A Polynomial Time Algorithm for Rank-1 Bimatrix Games

Ruta Mehta
College of Computing, Georgia Institute of Technology.
rmehta@cc.gatech.edu

Two player normal form game is the most basic form of game, studied extensively in game theory. Such a game can be represented by two payoff matrices \((A, B)\), one for each player, hence they are also known as bimatrix games. The rank of a bimatrix game \((A, B)\) is defined as the rank of matrix \((A + B)\).

For zero-sum games, i.e., rank-0, von Neumann (1928) \cite{von_neumann} showed that Nash equilibrium are min-max strategies, which is equivalent to the linear programming duality \cite{duality,kannan}. Kannan and Theobald (2005) \cite{kannan,thobald} gave an FPTAS for constant rank games and asked if there exists a polynomial time algorithm even for rank-1 games. The main difficulty is that unlike rank-0 games, rank-1 games can have disconnected set of equilibria; even exponentially many \cite{rank_one}. Adsul, Garg, Mehta and Sohoni (2011) \cite{adsul} settled this question together with other interesting structural results. In doing so they extend the polynomial time solvability of linear programming to a larger class of optimization problems with non-convex, even disconnected solution set. To the best of our knowledge no such result is known when the solution set is disconnected.

This note is aimed at giving a simpler exposition of the AGMS \cite{adsul} algorithm and analysis. We note that Nash equilibrium computation in general bimatrix game is PPAD-complete \cite{agrandi,agrandi2,agrandi3}.

1 Preliminaries

A bimatrix game is a single shot, two player game, each player having finitely many strategies (moves) to play from. Let \(S_1\) and \(S_2\) denote the set of pure strategies for the first and the second player respectively, and let \(m = |S_1|\) and \(n = |S_2|\). Such a game can be represented by two payoff matrices \(A\) and \(B\) such that, if the played strategy profile is \((i, j) \in S_1 \times S_2\), then the payoffs of the first player is \(A_{ij}\) and that of second player is \(B_{ij}\). Note that the rows of these matrices correspond to the strategies of the first player and the columns to the strategies of second player.

Players may randomize among their strategies; a randomized play is called a mixed strategy. The set of mixed strategies for the first-player is \(X = \{(x_1, \ldots, x_m) \mid x_i \geq 0, \forall i \in S_1, \sum_{i \in S_1} x_i = 1\}\) and for the second-player, it is \(Y = \{(y_1, \ldots, y_n) \mid y_j \geq 0, \forall j \in S_2, \sum_{j \in S_2} y_j = 1\}\). By playing \((x, y) \in X \times Y\) we mean strategies are picked independently at random as per \(x\) by the first-player and as per \(y\) by the second-player. Therefore the expected payoffs of the first-player and second-player are, respectively

\[
\sum_{i,j} A_{ij} x_i y_j = x^T A y
\]

\[
\sum_{i,j} B_{ij} x_i y_j = x^T B y
\]

Definition 1 (Nash Equilibrium \cite{von_neumann}) A strategy profile is said to be a Nash equilibrium strategy profile (NESP) if no player achieves a better payoff by a unilateral deviation \cite{ unilateral}. Formally, \((x, y) \in X \times Y\) is a NESP iff \(\forall x' \in X, x^T A y \geq x'^T A y\) and \(\forall y' \in Y, x^T B y \geq x^T B y'\).

Given strategy \(y\) for the second-player, the first-player gets \((Ay)_k\) from her \(k^{th}\) strategy. Clearly, her best strategies are \(\arg \max_k (Ay)_k\), and a mixed strategy fetches the maximum payoff only if she
randomize among her best strategies. Similarly, given $x$ for the first-player, the second-player gets $(x^T B)_k$ from $k^{th}$ strategy, and same conclusion applies. These can be equivalently stated as the following complementarity type conditions,

$$\forall i \in S_1, \ x_i > 0 \Rightarrow A_i y = \max_{k \in S_1} A_k y$$

$$\forall j \in S_2, \ y_j > 0 \Rightarrow x^T B_j = \max_{k \in S_2} x^T B_k$$

(1)

