18 Equilibrium Computation

Consider a bimatrix game with payoff matrices $P, Q \in \mathbb{R}^{m \times n}$, and assume without loss of generality that $P, Q > 0$. It will be convenient to index the actions of the row and column player by $M = \{1, \dots, m\}$ and $N = \{m+1, \dots, m+n\}$, respectively, and define the sets X and Y of strategies accordingly.

A pair $(x, y) \in X \times Y$ is an equilibrium if and only if all pure strategies in $S(x)$ are best responses to y and all pure strategies in $S(y)$ are best responses to x, i.e., if

$$\begin{aligned} &\text{for all } i \in M, \quad x_i > 0 \text{ implies } (Py)_i = \max_{k \in M} (Py)_k \text{ and} \\ &\text{for all } j \in N, \quad y_j > 0 \text{ implies } (Q^T x)_j = \max_{k \in N} (Q^T x)_k. \end{aligned} \tag{18.1}$$

18.1 Labeled Polytopes

Not every subset of actions can be the support of an equilibrium, and it makes sense to search for equilibrium supports in a more organized way. In the following we restrict our attention to non-degenerate games, but note that degeneracies can be handled by a slight perturbation of the payoff matrices. A bimatrix game is *non-degenerate* if for every $(x, y) \in X \times Y$, $|S(x)| \geqslant |S(y)|$ if y is a best response to x and $|S(y)| \geqslant |S(x)|$ if x is a best response to y.

Let

$$\begin{aligned} \hat{X} &= \{(x, u) \in \mathbb{R}^m \times \mathbb{R} : x \geqslant 0, x^T \mathbf{1} = 1, Q^T x \leqslant u\mathbf{1}\} \quad \text{and} \\ \hat{Y} &= \{(y, v) \in \mathbb{R}^n \times \mathbb{R} : y \geqslant 0, y^T \mathbf{1} = 1, Py \leqslant v\mathbf{1}\}, \end{aligned}$$

where $\mathbf{1}$ denotes an all-ones vector of appropriate dimension. The first two inequalities for $\hat{X}$ ensure that x is indeed a strategy of the row player, while the third inequality ensures that u is an upper bound on the expected payoff for every pure strategy of the column player. We will be interested in the extreme points of $\hat{X}$ and $\hat{Y}$, because they satisfy some of the constraints with equality and therefore correspond to strategies with specific properties: if $x_i = 0$, then the pure strategy $i \in M$ is not played in strategy x; if $(Q^T x)_j = u$, then the pure strategy $j \in N$ is a best response to x. In the case of $\hat{X}$ there are $m+1$ variables, so in addition to the equality constraint at least m of the inequality constraints have to hold with equality at every extreme point. An extreme point (x, u) of $\hat{X}$ thus corresponds to a situation where $|S(x)| = k$, for some $k \leqslant m$, and at least k pure strategies of the column player are a best response to x. By non-degeneracy, there in fact have to be exactly k such strategies. Analogously, at every extreme point (y, v) of $\hat{Y}$, $|S(y)| = k$ for some $k \leqslant n$ and k pure strategies of the row player are a best response to y. It is easy to see an equilibrium (x, y) of a non-degenerate game must satisfy $|S(x)| = |S(y)|$, so that every equilibrium of such a game can be described as a pair of extreme points of $\hat{X}$ and $\hat{Y}$.

Since $P, Q > 0$ implies that $u, v > 0$, we can simplify notation and consider

$$\begin{aligned}\overline{X} &= \{x \in \mathbb{R}^m : x \geqslant 0, Q^T x \leqslant \mathbf{1}\} \quad \text{and}\\ \overline{Y} &= \{y \in \mathbb{R}^n : y \geqslant 0, Py \leqslant \mathbf{1}\}.\end{aligned}$$

The vectors in $\overline{X}$ are no longer normalized, but there is a direct correspondence between the extreme points of $\hat{X}$ and $\overline{X}$, except for the zero vector: for each extreme point (x, u) of $\hat{X}$, x/u is an extreme point of $\overline{X}$, and for each extreme point x of $\overline{X}$, apart from the zero vector, $(x/\sum_{i=1}^m x_i, 1/\sum_{i=1}^m x_i)$ is an extreme point of $\hat{X}$. Analogous conditions hold for $\hat{Y}$ and $\overline{Y}$.

