

2-Player Nash and Nonsymmetric Bargaining Games: Algorithms and Structural Properties

*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 a non-succinct linear game whose solution can be found in strongly polynomial time.

1 Introduction

In game theory, 2-player games occupy a special place – not only because numerous applications involve 2 players but also because they 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 3 or more players may have only irrational solutions; an example of the latter, called “a three-man poker game,” was given by Nash [Nas50b]). Finally, von Neumann’s minimax theorem for 2-player zero-sum games yields a polynomial time algorithm using LP. On the other hand, 3-player zero-sum games are PPAD-hard, since 2-player non-zero-sum games can be reduced to them; the reduction is due to [vNM44] and PPAD-hardness is due to [CDT09].

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].

Recently, Vazirani [Vaz09] initiated a systematic algorithmic study of Nash bargaining games and also carried this program over to solving nonsymmetric bargaining games of Kalai [Kal77]. In this paper we carry the program further, though only for the case of 2-player games. Our

*College of Computing, Georgia Institute of Technology, Atlanta, GA 30332-0280, E-mail: vazirani@cc.gatech.edu.

findings indicate that this case exhibits a rich set of possibilities algorithmically and calls for a further investigation.

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. Two basic classes of these games defined in [Vaz09] are NB and LNB. The convex program for a game in NB admits a polynomial time separation oracle and hence its solution can be obtained to any desired accuracy using the ellipsoid algorithm. All constraints in the convex program for a game in LNB are linear and [Vaz09] gives combinatorial polynomial time algorithms for several games in this class; by a *combinatorial algorithm* we mean an algorithm that performs an efficient search over a discrete space. Let NB2 and LNB2, respectively, be the restrictions of these classes to 2-player games.

We show that for solving any game in LNB2, it is not essential to solve a convex program – an LP solver suffices. As a consequence, all games in LNB2 have rational solutions; this property does not hold for 3-player games in LNB. We then define a subclass of LNB2 called SLNB2, consisting of *succinct* games, i.e., the coefficients in its convex program are “small”. 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 game **DG2**, which consists of a directed graph with edge capacities and each player is a source-sink pair desiring flow (see Section 3 for definition). This game is derived from Kelly’s *flow markets* [Kel97].

Next, we ask if there is a game in (LNB2 - SLNB2) that admits a strongly polynomial time algorithm. The answer turns out to be “yes”. We show that the 2-player version of the game **ADNB**, for which a combinatorial polynomial time algorithm is given in [Vaz09], admits a combinatorial strongly polynomial algorithm. This game is derived from the linear case of the Arrow-Debreu market model (see Section 8 for definition).

Finally, we ask if there is a game in (NB2 - LNB2) that can be solved in polynomial time without a convex program solver. Once again, the answer turns out to be “yes”. We give a game whose solution reduces to solving a degree 4 equation. Alternatively, it also admits an elegant geometric solution.

Our last 2 results raise interesting questions, e.g., is there a characterization of the subclass of LNB2 which consists of all games that admit strongly polynomial algorithms? They also indicate that the class NB2, in particular (NB2 - LNB2), may be worth exploring further algorithmically and structurally, e.g., does (NB2 - LNB2) contain a game that always has a rational solution? And are there 2-player games, not in NB2, whose solution can be computed in polynomial time?

Building on a remarkable convex program of Eisenberg and Gale [EG59], [JV08] gave the notion of *Eisenberg-Gale markets*; see Section 5. In answering an open question of [JV08] affirmatively, [CDV06] showed that EG(2) markets, i.e., the restriction of Eisenberg-Gale markets to 2 buyers, always admit a rational solution and it can be found using only an LP solver. Our first result is obtained by extending their algorithm. For the second result, we use the notion of a *flexible budget market* given in [Vaz09]; see Section 8. We reduce **ADNB2** to such a market and give a combinatorial algorithm for finding an equilibrium in it.

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 agent 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 $\mathbf{v} \in \mathcal{N}$.*

Thus 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 2 conditions given above. For the purposes of computability, we will restrict to rational p_i ’s. 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 $\mathbf{v} \in \mathcal{N}$.

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 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

Before defining the classes NB2 and LNB2, we recall the definition of the classes NB and LNB from [Vaz09]. Let \mathcal{G} be an n -person Nash or nonsymmetric bargaining game whose solution is given by the optimal solution to the following convex program, where \mathbf{x} are m auxiliary variables, the functions f_i are convex. (Clearly, \mathcal{G} is a Nash bargaining game if each $w_i = 1$.)

