
- John Fearnley, Paul W. Goldberg, Alexandros Hollender, and Rahul Savani. The complexity of computing KKT solutions of quadratic programs. *arXiv preprint arXiv:2311.13738*, 2023.
- Aris Filos-Ratsikas, Kristoffer A. Hansen, Kasper Høgh, and Alexandros Hollender. FIXP-membership via convex optimization: Games, cakes, and markets. *SIAM Journal on Computing*, pages FOCS21–30–FOCS21–84, 2023.
- George Gamow and Marvin Stern. Puzzle-math. Viking Press, 1958.
- Martin Gardner. aha! Insight. Scientific American, W. H. Freeman and Company, 1978.
- Jugal Garg. Market equilibrium under piecewise Leontief concave utilities. *Theoretical Computer Science*, 703: 55–65, 2017.
- Jugal Garg and Vijay V. Vazirani. On computability of equilibria in markets with production. In *Proceedings* of the 25th ACM-SIAM Symposium on Discrete Algorithms (SODA), pages 1329–1340, 2014.
- Jugal Garg, Albert Xin Jiang, and Ruta Mehta. Bilinear games: Polynomial time algorithms for rank based subclasses. In *Proceedings of the 7th International Workshop on Internet and Network Economics (WINE)*, pages 399–407, 2011.
- Jugal Garg, Ruta Mehta, Milind Sohoni, and Vijay V. Vazirani. A complementary pivot algorithm for market equilibrium under separable, piecewise-linear concave utilities. *SIAM Journal on Computing*, 44(6): 1820–1847, 2015.
- Jugal Garg, Ruta Mehta, and Vijay V. Vazirani. Substitution with satiation: A new class of utility functions and a complementary pivot algorithm. *Mathematics of Operations Research*, 43(3):996–1024, 2018.
- Irving L. Glicksberg. A further generalization of the Kakutani fixed theorem, with application to Nash equilibrium points. *Proceedings of the American Mathematical Society*, 3(1):170–174, 1952.
- Michel X. Goemans. Smallest compact formulation for the permutahedron. *Mathematical Programming*, 153 (1):5–11, 2015.
- Paul Goldberg, Alexandros Hollender, and Warut Suksompong. Contiguous cake cutting: Hardness results and approximation algorithms. *Journal of Artificial Intelligence Research*, 69:109–141, 2020.
- P. Hall. On representatives of subsets. Journal of the London Mathematical Society, s1-10(1):26-30, 1935.
- Kristoffer Arnsfelt Hansen and Troels Bjerre Lund. Computational complexity of proper equilibrium. In *Proceedings of the 19th ACM Conference on Economics and Computation (EC)*, pages 113–130, 2018.

- Joseph T. Howson. Equilibria of polymatrix games. Management Science, 18(5-part-1):312–318, 1972.
- Kamal Jain. A polynomial time algorithm for computing an Arrow-Debreu market equilibrium for linear utilities. *SIAM Journal on Computing*, 37(1):303–318, 2007.
- Elena Janovskaja. Equilibrium points in polymatrix games. *Lithuanian Mathematical Journal*, 8(2):381–384, 1968.
- David S. Johnson, Christos H. Papadimitriou, and Mihalis Yannakakis. How easy is local search? *Journal of computer and system sciences*, 37(1):79–100, 1988.
- Shizuo Kakutani. A generalization of Brouwer's fixed point theorem. *Duke Mathematical Journal*, 8(3): 457–459, 1941.
- George Karakostas and Anastasios Viglas. Equilibria for networks with malicious users. *Mathematical Programming*, 110(3):591–613, 2007.
- Shiva Kintali, Laura J. Poplawski, Rajmohan Rajaraman, Ravi Sundaram, and Shang-Hua Teng. Reducibility among fractional stability problems. *SIAM Journal on Computing*, 42(6):2063–2113, 2013.
- Max Klimm and Maximilian J. Stahlberg. Complexity of equilibria in binary public goods games on undirected graphs. In *Proceedings of the 24th ACM Conference on Economics and Computation (EC)*, pages 938–955, 2023.
- Max Klimm and Philipp Warode. Complexity and parametric computation of equilibria in atomic splittable congestion games via weighted block Laplacians. In *Proceedings of the 14th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2728–2747, 2020.
- Bronisław Knaster, Casimir Kuratowski, and Stefan Mazurkiewicz. Ein Beweis des Fixpunktsatzes für *n*-dimensionale Simplexe. *Fundamenta Mathematicae*, 14:132–137, 1929.
- Daphne Koller, Nimrod Megiddo, and Bernhard Von Stengel. Efficient computation of equilibria for extensive two-person games. *Games and Economic Behavior*, 14(2):247–259, 1996.
