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.

The notion of flexible budget markets, introduced in [Vaz09], plays a crucial role in the design of these algorithms.

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-complete, since 2-player non-zero-sum games can be reduced to them; the reduction is due to [vNM44] and PPAD-completeness 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

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

this paper we carry the program further, though only for the case of 2-player games. Our findings indicate that this case exhibits a rich set of possibilities 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. Three basic classes of these games defined in [Vaz09] are NB, LNB, and RNB. 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. As shown in [Vaz09], finding the solution to a game in RNB, a subclass of LNB, reduces to finding the equilibrium in a new market model, called *flexible budget market*. The algorithms in [Vaz09] and in the present paper work by finding equilibria in such markets. Let NB2, LNB2, and RNB2, respectively, be the restrictions of these classes to 2-players 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 10 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?

1.1 Algorithmic contributions

Recently, [CDV06] gave polynomial time algorithms for finding equilibria for Eisenberg-Gale markets with exactly 2 buyers, using only an LP solver. It is easy to see that their algorithm will find the solution to all instances of games in LNB2 in which both players' disagreement utilities are zero (see Section 4). Handling an arbitrary instance of a game in LNB2 requires a non-trivial extension; in this section, we will outline the sequence of algorithmic ideas that led us to the solution.

We first give a polynomial time algorithm for **DG2** using only an LP solver. The KKT conditions of the convex program for this game are difficult to work with. A key step was to use the constraints of a different LP formulation for flow which helps show that this problem is in RNB2. We then use the reduction given in [Vaz09] to obtain the corresponding flexible budget market for this problem. Unlike traditional market models, the money spent by each buyer in this market is not fixed but is a certain function of the prices of goods.

The equilibrium of an Eisenberg-Gale market with 2 buyers occurs on a face of a certain 2-dimensional polytope, called the *flow polytope*. The algorithm of [CDV06] conducts a binary search on the quantity $m_1/(m_1+m_2)$ to determine the correct face, where m_1 and m_2 are the moneys of the 2 buyers in the market.

To find the equilibrium of our flexible budget market also, we also need to determine the correct face of a similar polytope. However, in our case the money, m_1 and m_2, of the two buyer is a function of prices of goods and therefore can be determined only after knowing the correct face. This leads to the proverbial "chicken and egg" problem. We show how to circumvent this problem by conducting the binary search on a different parameter, z, which is the ratio of the rates at which the two players buy flow at equilibrium. It turns out that determining z also requires knowledge of the correct face. Fortunately, given a face, we can determine if it is correct and if not, we show how to determine which of the two sides contains the correct face (see Lemma 9). Hence, binary search can still be made to work. Once the correct face is determined, z and the equilibrium can be computed. Rationality follows as a corollary of the fact that all these computations need only an LP solver.

The KKT conditions of the convex program for an arbitrary game in LNB2 also do not have a nice interpretation. However, the ideas built up for **DG2**, including parameter z, can be extended suitably in the abstract setup of this game to arrive at a similar algorithm. Of course, the latter algorithm could have been given without giving the algorithm for **DG2**; however, this would have rendered the paper devoid of intuition and hard to follow. Hence we decided against it.

2 Nash and Nonsymmetric Bargaining Games

An *n-person Nash bargaining game* consists of a pair $(\mathcal{N}, \boldsymbol{c})$, where $\mathcal{N} \subseteq \mathbf{R}_+^n$ is a compact, convex set and $\boldsymbol{c} \in \mathcal{N}$. Set $\mathcal{N}$ is the *feasible set* and its elements give utilities that the n players can simultaneously accrue. Point $\boldsymbol{c}$ 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 $1, 2, \ldots n$. Game $(\mathcal{N}, \boldsymbol{c})$ is said to be *feasible* if there is a point $\boldsymbol{v} \in \mathcal{N}$ such that $\forall i \in B,\ v_i > c_i$.

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

1. **Pareto optimality:** No point in $\mathcal{N}$ can weakly dominate $\boldsymbol{v}$.

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 $\boldsymbol{v}$ is the solution for $(\mathcal{N}, \boldsymbol{c})$, and $\mathcal{S} \subseteq \mathbf{R}^n_+$ is a compact, convex set satisfying $\boldsymbol{c} \in \mathcal{S}$ and $\boldsymbol{v} \in \mathcal{S} \subseteq \mathcal{N}$, then $\boldsymbol{v}$ is also the solution for $(\mathcal{S}, \boldsymbol{c})$.

Via an elegant proof, Nash proved:

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

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 $\boldsymbol{v} \in \mathcal{N}$. 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 $\Pi_{i \in B}(v_i - c_i)^{p_i}$, over all $\boldsymbol{v} \in \mathcal{N}$, 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 $B, \mathcal{N}, \boldsymbol{c}$ 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 $\Pi_{i \in B}(v_i - c_i)^{w_i}$, over all $\boldsymbol{v} \in \mathcal{N}$.

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.