The strategies fetching maximum payoff are called the best response strategies (w.r.t. the opponent’s play). Based on conditions (1) next we define best response polyhedra (BRPs), $P$ for the first-player and $P$ for the second-player. In the following expression, $x$ and $y$ are vector variables representing the mixed-strategies, and $\pi_1$ and $\pi_2$ are scalar variables supposed to capture payoffs.

$$P = \{(x, \pi_2) \in \mathbb{R}^{m+1} \mid x \in X; \ (x^T B) j - \pi_2 \leq 0, \ \forall j \in S_2\}$$

$$Q = \{(y, \pi_1) \in \mathbb{R}^{n+1} \mid y \in Y; \ (Ay)_i - \pi_1 \leq 0, \ \forall i \in S_1\}$$

(2)

Consider the following complementarity conditions for a point $((x, \pi_2), (y, \pi_1)) \in P \times Q$.

$$x_i ((Ay)_i - \pi_1) = 0, \ \forall i \in S_1 \quad \text{and} \quad y_j ((x^T B)_j - \pi_2) = 0, \ \forall j \in S_2$$

(3)

For any given point in $P \times Q$, clearly, we have $x_i ((Ay)_i - \pi_1) \leq 0, \ \forall i$ and $y_j ((x^T B)_j - \pi_2) \leq 0, \ \forall j$. Summing all of them up gives $x^T (A + B) y - \pi_1 - \pi_2 \leq 0$. Clearly, it is exactly zero if and only if (3) is satisfied. Therefore, the following quadratic program captures points satisfying (3).

$$\max : \ x^T (A + B) y - \pi_1 - \pi_2$$

s.t. \quad $$(x, \pi_2) \in P; \quad (y, \pi_1) \in Q$$

(4)

The next lemma follows from the above discussion and the fact that every NESP satisfies (1).

**Lemma 2** The following are equivalent

1. $(x, y)$ is a NESP of game $(A, B)$ with payoffs $\pi_1 = x^T Ay$ and $\pi_2 = x^T By$.

2. $(x, \pi_2) \in P$, $(y, \pi_1) \in Q$, and they satisfy (3), equivalently $x^T (A + B) y - \pi_1 - \pi_2 = 0$.

3. $(x, y, \pi_1, \pi_2)$ is an optimal solution of the quadratic program (4).

## 2 Rank-1 Game Space

Rank of a bimatrix game $(A, B)$ is $\text{rank}(A + B)$. Zero-sum games have rank-0, and are polynomial time solvable, as they are equivalent to linear programs [5][1]. In this section we consider the smallest extension of zero-sum games, namely rank-1 games, and analyze them with a perspective of designing a polynomial time algorithm to solve them.

If the game has rank-1, then matrix $A + B$ can be represented as $a \cdot b^T$, where $a \in \mathbb{R}^m$ and $b \in \mathbb{R}^n$. Note that tuple $(A, a, b)$ is enough to represent such a game, since $B$ can be written as $-A + a \cdot \cdot b^T$. Replacing $A + B$ by $a \cdot b^T$ and $B$ by $-A + a \cdot b^T$ in (4), we get,

$$\max : \ (x^T \cdot a)(b^T \cdot y) - \pi_1 - \pi_2$$

s.t. \quad $(y, \pi_1) \in Q$

$$x \in X; \quad \forall j \in S_2, \quad -(x^T A)_j + (x^T \cdot a)b_j \leq \pi_2$$

(5)

The formulation of (5) is a rank-1 quadratic program. Solving a rank-1 quadratic programs is NP-hard in general [7]. One major issue with rank-1 games is that they may have disconnected set

---

1. By solving a game, we mean computing its Nash equilibrium.
of equilibria. We consider space \( S \) of rank-1 games, defined next, so that all of their equilibria form a well-behaved set.

\[
S = \{(A, u, b) \mid u \in \mathbb{R}^m\}
\]