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

The game \mathcal{G} is said to be in the class NB if each of the k constraints of program (1) can be checked in polynomial time at any given point (\mathbf{v}, \mathbf{x}) . This gives a separation oracle for the program and therefore, using the ellipsoid algorithm, the Nash or nonsymmetric bargaining solution to the game \mathcal{G} can be obtained to any desired accuracy, assuming the game is feasible. Furthermore, \mathcal{G} is feasible iff the optimal solution to the following convex program is > 0 , which can also be checked in polynomial time.

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

The restriction of class NB to 2-player games yields the class NB2.

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 LNB to 2-player games gives us the class LNB2. We will assume w.l.o.g. that the convex program for game \mathcal{G} in LNB2 has the following form:

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

where \mathbf{A} is an $m \times n$ matrix, \mathbf{x} is a vector consisting of allocation and auxiliary variables, say n in number, and $\mathbf{b}_1, \mathbf{b}_2, \mathbf{e}$ are 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 n . 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 3 classes defined above. We will study all 3 games in detail in this paper.

4.1 The game DG2

The game **DG2** 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 in the game and has its own disagreement utility (flow value) c_i , for $i = 1, 2$. In the nonsymmetric version, we are also given the clouts w_1 and w_2 of the two players. The object is to find the Nash or nonsymmetric bargaining solution. Let \mathcal{G} denote the given instance of **DG2**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 [Vaz09]. 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.

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 .

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 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 question 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.

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 from consideration.

In [Vaz09], 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.

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 [Vaz09] is not strongly polynomial.

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

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

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:

$$\text{maximize} \quad \sum_{i=1,2} \log(v_i - c_i) \quad (6)$$

$$\begin{aligned} \text{subject to} \quad & v_1^2 + v_2^2 \leq 1 \\ & \forall i = 1, 2: v_i \geq 0 \end{aligned}$$

5 Fisher and Eisenberg-Gale Market Models

We will first state Fisher's market model for the case of linear utility functions [BS00]. 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 following is the Eisenberg-Gale convex program. Using KKT conditions, one can show that its optimal solution is an equilibrium allocation for Fisher's linear market and the Lagrange variables corresponding to the inequalities give equilibrium prices of goods (e.g., see Theorem 5.1 in [Vaz07]).

$$\begin{aligned} \text{maximize} \quad & \sum_{i \in B} m_i \log v_i & (7) \\ \text{subject to} \quad & \forall i \in B: v_i = \sum_{j \in G} u_{ij} x_{ij} \\ & \forall j \in G: \sum_{i \in B} x_{ij} \leq 1 \\ & \forall i \in B, \forall j \in G: x_{ij} \geq 0 \end{aligned}$$

Next, we state the definition of Eisenberg-Gale markets as given in [JV08]. Let us say that a convex program is an *Eisenberg-Gale-type* convex program if its objective function is of the form

$$\max \sum_{i \in B} m_i \log v_i,$$

subject to linear packing constraints, i.e., constraints of the form \leq in which all coefficients and the r.h.s. are non-negative. Let \mathcal{M} be a Fisher market, with an arbitrary utility function, whose set of feasible allocations and buyers' utilities are captured by a polytope Π . We will assume that the linear constraints defining Π are packing constraints. As a result, \mathcal{M} satisfies the *free disposal property*, i.e., if \mathbf{v} is a feasible utility vector then so is any vector dominated by \mathbf{v} . We will say that an allocation x_1, \dots, x_n made to the buyers is a *clearing allocation* if it

uses up all goods exactly to the extent they are available in \mathcal{M} . Finally, we will say that \mathcal{M} is an *Eisenberg-Gale market* if any clearing allocation x_1, \dots, x_n that maximizes

$$\max \sum_{i \in B} m_i \log v_i(x_i)$$

is an equilibrium allocation, i.e., there are prices p_1, \dots, p_g for the goods such that for each buyer i , x_i is a utility maximizing bundle for i at these prices. The class EG(2), defined in [CDV06], is essentially the restriction of Eisenberg-Gale markets to the case of 2 buyers; see [CDV06] for the precise definition.

