

Rational Convex Programs and Efficient Algorithms for 2-Player Nash and Nonsymmetric Bargaining Games*

Vijay V. Vazirani[†]

Abstract

The solution to a Nash or a nonsymmetric bargaining game is obtained by maximizing a concave function over a convex set, i.e., it is the solution to a convex program. We show that each 2-player game whose convex program has linear constraints, admits a rational solution and such a solution can be found in polynomial time using only an LP solver. If in addition, the game is succinct, i.e., the coefficients in its convex program are “small”, then its solution can be found in strongly polynomial time. We also give non-succinct linear games whose solution can be found in strongly polynomial time.

1 Introduction

Recently, Vazirani [Vaz12] introduced the notion of a *rational convex program (RCP)* – a non-linear convex program that “behaves like” an LP in that it admits rational solutions – and identified two classes of RCPs, quadratic and logarithmic, which contain all the known RCPs. Whereas convex programs belonging to the first class, quadratic RCPs, are always rational and have been studied for several decades, rationality for programs in the second class has to be established piecemeal, and other than the classical Eisenberg-Gale program, found in 1956 [EG59], the rest were discovered only in the last decade in the context of the study of equilibria for various market models. A convex program solver, based on either the ellipsoid algorithm or interior point methods [GLS88], can find an optimal solution to an RCP in polynomial time; however, at present, convex program solvers are considerably slower than LP-solvers. On the other hand, rationality indicates the existence of rich combinatorial structure in the problem, which may be gainfully exploited algorithmically. For issues of not only efficiency, but also to gain deeper structural insights into problems, [Vaz12] proposed two programs of study:

Program A: Design polynomial time (or better, strongly polynomial) combinatorial algorithms that solve individual RCPs.

Program B: Design polynomial time (or better, strongly polynomial) algorithms that solve individual RCPs, given an LP-solver (each call counts as 1 step).

*A preliminary version of this paper appeared as [Vaz10].

[†]College of Computing, Georgia Institute of Technology, Atlanta, GA 30332-0280, E-mail: vazirani@cc.gatech.edu. Research supported by NSF Grants CCF-0728640 and CCF-0914732, ONR Grant N000140910755, and a Google Research Grant.

In this paper we study, from this viewpoint, Nash bargaining games and their generalization to nonsymmetric bargaining games of Kalai [Kal77], when restricted to two players. The solution to a Nash or a nonsymmetric bargaining game is obtained by maximizing a concave function over a convex set, i.e., it is the solution to a convex program (see Section 2 for an explanation). We start by giving a classification of all Nash and nonsymmetric bargaining games, based on structural properties of their convex programs. The restriction of these classes to two players admits a rich theory, as described below.

Let NB be the class of Nash and nonsymmetric bargaining games that can be solved in polynomial time, and let LNB to be the subclass of NB consisting of those games whose constraints (in the convex program for the game) are linear (these can be solved exactly in polynomial time using, e.g., the ellipsoid algorithm [GLS88]). We let NB2 and LNB2 be the restrictions of these classes to 2-player games. Our main result is a polynomial time algorithm for solving any game in LNB2, given an LP-solver (i.e., program B). As a corollary, each game in LNB2 has a rational solution and hence its convex program is a logarithmic RCP; this property does not hold for 3-player games in LNB.

Next, we define a subclass of LNB2 called SLNB2, consisting of *succinct* games, i.e., the coefficients in the convex program of such a game are “small” (having bit-size that is polynomial in the number of variables and constraints of the convex program). We show that all games in SLNB2 admit strongly polynomial algorithms; however, these algorithms are not combinatorial. This class includes nontrivial and interesting games, e.g., the *2-player directed graph multicommodity flow game*, **DMF2**, which consists of a directed graph with edge capacities and each player is a source-sink pair desiring flow. This game is derived from Kelly’s flow markets [Kel97].

This raises the question of whether there are games in (LNB2 - SLNB2) that admit strongly polynomial time algorithms. We give an affirmative answer by studying a Nash bargaining game introduced in [Vaz12] and called the *Arrow-Debreu Nash bargaining game*, **ADNB**; this game was derived from the linear utilities case of the Arrow-Debreu market model. [Vaz12] showed that the convex program for this game is a logarithmic RCP, and gave a combinatorial polynomial time algorithm for solving it exactly. The 2-player version of this game, which we call **ADNB2**, is in (LNB2 - SLNB2) and admits a combinatorial strongly polynomial algorithm. Finally, we ask if the class (NB2 - LNB2) contains a game that admits a combinatorial algorithm. We provide an affirmative answer by giving the *circle game*. Its solution reduces to solving a degree 4 equation.

1.1 2-player vs. multi-player games

In game theory, 2-player games occupy a special place – not only because numerous applications involve two players but also because 2-player games often have remarkable properties that are not possessed by extensions to more players. For instance, in the case of Nash equilibrium, the 2-player case is the most extensively studied and used, and captures a rich set of possibilities, e.g., those encapsulated in canonical games such as prisoner’s dilemma, battle of the sexes, chicken, and matching pennies. In terms of properties, 2-player Nash equilibrium games always have rational solutions whereas games with three or more players may have only irrational solutions; an example of the latter, called “a three-man poker game,” was given by Nash [Nas50b].

From a computational viewpoint, the difference is even more stark. The problem of finding a

Nash equilibrium is PPAD-complete for two players [DGP09, CDT09] and FIXP-complete for three or more players [EY10]. Note that whereas PPAD is in $\text{FNP} \cap \text{co-FNP}$, the only fact known about FIXP is that $\text{P} \subseteq \text{FIXP} \subseteq \text{PSPACE}$. Next, let us restrict to zero-sum games. For two players, von Neumann’s minimax theorem yields a polynomial time algorithm using LP [vNM44]. On the other hand, 3-player zero-sum games are PPAD-hard, since any 2-player non-zero-sum game can be reduced to a 3-player zero-sum game [vNM44].

Our paper establishes a similar qualitative difference between the 2-player case and the multi-player case of Nash and nonsymmetric bargaining games. It is interesting to note that John Nash’s seminal paper, defining the bargaining game, dealt only with the case of 2-players [Nas50a]. Later, it was observed that his entire setup, and theorem characterizing the bargaining solution, easily generalize to the case of more than 2 players, e.g., see [Kal77]. Today, Nash bargaining is regarded as a central solution concept within game theory for “fair” allocation of utility among competing players in the presence of complete information, e.g., see [Kal85, TL89, OR94].

1.2 Outline of the paper

Section 2 defines Nash and Nonsymmetric bargaining games and shows why their solution is captured by a convex program. Section 3 gives a classification of these games, and their restriction to 2-players, that is relevant to our work, and Section 4 gives examples of 2-player games in each class. Section 5 gives the notion of rational convex programs from [Vaz12].

Our main result is given in Section 6 – a polynomial time algorithm for games in LNB2, using an LP-solver. After defining some important procedures in Section 6.1, we give the algorithm in Section 6.4. We note that our algorithm is a generalization of that of [CDV10] for EG(2) markets, the restriction of Eisenberg-Gale markets to 2 buyers.

Section 7 gives a strongly polynomial algorithm for the game **ADNB2**, using the notion of a *flexible budget market* introduced in [Vaz12] and defined in Section 7.1. We reduce **ADNB2** to such a market and give a combinatorial algorithm for finding an equilibrium in it. Section 8 extends this algorithm to give a strongly polynomial algorithm for the game **plc-ADNB2**. It reduces **plc-ADNB2** to a generalization of flexible budget market in which buyers pay for the amount of utility they accrue rather than the amount of goods they obtain; the latter notion was defined in [GV11]. We decided against giving an algorithm only for the more general problem, **plc-ADNB2**, because the algorithm for it is considerably more involved than that for **ADNB2**. Finally, Section 9 gives combinatorial algorithms for the circle game.

A number of interesting questions remain open, e.g., is there a characterization of the subclass of LNB2 which consists of all games that admit strongly polynomial algorithms? Also, the class (NB2 - LNB2) needs to be properly explored and understood, both structurally and algorithmically.

2 Nash and Nonsymmetric Bargaining Games

An n -person Nash bargaining game consists of a pair $(\mathcal{N}, \mathbf{c})$, where $\mathcal{N} \subseteq \mathbf{R}_+^n$ is a compact, convex set and $\mathbf{c} \in \mathcal{N}$. Set \mathcal{N} is the *feasible set* and its elements give utilities that the n players

can simultaneously accrue. Point \mathbf{c} is the *disagreement point* – it gives the utilities that the n players obtain if they decide not to cooperate. The set of n agents will be denoted by B and the agents will be numbered $1, 2, \dots, n$. Game $(\mathcal{N}, \mathbf{c})$ is said to be *feasible* if there is a point $\mathbf{v} \in \mathcal{N}$ such that $\forall i \in B, v_i > c_i$.

The solution to a feasible game is the point $\mathbf{v} \in \mathcal{N}$ that satisfies the following four axioms:

1. **Pareto optimality:** No point in \mathcal{N} can weakly dominate \mathbf{v} .
2. **Invariance under affine transformations of utilities.**
3. **Symmetry:** The numbering of the players should not affect the solution.
4. **Independence of irrelevant alternatives:** If \mathbf{v} is the solution for $(\mathcal{N}, \mathbf{c})$, and $\mathcal{S} \subseteq \mathbf{R}_+^n$ is a compact, convex set satisfying $\mathbf{c} \in \mathcal{S}$ and $\mathbf{v} \in \mathcal{S} \subseteq \mathcal{N}$, then \mathbf{v} is also the solution for $(\mathcal{S}, \mathbf{c})$.