The quadratic program (QP) of (5) is the same for all the games \((A, u, b) \in S\) except for term \((x^T \cdot a)\) appearing in the cost function and in inequality of polyhedron \(Q\). This term becomes \((x^T \cdot u)\) and changes as \(u\) changes. Let us replace this term by a scalar variable \(\lambda\) at both the places,

\[
\text{max} : \lambda(b^T \cdot y) - \pi_1 - \pi_2 \\
\text{s.t.} \quad (y, \pi_1) \in Q; \quad (x, \pi_2, \lambda) \in P'
\]

where \(P' = \{(x, \pi_2, \lambda) \mid x \in X; \quad -(x^T A)_j + \lambda d_j \leq \pi_2, \forall j \in S_2\}\)

(6)

Similar to (3) consider the following complementarity conditions for \(P' \times Q\),

\[
x_i((Ay)_i - \pi_1) = 0, \forall i \in S_1 \quad \text{and} \quad y_j(-(x^T A)_j + \lambda b_j - \pi_2) = 0, \forall j \in S_2
\]

(8)

Using Lemma 2 and the above construction we show the next lemma.

Lemma 3 The following are equivalent

1. \((x, y)\) is a NESP of game \((A, u, b) \in S\) with payoffs \(\pi_1 = x^T Ay\) and \(\pi_2 = x^T (-A + u \cdot b^T)y\).

2. For \(\lambda = x^T \cdot u\), we have \((x, \pi_2, \lambda) \in P', (y, \pi_1) \in Q\), and they satisfy (8), equivalently \(\lambda(d^T \cdot y) - \pi_1 - \pi_2 = 0\).

3. For \(\lambda = x^T \cdot u\), \((x, y, \pi_1, \pi_2, \lambda)\) is an optimal solution of QP (6).

Proof : \((1 \Rightarrow 2)\) Lemma 2 implies that \((x, y)\) satisfies (3) and \((x, \pi_2) \in P\). Replacing \(B\) with \(-A + u \cdot b^T\) the second matrix of game \((A, u, b)\) and then \(x^T \cdot u\) with \(\lambda\) in (3) gives (8), and in P of (11) gives \(P'\) of (7). Summing all of the equalities of (8) gives \(\lambda(b^T \cdot y) - \pi_1 - \pi_2 = 0\).

\((2 \Rightarrow 3)\) At any feasible point of the QP (6) we have \(x_i((Ay)_i - \pi_1) \leq 0, \forall i\) and \(y_j(-(x^T A)_j + \lambda b_j - \pi_2) \leq 0, \forall j\). Summing all of these up gives \(\lambda(b^T \cdot y) - \pi_1 - \pi_2 \leq 0\), and equality holds if (8) is satisfied. Further, from the first part we know that there are points in \(P' \times Q\) where (8) is satisfied.

\((3 \Rightarrow 1)\) Let \(v = (x, y, \pi_1, \pi_2, \lambda)\) be an optimal solution on (6); \(v\) is feasible point with objective value zero. Since, replacing \(\lambda\) with \(x^T \cdot u\) gives the (4) for the game \((A, -A + u \cdot b^T)\), point \(((y, \pi_1), (x, \pi_2))\) is feasible in it and with objective value zero. Hence proved using Lemma 2. \(\square\)

Let \(N\) denote the set of solutions of QP (6). Except for term \(\lambda(b^T \cdot y)\) everything else is linear in this QP. If we replace \(\lambda\) by a number, say \(c \in \mathbb{R}\), then it indeed becomes a linear program (LP). Therefore, let \(LP(\lambda)\) be the linear program obtained by replacing \(\lambda\) with the given parameter at both the places in (6).

Lemma 4 \((x', y', \pi'_1, \pi'_2, \lambda') \in N\) iff \((x', y', \pi'_1, \pi'_2)\) is an optimal solution of \(LP(\lambda')\).

Proof : The polyhedron of \(LP(c)\) for any \(c \in \mathbb{R}\), is polyhedron of the QP (6) intersected with hyper-plane \(\lambda = c\); a subset of \(P' \times Q\) where \(\lambda\) takes value \(c\). Therefore, the optimal value of \(LP(c)\) is at most zero (Lemma 3).