6 The Feasible Polytope and Some Basic Procedures

Let \mathcal{G} be a game in LNB2 whose solution is captured by convex program (3). We can test if \mathcal{G} is feasible by solving the following LP:

$$\begin{aligned} & \text{maximize} && t && (8) \\ & \text{subject to} && v_1 \geq c_1 + t \\ & && v_2 \geq c_2 + t \\ & \text{subject to} && \mathbf{Ax} + \mathbf{b}_1 v_1 + \mathbf{b}_2 v_2 \leq \mathbf{e} \\ & && \mathbf{x} \geq 0 \end{aligned}$$

Now, \mathcal{G} is feasible iff the optimal value of $t > 0$. 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 (3).

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

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 LP's optimizing over the polytope Π .

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

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*; since \mathcal{G} is feasible, this polytope is full dimensional. In this section, we will describe its *useful facets*, i.e., facets on which the solution to game \mathcal{G} can lie, and we will give some basic procedures for operating on these facets.

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} . Finally, the set of facets 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 facets.

Each of the useful facets has the form

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

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

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

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

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

The solution to \mathcal{G} must lie on a face which is either a useful facet or a vertex at the intersection of 2 useful facets. These 2 possibilities give rise to distinct procedures and proofs throughout.

6.1 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} , $y_1 + \alpha y_2 \leq \beta$, 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.

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

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

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

Proof : Let A and B denote the sets on the l.h.s. and r.h.s. of the equality, respectively. Let $f \in A$. 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 $f \in B$ and $A \subseteq B$.

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

By Lemma 2, 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 .

6.2 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, solve LP (15) for finding a feasible allocation gives the 2 buyers utilities of $y_1 = a$ and $y_2 = b$.

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

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 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, $\alpha^2 \leftarrow \alpha_2$.

7 Binary Search on Parameter z

We first give some crucial definitions. Let (f_1, f_2) be the solution to game \mathcal{G} . For player i define

$$\gamma_i = \frac{f_i}{m_i}.$$

Define parameter z to be

$$z = \frac{\gamma_1}{\gamma_2}.$$

The next lemma relates z to the point where the solution lies.

Lemma 3 *If the solution to game \mathcal{G} lies on:*

1. *the facet $y_1 + \alpha y_2 \leq \beta$, 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 z , on the interval $[\alpha^1, \alpha^2]$, to find the right 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.

Lemma 4 1. *The solution to game \mathcal{G} lies on the facet $y_1 + \alpha y_2 \leq \beta$, having endpoints (a_1, b_1) and (a_2, b_2) , with $a_1 < a_2$, iff*

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

2. *The solution to market \mathcal{M} lies on the vertex (α_1, α_2) , having coordinates (a, b) , which is at the intersection of facets $y_1 + \alpha_1 y_2 \leq \beta_1$ and $y_1 + \alpha_2 y_2 \leq \beta_2$, iff*

$$\frac{w_1}{w_1 + w_2} \in \left(\frac{a}{\beta_2}, \frac{a}{\beta_1} \right).$$

Proof : In the first case, substituting $w_i = f_i/\gamma_i$ and $\alpha = \gamma_1/\gamma_2$ (this follows from Lemma 3), we get

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

For the other direction, if $w_1/(w_1 + w_2)$ lies in the interval given, then by the equation given above, $f_1 \in [a_1, a_2]$, thereby showing that the solution lies on the facet $y_1 + \alpha y_2 \leq \beta$.

In the second case, by Lemma 3, $\gamma_1/\gamma_2 \in (\alpha_1, \alpha_2)$, and this leads to the interval in which $w_1/(w_1 + w_2)$ lies. The proof of the other direction also follows in the same manner. \square

The operation in Step 2 in Algorithm 6, $\lfloor x \rfloor_\kappa$, truncates x to accuracy $2^{-\kappa}$ where κ is defined in the proof of Lemma 5.

Lemma 5 *Binary search executes 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 6.1), determine if α lies on:

Case 1: A facet, say $y_1 + \alpha y_2 \leq \beta$, with endpoints (a_1, b_1) and (a_2, b_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 :

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

If the solution is $y_1 = a, y_2 = b$, output the solution to game \mathcal{G} :

$v_1 = a + c_1$ and $v_2 = b + c_2$, and HALT.

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

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

Else if $r \geq (a/\beta_1)$ 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 to game \mathcal{G} : $v_1 = a + c_1$ 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. 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 4, 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 only 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 **DG2**, which lies in SLNB2, can be solved in strongly polynomial time. [JV08] give examples of Eisenberg-Gale markets with 3 buyers which do not have rational solutions. Hence, instances of the corresponding Nash bargaining games, with zero disagreement utilities, do not possess rational solutions; this includes the modification of **DG2** to a 3-player game, with 3 source-sink pairs.