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 $\boldsymbol{x}$ are m auxiliary variables, the functions f_i are convex and the functions h_i are affine. (Clearly, $\mathcal{G}$ is a Nash bargaining game if each $w_i = 1$.)

$$\text{maximize} \quad \sum_{i \in B} w_i \log(v_i - c_i) \tag{1}$$

[1] The 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 $1 \leq i \leq n$.

$$
\begin{aligned}
\text{subject to} \quad & \text{for } \; i = 1 \dots k: \quad f_i(\boldsymbol{v}, \boldsymbol{x}) \le 0 \\
& \text{for } \; i = 1 \dots l: \quad h_i(\boldsymbol{v}, \boldsymbol{x}) = 0 \\
& \boldsymbol{v} \ge 0 \\
& \boldsymbol{x} \ge 0
\end{aligned}
$$

The game $\mathcal{G}$ is said to be in the class NB if each of the $k+l$ constraints of program (1) can be checked in polynomial time at any given point $(\boldsymbol{v}, \boldsymbol{x})$. 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.

$$
\begin{aligned}
\text{maximize} \quad & t \qquad (2) \\
\text{subject to} \quad & \text{for } \; i = 1 \dots n: \quad v_i \ge c_i + t \\
& \text{for } \; i = 1 \dots k: \quad f_i(\boldsymbol{v}, \boldsymbol{x}) \le 0 \\
& \text{for } \; i = 1 \dots l: \quad h_i(\boldsymbol{v}, \boldsymbol{x}) = 0 \\
& \boldsymbol{v} \ge 0 \\
& \boldsymbol{x} \ge 0
\end{aligned}
$$

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 $\boldsymbol{v}$, 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:

$$
\begin{aligned}
\text{maximize} \quad & \sum_{i=1,2} w_i \log(v_i - c_i) \qquad (3) \\
\text{subject to} \quad & \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}_1 v_1 + \boldsymbol{b}_2 v_2 \le \boldsymbol{d} \\
& \text{for } \; i = 1, 2: \quad v_i \ge 0 \\
& \boldsymbol{x} \ge 0
\end{aligned}
$$

where $\boldsymbol{A}$ is an $m \times n$ matrix, $\boldsymbol{x}$ is a vector consisting of n auxiliary variables and $\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{d}$ are m-dimensional vectors. We will denote by Π the polyhedron in $\mathbf{R}^{n+2}$ which is defined by the constraints of program (3); its projection onto the coordinates of v_1, v_2 give us the feasible set $\mathcal{N}$, which is a polytope in $\mathbf{R}^2$.

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

4 Fisher's Model and Eisenberg-Gale Markets

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 $B = \{1, 2, \ldots, n\}$, and a set of g divisible goods, $G = \{1, 2, \ldots, g\}$; 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, $i \in B$. 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 $1 \le j \le g$, 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 $\boldsymbol{p} = \{p_1, p_2, \ldots, p_g\}$ 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]).

$$\begin{aligned}
\text{maximize} \quad & \sum_{i \in B} m_i \log v_i && (4)\\
\text{subject to} \quad & \forall i \in B: \; v_i = \sum_{j \in G} u_{ij} x_{ij} \\
& \forall j \in G: \; \sum_{i \in B} x_{ij} \le 1 \\
& \forall i \in B, \; \forall j \in G: \; x_{ij} \ge 0
\end{aligned}$$

We will 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 $\le$ 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 P. We will assume that the linear constraints defining P are packing constraints. As a result, $\mathcal{M}$ satisfies the *free disposal property*, i.e., if $\boldsymbol{v}$ is a feasible utility vector then so is any vector dominated by $\boldsymbol{v}$. We will say that an allocation $x_1, \ldots, x_n$ 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 $x_1, \ldots, x_n$ that maximizes

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

is an equilibrium allocation, i.e., there are prices $p_1, \ldots p_g$ for the goods such that for each buyer i, x_i is a utility maximizing bundle for i at these prices.

In general, the variables defining polytope P consist of utility variables for the n buyers and allocation variables. Let $\mathcal{N}$ be the projection of P onto the utility variables. Then the Eisenberg-Gale-type convex program for $\mathcal{M}$ can also be written as

$$\begin{aligned} \text{maximize} \quad & \sum_{i \in B} m_i \log v_i \\ \text{subject to} \quad & \boldsymbol{v} \in \mathcal{N} \end{aligned} \tag{5}$$

If we view $\mathcal{N}$ as the feasible set of a Nash or nonsymmetric bargaining game, then program (5) captures the solution to instances of this game in which the clout of player i is m_i and all disagreement utilities are zero. Hence, an equilibrium of $\mathcal{M}$ is the solution to this bargaining game for these instances.

[CDV06] show that each Eisenberg-Gale market with 2 buyers admits a rational equilibrium and it can be computed with an LP solver. Hence the same holds for instances of 2-person bargaining games having zero disagreement utilities.

5 The Class RNB and Flexible Budget Markets

The class RNB was motivated by the notion of resource allocation markets given by Kelly [Kel97], in the context of modeling and understanding TCP congestion control, and by a slight extension given in [KV] (in which agents may derive different utilities from different objects). RNB is short for *resource allocation Nash and nonsymmetric bargaining games* and is a subclass of LNB.