Via an elegant proof, Nash proved:

Theorem 1 Nash [Nas50a] *If game $(\mathcal{N}, \mathbf{c})$ is feasible then there is a unique point in \mathcal{N} satisfying the axioms stated above. This is also the unique point that maximizes $\prod_{i \in B} (v_i - c_i)$ over all $\mathbf{v} \in \mathcal{N}$.*

Now, maximizing $\prod_{i \in B} (v_i - c_i)$ is equivalent to maximizing the log of this function, i.e., $\sum_{i \in B} \log(v_i - c_i)$, which is concave. Hence, Nash’s solution involves maximizing a concave function over a convex domain, and is therefore the optimal solution to the convex program that maximizes $\sum_{i \in B} \log(v_i - c_i)$ subject to $\mathbf{v} \in \mathcal{N}$. As a consequence, if for a specific game, a separation oracle can be implemented in polynomial time, then using the ellipsoid algorithm one can get as good an approximation to the solution as desired [GLS88].

Kalai [Kal77] generalized Nash’s bargaining game by removing the axiom of symmetry and showed that any solution to the resulting game is the unique point that maximizes $\prod_{i \in B} (v_i - c_i)^{p_i}$, over all $\mathbf{v} \in \mathcal{N}$, for some choice of positive numbers p_i , for $i \in B$, such that $\sum_{i \in B} p_i = 1$. Thus, any particular nonsymmetric bargaining solution is specified by giving the p_i ’s satisfying the two conditions, i.e., $\forall i \in B, p_i > 0$ and $\sum_{i \in B} p_i = 1$. For the purposes of computability, we will restrict to nonsymmetric games in which the p_i ’s are rational. Equivalently, let us define the *n -person nonsymmetric bargaining game* as follows. Assume that $B, \mathcal{N}, \mathbf{c}$ are as defined above. In addition, we are given the *clout*¹ of each player: a positive integer w_i for each player i .

Assuming the game is feasible, the solution to this nonsymmetric bargaining game is the unique point that maximizes $\prod_{i \in B} (v_i - c_i)^{w_i}$ over all $\mathbf{v} \in \mathcal{N}$. As before, we will view this as the solution to a convex program by maximizing $\sum_{i \in B} w_i \log(v_i - c_i)$ over all $\mathbf{v} \in \mathcal{N}$.

One more remark is in order. As shown by Kalai [Kal77], any nonsymmetric game can be reduced to a Nash bargaining game over a larger number of players. However, this reduction is not useful for our purpose because once the number of players increases, the special properties of 2-player games are lost.

¹The choice of the term “clout of a player” is justified by a theorem of Kalai stating that the solution to this game corresponds precisely to the solution of a k -person symmetric Nash bargaining game, with $k = \sum_{i \in B} w_i$, which is obtained by taking w_i copies of player i , for $1 \leq i \leq n$.

3 The Classes NB2, LNB2 and SLNB2

We first define the classes NB and LNB. Let \mathcal{G} be an n -person Nash or nonsymmetric bargaining game. As stated in Section 2, the solution to this game is given by the optimal solution to the following convex program, where \mathbf{x} is a vector of r allocation and auxiliary variables, and the functions f_i are convex. (Clearly, \mathcal{G} is a Nash bargaining game if all the w_i s are equal.)

$$\begin{aligned}
 & \text{maximize} && \sum_{i \in B} w_i \log(v_i - c_i) && (1) \\
 & \text{subject to} && \text{for } i = 1 \dots k : && f_i(\mathbf{v}, \mathbf{x}) \leq 0 \\
 & && && \mathbf{v} \geq \mathbf{c} \\
 & && && \mathbf{x} \geq 0
 \end{aligned}$$

Furthermore, \mathcal{G} is feasible iff the optimal solution to the following convex program is greater than zero:

$$\begin{aligned}
 & \text{maximize} && t && (2) \\
 & \text{subject to} && \text{for } i = 1 \dots n : && v_i \geq c_i + t \\
 & && \text{for } i = 1 \dots k : && f_i(\mathbf{v}, \mathbf{x}) \leq 0 \\
 & && && \mathbf{x} \geq 0
 \end{aligned}$$

The game \mathcal{G} is said to be in the class NB if its feasibility can be determined in polynomial time, and its solution can be found in polynomial time (if feasible).

If all constraints in (1) are linear, then game \mathcal{G} is said to be *linear*. If so, the constraints form a polyhedron in \mathbf{R}^{n+m} . Its projection on the first n coordinates, corresponding to \mathbf{v} , is a polytope, which is also the feasible set \mathcal{N} . The class of these games is called *linear Nash and nonsymmetric bargaining games*, and abbreviated to LNB.

Finally, the restriction of NB and LNB to 2-player games will be called NB2 and LNB2, respectively. We will assume w.l.o.g. that the convex program for game \mathcal{G} in LNB2 has the following form:

$$\begin{aligned}
 & \text{maximize} && \sum_{i=1,2} w_i \log(v_i - c_i) && (3) \\
 & \text{subject to} && \mathbf{A}\mathbf{x} + \mathbf{b}_1 v_1 + \mathbf{b}_2 v_2 \leq \mathbf{q} \\
 & && \text{for } i = 1, 2 : && v_i \geq c_i \\
 & && && \mathbf{x} \geq 0
 \end{aligned}$$

where \mathbf{A} is an $m \times r$ matrix, \mathbf{x} is a vector consisting of a total of r allocation and auxiliary variables, and $\mathbf{b}_1, \mathbf{b}_2, \mathbf{q}$ are arbitrary m -dimensional vectors.

We will say that \mathcal{G} is *succinct* if all the entries in $\mathbf{A}, \mathbf{b}_1, \mathbf{b}_2$ are polynomially bounded in m and r . The subclass of LNB2 consisting of all succinct games will be called SLNB2.

4 Some Representative 2-Player Games

In this section, we provide representative games for the three classes defined above. We will study all three games in detail in this paper.

4.1 The game **DMF2**

The game **DMF2** lies in SLNB2. We are given a directed graph $G = (V, E)$, with $c_e \in \mathbf{Q}^+$ specifying the capacity of edge $e \in E$. Two source-sink pairs are also specified, (s_1, t_1) and (s_2, t_2) . Each source-sink pair represents a player and the utility derived by player i is the flow sent from s_i to t_i . Each player has its own disagreement utility (flow value) c_i , for $i = 1, 2$; c_i can be thought of as a strict lower bound on the amount of flow player i desires (perhaps because of the resources player i has invested in building the network). In the nonsymmetric version, we are also given the clouts w_1 and w_2 of the two players. The objective is to find the Nash or nonsymmetric bargaining solution. Let \mathcal{G} denote the given instance of **DMF2**².

Next, we give a convex program that captures the solution to \mathcal{G} . The flow going from s_i to t_i will be referred to as commodity i , for $i = 1, 2$, and f_i will denote the total flow of commodity i . For each edge $e \in E$, we have 2 variables, f_e^1 and f_e^2 which denote the amount of each commodity flowing through e . The constraints ensure that the total flow going through an edge does not exceed its capacity and that for each commodity, at each vertex, other than the source-sink pair of this commodity, flow conservation holds. For vertex $v \in V$, $\text{out}(v) = \{(v, u) \mid (v, u) \in E\}$ and $\text{in}(v) = \{(u, v) \mid (u, v) \in E\}$. The constraints of this program are simply ensuring that (f_1, f_2) lies in the feasible set \mathcal{N} .

$$\begin{aligned}
 & \text{maximize} && \sum_{i=1,2} w_i \log(f_i - c_i) && (4) \\
 & \text{subject to} && \text{for } i = 1, 2 : && f_i = \sum_{e \in \text{out}(s_i)} f_e^i \\
 & && \forall e \in E : && f_e^1 + f_e^2 \leq c_e \\
 & && \text{for } i = 1, 2 : && \forall v \in V - \{s_i, t_i\} : \sum_{e \in \text{in}(v)} f_e^i = \sum_{e \in \text{out}(v)} f_e^i \\
 & && \text{for } i = 1, 2 : && \forall e \in E : f_e^i \geq 0
 \end{aligned}$$

4.2 The game **ADNB2**

The game **ADNB2** lies in (LNB2 - SLNB2). To define it, we first need to define the game **ADNB**, introduced in [Vaz12]. This game was derived from the linear case of the Arrow-Debreu model, which differs from Fisher's linear case in that each agent comes to the market not with money but with an initial endowment of goods. We first state it formally.

Let $B = \{1, 2, \dots, n\}$ be a set of agents and $G = \{1, 2, \dots, g\}$ be a set of divisible goods. We will assume w.l.o.g. that there is a unit amount of each good. Let u_{ij} be the utility derived by

²Note that there will be no confusion in using "c" to denote capacities of edges as well as disagreement utilities of players since in the former case, the subscript will always be e and in the latter case, it will be $1, 2$ or i .

agent i on receiving one unit of good j ; w.l.o.g., we will assume that u_{ij} is integral. If x_{ij} is the amount of good j that agent i gets, for $1 \leq j \leq g$, then the total utility derived by her is

$$v_i(x) = \sum_{j \in G} u_{ij} x_{ij}.$$

Finally, we assume that each agent has an initial endowment of these goods; the total amount of each good possessed by the agents is 1 unit. The problem is to find prices for these goods so that if each agent sells her entire initial endowment at these prices and uses the money to buy an optimal bundle of goods, the market clears, i.e., there is no deficiency or surplus of any good.