We will now give a characterization of those extreme points of $\overline{X}$ and $\overline{Y}$ that correspond to an equilibrium of the underlying game. For given extreme points x and y of $\overline{X}$ and $\overline{Y}$, let $L(x)$ and $L(y)$ be the index sets of those constraints that hold with equality, i.e.,

$$\begin{aligned}L(x) &= \{i \in M : x_i = 0\} \cup \{j \in N : (Q^T x)_j = 1\} \quad \text{and}\\ L(y) &= \{j \in N : y_j = 0\} \cup \{i \in M : (Py)_i = 1\}.\end{aligned}$$

We refer to the sets $L(x)$ and $L(y)$ as the *labels* of x and y and call a pair (x, y) *fully labeled* if $L(x) \cup L(y) = M \cup N$.

THEOREM 18.1. *A pair of extreme points* $(x, y) \in \overline{X} \times \overline{Y}$ *with* $(x, y) \neq (0, 0)$ *corresponds to an equilibrium if and only if it is fully labeled.*

Proof. Suppose that $L(x) \cup L(y) = M \cup N$, and let

$$\begin{aligned}M_1 &= \{i \in M : x_i = 0\}, & N_1 &= \{j \in N : y_j = 0\},\\ M_2 &= \{i \in M : (Py)_i = 1\}, & N_2 &= \{j \in N : (Q^T x)_j = 1\}.\end{aligned}$$

Then, $M_1 \cup M_2 = M$ and $N_1 \cup N_2 = N$, and it is easy to see that (18.1) is satisfied.

Conversely assume that (x, y) corresponds to an equilibrium. Then, by (18.1), $(M \setminus S(x)) \cup S(y) \subseteq L(x)$ and $(N \setminus S(y)) \cup S(x) \subseteq L(y)$, and thus $L(x) \cup L(y) = M \cup N$. □

18.2 The Lemke-Howson Algorithm

The relationship between completely labeled pairs of extreme points and equilibria of the underlying game can be used to obtain a combinatorial algorithm for finding an equilibrium. The idea is to start from $(0, 0)$ and pivot alternatingly in $\overline{X}$ and $\overline{Y}$ until a completely labeled pair is found. To make this idea precise, let V_X and V_Y be the sets of extreme points of $\overline{X}$ and $\overline{Y}$, and let E_X and E_Y be the set of edges between adjacent extreme points, i.e.,

$$\begin{aligned}E_X &= \{(x, x') \in V_X \times V_X : |L(x) \cap L(x')| = m - 1\}\\ E_Y &= \{(y, y') \in V_Y \times V_Y : |L(y) \cap L(y')| = n - 1\}.\end{aligned}$$

Further let $V = V_X \times V_Y$ and $E = \{((x,y),(x',y')) \in V \times V : (x,x') \in E_X \text{ or } (y,y') \in E_Y\}$.

The key observation is that if we restrict our attention to extreme points that are almost fully labeled, with the possible exception of a particular label ℓ, then there is always a unique way in which we can proceed. For $\ell \in M \cup N$, let $V_\ell = \{(x,y) \in V : L(x) \cup L(y) \supseteq M \cup N \setminus \{\ell\}\}$ and $E_\ell = E \cap (V_\ell \times V_\ell)$.

Theorem 18.2. *Let $\ell \in M \cup N$. Then, V_ℓ contains $(0,0)$ as well as every pair $(x,y) \in V$ that corresponds to an equilibrium of the underlying game. If the underlying game is non-degenerate, then $(0,0)$ and the elements of V_ℓ corresponding to an equilibrium have degree one in the graph (V_ℓ, E_ℓ), and all other elements of V_ℓ have degree two.*

Proof. Both $(0,0)$ and all pairs corresponding to equilibria are fully labeled, so the first property is obvious.