RNB consists of games of the following form. Let B be a set of agents, $|B| = n$, and G a set of divisible goods, $|G| = g$. Let b_j denote the amount of good $j \in G$ that is available. Each agent can "make" *objects* using the goods. Let T_i denote the set of objects that agent i can make and assume that in order to make one unit of object $k \in T_i$, i must use a_{ijk} amount of good j, for each $j \in G$, and derives utility u_{ik} from one unit of this object. Note that i is allowed to make fractional units of any of the objects. For each $k \in T_i$, if y_{ik} denotes the total amount of object k that i makes, then the total utility derived by i is $\sum_{k \in T_i} u_{ik} y_{ik}$. In some games, T_i may be exponentially large in n and g, though it may be succinctly specified, e.g., game **DG2** in Section 6. The assignments to y_{ik}'s will be called *allocations*. Finally, in the case of nonsymmetric games, let w_i be the clout of $i \in B$.

Consider all possible ways of distributing the goods among the agents, and for each way, the utilities derived by all the agents. Let $\mathcal{N}$ denote the set of all utility vectors obtainable in this manner and let $\boldsymbol{c} \in \mathcal{N}$ denote the disagreement point. The problem is to find the Nash or nonsymmetric bargaining solution. This is captured by the following convex program.

$$\text{maximize} \quad \sum_{i \in B} w_i \log(v_i - c_i) \tag{6}$$

$$\text{subject to} \quad \forall i \in B: \; v_i = \sum_{k \in T_i} u_{ik} y_{ik}$$
$$\forall j \in G: \; \sum_{i \in B} \sum_{k \in T_i} a_{ijk} y_{ik} \leq b_j$$
$$\forall i \in B, \; \forall k \in T_i: \; y_{ik} \geq 0$$

We will say that this program is *feasible* iff the underlying bargaining game is feasible. Let $p_j, \; j \in G$, be the Lagrange variables corresponding to the second set of (packing) constraints; we will interpret these as prices of the goods. Denote the cost of making one unit of object k by agent i at prices $\boldsymbol{p}$ by $P(i, k, \boldsymbol{p})$. Clearly,

$$P(i, k, \boldsymbol{p}) = \sum_{j \in G} a_{ijk} p_j.$$

By the KKT conditions, optimal solutions to y_{ik}'s and p_j's must satisfy:

(1) $\forall j \in G: \; p_j \geq 0.$

(2) $\forall j \in G: \; p_j > 0 \Rightarrow \; \sum_{i \in B} \sum_{k \in T_i} a_{ijk} y_{ik} = b_j.$

(3) $\forall i \in B, \; \forall j \in G: P(i, k, \boldsymbol{p}) \geq \frac{w_i \cdot u_{ik}}{v_i - c_i}.$

(4) $\forall i \in B, \; \forall j \in G: y_{ik} > 0 \Rightarrow P(i, k, \boldsymbol{p}) = \frac{w_i \cdot u_{ik}}{v_i - c_i}.$

Strict concavity of the objective function of program (6) and the fourth KKT condition imply the following in a straightforward manner (e.g., see Theorem 5.1 in [Vaz07]).

Proposition 2 *If convex program (6) is feasible, it has a unique optimal solution $\boldsymbol{v} \in \mathcal{N}$ and $\forall i \in B, \; \forall k \in T_i$, the prices, $P(i, k, \boldsymbol{p})$ are unique.*

Next, given an instance I of a game in RNB, we define its corresponding *flexible budget market*, $\mathcal{M}$. The set of buyers in $\mathcal{M}$ will be B and goods will be G. The parameters w_i, a_{ijk}, u_{ik} and sets T_i will have the same meaning in $\mathcal{M}$ as in I; however, instead of representing the disagreement utility of buyer i, parameter c_i gives a strict lower bound on the utility that buyer i wants to derive.

The special feature of a flexible budget market is that the money of buyers is a function of prices of goods. Relative to prices $\boldsymbol{p}$, define the *maximum bang-per-buck* of buyer i to be

$$\gamma_i = max_{k \in T_i} \left\{ \frac{u_{ik}}{P(i, k, \boldsymbol{p})} \right\}.$$

Now, the money of agent i is defined to be

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

At any given prices, each buyer is interested in maximizing the total utility accrued. Clearly, this will be achieved by spending all money on goods to build maximum bang-per-buck objects.

The problem is to find market clearing prices, i.e., prices such that if each buyer is sold a bundle of goods that maximizes her utility, all goods having positive price are sold exactly, i.e., there is no deficiency or surplus of such goods. Unlike an Eisenberg-Gale market, which always has an equilibrium, assuming some mild conditions, a flexible budget market may not admit an equilibrium. We will say that market $\mathcal{M}$ is *feasible* iff it admits an equilibrium.

Using KKT conditions of program (6), the following can be proved.

Theorem 3 *([Vaz09]) Instance I is feasible iff market $\mathcal{M}$ is feasible. Moreover, if I and $\mathcal{M}$ are both feasible, then allocations $\boldsymbol{y}$ and dual $\boldsymbol{p}$ are optimal for RNB game I iff they are equilibrium allocations and prices for the flexible budget market $\mathcal{M}$.*

6 The Game DG2 and its Flexible Budget Market

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**.

Note 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.