W.l.o.g. we may assume that each good is desired by at least one agent and each agent desires at least one good, i.e.,

$$\forall j \in G, \exists i \in B : u_{ij} > 0 \quad \text{and} \quad \forall i \in B, \exists j \in G : u_{ij} > 0.$$

If not, we can remove the good or the agent (together with the her initial endowments) from consideration.

In [Vaz12], we explored a different solution concept for this setting: for each agent i , compute the utility she accrues from her initial endowment, say c_i . Let \mathcal{N} in \mathbf{R}_+^n denote the set of all possible utility vectors obtained by distributing the goods among the agents in all possible ways. Now seek the Nash bargaining solution for instance $(\mathcal{N}, \mathbf{c})$. The setup was made more general by assuming that c_i 's are arbitrary numbers given with the problem instance, i.e., they do not come from initial endowments. Clearly, in this more general form, the given game may be infeasible. Hence, the problem is to first determine feasibility and, if the given game is feasible, to find its Nash bargaining solution.

Let **ADNB2** denote the restriction of this game to 2 players. We will assume these are non-symmetric games, i.e., we are also given the clout, w_1 and w_2 of the two players. We give a combinatorial strongly polynomial algorithm for this game; the algorithm in [Vaz12] is not strongly polynomial.

The bargaining solution to **ADNB2** is the optimal solution to the following convex program:

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1,2} w_i \log(v_i - c_i) & (5) \\ \text{subject to} \quad & \forall i = 1, 2 : \quad v_i = \sum_{j \in G} u_{ij} x_{ij} \\ & \forall j \in G : \quad \sum_{i=1,2} x_{ij} \leq 1 \\ & \forall i = 1, 2, \forall j \in G : \quad x_{ij} \geq 0 \end{aligned}$$

4.3 The circle game

The circle game lies in (NB2 - LNB2). Its feasible set is the intersection of the unit disk with the positive orthant. We will consider only its Nash bargaining version. Its convex program is:

$$\begin{aligned}
& \text{maximize} && \sum_{i=1,2} \log(v_i - c_i) && (6) \\
& \text{subject to} && v_1^2 + v_2^2 \leq 1 \\
& && \forall i = 1, 2 : v_i \geq 0
\end{aligned}$$

5 Rational convex programs

A *rational convex program (RCP)* is a nonlinear convex program such that for any setting of its parameters to rational numbers such that it has a finite optimal solution, it admits an optimal solution that is rational and can be written using polynomially many bits in the number of bits needed to write all the parameters [Vaz12].

[Vaz12] identifies 2 classes of RCPs: quadratic and logarithmic. A convex program in either class has only linear constraints; its nonlinearity lies in its objective function. First assume that the objective function is of the form $x^T P x + q^T x$ where P is a positive semidefinite matrix. If so, the convex program is rational and this gives us the class of *quadratic RCPs*.

Next, assume that the objective function is of the form

$$f_0(x) = - \sum_{i=1}^n c_i \log g_i(x),$$

where g_1, \dots, g_n are linear functions and c_i 's are constants. Clearly, this yields a convex program; however, it is not always rational. For specific choices of the functions g_1, \dots, g_n and the linear constraints, it turns out to be an RCP, and this needs to be established piecemeal. This yields the class of *logarithmic RCPs*.

6 An algorithm for LNB2, using an LP-solver

In this section, we will obtain our main result, that any game in LNB2 can be solved in polynomial time, given an LP-solver. Let \mathcal{G} be a game in LNB2 whose solution is captured by convex program (3); for convenience we restate this program below.

$$\begin{aligned}
& \text{maximize} && \sum_{i=1,2} w_i \log(v_i - c_i) && (7) \\
& \text{subject to} && \mathbf{A}\mathbf{x} + \mathbf{b}_1 v_1 + \mathbf{b}_2 v_2 \leq \mathbf{q} \\
& && \text{for } i = 1, 2 : v_i \geq c_i \\
& && \mathbf{x} \geq 0
\end{aligned}$$

where \mathbf{A} is an $m \times r$ matrix, \mathbf{x} is a vector consisting of a total of r allocation and auxiliary variables, and $\mathbf{b}_1, \mathbf{b}_2, \mathbf{q}$ are arbitrary m -dimensional vectors. We can test if \mathcal{G} is feasible by solving the following LP:

$$\text{maximize} \quad t \quad (8)$$

$$\begin{aligned}
& \text{subject to} && v_1 \geq c_1 + t \\
& && v_2 \geq c_2 + t \\
& \text{subject to} && \mathbf{Ax} + \mathbf{b}_1 v_1 + \mathbf{b}_2 v_2 \leq \mathbf{e} \\
& && \mathbf{x} \geq 0
\end{aligned}$$

Now, \mathcal{G} is feasible iff the optimal value of t is greater than 0. Henceforth, assume that \mathcal{G} is feasible.

Next, we make the following change of variables,

$$\text{for } i = 1, 2 : \quad y_i = v_i - c_i,$$

hence obtaining the following program which is equivalent to (7).

$$\begin{aligned}
& \text{maximize} && \sum_{i=1,2} w_i \log y_i && (9) \\
& \text{subject to} && \mathbf{Ax} + \mathbf{b}_1(y_1 + c_1) + \mathbf{b}_2(y_2 + c_2) \leq \mathbf{e} \\
& && \text{for } i = 1, 2 : \quad y_i \geq 0 \\
& && \mathbf{x} \geq 0
\end{aligned}$$

Henceforth, we will denote $(\mathbf{e} - c_1 \mathbf{b}_1 - c_2 \mathbf{b}_2)$ by \mathbf{d} . We will denote by Π the polyhedron in \mathbf{R}^{n+2} which is defined by the constraints of program (9). In this paper, we will write the constraints of (9) concisely as follows. This notation will also be used for LPs optimizing over the polytope Π .

$$\begin{aligned}
& \text{maximize} && \sum_{i=1,2} w_i \log y_i && (10) \\
& \text{subject to} && (\mathbf{x}, y_1, y_2) \in \Pi
\end{aligned}$$

6.1 The Feasible Polytope and its Useful Faces

The projection of Π onto the coordinates y_1, y_2 gives a polytope, \mathcal{N} in \mathbf{R}^2 , which we will call the *feasible polytope*. This is precisely the feasible set of this Nash bargaining game, as defined in Section 2. In this section, we will describe the *useful faces* of this polytope, i.e., faces on which the solution to \mathcal{G} can lie.

We first compute the point (l_1, l_2) by first maximizing y_1 over Π to get l_1 and then maximizing y_2 over Π , subject to $y_1 = l_1$, to get l_2 . Similarly, compute the point (h_1, h_2) by first maximizing y_2 over Π to get h_2 and then maximizing y_1 over Π , subject to $y_2 = h_2$, to get h_1 . Clearly, both these points are vertices of \mathcal{N} . The set of faces encountered in moving, on the boundary of \mathcal{N} , from (l_1, l_2) to (h_1, h_2) , by increasing the second coordinate are the useful faces. If polytope \mathcal{N} is not full dimensional, we will already obtain the vertex or facet on which the solution lies. For the rest of this section, assume that \mathcal{N} is full dimensional.