For the second property, consider $(x,y) \in V_\ell$ and let $\bar{x}$ and $\bar{y}$ be the strategies in the underlying game corresponding to x and y. Since the game is non-degenerate, at most $|S(\bar{x})|$ pure strategies can be a best response to $\bar{x}$, and at most $|S(\bar{y})|$ pure strategies can be a best response to $\bar{y}$, so $L(x) \leqslant |M \setminus S(x)| + S(x) = m$ and $L(y) \leqslant |N \setminus S(y)| + S(y) = n$. Since x and y are extreme points of $\overline{X}$ and $\overline{Y}$, respectively, we have that $|L(x)| \geqslant m$ and $|L(y)| \geqslant n$, and thus $L|(x)| = m$ and $|L(y)| = n$.

If $(x,y) = (0,0)$, or if $(\bar{x},\bar{y})$ is an equilibrium, then (x,y) is fully labeled and therefore $L(x) \cap L(y) = \emptyset$. The neighbors of (x,y) are those elements of V_ℓ that replace ℓ with some other label, i.e., those where some other constraint holds with equality instead of the one corresponding to ℓ. Since the constraints are linear, and by non-degeneracy, there is exactly one such element.

Otherwise $L(x) \cap L(y) = \{k\}$ for some $k \in M \cup N$. The neighbors of (x,y) can again be obtained by replacing k with another label. This replacement can be done either in $\overline{X}$ or in $\overline{Y}$, and for each of the two there is one neighbor by the same reasoning as before. □

This means that the graph (V_ℓ, E_ℓ) for each $\ell \in M \cup N$ consists of paths and cycles that are pairwise disjoint, and the ends of the paths correspond to the pair $(0,0)$ and to the equilibria of the underlying game. The Lemke-Howson algorithm now takes the obvious steps: it starts from $(0,0)$ and follows the path until it reaches the pair of extreme points at the other end, which must correspond to an equilibrium. Moving from a vertex $(x,y) \in V_\ell$ to a neighboring vertex $(x',y') \in V_\ell$ corresponds to dropping a label from either x or y and picking up another label with the same element. In the first round the label that is dropped is ℓ. In subsequent rounds it is the duplicate label that has been picked up in the previous round, but it is dropped from the respective other element. Eventually ℓ will be picked up, leading to a fully labeled pair. Formally the algorithm proceeds as follows:

1. Pick $\ell \in M \cup N$. Let (x,y) be the pair obtained from $(0,0)$ by dropping label ℓ.

2. Let $k \in L(x') \cap L(y')$ be the label that was just picked up. If it was picked up by x, then drop it from y; otherwise drop it from x. Let (x, y) be the resulting pair.
3. If (x, y) is fully labeled, then stop. Otherwise go to Step 2.

Theorem 18.2 also shows that every non-degenerate game has at least one equilibrium, and in fact that the number of equilibria in such games is odd.

COROLLARY 18.3. *Every non-degenerate bimatrix game has an odd number of equilibria.*

18.3 Complementary Pivoting

Adding slack variables $r \in \mathbb{R}^m$ and $s \in \mathbb{R}^n$ turns the best response conditions into $Q^T x + s = 1$ and $Py + r = 1$, where $x, y, r, s \geqslant 0$. The pair (x, y) then is completely labeled if and only if $x^T r = 0$ and $y^T s = 0$. Dropping a label corresponds to increasing or decreasing one of the variables such that one of the constraints, the one corresponding to the label being dropped, no longer holds with equality. When the variable is increased or decreased as much as possible, a different constraint starts to hold with equality, and the label corresponding to this constraint is picked up. This procedure can be carried out through pivoting in a tableau, in a very similar way to pivoting in the simplex method.