We start by giving the convex program which 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 $\mathcal{N}$.

$$
\begin{aligned}
\text{maximize} \quad & \sum_{i=1,2} w_i \log(f_i - c_i) && (7) \\
\text{subject to} \quad & \text{for } \; i = 1, 2 : \quad f_i = \sum_{e \in \text{out}(s_1)} f_e^i \\
& \forall e \in E : \quad f_e^1 + f_e^2 \le c_e \\
& \text{for } \; i = 1, 2 : \quad \forall v \in V - \{s_i, t_i\} : \quad \sum_{e \in \text{in}(v)} f_e^i = \sum_{e \in \text{out}(v)} f_e^i \\
& \text{for } \; i = 1, 2 : \quad \forall e \in E : \quad f_e^i \ge 0
\end{aligned}
$$

Let $\boldsymbol{f}$ be the vector consisting of all the variables f_e^1, f_e^2 for $e \in E$ and let Π be the polyhedron defined by the constraints of program (7). We will use the following succinct way of writing this program.

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1,2} w_i \log(f_i - c_i) \\ \text{subject to} \quad & (\boldsymbol{f}, f_1, f_2) \in \Pi \end{aligned} \tag{8}$$

We now make an important transformation. By standard flow theory, the following is an equivalent convex program. Let $\mathcal{P}_i$ be the set of all paths from s_i to t_i, for $i = 1, 2$, and for any such path q, let variable f_q denote the flow sent on this path.

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1,2} w_i \log(f_i - c_i) \\ \text{subject to} \quad & \text{for } \ i = 1, 2 : \quad f_i = \sum_{q \in \mathcal{P}_i} f_q \\ & \forall e \in E : \quad \sum_{q \in (\mathcal{P}_1 \cup \mathcal{P}_2) \ s.t. \ e \in q} f_q \leq c_e \\ & \forall q \in (\mathcal{P}_1 \cup \mathcal{P}_2) : \ f_q \geq 0 \end{aligned} \tag{9}$$

Thus in game **DG2** edges in E can be viewed as goods and s_i to t_i paths can be viewed as objects, i.e., **DG2** can be viewed to be in the class RNB. Therefore, we can use Theorem 3 to reduce it to a flexible budget market. Notice however, that whereas convex program (7) has polynomially many constraints and variables, (9) has exponentially many variables and constraints. Hence, all computations will still be done using (7).

The flexible budget market, $\mathcal{M}$, for game **DG2** is: We are given a directed graph $G = (V, E)$ whose edges are the goods in the market. The capacity of edge $e \in E$ is specified by $c_e \in \mathbf{Q}^+$. The market has two agents, $1 = 1, 2$. Corresponding to agent i, we are given a source-sink pair of vertices, (s_i, t_i). Agent i wants to buy flow paths from s_i to t_i and c_i is a strict lower bound on the flow desired by this agent. We are also given the clouts w_1 and w_2 of the two agents. We to find prices for the edges of G, p_e. The money of each agent is a function of these prices in the following manner. Let r_i denote the cost of the cheapest path from s_i to t_i w.r.t. these prices. Then, the money of agent i is

$$m_i = w_i + c_i r_i.$$

We will say that flows and edge prices form an *equilibrium in this market* iff they satisfy the following conditions:

1. Only saturated edges have positive prices.
2. All the flows go over cheapest source-sink paths, w.r.t. edge prices.
3. The flow obtained by each agent fully uses up its money.

Testing feasibility of game $\mathcal{G}$ or market $\mathcal{M}$ involves solving the following LP. Observe that the abbreviated form of the constraints of (7) used in (8) have also been used in this LP. We will use this notation in future as well.

$$\begin{aligned} \text{maximize} \quad & t \\ \text{subject to} \quad & \text{for } i = 1,2: \; f_i \geq c_i + t \\ & (\boldsymbol{f}, f_1, f_2) \in \Pi \end{aligned} \tag{10}$$

Now, $\mathcal{G}$ and $\mathcal{M}$ are feasible iff at optimality, $t > 0$. Henceforth, we will assume that they are both feasible.

7 The Flow Polytope and Some Basic Procedures

The projection of Π onto the coordinates f_1, f_2 gives a polytope in $\mathbf{R}^2$. This polytope contains the set of feasible flows (f_1, f_2) defined by the constraints of program (7). We will it call the *flow polytope* and will denote it by $\mathcal{N}$, since it is also the feasible set of the bargaining game. In this section, we will give some basic procedures for operating on this polytope.

The flow polytope has 2 trivial facets, $f_1 \geq 0$ and $f_2 \geq 0$; we will be concerned with the rest of the facets. [CDV06] show that the latter can be exponentially many for the case of an Eisenberg-Gale market with 2 buyers which, as mentioned in the Introduction, corresponds to an instance of a game in LNB2 in which both players' disagreement utilities are zero.

Among the non-trivial facets, there can be at most one with the form, $f_1 \leq \beta$, for $\beta > 0$. Each of the remaining non-trivial facets has the form

$$f_1 + \alpha f_2 \leq \beta,$$

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

$$f_1 + \alpha_1 f_2 \leq \beta_1 \quad \text{and} \quad f_1 + \alpha_2 f_2 \leq \beta_2,$$

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

The solution to the given game must lie on a face which is either a non-trivial facet or a vertex at the intersection of 2 non-trivial facets. These 2 possibilities give rise to distinct procedures and proofs throughout, including the basic procedures given below.