\((\Rightarrow)\) At \((x', y', \pi'_1, \pi'_2)\) the cost function of \(LP(\lambda')\) evaluates to zero, and has to be optimal.

\((\Leftarrow)\) Let \((x', y')\) be a Nash equilibrium of rank-1 game \((A, u, b)\) where \(u = [\lambda', \ldots, \lambda']^T\). Clearly, \(x^T \cdot u\) is \(\lambda'\) for all \(x \in X\). Therefore, using Lemma 3 we have that \(\lambda'(b^T \cdot y') - \pi'_1 - \pi'_2 = 0\) and \((x', y', \pi'_1, \pi'_2)\) is feasible in \(LP(\lambda')\), hence also optimal. The lemma follows since cost function evaluates to zero at every solution of \(LP(\lambda')\) and its feasible set padded with \(\lambda'\) is a subset of \(P' \times Q\). \(\square\)
3 A Polynomial-Time Algorithm

Given a rank-1 game \((A, a, b)\), let \(I\) denote the interval \([\min_i a_i, \max_i a_i]\). Since, \(x \in X\) is a probability distribution vector, we have \(x^T \cdot c \in I\). Consider function \(f : I \rightarrow I\) such that

\[
f(c) = x^T \cdot a, \quad \text{where } (x, y, \pi_1, \pi_2) \text{ is a solution of } LP(c)
\]  

(9)

If there are multiple values of \(x\) in the solution set of \(LP(c)\) then it is not clear which \(x\) to use to compute \(f(c)\). To avoid any such ambiguity we need to show that \(x\) takes a unique value in all the solutions of \(LP(c)\). For this we will use the structure of \(\mathcal{N}\) and convexity of the solution set of \(LP(c)\). Let us assume that polyhedra \(P'\) and \(Q\) are non-degenerate.

Lemma 5 Given an \(a \in \mathbb{R}\), there is a unique \(x\) in the solutions of \(LP(c)\).

Proof: Note that the polyhedron \(P' \times Q\) is in \(m + n + 1\) dimension; two equalities compensate for two variables. At a point \(v \in \mathcal{N}\), either \(x_i = 0\) or \((Ay)_i - \pi_1 = 0\) for every \(i \in S_1\), and either \(y_j = 0\) or \(-(x^T A)_j + \lambda b_j - \pi_2 = 0\) for every \(j \in S_2\) (Lemma 3). In other words at least \(m + n\) inequalities of \(P' \times Q\) hold with equality at \(v\). Therefore, \(v\) has to be either on an edge or at a vertex of polyhedron \(P' \times Q\). If it is a vertex, then exactly \(m + n + 1\) inequalities are tight at \(v\), implying that for some \(i\) or \(j\) both the corresponding inequalities are tight. If it is an \(i\) then the edges we obtain by relaxing \(x_i = 0\) or \((Ay)_i - \pi_1 = 0\) still satisfy \(\mathcal{N}\) and hence are in \(\mathcal{N}\). Similarly for \(j\). Putting all of these together we get that \(\mathcal{N} \subset 1\)-skeleton of \(P' \times Q\), every vertex of which has degree two in \(\mathcal{N}\). Essentially set \(\mathcal{N}\) forms paths and cycles on 1-skeleton of \(P' \times Q\), with unbounded edges at either end of every path.

Let \(\mathcal{N}(a)\) denote the set of points of \(\mathcal{N}\) with \(\lambda = a\). This set is exactly the solutions of \(LP(a)\) (Lemma 1), and hence is convex. In that case, \(\mathcal{N}\) can have only paths, where \(\lambda\) changes monotonically on each path. Since, the values of \(\lambda\) covered by a path will form a closed interval, and \(\mathcal{N}(a)\) is entirely contained in one path, there can not be more than one path in \(\mathcal{N}\).

An edge of polyhedron \(P' \times Q\) is formed by either vertex of \(Q\) and edge of \(P'\), or edge of \(Q\) and vertex of \(P'\). Let these two type of edges be called \(T_1\) and \(T_2\) respectively. Clearly, \(\lambda\) remains constant on any edge of type \(T_2\). However, this can not be the case for type \(T_1\) edge in \(\mathcal{N}\) as shown next.

