CS 573: Algorithmic Game Theory	Lecture date: January 23rd, 2008
Instructor: Chandra Chekuri	Scribe: Bolin Ding

Contents

1 2-Player Zero-Sum Game

In this section, we will first prove that any *2-player zero-sum finite game* has a Nash equilibrium of *mixed strategies* (a special case of Nash's theorem), and a Nash equilibrium can be found through solving linear programs. We will further show how to prove lower bounds on randomized complexity of Las Vegas algorithms using this result.

A 2-player game is a *zero-sum* game if the sum of the payoffs of the 2 players (*row player* and *column player*) is zero for any choices of strategies. We consider *finite game* here, i.e., the number of players and the strategy set of each player are both finite. Let $A = \{A\}_{m\times n}$ be the *payoff matrix* of row player. From the definition of *zero-sum* game, we have $-A$ is the payoff matrix of column player. Let $X = \{\text{row vector } x \in [0,1]^m \mid \sum x_i = 1\}$ and $Y = \{\text{column vector } y \in [0,1]^n \mid \sum y_i = 1\}$ be the mixed strategy sets of the row player and the column player, respectively.

1.1 Min-Max Theorem for 2-Player Zero-Sum Game

Theorem 1.1 *Any 2-player zero-sum finite game has a Nash equilibrium of mixed strategies, and a Nash equilibrium can be found in polynomial time.*

Proof: We prove this theorem using the notations given at the beginning of this section.

First, we prove the following equality:

$$\max_{x\in X}\min_{y\in Y} xAy = \min_{y\in Y}\max_{x\in X} xAy.$$

Let e_i be a row vector $[0,...,0,1,0,...,0]$, where the i^{th} entry, the only nonzero entry in e_i, is 1. Let $\bar{e}_j$ be a column vector $[0,...,0,1,0,...,0]^T$, where the j^{th} entry, the only nonzero entry in $\bar{e}_j$, is 1. Fix $x \in X$, we have

$$\min_{y\in Y}(xA)y = \min_{j\in\{1,...,n\}}(xA)\bar{e}_j,$$

because on one hand, $\{\bar{e}_1,...,\bar{e}_n\} \subseteq Y$, which implies

$$\min_{y\in Y}(xA)y \le \min_{j\in\{1,...,n\}}(xA)\bar{e}_j,$$

and on the other hand, for any $y \in Y$, $\sum_j (xA)_j y_j \geq \min_j (xA)_j$ ($(xA)_j$ is the j^{th} entry of vector (xA), i.e., $(xA)_j = (xA)\bar{e}_j$), which implies

$$\min_{y \in Y} (xA)y \geq \min_{j \in \{1,...,n\}} (xA)\bar{e}_j.$$

Therefore, let $\lambda(x) = \min_j (xA)\bar{e}_j$, and then we have

$$\max_{x \in X} \min_{y \in Y} xAy = \max_{x \in X} \min_{j \in \{1,...,n\}} xA\bar{e}_j = \max_{x \in X} \lambda(x).$$

We can use the following linear programming to find $\lambda^* = \max_x \lambda(x)$: (PLP)

$$\begin{aligned} \lambda^* &= \max \lambda \\ \text{s.t.} \qquad xA\bar{e}_j = \sum_{i=1}^{m} x_i A_{i,j} &\geq \lambda, \text{ for } j = 1, ..., n \\ \sum_{i=1}^{m} x_i &= 1 \\ x_i &\geq 0, \text{ for } i = 1, ..., m. \end{aligned}$$

Similarly, for fixed $y \in Y$, let $\gamma(y) = \max_i e_i(Ay)$, and then we have

$$\min_{y \in Y} \max_{x \in X} xAy = \min_{y \in Y} \max_{i \in \{1,...,m\}} e_i Ay = \min_{y \in Y} \gamma(y).$$

We can use the following linear programming to find $\gamma^* = \min_y \gamma(y)$: (DLP)

$$\begin{aligned} \gamma^* &= \min \gamma \\ \text{s.t.} \qquad e_i Ay = \sum_{j=1}^{n} A_{i,j} y_j &\leq \gamma, \text{ for } i = 1, ..., m \\ \sum_{j=1}^{n} y_j &= 1 \\ y_j &\geq 0, \text{ for } j = 1, ..., n. \end{aligned}$$

Since (DLP) is the dual LP to (PLP), suppose x^* and y^* are the optimal solutions to (PLP) and (DLP), respectively, we have

$$\max_{x \in X} \min_{y \in Y} xAy = \lambda^* = x^* Ay^* = \gamma^* = \min_{y \in Y} \max_{x \in X} xAy.$$