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

Let α^1 and α^2 be the α values of the two extreme facets of the flow polytope, with $\alpha^1 < \alpha^2$; observe that α^2 may equal ∞. 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 the flow polytope, $f_1 + \alpha f_2 \leq \beta$, for a suitable value of β. If so, find this facet.
2. There is a vertex of the flow polytope, (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 f_1 and f_2, respectively.

$$\begin{aligned} &\text{maximize} && f_1 + \alpha f_2 \\ &\text{subject to} && (\boldsymbol{f}, f_1, f_2) \in \Pi \end{aligned} \tag{11}$$

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

$$\begin{aligned} &\text{minimize} && f_1 \\ &\text{subject to} && f_1 + \alpha f_2 = \beta \\ &&& (\boldsymbol{f}, f_1, f_2) \in \Pi \end{aligned} \tag{12}$$

Next, change the objective in LP (12) to maximize f_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 $f_1 + \alpha f_2 = \beta$ 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 f_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. For this, we first write the dual of LP (11); we will assume that the constraint $(f_1, f_2) \in \mathcal{N}$ is replaced by the polynomially many constraints of convex program (7). In the dual, there is a variable for each edge $e \in E$, d_e, which will be interpreted as the length of this edge. For each vertex $v \in V$, there are two variables, γ_v and δ_v, representing the length of the shortest path from s_1 and s_2, respectively, to v.

$$\begin{aligned} &\text{minimize} && \sum_e c_e d_e \\ &\text{subject to} && \gamma_{s_1} = 0 \quad \gamma_{t_1} \geq 1 \\ &&& \delta_{s_2} = 0 \quad \delta_{t_2} \geq \alpha \\ &&& \forall e = (u, v) \in E: \ \gamma_v - \gamma_u \leq d_e \\ &&& \forall e = (u, v) \in E: \ \delta_v - \delta_u \leq d_e \\ &&& \forall e \in E: d_e \geq 0 \\ &&& \forall v \in V: \gamma_v \geq 0 \\ &&& \forall v \in V: \delta_v \geq 0 \end{aligned} \tag{13}$$

The next LP is derived from LP (13) by adding in constraints on d_e 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 concerned with the set of values that variable x_α can take.

$$\begin{aligned} &\gamma_{s_1} = 0; \quad \gamma_{t_1} = 1 \\ &\delta_{s_2} = 0; \quad \delta_{t_2} = x_\alpha \\ &\forall e = (u, v) \in E: \ \gamma_v - \gamma_u \leq d_e \\ &\forall e = (u, v) \in E: \ \delta_v - \delta_u \leq d_e \end{aligned} \tag{14}$$

$$\forall e = (u, v) \in E \ s.t. \ f^1_{(u,v)} > 0 : \ \gamma_v - \gamma_u = d_e$$
$$\forall e = (u, v) \in E \ s.t. \ f^2_{(u,v)} > 0 : \ \delta_v - \delta_u = d_e$$
$$\forall e \in E \ s.t. \ f^1_e + f^2_e < c_e : \ d_e = 0$$
$$\forall e \in E : d_e \geq 0$$

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

Lemma 4 $\{\alpha \mid LP\ (11)$ *attains its optimal solution at* $(a, b)\} = \{x_\alpha \mid LP\ (14)$ *is feasible*$\}$.

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

7.2 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 way of routing the 2 commodities $f_1 = a$ and $f_2 = b$ and obtain flows on edges f^1_e and f^2_e.

$$f_1 = a \tag{15}$$
$$f_2 = b$$
$$(\boldsymbol{f}, f_1, f_2) \in \Pi$$

Next, solve the minimization and maximization versions, with objective function x_α, of LP (14) to find α_1 and α_2, respectively. If $\alpha_1 = \alpha_2 = \alpha$, (a, b) lies on the facet $f_1 + \alpha f_2 \leq a + \alpha b$. Otherwise, $\alpha_1 < \alpha_2$ and (a, b) lies on the vertex (α_1, α_2).

8 Binary Search on Parameter z

The crucial parameter for our algorithm is

$$z = \frac{r_2}{r_1}.$$

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

Lemma 5 *If the solution to the given game lies on:*

1. *the facet* $f_1 + \alpha f_2 \leq \beta$*, 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 (7),

$$g = w_1 \log(f_1 - c_1) + w_2 \log(f_2 - c_2)$$

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 $f_1 + \alpha f_2 = \beta$ w.r.t. f_2 and f_1, we get

$$\frac{w_2/(b - c_2)}{w_1/(a - c_1)} = \alpha.$$

But the l.h.s. is $r_2/r_1 = z$, thereby giving $z = \alpha$.

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

Our algorithm will find the value of z, and hence the right face where the solution lies, by conducting a binary search on an appropriately chosen interval $[L, H]$, where L and H are defined as follows. Find the unique point b such that (c_1, b) lies on the boundary of the flow polytope by solving the following LP:

$$\begin{aligned} &\text{maximize} && f_2 \\ &\text{subject to} && f_1 = c_1 \\ & && (\boldsymbol{f}, f_1, f_2) \in \Pi \end{aligned} \tag{16}$$

Next, use Procedure 2 of Section 7.2 to determine the facet or vertex on which (c_1, b) lies. If it lies on the facet $f_1 + \alpha f_2 \leq \beta$, then define $H = \alpha$. Else, if it is the vertex (α_1, α_2), then define $H = \alpha_1$.