Each of the useful facets has the form

$$y_1 + \alpha y_2 = \beta,$$

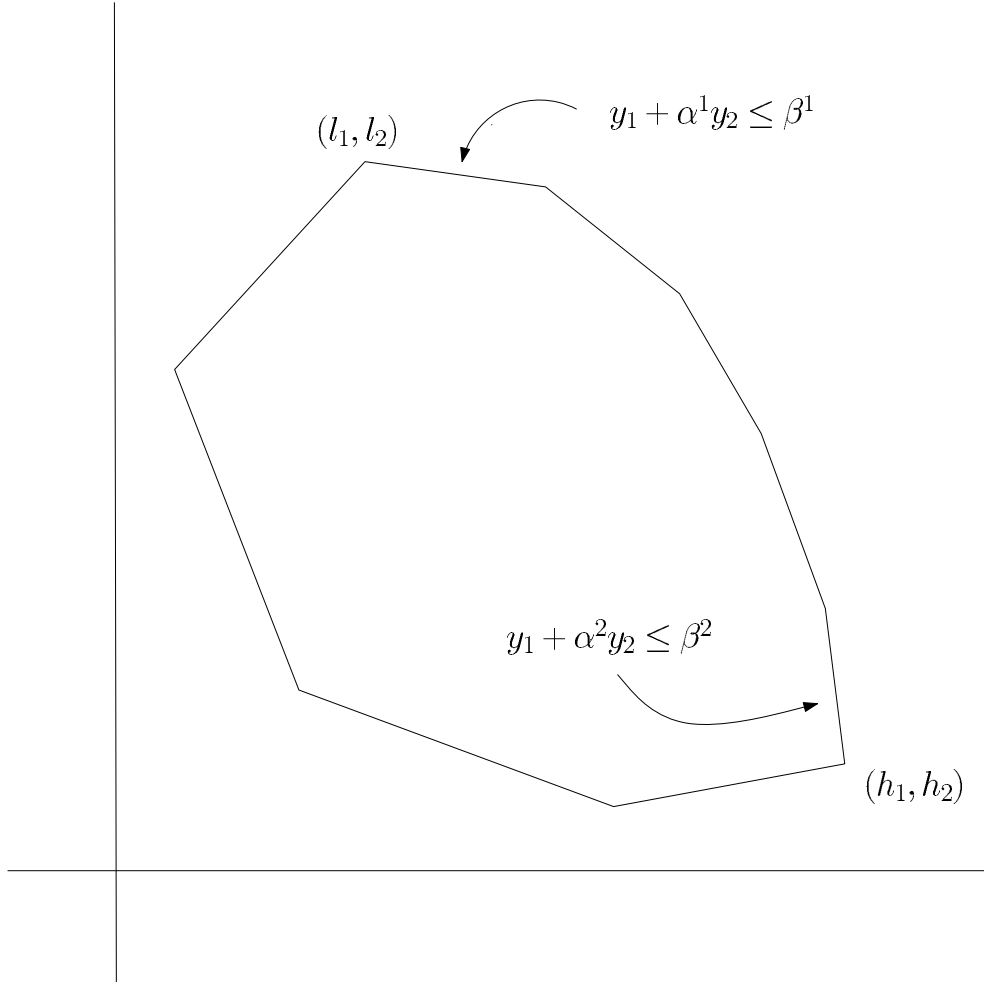


Figure 1: The feasible polytope and its useful faces. In this figure, the y_1 -axis is vertical and the y_2 -axis is horizontal.

where $\alpha > 0$ and $\beta > 0$. We will denote a useful vertex which lies at the intersection of the two facets

$$y_1 + \alpha_1 y_2 = \beta_1 \quad \text{and} \quad y_1 + \alpha_2 y_2 = \beta_2,$$

by (α_1, α_2) ; we will assume $\alpha_1 < \alpha_2$.

Let α^1 and β^1 (α^2 and β^2) be the α and β values of the first (last) facet encountered in moving from (l_1, l_2) to (h_1, h_2) ; clearly, $\alpha^1 < \alpha^2$. Our algorithm performs a binary search on α in the interval $[\alpha^1, \alpha^2]$. In Procedure 3 below, we show how to compute α^1 and α^2 .

Assume there are τ useful faces; clearly, τ is odd, since the first and last faces are facets. Now

we will partition the interval

$$\left[\frac{h_1}{\beta^2}, \frac{l_1}{\beta^1} \right]$$

into τ sub-intervals, one corresponding to each useful face; their importance will become clear in Lemma 3.

Let F be a useful face. If F is the facet $y_1 + \alpha y_2 = \beta$ having endpoints (a_1, b_1) and (a_2, b_2) , with $a_1 < a_2$, then the interval corresponding to it is

$$\text{Int}(F) = \left[\frac{a_1}{\beta}, \frac{a_2}{\beta} \right].$$

If F is the vertex (α_1, α_2) having coordinates (a, b) , which is at the intersection of facets $y_1 + \alpha_1 y_2 = \beta_1$ and $y_1 + \alpha_2 y_2 = \beta_2$, then the interval corresponding to it is

$$\text{Int}(F) = \left(\frac{a}{\beta_2}, \frac{a}{\beta_1} \right).$$

6.2 A Test for the “Right” Face

In this section, we will state the test made in each iteration of the binary search to determine if the face containing the solution has been found. Let (f_1^*, f_2^*) be the solution to game \mathcal{G} . For player i define

$$\gamma_i = \frac{f_i^*}{w_i}, \quad \text{and} \quad z = \frac{\gamma_1}{\gamma_2}.$$

The next lemma relates z to the α value of the face on which the solution lies.

Lemma 2 *If the solution to game \mathcal{G} lies on:*

1. *the facet $y_1 + \alpha y_2 = \beta$, then $z = \alpha$.*
2. *the vertex (α_1, α_2) , then $\alpha_1 < z < \alpha_2$.*

Proof : In the first case, the objective function of the convex program (4),

$$g = w_1 \log y_1 + w_2 \log y_2$$

must be tangent to the facet at the solution point, say (a, b) . Equating the ratio of the partial derivatives of g and the line $y_1 + \alpha y_2 = \beta$ w.r.t. y_2 and y_1 , we get

$$\frac{a/w_1}{b/w_2} = \alpha.$$

But the l.h.s. is $\gamma_1/\gamma_2 = z$, thereby giving $z = \alpha$.

In the second case, the tangent to g at the solution must be intermediate between the slopes of the adjacent facets, giving $\alpha_1 < z < \alpha_2$. \square

Our algorithm will conduct a binary search on α , on the interval $[\alpha^1, \alpha^2]$, to find the face on which the solution lies. The test given in the next lemma helps determine, in each iteration, if the current face is the right one. It also indicates how to shrink the search interval if the current face is not the right one.

Lemma 3 *Let F be the face on which the solution lies. Then,*

$$\frac{w_1}{w_1 + w_2} \in \text{Int}(F).$$

Proof : Consider the two cases:

1. **Case 1:** The solution to game \mathcal{G} , (f_1^*, f_2^*) , lies on the facet $y_1 + \alpha y_2 = \beta$, having endpoints (a_1, b_1) and (a_2, b_2) , with $a_1 < a_2$. Substituting $w_i = f_i/\gamma_i$ and using the fact that $\alpha = \gamma_1/\gamma_2$, which follows from Lemma 2, we get

$$\frac{w_1}{w_1 + w_2} = \frac{f_1}{f_1 + \alpha f_2} = \frac{f_1}{\beta} \in \left[\frac{a_1}{\beta}, \frac{a_2}{\beta} \right].$$

2. **Case 2:** The solution to game \mathcal{G} lies on the vertex (α_1, α_2) , having coordinates (a, b) , which is at the intersection of facets $y_1 + \alpha_1 y_2 = \beta_1$ and $y_1 + \alpha_2 y_2 = \beta_2$. By Lemma 2, $\gamma_1/\gamma_2 \in (\alpha_1, \alpha_2)$, and this leads to the interval in which $w_1/(w_1 + w_2)$ lies. \square

6.3 Some Basic Procedures for Operating on the Feasible Polytope

We will give some basic procedures for operating on the useful faces of \mathcal{N} , e.g., given a particular value of α , find the facet or vertex corresponding to it. These procedures will be used in the algorithm, which performs a binary search on α . The two possibilities of whether the solution lies on a facet or a vertex give rise to distinct procedures and proofs throughout.

6.3.1 Procedure 1: Given α , find the face it lies on

We give an algorithm for the following task: Given a number α s.t. $\alpha^1 \leq \alpha \leq \alpha^2$, determine which of the following possibilities holds:

1. α defines a facet of \mathcal{N} , $y_1 + \alpha y_2 = \beta$, for a suitable value of β . If so, find this facet.
2. There is a vertex of \mathcal{N} , (α_1, α_2) , such that $\alpha_1 < \alpha < \alpha_2$. If so, find this vertex.

First solve the following LP and let its optimal objective function value be denoted by β and let a and b denote the optimal values of y_1 and y_2 , respectively.

$$\begin{aligned}
& \text{maximize} && y_1 + \alpha y_2 && (11) \\
& \text{subject to} && (\mathbf{x}, y_1, y_2) \in \Pi
\end{aligned}$$

Having computed β , solve the following LP and let its objective function value be denoted by a_1 .

$$\begin{aligned}
& \text{minimize} && y_1 && (12) \\
& \text{subject to} && y_1 + \alpha y_2 = \beta \\
& && (\mathbf{x}, y_1, y_2) \in \Pi
\end{aligned}$$

Next, change the objective in LP (12) to maximize y_1 , and let its optimal objective function value be a_2 . If $a_1 < a_2$, we are in the first case. Define $b_1 = (\beta - a_1)/\alpha$ and $b_2 = (\beta - a_2)/\alpha$. Then, the endpoints of the facet $y_1 + \alpha y_2 = \beta$ are (a_1, b_1) and (a_2, b_2) . Otherwise, $a_1 = a_2 = a$, say, and we are in the second case. Let b be the value of y_2 computed in LP (12). Then, the vertex has coordinates (a, b) .

Next, we need to find α_1 and α_2 for this vertex. Let us begin by writing the dual for LP (11).

$$\begin{aligned}
& \text{minimize} && \sum_j d_j p_j && (13) \\
& \text{subject to} && \sum_j b_{1j} p_j \geq 1 \\
& && \sum_j b_{2j} p_j \geq \alpha \\
& && \text{for } 1 \leq i \leq n : \sum_j A_{ji} p_j \geq 0 \\
& && \text{for } 1 \leq j \leq m : p_j \geq 0
\end{aligned}$$

Let $(\mathbf{x}^*, y_1^*, y_2^*)$ be an optimal solution to LP (11). Since \mathcal{G} has been assumed to be feasible, $y_1^* > 0$ and $y_2^* > 0$. The next LP is derived from LP (13) by adding constraints on p_j which are implied by the complementary slackness conditions of the primal and dual pair of LP's (11) and (13). It is not optimizing any function, since we are only concerned with its feasible solutions.

$$\begin{aligned}
& \sum_j b_{1j} p_j = 1 && (14) \\
& \sum_j b_{2j} p_j = r \\
& \text{for } 1 \leq i \leq n : \sum_j A_{ji} p_j \geq 0 \\
& \text{for } 1 \leq i \leq n \text{ s.t. } x_i^* > 0 : \sum_j A_{ji} p_j = 0 \\
& \text{for } 1 \leq j \leq m \text{ s.t. } \sum_i A_{ji} x_i^* + b_{ij} y_1^* + b_{2j} y_2^* < d_j : p_j = 0 \\
& \text{for } 1 \leq j \leq m : p_j \geq 0
\end{aligned}$$

The next lemma follows from the complementary slackness conditions of the primal and dual pair of LP's (11) and (13).