Second, we prove (x^*, y^*) found above is a Nash equilibrium. The payoff of row player w.r.t. (x^*, y^*) is $x^* Ay^*$. For the sake of contradiction, suppose there exists $x' \in X$, s.t. $x' Ay^* > x^* Ay^*$. We can find $i' \in \{1, ..., n\}$, s.t. $e_{i'} Ay^* = (Ay^*)_{i'} \geq x' Ay^* > x^* Ay^* = \gamma^*$ ($(Ay^*)_i$ is the i^{th} entry of vector (Ay^*), and pick $(Ay^*)_{i'}$ as the largest entry of (Ay^*)), which implies $e_{i'} Ay^* > \gamma^*$. However, we have $e_{i'} Ay^* \leq \gamma^*$ in (DLP)'s constraints. So for any $x' \in X$, we have $x' Ay^* \leq x^* Ay^*$. Similarly, we can show that for any $y' \in Y$, we have $x^* Ay' \geq x^* Ay^*$. Therefore, (x^*, y^*) is a Nash equilibrium, and can be found through solving two LPs, which completes our proof.

We leave the following claim as an exercise: for any Nash equilibrium (x, y), $xAy = x^* Ay^*$. $\square$

1.2 Application: Lower Bounds on Las Vegas Randomized Algorithms

A *Las Vegas algorithm* for a problem is a randomized algorithm that always gives correct results, but its running time on any given input is a random variable. In other words, a Las Vegas algorithm does not gamble with the verity of the result — it only gambles with the resources used for the computation (from `http://en.wikipedia.org/wiki/Las_Vegas_algorithm`). A simple example is randomized *quicksort*, where the *pivot* is chosen randomly. The running time of quicksort is a random variable of the way pivots are picked, but the result is always sorted.

Given problem Π, we will show how to prove a lower bound on the randomized complexity of Π in the following part. Let's first define the *randomized complexity* of problem Π: Given a collection of randomized algorithms $\mathcal{R}$ for problem Π, let $\mathcal{I}(n)$ be the set of all instances of Π of size n, and $\mathcal{R}(n)$ be the set of all randomized algorithms for Π that work correctly for instances in $\mathcal{I}(n)$.

Definition 1.2 (Randomized Complexity) *For an instance $I \in \mathcal{I}(n)$ and an algorithm $R \in \mathcal{R}(n)$, let $R(I)$ be the running time random variable of algorithm R on instance I, i.e., a mapping from the possible results of random experiments (the sample space) used in R to the running time of R on instance I. Let*

$$R(n) = \max_{I \in \mathcal{I}(n)} \mathbf{E}[R(I)]$$

be the maximum expected running time of R on an instance I of Π of size n. The randomized complexity of Π is defined as

$$RC(n) = \min_{R \in \mathcal{R}(n)} R(n) = \min_{R \in \mathcal{R}(n)} \max_{I \in \mathcal{I}(n)} \mathbf{E}[R(I)],$$

i.e., the best worst-case running time of a randomized algorithm for problem Π.

Note we can restrict $\mathcal{R}$ to be the collection of Las Vegas algorithms in the following discussion.

The general idea to prove a lower bound on the randomized complexity of problem Π is: Imagine that in a 2-player zero-sum game (two players use mixed strategies), row player generates a problem instance of Π, and column player picks an algorithm in his/her bag to solve this problem. The payoff of row player is the running time of the algorithm picked by column player, and he/she wants to maximize the payoff, i.e., to deteriorate the performance of the algorithm. Intuitively, "how bad the performance can deteriorated" is a lower bound on the randomized complexity of problem Π. We will formally state and prove this intuition in the following part.

Let $\mathcal{D}(n)$ be the set of all deterministic algorithms for Π that work correctly for instances in $\mathcal{I}(n)$. We have the following proposition about the relationship between $\mathcal{R}(n)$ and $\mathcal{D}(n)$.

Proposition 1.3 *Any randomized algorithm $R \in \mathcal{R}(n)$ is a probability distribution over $\mathcal{D}(n)$.*

Intuitively, once the results of all random experiments used in a randomized algorithm $R \in \mathcal{R}(n)$ are fixed, R becomes a deterministic one. In other words, a deterministic algorithm $D \in \mathcal{D}(n)$ corresponds to a sample from random experiments used in a randomized algorithm R, and thus any $R \in \mathcal{R}(n)$ can be considered as a probability distribution over $\mathcal{D}(n)$.