In a similar manner, find the unique point a such that (a, c_2) lies on the boundary of $\mathcal{N}$ and then find the facet or vertex on which (a, c_2) lies. If the point (a, c_2) lies on the facet $f_1 + \alpha f_2 \leq \beta$, then define $L = \alpha$. Else, if it is the vertex (α_1, α_2), then define $L = \alpha_2$.

The operation in Step 2 in Algorithm 7, $\lfloor x \rfloor_\kappa$, truncates x to accuracy $2^{-\kappa}$ where κ is defined in the proof of Lemma 6.

Lemma 6 *Binary search executes polynomial in n iterations.*

Proof : First, we place an upper bound on the size of the interval $[L, H]$. 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 7, $\lfloor x \rfloor_\kappa$, 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. □

Algorithm 7 (Binary Search)

1. **(Initialization:)** $l \leftarrow L$ and $h \leftarrow H$.
2. $$\alpha \leftarrow \lfloor \frac{l+h}{2} \rfloor_\kappa.$$
3. Using Procedure 1 (Section 7.1), determine if α lies on:

   Case 1: A facet, say $f_1 + \alpha f_2 \le \beta$, with endpoints (a_1, b_1) and $(a_2.b_2)$.

   $$\text{Let } c \leftarrow \frac{w_2}{w_1}\left(\frac{a_1 - c_1}{b_1 - c_2}\right) \text{ and } d \leftarrow \frac{w_2}{w_1}\left(\frac{a_2 - c_1}{b_2 - c_2}\right).$$

   If $\alpha \in [c, d]$, then use Procedure 3 (Section 8.1) to find equilibrium flows and edge prices and HALT.
   Else if $\alpha < c$ then $h \leftarrow c$ and go to step 2.
   Else if $\alpha > d$ then $l \leftarrow d$ and go to step 2.

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

   $$c \leftarrow \frac{w_2(a - c_1)}{w_1(b - c_2)}.$$

   If $\alpha_1 < c < \alpha_2$, then use Procedure 4 (Section 8.2) to find equilibrium flows and edge prices and HALT.
   Else if $c \le \alpha_1$ then $h \leftarrow \alpha_1$ and go to step 2.
   Else if $c \ge \alpha_2$ then $l \leftarrow \alpha_2$ and go to step 2.

4. End.

We next give a proof of correctness of Algorithm 7. Let (d_2, c_2) be the unique point on the boundary of the feasible set, $\mathcal{N}$, having second coordinate c_2. Since the given game is feasible, its f_1 value must lie in the interval (c_1, d_2). Define the following on this interval.

- Function $\Gamma : (c_1, d_2) \to \mathbf{R}_+$ is defined as $h(x) = y$ such that (x, y) lies on the boundary of $\mathcal{N}$. Intuitively, $\Gamma(x)$ gives the f_2 value of the point on the boundary of $\mathcal{N}$ whose f_1 value is x.
- Function $g : (c_1, d_2) \to \mathbf{R}_+$ is defined as

  $$g(x) = \frac{w_2(x - c_1)}{w_1(\Gamma(x) - c_2)}.$$

- Multifunction h on the interval (c_1, d_2). At point $x \in [c_1, d_2)$, if $(x, h(x))$ lies on the facet $f_1 + \alpha f_2 \le \beta$, then $h(x)$ attains the single value α. If $(x, h(x))$ lies on the vertex (α_1, α_2), then $h(x)$ is the interval (α_1, α_2). Intuitively, h gives the α values of facets and vertices.

Observe that h is fully defined by the graph and its edge capacities. On the other hand, g is defined by the parameters w_1, w_2, c_1, c_2 of the given game. Also observe that h is decreasing and

g is monotonically increasing.

Lemma 8 *Let x be the unique point such that $g(x) \in h(x)$. Then the solution to the given game is $(x, \Gamma(x))$.*

Proof : Let the solution of the game be at point (f_1, f_2). By definition of g and z, $g(f_1) = z$. Since h is decreasing and g is monotonically increasing, g and h must "intersect" at a unique point. Finally, by Lemma 5 and the definition of h, the first coordinate of this point must be f_1. □

Lemma 9 *Algorithm 7 performs binary search correctly.*

Proof : Let the solution to the game be (f_1, f_2). Since h is decreasing and g is monotonically increasing, $g(x) < h(x)$ for $x < f_1$ and $g(x) > h(x)$ for $x > f_1$. Therefore by comparing $g(x)$ with $h(x)$, binary search can determine which half contains the solution to the game. □

8.1 Procedure 3: Solution lies on a facet

Solve the following 2 equations for f_1 and f_2:

$$f_1 + \alpha f_2 = \beta$$

$$\frac{w_2/(f_2 - c_2)}{w_1/(f_1 - c_1)} = \alpha.$$

Next, compute

$$r_1 = \frac{w_1}{f_1 - c_1}.$$

Let the solution be $f_1 = a, f_2 = b$. Next, solve LP (13) to obtain d_e's and LP (15) to route flow (a, b) as a valid flow in G. Finally, output the flow (a, b) and edge prices $\forall e \in E : \; p_e = r_1 d_e$.