Claim 6 On every type \(T_1\) edge of \(\mathcal{N}\), \(\lambda\) strictly changes.

Proof: From the discussion it is clear that \(\mathcal{N}\) forms a path on 1-skeleton of \(P' \times Q\), with alternating edges of type \(T_1\) and \(T_2\). Let \(e_1\) and \(e_2\) be two adjacent edges of type \(T_1\) and \(T_2\). To the contrary suppose \(\lambda\) is constant on \(e_1\), and takes value \(a\). In that case \(\lambda = a\) on both \(e_1\) and \(e_2\) since \(e_2\) is of type \(T_2\). Hence \(e_1 \in \mathcal{N}(a)\) and \(e_2 \in \mathcal{N}(a)\) contradicting its convexity. 

Claim 6 implies that set \(\mathcal{N}(a)\) consists of either a point on an edge of type \(T_1\), or an entire edge \(T_2\). In both the cases, there is a unique \(x\).

Lemma 5 implies that function \(f\) is well defined. Next we show how this function relates to the Nash equilibria of game \((A, a, b)\).

Lemma 7 \((x, y)\) is a NE of game \((A, a, b)\) iff \(a = x^T \cdot c\) is a fixed-point of function \(f\).

Proof: Due to Lemma 3 \((x, y)\) is a NE of game \((A, a, b)\) if and only if \((x, y, \pi_1, \pi_2, a) \in \mathcal{N}\), where \(c = x^T \cdot a, \pi_1 = x^T Ay\) and \(\pi_2 = x^T (-A + a \cdot b^T) y\). This in turn is possible if and only if \((x, y, \pi_1, \pi_2)\) is a solution of \(LP(c)\) (Lemma 3). The later case happens only at fixed points of \(f\). 

Thus, computing a NE of game \((A, a, b)\) reduces to finding a fixed point of function \(f\) with one-dimensional domain. The latter is known to be doable in polynomial time using binary search, as follows:
1. Initialize: \( c_1 = \min_i a_i \) and \( c_2 = \max_i a_i \).

2. For \( i = 1, 2\), if \( f(c_i) = c_i \) then output \( LP(c_i) \) and exit.

3. \( c \leftarrow c_1 + c_2 / 2 \). If \( f(c) = c \) then output \( LP(c) \) and exit.

4. else if \( f(c) > c \) then \( c_1 \leftarrow c \) else \( c_2 \leftarrow c \). Go to Step 3.

Let \( L \) be the total bit length of input \((A,a,b)\).

**Theorem 8** The algorithm terminates in at most \( \text{poly}(m,n,L) \) many iterations.

**Proof:** Since, Nash equilibrium strategy profile of a bimatrix game consists of only rational numbers of polynomial size \([14]\), fixed points of function \( f \) are also rational numbers of polynomial size (Lemma \([7]\)). The number of pivots binary search takes is at most the size of fixed points. \( \square \)

### 3.1 Degeneracy

If polyhedron \( P' \) or \( Q \) is degenerate, then Lemma \([5]\) does not apply, and \( f \) becomes a correspondence. However the set \( f(a) \) is always convex and compact. Therefore, we can use Kakutani fixed-point theorem \([8]\) instead and Lemma \([7]\) will still hold. An obvious modification in the algorithm are: \( i) \) instead of checking \( f(c) = c \), check if \( c \in f(c) \) which can be done in polynomial time. \( ii) \) Similarly, \( f(c) > c \) will imply checking if \( d > c \), \( \forall d \in f(c) \).

### 4 Discussion

The next step is to consider rank-2 games, where \( A + B \) can be written as \( a \cdot b^T + e \cdot f^T \). The analysis of Section \([2]\) can be extended to get an LP with two parameters, in turn the fixed-point formulation of Section \([3]\) will be on two-dimensional domain. However, unlike 1-D fixed-point, 2-D fixed-point computation is PPAD-hard in general \([3]\). It would be very interesting to know if this hardness carries back to rank-2 games, or not.

**References**