Lemma 4 $\{\alpha \mid LP (11) \text{ attains its optimal solution at } (a, b)\}$
 $= \{r \mid \exists \text{ a feasible solution to LP (14) in which } \sum_j b_{2j}p_j = r\}$.

Proof : Let C and D denote the sets on the l.h.s. and r.h.s. of the equality, respectively. Let $f \in C$. Clearly, (\mathbf{x}^*, a, b) is an optimal solution to LP (11) with α substituted by f . Let \mathbf{p}^* be an optimal solution to LP (13). Since \mathbf{p}^* satisfies all complementary slackness conditions of LP's (11) and (13), it is a feasible solution to LP (14) with r replaced by f . Hence $f \in D$ and $C \subseteq D$.

The reverse inclusion follows in a similar manner, again using complementary slackness conditions of LP's (11) and (13). \square

By Lemma 4, we can obtain α_1 and α_2 as follows. First, minimize r subject to the constraints of LP (14); this gives α_1 . Next, maximize r subject to the constraints of LP (14); this gives α_2 .

6.3.2 Procedure 2: Given (a, b) , find the face it lies on

Given a point (a, b) on the boundary of \mathcal{N} , we give a procedure for finding the facet or vertex it lies on. First, assume that the two buyers' utilities are a and b , respectively, and solve LP (15) to get the corresponding feasible allocation.

$$\begin{aligned} y_1 &= a \\ y_2 &= b \\ (\mathbf{x}, y_1, y_2) &\in \Pi \end{aligned} \tag{15}$$

Next, solve the minimization and maximization versions, with objective function r , of LP (14) to find α_1 and α_2 , respectively. If $\alpha_1 = \alpha_2 = \alpha$, (a, b) lies on the facet $y_1 + \alpha y_2 \leq a + \alpha b$. Otherwise, $\alpha_1 < \alpha_2$ and (a, b) lies on the vertex (α_1, α_2) .

6.3.3 Procedure 3: Computing α^1 and α^2

We now show how to compute α^1 and α^2 , defined at the beginning of this section. As stated there, our binary search will be performed on the interval $[\alpha^1, \alpha^2]$.

First, use Procedure 2 to find the vertex, say (α_1, α_2) , on which (l_1, l_2) lies. Set, $\alpha^1 \leftarrow \alpha_1$. Next, use Procedure 2 to find the vertex, say (α_1, α_2) , on which (h_1, h_2) lies. Set, $\alpha^2 \leftarrow \alpha_2$.

6.4 Binary Search on α

The algorithm, which performs a binary search on α , is presented in Algorithm 6. The operation in Step 2, $\lfloor x \rfloor_\kappa$, truncates x to accuracy $2^{-\kappa}$ where κ is defined in the proof of Lemma 5.

Lemma 5 *The binary search terminates in polynomial in n iterations.*

Proof : First, we place an upper bound on the size of the interval $[\alpha^1, \alpha^2]$. By Cramer's rule, the number of bits in the solution to LP (14) is polynomial in n . Let this number be κ . Therefore, for each of the facets, α can be written in κ bits. However, we do not know where the binary point lies. So, let us assume that we will only deal with 2κ bit long numbers, with κ bits before and κ bits after the binary point. The operation in Step 2 in Algorithm 6, $\lfloor x \rfloor_\kappa$, is meant to truncate x to this form. Therefore, the size of the interval is bounded by $2^{2\kappa}$. Hence binary search will execute $O(\kappa)$, i.e., polynomial in n iterations. \square

Algorithm 6 (Binary Search)

1. **(Initialization:)** $l \leftarrow \alpha^1$ and $h \leftarrow \alpha^2$.

$$\text{Let } r \leftarrow \frac{w_1}{w_1 + w_2}.$$

2. $\alpha \leftarrow \lfloor \frac{l+h}{2} \rfloor_\kappa$.

3. Using Procedure 1 (Section 6.3.1), determine if α lies on:

Case 1: A facet, say $y_1 + \alpha y_2 = \beta$, with endpoints (a_1, b_1) and (a_2, b_2) , with $a_1 < a_2$.

If $r > (a_2/\beta)$ then $l \leftarrow \alpha$ and go to step 2.

Else if $r < (a_1/\beta)$ then $h \leftarrow \alpha$ and go to step 2.

Else if $r \in \left[\frac{a_1}{\beta}, \frac{a_2}{\beta} \right]$, then solve the following 2 equations for y_1 and y_2 :

$$y_1 + \alpha y_2 = \beta \quad \text{and} \quad \frac{y_1/w_1}{y_2/w_2} = \alpha.$$

Let the solution be $y_1 = a, y_2 = b$.

Output the solution: $v_1 = a + c_1$ and $v_2 = b + c_2$, and HALT.

Case 2: A vertex, say (α_1, α_2) , with coordinates (a, b) ,

If $r \geq (a/\beta_2)$ then $l \leftarrow \alpha_2$ and go to step 2.

Else if $r \leq (a/\beta_1)$ then $h \leftarrow \alpha_1$ and go to step 2.

Else if $r \in \left(\frac{a}{\beta_2}, \frac{a}{\beta_1} \right)$,

then output the solution: $v_1 = a + c_1$ and $v_2 = b + c_2$, and HALT.

4. End.

Lemma 7 *Algorithm 6 performs binary search correctly.*

Proof : We first justify restricting search to α -values in the range $[\alpha^1, \alpha^2]$, i.e., the faces encountered on the boundary of \mathcal{N} in moving from (l_1, l_2) to (h_1, h_2) by increasing the second coordinate, as stated in Section 6.1. Observe that starting at (l_1, l_2) and moving in the other direction on the boundary of \mathcal{N} , the second coordinate must decrease and the first either remains the same or decreases, hence decreasing the objective function value of (3). Similarly, moving beyond (h_1, h_2) on the boundary of \mathcal{N} , the first coordinate must decrease and the second either remains the same or decreases, again decreasing the objective function value of (3).

By the necessary and sufficient conditions established in Lemma 3, the algorithm can determine if the current face is the right one or not. Next, observe that value of $w_1/(w_1 + w_2)$ decreases monotonically in moving from (l_1, l_2) to (h_1, h_2) on the boundary of \mathcal{N} . With this observation, one can check that if the current face is not the right one, in each case, the algorithm moves to the side of this face that contains the right face. \square

Hence we get:

Theorem 8 *Every game in LNB2 has a rational solution; moreover, such a solution can be found in polynomial time using an LP-solver.*

Next assume that the coefficients in the constraints of convex program (3) are “small”, i.e., polynomially bounded in n . Then all LP’s that need to be solved will also have “small” coefficients (the objective function and right hand side don’t need to be “small”). Such LP’s can be solved in strongly polynomial time [Tar86]. By Lemma 5, binary search will execute only polynomial in n iterations. Hence we get:

Theorem 9 *Every game in SLNB2 can be solved in strongly polynomial time.*

In particular, the game **DMF2**, which lies in SLNB2, can be solved in strongly polynomial time. [JV10] give examples of Eisenberg-Gale markets with three buyers which do not have rational solutions. In particular, let **DMF3** be the extension of **DMF2** to three players, with 3 source-sink pairs. Consider instances of **DMF3** in which each player’s disagreement utility is zero. One can check that these constitute an Eisenberg-Gale market that is studied in [JV10] and it is shown that this market does not have rational solutions. Hence the game **DMF3**, which is in (LNB - LNB2), does not admit a rational convex program.

7 A Strongly Polynomial Algorithm for the Game ADN2

In this section, we will present a combinatorial, strongly polynomial algorithm for the game **ADN2** which was defined in Section 4.2. The solution to this game is given by the optimal solution to the convex program (5). For convenience, we have restated this program below.

$$\begin{aligned}
& \text{maximize} && \sum_{i=1,2} w_i \log(v_i - c_i) && (16) \\
& \text{subject to} && \forall i = 1, 2: v_i = \sum_{j \in G} u_{ij} x_{ij} \\
& && \forall j \in G: \sum_{i=1,2} x_{ij} \leq 1 \\
& && \forall i = 1, 2, \forall j \in G: x_{ij} \geq 0
\end{aligned}$$

We start by giving the KKT conditions for this convex program. We will assume that the Lagrangian variables corresponding to the inequality constraints are p_j s. Since objective is strictly concave, if the program is feasible, it has a unique dual solution \mathbf{p} .

- (1) $\forall j \in G: p_j \geq 0$.
- (2) $\forall j \in G: p_j > 0 \Rightarrow \sum_{i=1,2} x_{ij} = 1$.
- (3) $\forall i = 1, 2, \forall j \in G: p_j \geq \frac{w_i \cdot u_{ij}}{v_i - c_i}$.
- (4) $\forall i = 1, 2, \forall j \in G: x_{ij} > 0 \Rightarrow p_j = \frac{w_i \cdot u_{ij}}{v_i - c_i}$.