Consider for example the bimatrix game given by

$$P = \begin{pmatrix} 3 & 3 \\ 2 & 5 \\ 0 & 6 \end{pmatrix} \qquad \text{and} \qquad Q = \begin{pmatrix} 3 & 2 \\ 2 & 6 \\ 3 & 1 \end{pmatrix},$$

Indexing r and s by $M = \{1, \dots, m\}$ and $N = \{m + 1, \dots, m + n\}$, respectively, the constraint $Q^T x + s = 1$ can be written in tableau form as follows:

x_1	x_2	x_3	s_4	s_5	
3	2	3	1	0	1
2	6	1	0	1	1

Assume that label $\ell = 2$ is dropped. In that case, want to add variable x_2 to the basis, by pivoting on the column that corresponds to this variable. The corresponding pivot row is the one that minimizes the ratio between the entry in the last column and the entry in the pivot column among those that have a positive entry in the latter. Pivoting then works by dividing the pivot row by the entry in the pivot column in order to make that entry equal to one, and adding a multiple of the pivot row to the other rows in order to make all other entries in the pivot column equal to zero. We obtain the following tableau:

$\frac{7}{3}$	0	$\frac{8}{3}$	1	$-\frac{1}{3}$	$\frac{2}{3}$
$\frac{1}{3}$	1	$\frac{1}{6}$	0	$\frac{1}{6}$	$\frac{1}{6}$

The second row now corresponds to variable x_2 that has entered the basis. On the other hand, variable s_5 has left the basis, and $5 \in L(x) \cap L(y)$. We thus want to turn to the constraint $Py + r = 1$ and drop the duplicate label 5. The initial tableau for this constraint looks as follows:

y_4	y_5	r_1	r_2	r_3	
3	3	1	0	0	1
2	5	0	1	0	1
0	6	0	0	1	1

By pivoting on the second column, corresponding to y_5, and on the third row, we pick up label 3 and obtain the following tableau:

3	0	1	0	$-\frac{1}{2}$	$\frac{1}{2}$
2	0	0	1	$-\frac{5}{6}$	$\frac{1}{6}$
0	1	0	0	$\frac{1}{6}$	$\frac{1}{6}$

Pivoting one more time in each of the two polytopes, we drop label 3 to pick up label 4, and drop label 4 to pick up label $\ell = 2$ and obtain a fully labeled pair. The final tableaus look as follows:

$\frac{7}{8}$	0	1	$\frac{3}{8}$	$-\frac{1}{8}$	$\frac{1}{4}$
$\frac{3}{16}$	1	0	$-\frac{3}{48}$	$\frac{3}{16}$	$\frac{1}{8}$

0	0	1	$-\frac{3}{2}$	$\frac{3}{4}$	$\frac{1}{4}$
1	0	0	$\frac{1}{2}$	$-\frac{5}{12}$	$\frac{1}{12}$
0	1	0	0	$\frac{1}{6}$	$\frac{1}{6}$

Reading off the values of x and y from the last column of each tableau and scaling them appropriately yields the equilibrium $((0, 1/3, 2/3), (1/3, 2/3))$.

18.4 The Complexity of Finding an Equilibrium Revisited

The Lemke-Howson algorithm creates a graph consisting of disjoint paths and cycles, such that one endpoint of a path can be found very easily and any other endpoint corresponds to an equilibrium. Clearly, there must be at least one other endpoint, and this endpoint can be found by following the path from the initial one. The problem is that the size of the graph, and thus potentially the length of the path in question, is exponential. Indeed, there exist games for which the Lemke-Howson algorithm takes an exponential number of steps until it terminates.

It turns out that there is a variety of other computational problems for which the existence of a solution is guaranteed by the same type of argument. One of them is the problem of finding a fixed point of a function satisfying the conditions of Brouwer's theorem. These problems comprise the complexity class PPAD (for polynomial parity argument, directed case), and the equilibrium problem is complete for this class under an appropriate type of reduction.

Theorem 18.4 (Chen and Deng, 2006). *Finding an equilibrium of a bimatrix game is PPAD-complete.*

The question whether equilibria can be found in polynomial time is thus equivalent to the question whether PPAD $=$ P, or whether there exists a general method that avoids following paths in the graph and instead takes a shortcut to a solution. While it could in principle be the case that PPAD $=$ P even when P $\neq$ NP, this is still considered rather unlikely, one of the reasons being that many of the problems in PPAD are notoriously hard.