- Carlton E. Lemke. Bimatrix equilibrium points and mathematical programming. *Management science*, 11(7): 681–689, 1965.
- Carlton E. Lemke and Joseph T. Howson, Jr. Equilibrium points of bimatrix games. *Journal of the Society for industrial and Applied Mathematics*, 12(2):413–423, 1964.
- Juncheng Li and Pingzhong Tang. Auto-bidding equilibrium in ROI-constrained online advertising markets. *arXiv preprint arXiv*:2210.06107, 2023.
- Robert R. Maxfield. General equilibrium and the theory of directed graphs. *Journal of Mathematical Economics*, 27(1):23–51, 1997.
- Nimrod Megiddo and Christos H. Papadimitriou. On total functions, existence theorems and computational complexity. *Theoretical Computer Science*, 81(2):317–324, 1991.
- Frédéric Meunier and Thomas Pradeau. Computing solutions of the multiclass network equilibrium problem with affine cost functions. *Annals of Operations Research*, 274:447–469, 2019.
- Igal Milchtaich. Generic uniqueness of equilibrium in large crowding games. *Mathematics of Operations Research*, 25(3):349–364, 2000.
- Roger B. Myerson. Refinements of the Nash equilibrium concept. *International Journal of Game Theory*, 7(2): 73–80, 1978.

- John F. Nash. Equilibrium points in *n*-person games. *Proceedings of the National Academy of Sciences*, 36(1): 48–49, 1950.
- Christos Papadimitriou and Binghui Peng. Public goods games in directed networks. In *Proceedings of the 22nd ACM Conference on Economics and Computation (EC)*, pages 745–762, 2021.
- Christos Papadimitriou, Emmanouil-Vasileios Vlatakis-Gkaragkounis, and Manolis Zampetakis. The computational complexity of multi-player concave games and Kakutani fixed points. In *Proceedings of the 24th ACM Conference on Economics and Computation (EC)*, page 1045, 2023.
- Christos H. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *Journal of Computer and System Sciences*, 48(3):498–532, 1994.
- Christos H. Papadimitriou and Tim Roughgarden. Computing correlated equilibria in multi-player games. *Journal of the ACM*, 55(3):1–29, 2008.
- Ariel D. Procaccia. Cake cutting: Not just child's play. Communications of the ACM, 56(7):78–87, 2013.
- Jack Robertson and William Webb. Cake-cutting algorithms: Be fair if you can. CRC Press, 1998.
- J. B. Rosen. Existence and uniqueness of equilibrium points for concave *n*-person games. *Econometrica*, 33 (3):520–534, 1965.
- Robert W. Rosenthal. A class of games possessing pure-strategy Nash equilibria. *International Journal of Game Theory*, 2(1):65–67, 1973.
- Rahul Savani. Finding Nash equilibria of bimatrix games. PhD thesis, London School of Economics and Political Science, 2006.
- David Schmeidler. Equilibrium points of nonatomic games. Journal of statistical Physics, 7:295–300, 1973.
- Reinhard Selten. Reexamination of the perfectness concept for equilibrium points in extensive games. *International Journal of Game Theory*, 4:25–55, 1975.
- Morton Slater. Lagrange multipliers revisited. Cowles Commission Discussion Paper: Mathematics 403, Cowles Foundation for Research in Economics, Yale University, 1950. URL https://elischolar.library.yale.edu/cowles-discussion-paper-series/304.
- Troels Bjerre Sørensen. Computing a proper equilibrium of a bimatrix game. In *Proceedings of the 13th ACM Conference on Electronic Commerce (EC)*, pages 916–928, 2012.
- Emanuel Sperner. Neuer Beweis für die Invarianz der Dimensionszahl und des Gebietes. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 6:265–272, 1928.
- Hugo Steinhaus. Sur la division pragmatique. Econometrica, pages 315–319, 1949.
- Walter Stromquist. How to cut a cake fairly. The American Mathematical Monthly, 87(8):640-644, 1980.
- Francis Edward Su. Rental harmony: Sperner's lemma in fair division. *The American Mathematical Monthly*, 106(10):930–942, 1999.
- Michael J. Todd. Orientation in complementary pivot algorithms. *Mathematics of Operations Research*, 1(1): 54–66, 1976.
- Vijay V. Vazirani and Mihalis Yannakakis. Market equilibrium under separable, piecewise-linear, concave utilities. *Journal of the ACM*, 58(3):1–25, 2011.
- J. G. Wardrop. Some theoretical aspects of road traffic research. *Proceedings of the Institution of Civil Engineers*, 1(3):325–362, 1952.
- Douglas R. Woodall. Dividing a cake fairly. *Journal of Mathematical Analysis and Applications*, 78(1):233–247, 1980.