Assumption 1.4 *$\mathcal{I}(n)$ and $\mathcal{D}(n)$ are both finite.*

The above assumption works for many problem. We clarify it as follows. Suppose the space, i.e., the memory of a computer, is bounded to be L bits. There may be at most 2^L different inputs of problem Π. Also, because a deterministic algorithm corresponds to a piece of binary code of length at most L, there are at most 2^L different deterministic algorithms for problem Π. Let $|\mathcal{I}(n)| = g$ and $|\mathcal{D}(n)| = h$. Then

$$\mathcal{R}(n) = \{\text{column vector } y \in [0,1]^h | \sum_{j=1}^{h} y_j = 1\}.$$

Let A be a $g \times h$ payoff matrix, where $A_{i,j}$ is the running time algorithm $j \in \mathcal{D}(n)$ on instance $i \in \mathcal{I}(n)$. Then from Definition 1.2, we have

$$RC(n) = \min_{y \in \mathcal{R}(n)} \max_{1 \le i \le g} e_i A y,$$

where e_i be a row vector $[0, ..., 0, 1, 0, ..., 0]$, whose i^{th} entry, the only nonzero entry in e_i, is 1. Then by Theorem 1.1, we have

$$RC(n) = \max_{x} \min_{1 \le j \le h} x A \bar{e}_j, \tag{1}$$

where $\bar{e}_j$ be a column vector $[0, ..., 0, 1, 0, ..., 0]^T$, whose j^{th} entry, the only nonzero entry in $\bar{e}_j$, is 1. Variable x is carried out over all probability distribution on $\mathcal{I}(n)$, i.e., $\{\text{row vector } x \in [0,1]^g | \sum_{i=1}^{g} x_i = 1\}$.

Conclusion: In Equation (1), $\min_j x A \bar{e}_j$ means, for some distribution x on $\mathcal{I}(n)$, the expected running time of the best deterministic algorithm. Therefore, to prove a lower bound $\omega(n)$ on the randomized complexity of problem Π, we only need to find a probability distribution x_0 on $\mathcal{I}(n)$, s.t. $\omega(n) \le \min_j x_0 A \bar{e}_j$. Then we have

$$\omega(n) \le \min_{1 \le j \le h} x_0 A \bar{e}_j \le \max_{x} \min_{1 \le j \le h} x A \bar{e}_j = RC(n). \tag{2}$$

Example 1.5 *In the sorting problem, we have known that for algorithms that use only comparisons, sorting n numbers takes $\Omega(n \log n)$ time for any deterministic algorithm. One can extend the proof to the case where the n numbers are uniformly picked from all the $n!$ inputs. Then we have*

$$\min_{1 \le j \le h} x_0 A \bar{e}_j \in \Omega(n \log n),$$

where x_0 is the uniform distribution on all the $n!$ inputs. Therefore, from (2), the randomized complexity of sorting problem is $\Omega(n \log n)$.

2 Nash's Theorem

Theorem 2.1 (Nash's Theorem) *Any game with a finite set of players and finite set of strategies has a Nash equilibrium of mixed strategies.*

In this section, we will first introduce Brouwer's Fixpoint Theorem, and then prove Nash's Theorem using it.

2.1 Brouwer's Fixed Point Theorem

Theorem 2.2 (Brouwer's Fixed Point Theorem) *Let $f : [0,1]^d \mapsto [0,1]^d$ be a continuous function. Then there exists a point $x \in [0,1]^d$ s.t. $f(x) = x$. More generally, this theorem holds for $f; X \mapsto X$, where X is a compact and convex set in R^d.*

Recall a set $X \subseteq R^d$ is *compact* iff it is closed and bounded. A set $X \subseteq R^d$ is *bounded* iff there exists $r \in R$ s.t. $X \subseteq B(r)$, where $B(r)$ is a ball of radius r. A set $X \subseteq R^d$ is *closed* iff any convergent sequence $x_1, x_2, ...,$ where $x_i \in X$ has limit $x \in X$. Following are some examples of functions which do not have any fixed point ($f(x) = x$) because some conditions in Theorem 2.2 are violated.

- Function $f : [0,1] \mapsto [0,1]$ is not continuous —

$$f(x) = \begin{cases} 1, & x \in [0, 1/2] \\ 0, & x \in (1/2, 1]. \end{cases}$$

- Set X is not closed — $f : (0,1) \mapsto (0,1)$

$$f(x) \quad = \quad x + (1-x)/2.$$

- Set X is not simply connected — $X = \{(x,y) | x^2 + y^2 = 1\}$

$$f((\cos\theta, \sin\theta)) = ((\cos(\theta + \Delta), \sin(\theta + \Delta)), \quad \text{for fixed } \Delta \in (0, 2\pi).$$

Brouwer's Fixed Point Theorem in 1-dimensional case can be easily proved as follows.