7.1 Linear Fisher and flexible budget market

The KKT conditions given above will help reduce an instance I of game **ADNB2** to an instance of a *flexible budget market*, introduced in [Vaz12]. This is a natural variant of linear Fisher markets [BS00], which we describe first.

Consider a market consisting of a set of n buyers $B = \{1, 2, \dots, n\}$, and a set of g divisible goods, $G = \{1, 2, \dots, g\}$; we may assume w.l.o.g. that there is a unit amount of each good. Let m_i be the money possessed by buyer i , $i \in B$; w.l.o.g. assume that each $m_i > 0$. Let u_{ij} be the utility derived by buyer i on receiving one unit of good j . Thus, if x_{ij} is the amount of good j that buyer i gets, for $1 \leq j \leq g$, then the total utility derived by i is

$$v_i(x) = \sum_{j=1}^g u_{ij} x_{ij}.$$

The problem is to find prices $\mathbf{p} = \{p_1, p_2, \dots, p_g\}$ for the goods so that when each buyer is given her utility maximizing bundle of goods, the market clears, i.e., each good having a positive price is exactly sold, without there being any deficiency or surplus. Such prices are called *market clearing prices* or *equilibrium prices*.

The main difference in a flexible budget market is that instead of having a fixed amount of money to spend in the market, buyers have a (strict) lower bound on the amount of utility they wish to derive, and at any given prices, they want to derive it in the most cost-effective manner. Thus, the money spent by buyers is a function of prices of goods. The objective again is to find market clearing or equilibrium prices.

Next, we give a precise definition of the flexible budget market, $\mathcal{M}(I)$, that instance I of **ADNB2** reduces to. The goods and utility functions of the two buyers are as in I . The disagreement

utility, c_i of agent i is transformed to a parameter c_i giving a strict lower bound on the amount of utility buyer i wants to derive. Given prices \mathbf{p} for the goods, define the *maximum bang-per-buck* of buyer i to be

$$\gamma_i = \max_j \left\{ \frac{u_{ij}}{p_j} \right\}.$$

Now, the amount of money of buyer i is defined to be

$$m_i = w_i + \frac{c_i}{\gamma_i}.$$

Unlike a Fisher market, which always admits an equilibrium, a flexible budget market may not admit one. If it does, we will say that the market is feasible.

Lemma 10 *Instance I is feasible iff $\mathcal{M}(I)$ is. Moreover, if I and $\mathcal{M}(I)$ are both feasible, then allocations \mathbf{x} and dual \mathbf{p} are solutions to instance I iff they are equilibrium allocations and prices for the flexible budget market $\mathcal{M}(I)$.*

Proof : We will use the fact that since convex program (5) has a concave objective and linear constraints, the KKT conditions are both necessary and sufficient for optimality.

(\Rightarrow) First assume that I is feasible and that allocations \mathbf{x} and dual \mathbf{p} are optimal for I . The latter must satisfy KKT conditions for convex program (5).

By the second KKT condition, each good having a positive price is fully sold. Assume that $x_{ij} > 0$. Then, by the definition of γ_i and the fourth KKT condition,

$$\gamma_i = \frac{u_{ij}}{p_j} = \frac{v_i - c_i}{w_i}.$$

The money of buyer i at prices \mathbf{p} in market $\mathcal{M}(I)$ is defined to be $m_i = w_i + c_i/\gamma_i$. The money spent by i in market $\mathcal{M}(I)$ is:

$$\sum_{j \in G} x_{ij} p_j = \frac{w_i}{v_i - c_i} \sum_{j \in G} x_{ij} u_{ij} = \frac{w_i \cdot v_i}{v_i - c_i} = w_i + \frac{c_i w_i}{v_i - c_i} = w_i + \frac{c_i}{\gamma_i} = m_i.$$

Furthermore, by the third and fourth KKT conditions, i buys only her maximum bang-per-buck objects. Hence, she gets an optimal bundle. This proves that \mathbf{x} and \mathbf{p} constitute equilibrium allocations and prices for market $\mathcal{M}(I)$.

(\Leftarrow) Next, assume that $\mathcal{M}(I)$ is feasible and that \mathbf{x} are equilibrium allocations and \mathbf{p} are its unique equilibrium prices. Now, \mathbf{x} is clearly feasible for program (5); we will show that \mathbf{x} and \mathbf{p} satisfy all KKT conditions for this program. The first two conditions are obvious.

Since i gets an optimal bundle of objects at prices \mathbf{p} ,

$$x_{ij} > 0 \Rightarrow \frac{u_{ij}}{p_j} = \gamma_i.$$

Since i spends all her money,

$$m_i = w_i + \frac{c_i}{\gamma_i} = \sum_{j \in G} x_{ij} p_j = \sum_{j \in G} x_{ij} \frac{u_{ij}}{\gamma_i} = \frac{v_i}{\gamma_i}.$$

Therefore, $\gamma_i = \frac{v_i - c_i}{w_i}$. This gives the last two conditions as well. \square

By the KKT conditions, if convex program (5) is feasible, the dual is unique. Hence, in this case, the market $\mathcal{M}(I)$ will have unique equilibrium prices.

7.2 The algorithm

We will first renumber the goods. Compute u_{1j}/u_{2j} for each good j and sort the goods in decreasing order of this ratio. Next, partition the goods by equality, i.e., goods having the same ratio lie in the same partition. We will replace all goods of a partition by one new good. Consider a partition and compute $\min_j \{u_{1j}\}$ for goods j in this partition. Assume the minimum is attained by u_{1k} . Then the utilities of the two players for the new good representing this partition, say g' will be u_{1k} and u_{2k} , respectively. Define the total number of units of g' that are available to be $\sum u_{1j}/u_{1k}$, where the sum over all goods j in the partition.

Let us assume that after this transformation, we have n goods available, $1, 2, \dots, n$ and the amount of good j is b_j and the goods are numbered in decreasing order of u_{1j}/u_{2j} . Clearly, after this transformation, we may have more than a unit amount of certain goods.

Next, we test for feasibility, i.e., we need to determine whether the two players can be given baskets providing c_1 and c_2 utility, respectively, without exhausting all goods. Clearly, the most efficient way of doing this is to give player 1 goods from the lowest index and to give player 2 goods from the highest index (the proof is a straightforward exchange argument, similar to the one given in Lemma 11). Assume that player 1 needs to be given all the goods $1, 2, \dots, k_1 - 1$ in their entirety and an amount x of good k_1 in order to make up c_1 utility. Next, assume that player 2 needs to be given all available goods $n, n - 1, \dots, k_2 + 1$ and an amount y of good k_2 to make up c_2 utility. Then, the game and the market are feasible iff $k_1 < k_2$ or $k_1 = k_2$ and $x + y < b_{k_1}$; observe that the last inequality needs to be strict, since c_1 and c_2 are strict lower bounds on the utilities desired by the two agents. Henceforth, assume that the given market is feasible and let us find an equilibrium for it. The following lemma will be crucial.

Lemma 11 *There are two cases for equilibrium allocation:*

- **Case 1:** *There is a k , $1 \leq k \leq n$ such that player 1 gets goods $1, 2, \dots, k$ and player 2 gets goods $k + 1, k + 2, \dots, n$.*
- **Case 2:** *There is a k , $1 \leq k \leq n$ such that player 1 gets goods $1, 2, \dots, k - 1$, player 2 gets goods $k + 1, k + 2, \dots, n$, and they both share good k .*

Proof : An equilibrium allocation must be an optimal solution to convex program (5). Suppose w.r.t. an allocation, there are goods j and j' , appearing in this order in the sorted order and such that buyer 1 get a non-zero amount of j' and buyer 2 gets a non-zero amount of j .

Then, by redistributing appropriate amounts of these goods, both buyers will get higher utility, hence contradicting the optimality of convex program (5). \square

Since each buyer must get a utility maximizing bundle of goods, we get:

Lemma 12 *Consider an equilibrium allocation to the two buyers. For each good j that is allocated to buyer i ,*

$$\gamma_i = \frac{u_{ij}}{p_j}$$

and for each good j that is not allocated to buyer i ,

$$\gamma_i \geq \frac{u_{ij}}{p_j}.$$

The set of feasible solutions to convex program (5) are closed under convex combinations (they form a polytope). Additionally, the objective function is strictly concave. Therefore, equilibrium utilities, v_1 and v_2 , accrued by the buyers are unique. Hence, only one of the $O(n)$ possibilities of Lemma 11 will satisfy all conditions of Lemma 12. For each possibility, our algorithm will compute corresponding prices to determine whether it is the right one.

Case 1: Let G_1 consist of the first k good and G_2 consist of the rest. Then,

$$\gamma_1 = \frac{u_{1j}}{p_j} \text{ for } j \in G_1 \text{ and } \gamma_2 = \frac{u_{2j}}{p_j} \text{ for } j \in G_2.$$

Let $\gamma_1 = 1/x$ and $\gamma_2 = 1/y$. The total money spent by player 1 is

$$m_1 = \sum_{j \in G_1} p_j b_j = x \sum_{j \in G_1} u_{1j} b_j = w_1 + c_1 x.$$

Similarly, the total money spent by player 2 is

$$m_2 = \sum_{j \in G_2} p_j b_j = y \sum_{j \in G_2} u_{2j} b_j = w_2 + c_2 y.$$

Solve these equations for x and y and compute the prices of goods p_j . If with these prices, each player gets a utility maximizing bundle of goods, i.e., the 2 conditions of Lemma 12 hold, these are equilibrium prices and allocations.