8 A Strongly Polynomial Algorithm for the Game ADNB2

We first give the notion of a *flexible budget market*, introduced in [Vaz09]. This is a natural variant on a Fisher market – the main difference 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 object again is to find market clearing or equilibrium prices.

Next, we give a precise definition of the flexible budget market, \mathcal{M} , that **ADNB2** reduces to (the reduction, derived by applying KKT conditions, appears in Theorem 4 in [Vaz09]). The

goods and utility functions of the two buyers are as in **ADNB2** and each buyer i has a parameter c_i giving a strict lower bound on the amount of utility she 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 buyer i spends is defined to be

$$m_i = 1 + \frac{c_i}{\gamma_i}.$$

8.1 The algorithm

We will first renumber the goods. Compute u_{1j}/u_{2j} for each good j , sort the goods in decreasing order of this ratio and partition by equality. For the purpose of this algorithm, it will suffice to replace each partition by one 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 this new good, say g' will be u_{1k} and u_{2k} , respectively. Next, we need to compute the number of units of g' that are available. Each good j in the partition will be represented by u_{1j}/u_{1k} units of g' . The sum over all goods in the partition is the total number of units of this good. 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} .

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. Assume that player 1 needs to be given all the available goods $1, 2, \dots, k_1 - 1$ 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 $x + y < b_{k_1}$.

Finally, assume that the given market is feasible and let us find an equilibrium for it. Since each buyer must get a utility maximizing bundle of goods, for each good j that is allocated to player i ,

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

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

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

This leads to two cases for the 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 .

Since the equilibrium prices are unique, only one of these $O(n)$ possibilities holds. We will check them all to determine which one it is.

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 given above 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 good 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 for equilibrium are satisfied.

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

Theorem 10 *There is a combinatorial strongly polynomial algorithm for solving **ADNB2**.*

9 The Circle Game

Using the KKT conditions of (6) it is easy to show that the Nash bargaining solution (x, y) satisfies the following equations:

$$(2y^2 - c_2 y - 1)^2 = c_1^2 (1 - y^2) \quad \text{and} \quad x^2 + y^2 = 1.$$

On the other hand, the problem also has a simple geometric solution. Let Q be the 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 11 Q is the Nash bargaining solution iff $\theta_1 = \theta_2$.

Proof : Let (a, b) be the point Q and let R be the intersection of the vertical line passing through Q and the horizontal line passing through P . Then the angle QPR is θ_1 .

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

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

From the triangle PQR we get that

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

The slope of the tangent to the circle at Q is $\tan \theta_2$.

By Nash's theorem, Q is the Nash bargaining solution iff the hyperbola $(x - c_1)(y - c_2) = \alpha$ is tangent to the unit circle at point Q , for a suitable value of α . Hence, by the above-stated facts, Q is the Nash bargaining solution iff $\theta_1 = \theta_2$. \square

10 Acknowledgment

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 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.
- [CDV06] D. Chakrabarty, N. Devanur, and V. V. Vazirani. New results on rationality and strongly polynomial solvability in Eisenberg-Gale markets. In *WINE*, 2006.
- [EG59] E. Eisenberg and D. Gale. Consensus of subjective probabilities: the Pari-Mutuel method. *The Annals of Mathematical Statistics*, 30:165–168, 1959.

- [GLS88] M. Grotschel, L. Lovasz, and A. Schirjver. *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag, 1988.
- [JV08] K. Jain and V. V. Vazirani. Eisenberg-Gale markets: Algorithms and game-theoretic properties. *Games and Economic Behavior (to appear)*, 2008.
- [Kal77] E. Kalai. Nonsymmetric Nash solutions and replications of 2-person bargaining. *International Journal of Game Theory*, 6(3):129–133, 1977.
- [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.
- [Tar86] E. Tardos. A strongly polynomial algorithm to solve combinatorial linear programs. *Operations Research*, 34(2):250–256, 1986.
- [Vaz07] V. V. Vazirani. Combinatorial algorithms for market equilibria. In N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, editors, *Algorithmic Game Theory*, pages 103–134. Cambridge University Press, 2007.
- [Vaz09] V. V. Vazirani. Nash bargaining via flexible budget markets. Submitted, 2009.
- [vNM44] J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1944.