Consider any continuous function $f : [0,1] \mapsto [0,1]$, if $f(0) = 0$ or $f(1) = 1$, then we have done. Otherwise, let $g(x) = x - f(x)$, we must have $g(0) = a < 0$ and $g(1) = b > 0$. Since $g(x)$ is continuous and $g(0) = a < 0 < b = g(1)$, by Intermediate Value Theorem, for any value $y \in [a,b]$, there exists $x \in [0,1]$ s.t. $g(x) = y$. Specifically, for $y = 0$, there exists $x_0 \in [0,1]$ s.t. $f(x_0) = x_0$.

2.2 Proof of Nash's Theorem

In the following part, we prove Nash's Theorem using Brouwer's Fixed Point Theorem.

Define a game as $(N, < S_i >, < u_i >)$, where $N = \{1, ..., n\}$ is the set of n players, S_i is the set of possible strategies of player i, and $u_i : S = S_1 \times ... \times S_n \mapsto R$ is the payoff function of player i. Note this is finite game ($|S_i|$ is bounded), and thus $B = \max_{s \in S} \max_i u_i(s)$ is finite.

In derived game $(N, < \Delta(S_i) >, < u_i >)$, let $\Delta(S_i)$ be the set of all possible mixed strategies which player i can pick,

$$\Delta(S_i) = \{p_i \in [0,1]^{|S_i|} | \sum_{a \in S_i} p_i(a) = 1\},$$

where $p_i(a)$ is the probability player i chooses strategy $a \in S_i$. Let $\mathcal{P} = \Delta(S_1) \times ... \times \Delta(S_1)$ be the set of all possible ways in which players can pick mixed strategies. Obviously, $\mathcal{P}$ is convex (and thus, simply connected) and compact. The idea to prove Nash's Theorem is to define a continuous function $f : \mathcal{P} \mapsto \mathcal{P}$ s.t. $f(p) = p$ iff p is a Nash equilibrium. From Brouwer's Fixed Point Theorem, there must exist $p \in \mathcal{P}$ s.t. $f(p) = p$, which completes the proof. In the following part, we will focus on the construction of such a function f.

For $p = (p_1, ..., p_n) \in \mathcal{P}$, where $p_i \in \Delta(S_i)$, define $f(p) = (f_1(p), ..., f_n(p))$, where $f_i(p) \in \Delta(S_i)$. Note f is a map from strategies to strategies. We want to construct a continuous function f, s.t. i) if p is a Nash equilibrium, then $f_i(p) = p_i$; ii) if p is not a Nash equilibrium, then $f_i(p) = q_i \neq p_i$, where q_i is an "improvement" based on p_i.

Recall $B_i(p_{-i}) = \{a \in S_i | a \text{ is a best pure strategy w.r.t. } p_{-i}\}$, where p_{-i} is the mixed strategies of players except player i. Let

$$\alpha = \sum_{a \in B_i(p_{-i})} (u_i(a, p_{-i}) - u_i(p_i, p_{-i})),$$

where $(u_i(a, p_{-i}) - u_i(p_i, p_{-i}))$ is the improvement of player i's payoff from mixed strategy p_i to pure strategy a while other players' strategies are fixed. Obviously, $\alpha = 0$ iff p_i is a best response to p_{-i}, and in general, α is bounded. Let $f_i(p) = q_i$, where

$$q_i(a) = \begin{cases} \frac{p_i(a) + [u_i(a, p_{-i}) - u_i(p_i, p_{-i})]}{1+\alpha} & a \in B_i(p_{-i}) \\ \frac{p_i(a)}{1+\alpha} & a \notin B_i(p_{-i}). \end{cases}$$

Note $p_i(a)$ and $q_i(a)$ are the probabilities player i chooses pure strategy a, in mixed strategy $p_i \in \Delta(S_i)$ and $q_i \in \Delta(S_i)$, respectively. We can prove:

- i) $f_i(p_i, p_{-i}) = p_i \Leftrightarrow \alpha = 0 \Leftrightarrow p_i$ is a best response to p_{-i};
- ii) $f_i(p)$ is continuous because $u_i(p)$ (as well as $u_i(a, p_{-i})$ and $u_i(p_i, p_{-i})$) is continuous, and α, as a function of p, is also continuous.

From i), we have $f(p) = p \Leftrightarrow p$ is a Nash equilibrium. From ii) and Brouwer's Fixed Point Theorem (recall $\mathcal{P}$ is convex and compact), there exists $p \in \mathcal{P}$ s.t. $f(p) = p$, i.e., p is a Nash equilibrium.