Case 2: Since good k is allocated to both buyers,

$$\gamma_1 = \frac{u_{1k}}{p_k} \text{ and } \gamma_2 = \frac{u_{2k}}{p_k}.$$

Let $u_{1k}/u_{2k} = \alpha$ and $\gamma_1 = 1/x$. Then $\gamma_2 = 1/(\alpha x)$. Let G_1 consist of the first k goods and G_2 consist of the rest. Then the total money spent by both players is

$$m_1 + m_2 = \sum_{j \in G} p_j b_j = x \left(\sum_{j \in G_1} u_{1j} b_j + \sum_{j \in G_2} \alpha u_{2j} b_j \right) = w_1 + c_1 x + w_2 + c_2 \alpha x.$$

Again, solve for x , compute prices of goods and check if the conditions of Lemma 12 are satisfied.

Hence we get:

Theorem 13 *There is a strongly polynomial algorithm for solving **ADNB2**.*

Observe that **ADNB2** is not in **SLNB2**, since the u_{ij} 's are not restricted to be polynomially bounded in n .

8 Extension to the Game **plc-ADNB2**

In the game **ADNB**, each player's utility function was assumed to be linear. Define **plc-ADNB** to be the generalization of **ADNB** to additively-separable (over goods), piecewise-linear, concave utility functions; *plc utilities* for short. An open problem of [Vaz12] asks for a combinatorial polynomial time algorithm for this generalization of **ADNB**, called **plc-ADNB**, in which agents have additively separable (over goods), piecewise-linear, concave utilities. In this section, we will sketch how the algorithm and proof given in Section 7 can be extended to yield a strongly polynomial algorithm for solving **plc-ADNB2**, the restriction of **plc-ADNB** to two players.

We will call the pieces of a buyer's utility function *segments*. Let f_{ij} denote the piecewise-linear, concave utility function of buyer i for good j . The k th segment in f_{ij} will be denoted s_{ijk} and S_{ij} will denote the set of all segments of function f_{ij} . Let u_{ijk} denote the utility accrued by i , per unit of j , corresponding to an allocation made in segment s_{ijk} and let l_{ijk} denote the amount of good represented by this segment.

Since there are exactly two buyers, for each good j , if buyer 1 gets x units, $0 \leq x \leq 1$, then buyer 2 must get $1 - x$ units. This observation motivates the following assumption and definition. We will assume that for each good j , buyer 1 has a segment going from a to b in function f_{1j} , for $0 \leq a < b \leq 1$, iff buyer 2 has a segment going from $(1 - b)$ to $(1 - a)$ in function f_{2j} . Clearly, this condition is easy to ensure by subdividing segments appropriately, and without increasing the number of segments by more than a factor of 2. Next, we will pair up segments in f_{1j} and f_{2j} : the *mate* of segment going from a to b in f_{1j} will be the segment going from $(1 - b)$ to $(1 - a)$ in f_{2j} and vice versa. The mate of segment s_{1jk} will be denoted as $s_{2j\bar{k}}$.

The convex program for **plc-ADNB2**, which generalizes convex program (5), is the following. The indices i, j, k will denote the buyer, good and segment number, respectively. The variable x_{ijk} denotes the amount of good j allocated to i from segment k .

$$\begin{aligned}
& \text{maximize} && \sum_{i=1,2} w_i \log(v_i - c_i) && (17) \\
& \text{subject to} && \forall i = 1, 2 : v_i = \sum_{j \in G} \sum_{k \in S_{ij}} u_{ijk} x_{ijk} \\
& && \forall j \in G : \sum_{i=1,2} \sum_{k \in S_{ij}} x_{ijk} \leq 1 \\
& && \forall i = 1, 2, \forall j \in G, \forall k \in S_{ij} : x_{ijk} \leq l_{ijk} \\
& && \forall i = 1, 2, \forall j \in G, \forall k \in S_{ij} : x_{ijk} \geq 0
\end{aligned}$$

Observe that utility maximization and the plc nature of functions f_{ij} 's implies that optimal allocations will be "left-justified", i.e., if segment $k + 1$ in f_{ij} is allocated then segment k must be fully allocated. Let the dual variable corresponding to the first and second set of inequalities be p_j 's and q_{ijk} 's, respectively. Then the KKT conditions for this program are:

- (1) $\forall j \in G : p_j \geq 0$,
- (2) $\forall i \in B, \forall j \in G, \forall k \in S_{ij} : q_{ijk} \geq 0$,
- (3) $\forall j \in G : p_j > 0 \Rightarrow \sum_{i=1,2} \sum_{k \in S_{ij}} x_{ijk} = 1$.
- (4) $\forall i = 1, 2, \forall j \in G, \forall k \in S_{ij} : q_{ijk} > 0 \Rightarrow x_{ijk} = l_{ijk}$,
- (5) $\forall i = 1, 2, \forall j \in G \forall k \in S_{ij} : p_j + q_{ijk} \geq \frac{w_i \cdot u_{ijk}}{v_i - c_i}$.
- (6) $\forall i = 1, 2, \forall j \in G \forall k \in S_{ij} : x_{ijk} > 0 \Rightarrow p_j + q_{ijk} = \frac{w_i \cdot u_{ijk}}{v_i - c_i}$.

As in the case of **ADNB2**, using these conditions, an instance of **plc-ADNB2** can be transformed to a market model; testing feasibility and if feasible, finding an equilibrium, for the market suffice to solve the instance of **plc-ADNB2**. The main difference is that in this market model, for given prices of goods, the amount of money a buyer pays depends not on the amount of goods she gets but the amount of utility she derives (which in turn depends on the u_{ijk} 's of segments allocated to her). In [GV11], this is interpreted as a perfect price discrimination market model; however, we will not need to go into that interpretation.

Let us say that segment s_{ijk} is *partially allocated* if $0 < x_{ijk} < l_{ijk}$, and it is *fully allocated* if $x_{ijk} = l_{ijk}$. By KKT condition (4), $q_{ijk} > 0$ only if this segment is fully allocated. Define buyer i 's rate to be

$$r_i = \max_{j \in G} \max_{k \in S_{ij}} \left\{ \frac{u_{ijk}}{p_j + q_{ijk}} \right\}.$$

This is the rate at which i accrues utility per dollar spent, at equilibrium.

The following conditions are analogous to those in Lemma 12 for **ADNB2** and characterize equilibrium allocations.

- (i) By KKT condition (6), for a fully allocated segment s_{ijk} ,

$$r_i = \frac{u_{ijk}}{p_j + q_{ijk}} = \frac{v_i - c_i}{w_i}.$$

- (ii) By KKT condition (6), for a partially allocated segment s_{ijk} ,

$$r_i = \frac{u_{ijk}}{p_j} = \frac{v_i - c_i}{w_i}.$$

- (iii) By KKT condition (5), for an unallocated segment s_{ijk} ,

$$r_i \geq \frac{u_{ijk}}{p_j}.$$

Via a proof similar to that in Lemma 10, one can show that the money of buyer i in this market

$$m_i = \sum_{j \in G} \sum_{k \in S_{ij}} x_{ijk}(p_j + q_{ijk}) = w_i + \frac{c_i}{r_i}.$$

Next, we show how to test for feasibility, i.e., if there is a way of distributing goods to the buyers so that each gets strictly more than c_i utility. We will only point out the differences from the process used for **ADNB2**. For each segment s_{1jk} of buyer 1 and its mate, $s_{2j\bar{k}}$, compute $u_{1jk}/u_{2j\bar{k}}$. Sort the segments in decreasing order of this ratio and partition by equality. Assume there are q partitions, numbered 1 to q . Analogous to **ADNB2**, the most efficient way of distributing goods is to allocate segments to buyer 1 from the smallest indices and to buyer 2 from the largest indices. Once again, the buyers can either be allocated separate partitions or they may have to share a “middle” partition.

Henceforth assume that the given instance is feasible. Now, corresponding to Lemma 11 we have the following two cases for an equilibrium allocation:

- **Case 1:** There is an l , $1 \leq l \leq q - 1$ such that player 1 is allocated all segments in partitions $1, 2, \dots, l$ and player 2 is allocated all segments in partitions $l + 1, l + 2, \dots, q$.
- **Case 2:** There is an l , $1 \leq l \leq q$ such that player 1 is allocated all segments in partitions $1, 2, \dots, l - 1$, player 2 is allocated all segments in partitions $l + 1, l + 2, \dots, q$, and they both share goods corresponding to segments in partition l .

As before, equilibrium utilities of the two buyers are unique and therefore only one of these $O(n)$ possibilities will satisfy all conditions stated above. For each possibility, our algorithm will compute corresponding rates and prices to determine whether it is the right one. The only difference is that although the rates are unique, the prices, p_j 's, are not. It is easy to see that if for good j one of the buyers has a partially allocated segment, then by condition (ii) given above, p_j is unique. However, if no buyer has a partially allocated segment corresponding to j , then the constraints on p_j are imposed by conditions (i), (iii) and non-negativity, and p_j can take any value from a fixed range. Once p_j is fixed, the q_{ijk} 's will compensate appropriately to ensure conditions (i) and (iii). Note however, that among all optimal \mathbf{p} 's, the one that maximizes the price of each good is unique. We will say that this price vector is *maximal*. It has the property that for each good j , there is a partial or fully allocated segment, s_{ijk} such that $r_i = u_{ijk}/p_j$. For each of the $O(n)$ allocations, we will compute corresponding rates and maximal prices.