Lemma 10 *The output of Procedure 3 constitutes equilibrium flows and edge prices for Case 1.*

Proof : We will show that the 3 conditions given in Section 6 are satisfied. The fact that only saturated edges have positive prices and all flows go over cheapest paths follows from complementary slackness conditions of the LP pair (11) and (13).

Finally, observe that the cost of the cheapest s_1, t_1 path is r_1 and that of the cheapest s_2, t_2 path is $\alpha r_1 = r_2$. Therefore, the money spent by the first player is

$$f_1 r_1 = f_1 \frac{w_1}{f_1 - c_1} = w_1 + r_1 c_1 = m_1.$$

Similarly, the money spent by the second player is $f_2 r_2 = m_2$ □

8.2 Procedure 4: Solution lies on a vertex

First, compute

$$z = \frac{w_2(a - c_1)}{w_1(b - c_2)}.$$

Next, solve the following equation for q:

$$z = \alpha_1 q + \alpha_2(1 - q).$$

Solve LP (13) twice, first with α replaced by α_1, then with α replaced by α_2. Let d_e and d'_e denote the two solutions, for each edge $e \in E$. Solve LP (15) to route flow (a, b) as a valid flow in G. Finally, output the flow (a, b) and edge prices $\forall e \in E: \; p_e = r_1(qd_e + (1 - q)d'_e)$.

Lemma 11 *The output of Procedure 4 constitutes equilibrium flows and edge prices for Case 2.*

Proof : Again, we will show that the 3 conditions given in Section 6 are satisfied. Consider the following 2 primal-dual pairs of LP's derived from the primal-dual pair (11) and (13). In the first pair, replace α by α_1 in (11) and (13). In the second pair, replace α by α_2 in (11) and (13). Since (f_1, f_2) lies on a vertex of the polytope, (f_1, f_2) is the optimal solution to the primal LP's in both pairs. By complementary slackness, the flow paths of (f_1, f_2) are the cheapest source-sink paths under both metrics, d_e and d'_e, and the same holds for any convex combination of these two metrics, and therefore also under edge prices p_e. Again, by complementary slackness applied to both pairs of LP's, only saturated edges have positive prices.

Finally, observe that the cost of the cheapest s_1, t_1 path is $r_1(q + (1 - q)) = r_1$ and that of the cheapest s_2, t_2 path is

$$r_1(q\alpha_1 + (1 - q)\alpha_2) = r_1 z = r_2.$$

Therefore, as in Lemma 10, the money spent by the two players is m_1 and m_2, respectively. □

Observe that the coefficients in all LP's that need to be solved are "small", i.e., polynomially bounded in n; in fact, in the case of **DG2**, they are 0/1. Such LP's can be solved in strongly polynomial time [Tar86] (the objective function and right hand side don't need to be "small"). By Lemma 6, binary search will execute only polynomial in n iterations. Furthermore, the solution involves solving LP's and hence must be rational. Hence we get:

Theorem 12 *Algorithm 7 finds the solution to game* **DG2** *in strongly polynomial time. Furthermore, the solution is rational.*

9 Generalizing to an Arbitrary Game in LNB2

Let $\mathcal{G}$ be an arbitrary game in LNB2 whose solution is given by the convex program (3). The KKT conditions of program (3) do not have a nice interpretation; however, we will be able to extending the algorithm obtained for **DG2** to solve $\mathcal{G}$. In this section we mention the changes needed.

As stated in Section 3, the polyhedron in $\mathbf{R}^{n+2}$ defined by the constraints of this program will be denoted by Π and its projection onto the coordinates of v_1, v_2 will be denoted by $\mathcal{N}$. The definition of parameter z is now abstract. Let the objective function of program (3) be denoted by

$$g(v_1, v_2) = \sum_{i=1,2} w_i \log(v_i - c_i).$$

Now define

$$z = \frac{\partial g/\partial v_2}{\partial g/\partial v_1} = \frac{w_2/(v_2 - c_2)}{w_1/(v_1 - c_1)}.$$

Observe that Lemma 5 still holds with this definition of z. As before, we will do a binary search for z on the facets of $\mathcal{N}$.

The procedures given in Section 7 carry over. The LP corresponding to LP (11) is

$$\begin{aligned} \text{maximize} \quad & v_1 + \alpha v_2 \\ \text{subject to} \quad & \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}_1 v_1 + \boldsymbol{b}_2 v_2 \le \boldsymbol{d} \\ & \text{for } \ i = 1, 2: \quad v_i \ge 0 \\ & \boldsymbol{x} \ge 0 \end{aligned} \tag{17}$$

Its dual, which corresponds to LP (13) is the following; we have used p_j to denote the dual corresponding to the jth inequality and j always sums from 1 to m.

$$\begin{aligned} \text{minimize} \quad & \sum_j d_j p_j \\ \text{subject to} \quad & \sum_j b_{ij} p_j \ge 1 \\ & \sum_j b_{2j} p_j \ge \alpha \\ & \text{for } \ 1 \le i \le n: \ \sum_j A_{ji} p_j \ge 0 \\ & \text{for } \ 1 \le j \le m: \ p_j \ge 0 \end{aligned} \tag{18}$$

The next LP corresponds to LP (14). It is derived from LP (18) by adding in constraints on p_j which are implied by the complementary slackness conditions of the primal and dual pair of LP's (17) and (18).