Case 1: Let S_1 consist of all of buyer 1's segments in the first l partitions and S_2 consist of buyer 2's segments in the rest of the $q - l$ partitions. Let U_1 (U_2) be the total utility accrued to buyer 1 (2) from all segments in S_1 (S_2). The total money spent by player i is

$$m_i = \frac{U_i}{r_i} = w_i + \frac{c_i}{r_i}.$$

These equations yield r_1 and r_2 . For each good j , its maximal price is given by

$$p_j = \min_{s_{ijk} \in S_1 \cup S_2} \left\{ \frac{u_{ijk}}{r_i} \right\}.$$

Using these rates and prices, check if all conditions for equilibrium are satisfied.

Case 2: Let S_1 consist of all of buyer 1's segments in the first $l - 1$ partitions and S_2 consist of buyer 2's segments in the last $q - l$ partitions. Let U_1 (U_2) be the total utility accrued to buyer 1 (2) from all segments in S_1 (S_2). Let S be buyer 1's segments in the l th partition. There is full freedom in distributing goods corresponding to segments in S among the two buyers, as long as each buyer gets the right amount of total utility. Therefore, all these segments can be regarded as partially allocated. Hence $q_{1jk} = q_{2j\bar{k}} = 0$, corresponding to each segment $s_{1jk} \in S$, and the maximal price of good j will satisfy:

$$r_1 = \frac{u_{1jk}}{p_j} \quad \text{and} \quad r_2 = \frac{u_{2j\bar{k}}}{p_j}.$$

Let $\alpha = u_{ijk}/u_{2j\bar{k}}$. Then $r_1/r_2 = \alpha$.

Let U denote the utility accrued by buyer 1 if all these segments were allocated to her. Next, we will show that no matter how goods corresponding to segments in S are divided among the two buyers, the total money spent by them on these goods is U/r_1 . Let $s_{1jk} \in S$. If buyer 1 is given x amount of j corresponding to this segment, and buyer 2 is given the remaining, i.e., $l_{1jk} - x$, then the total amount spent by the two buyers is:

$$\frac{x \cdot u_{1jk}}{r_1} + \frac{(l_{1jk} - x) \cdot u_{2j\bar{k}}}{r_2} = \frac{l_{1jk} \cdot u_{1jk}}{r_1},$$

where we replaced $u_{2j\bar{k}}/r_2$ by u_{1jk}/r_1 in the second term in the sum. This justifies the claim that the total money spent by both buyers together on these goods is U/r_1 .

Hence, the total money spent by both players is

$$m_1 + m_2 = (w_1 + \frac{c_1}{r_1}) + (w_2 + \frac{c_2}{r_2}) = \frac{U_1}{r_1} + \frac{U_2}{r_2} + \frac{U}{r_1}.$$

Since $r_1 = \alpha \cdot r_2$, we get an equation in one variable, which yields r_2 , and eventually r_1 . It also yields the maximal prices of all good that are represented in S . For each of the rest of the goods, j , its maximal price is given by

$$p_j = \min_{s_{ijk} \in S_1 \cup S_2} \left\{ \frac{u_{ijk}}{r_i} \right\}.$$

Using these rates and prices, check if all conditions for equilibrium are satisfied.

Hence we get:

Theorem 14 *There is a strongly polynomial algorithm for solving **plc-ADNB2**.*

9 The Circle Game

The circle game was defined in Section 4.3. Its feasible set is the intersection of the unit disk with the positive orthant. We will not consider its nonsymmetric version. Its convex program (6) is restated below.

$$\begin{aligned}
& \text{maximize} && \log(v_1 - c_1)(v_2 - c_2) && (18) \\
& \text{subject to} && v_1^2 + v_2^2 \leq 1 \\
& && \forall i = 1, 2 : v_i \geq 0
\end{aligned}$$

Using the KKT conditions for convex program (6), it is easy to see that the Nash bargaining solution (x, y) satisfies the following equations:

$$(2y^2 - c_2y - 1)^2 = c_1^2(1 - y^2) \quad \text{and} \quad x^2 + y^2 = 1.$$

On the other hand, the solution to the game also has a simple geometric characterization. Let Q be a point on the unit circle in the positive orthant. Let O denote the origin and P denote the point (c_1, c_2) . Let θ_1 be the angle made by PQ with the x -axis and θ_2 be the angle made by OQ with the y -axis.

Proposition 15 *Q is the Nash bargaining solution iff $\theta_1 = \theta_2$.*

Proof : The solution to the circle game is the point on the unit circle that maximizes $(x - c_1)(y - c_2)$. Consider the class of hyperbolas $(x - c_1)(y - c_2) = \alpha$ for different values of α . Clearly, the solution corresponds to the hyperbola in this class that is tangent to the unit circle.

The slope of the tangent to the hyperbola $(x - c_1)(y - c_2) = \alpha$ at (x, y) , taking the ratio of partial derivatives w.r.t. y and x , is

$$\frac{y - c_2}{x - c_1}.$$

Let the point Q be (a, b) and consider the appropriate value of α such that the hyperbola $(x - c_1)(y - c_2) = \alpha$ contains Q. Then, the tangent to this hyperbola at Q is given by $\frac{b - c_2}{a - c_1}$. The slope of the tangent to the circle at Q is $\tan \theta_2$. Since Q is the solution to the circle game, the two tangents must be the same, hence

$$\tan \theta_2 = \frac{b - c_2}{a - c_1}.$$

Let R be the intersection of the vertical line passing through Q and the horizontal line passing through P; clearly, the angle QPR is θ_1 . From the triangle PQR we get that

$$\tan \theta_1 = \frac{b - c_2}{a - c_1}.$$

Hence, $\tan(\theta_1) = \tan(\theta_2)$. Finally, since θ_1 and θ_2 are both in the range $[0, \pi/2]$, we get that $\theta_1 = \theta_2$. \square

10 Acknowledgments

A special thanks to Inbal Talgam for pointing out that my previous approach to Theorem 8 was needlessly complicated and to Leonard Schulman for the pointing out the attractive geometric

characterization of the solution to the circle game.

References

- [BS00] W. C. Brainard and H. E. Scarf. How to compute equilibrium prices in 1891. *Cowles Foundation Discussion Paper*, (1270), 2000.
- [CDT09] X. Chen, X. Deng, and S.-H. Teng. Settling the complexity of computing two-player Nash equilibria. *Journal of the ACM*, 56(3), 2009.
- [CDV10] D. Chakrabarty, N. Devanur, and V. V. Vazirani. New results on rationality and strongly polynomial solvability in Eisenberg-Gale markets. *SIAM Journal on Discrete Mathematics*, 24(3), 2010.
- [DGP09] C. Daskalakis, P. Goldberg, and C. Papadimitriou. The complexity of computing a Nash equilibrium. *SIAM Journal on Computing*, 2009.
- [EG59] E. Eisenberg and D. Gale. Consensus of subjective probabilities: the Pari-Mutuel method. *The Annals of Mathematical Statistics*, 30:165–168, 1959.
- [EY10] K. Etessami and M. Yannakakis. On the complexity of Nash equilibria and other fixed points. *SIAM Journal on Computing*, 39(6), 2010.
- [GLS88] M. Grotschel, L. Lovasz, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag, 1988.
- [GV11] G. Goel and V. V. Vazirani. A perfect price discrimination market model with production, and a rational convex program for it. *Mathematics of Operations Research*, 36, 2011.
- [JV10] K. Jain and V. V. Vazirani. Eisenberg-Gale markets: Algorithms and game-theoretic properties. *Games and Economic Behavior*, 70(1), 2010.
- [Kal77] E. Kalai. Nonsymmetric Nash solutions and replications of 2-person bargaining. *International Journal of Game Theory*, 6(3):129–133, 1977.
- [Kal85] E. Kalai. Solutions to the bargaining problem. In L. Hurwicz, D. Schmeidler, and H. Sonnenschein, editors, *Social Goals and Social Organization*, pages 75–105. Cambridge University Press, 1985.
- [Kel97] F. P. Kelly. Charging and rate control for elastic traffic. *European Transactions on Telecommunications*, 8:33–37, 1997.
- [Nas50a] J. F. Nash. The bargaining problem. *Econometrica*, 18:155–162, 1950.
- [Nas50b] J. F. Nash. Equilibrium points in n -person games. *Proceedings National Academy of Sciences*, 36:48–49, 1950.
- [OR94] M. Osborne and A. Rubinstein. *A Course in Game Theory*. The MIT Press, 1994.

- [Tar86] E. Tardos. A strongly polynomial algorithm to solve combinatorial linear programs. *Operations Research*, 34(2):250–256, 1986.
- [TL89] W. Thomson and T. Lensberg. *Axiomatic Theory of Bargaining With a Variable Population*. Cambridge University Press, 1989.
- [Vaz10] V. V. Vazirani. 2-player Nash and nonsymmetric bargaining: Algorithms and structural properties. In *Lecture Notes in Computer Science, 3rd International Symposium on Algorithmic Game Theory*, volume 6386/2010, pages 323–334, 2010.
- [Vaz12] V. V. Vazirani. The notion of a rational convex program, and an algorithm for the Arrow-Debreu Nash bargaining game. *Journal of the ACM*, 59(2), 2012.
- [vNM44] J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1944.