$$\begin{aligned} & \sum_j b_{ij} p_j \ge 1 \\ & \sum_j b_{2j} p_j \ge x\alpha \end{aligned} \tag{19}$$

$$\text{for } \ 1 \le i \le n: \ \sum_j A_{ji} p_j \ge 0$$

$$\text{for } \ 1 \le i \le n \ s.t. \ x_i > 0: \ \sum_j A_{ji} p_j = 0$$

$$\text{for } \ 1 \le j \le m \ s.t. \ \sum_i A_{ji} x_i + b_{ij} v_1 + b_{2j} v_2 < d_j: \quad p_j = 0$$

$$\text{for } \ 1 \le j \le m: \ p_j \ge 0$$

The only difference in the binary search is the following. Observe that in the case of game **DG2**, polytope $\mathcal{N}$ is downward closed, i.e., if $(a,b) \in \mathcal{N}$ and $c \le a$ and $d \le b$ are non-negative then $(c,d) \in \mathcal{N}$. As a result, all points on its non-trivial facets, possibly excluding the first and last facets, are Pareto optimal. This may not hold for game $\mathcal{G}$. On the other hand, the Nash or nonsymmetric bargaining solution is attained at a Pareto optimal point.

Therefore, we will define the endpoints, L, H of the binary search as follows. First find points (c_1, b) and (a, c_2) as in Section 8. Next, for $i = 1, 2$, find

$$\mu_i = \max\{v_i \mid v_i \ \text{ lies on the boundary of } \ \mathcal{N}\}.$$

Next find

$$d_1 = max\{x \mid (\mu_1, x) \in \mathcal{N}\}$$

$$d_2 = max\{x \mid (x, \mu_2) \in \mathcal{N}\}.$$

Finally, out of (c_1, b) and (d_2, μ_2) use the α value of the point having larger first coordinate as H, and out of (a, c_2) and (μ_1, d_1) use the α value of the point having larger second coordinate as L.

The rest of the algorithm is the same as algorithm 7. Furthermore, if the game is in SLNB2, all LP's that need to be solved require strongly polynomial time. Hence we get:

Theorem 13 *Each game in LNB2 is rational and there is a polynomial algorithm for finding its solution. Furthermore, if the game is in SLNB2, the algorithm is actually strongly polynomial.*

[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.

10 A Strongly Polynomial Algorithm for the Game ADNB2

The game **ADNB**, defined in [Vaz09] was derived from the linear case of the Arrow-Debreu model. This model 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 $B = \{1, 2, \ldots, n\}$ be a set of agents and $G = \{1, 2, \ldots, g\}$ 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 $1 \le j \le g$, 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.,

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

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 $(\mathcal{N}, \boldsymbol{c})$. 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 **ADNB**2 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.

Clearly, the bargaining solution to this game is the optimal solution to the following convex program:

$$\begin{aligned}
\text{maximize} \quad & \sum_{i=1,2} w_i \log(v_i - c_i) && (20)\\
\text{subject to} \quad & \forall i = 1,2:\ \ v_i = \sum_{j \in G} u_{ij} x_{ij} \\
& \forall j \in G:\ \ \sum_{i=1,2} x_{ij} \le 1 \\
& \forall i = 1,2,\ \forall j \in G:\ \ x_{ij} \ge 0
\end{aligned}$$

Observe that **ADNB**2 is in RNB. By Theorem 3 we can reduce it to the following flexible budget market, $\mathcal{M}$. The goods and utility functions of the two buyers are as in **ADNB**2 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 $\boldsymbol{p}$ 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, buyer i's money is defined to be

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

10.1 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, $1, 2, \ldots, n$ 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 $1, 2, \ldots k_1 - 1$ 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 $n, n-1, \ldots, k_2 + 1$ 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 $1, 2, \ldots, k$ and player 2 gets goods $k+1, k+2, \ldots, n$.
- **Case 2:** There is a k, $1 \leq k \leq n$ such that player 1 gets goods $1, 2, \ldots, k-1$, player 2 gets goods $k+1, k+2, \ldots, n$, 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

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

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

Observe that **ADNB**2 is not in SLNB2, since the u_{ij}'s are not restriced to be polynomially bounded in n. Even so, we get:

Theorem 14 *There is a combinatorial strongly polynomial algorithm for solving* **ADNB***2.*

11 The Circle Game

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

$$\begin{aligned} &\text{maximize} && \sum_{i=1,2} \log(v_i - c_i) && (21) \\ &\text{subject to} && v_1^2 + v_2^2 \le 1 \\ & && \forall i = 1,2 : \quad v_i \ge 0 \end{aligned}$$

Using the KKT conditions of (21) 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 15 *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 $\theta_1 = \theta_2$. □

12 Acknowledgment

I wish to thank Leonard Schulman for the pointing out the 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.

[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.

[KV] F. P. Kelly and V. V. Vazirani. Rate control as a market equilibrium. Unpublished manuscript 2002. Available at:
`http://www.cc.gatech.edu/ vazirani/KV.pdf`.

[Nas50a] J. F. Nash. The bargaining problem. *Econometrica*, 18:155–162, 1950.

[Nas50b] J. F. Nash. Equilibrium points in n-person games. *Porceedings 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.