Srikar Varadaraj

Department of Computer Science
Columbia University

May 2016

Abstract

This document provides a self-contained introduction to Game Theory, Nash's Equilibrium Existence Theorem and the use of Homotopic Methods to find Nash Equilibria. The focus of the work is to look at the hardness of finding Nash Equilibria in two-person games using the Lemke-Howson algorithm. We closely describe the arguments in [11],[12] and [2]. The intended aim of the document is to provide strong intuition for the working of Lemke-Howson, survey the few major results on its complexity, and attempt to pave a path for understanding "average-case" hardness.

Table of Contents

Page

CHAPTER 1

Existence of Nash Equilibria

The story begins with John Nash's 1950 paper [9] "Equilibrium Points in n-Person Games" and an elaboration of the argument in a 1951 paper. The initial communication, which is only a page long, cites two references (Kakutani and John-Von-Neumann) and states the existence of a Nash Equilibrium as a direct consequence of Kakutani's fixed point theorem. It introduces n-person games:

> "One may define a concept of an n-person game in which each player has a finite set of pure strategies and in which a definite set of payments to the n-players corresponds to each n-tuple of pure strategies, one strategy being taken for each player. For mixed strategies, which are probability distributions over the pure strategies, the pay-off functions are the expectations of the players, thus becoming polylinear forms in the probabilities with which the various players play their various pure strategies..."

and proceeds to conclude after a short argument:

> "Since the image of each point under the mapping is convex, we infer from Kakutani's theorem' that the mapping has a fixed point (i.e., point contained in its image). Hence there is an equilibrium point."

Indeed, Nash's revolutionary theorem is nothing but a simple application of a fixed-point theorem. It is often the case that with the genesis of a new line of research, there is a marked evolution of terms, ideas and the language in which these ideas are communicated. However the language in [9] is surprisingly modern and easy to understand. We begin our journey here by quickly introducing n-person games, the idea of an equilibrium and fixed point theorems.
We use some of the definitions and notation from [6] for the initial part of the introduction.

1.1 Definitions and Introduction

Definition 1. *A finite n-person game in normal form is a tuple* (N, A, O, μ, u) *such that:*

- *N is a finite set of players*
- *$A = (A_1, ..., A_n)$ is the set of action profiles. A_i is a finite set of actions available to player i.*
- *O is a set of outcomes*
- *Given an action profile, function $\mu : A \to O$ gives the outcome.*
- *$u = (u_1, ..., u_n)$ where $u_i : O \to \mathbb{R}$ is the payoff function, or utility function for player i.*

In general, we do not really need the notion of O or μ. Typically, every possible set of actions corresponds to some utility vector for the players. Each player attempts to maximize their own utility.
We typically only consider the normal form game (N, A, u) as a result. We can write $[N]$ instead of N, indicating that each player is indexed by $i \in [N]$.

(Player 2)

		Confess	Silent
(Player 1)	Confess	4 , 4	1 , 5
	Silent	5 , 1	2 , 2

The prisoner's dilemma gives a concrete example of a 2-player game. Player 1 and Player 2 have actions "Confess" and "Silent". Their respective prison time is given as ordered pairs. Their utilities would be $-1*$(time).

Definition 2. *A pure strategy profile for player i corresponds to one specific action $a_i \in A_i$ from the player's set of possible actions.*

If a player is allowed to randomize over the possible set of actions, we instead get a mixed strategy, which allows more variability than pure strategies. The concept of a mixed strategy guarantees the existence of a Nash equilibrium, as we will see soon.

Definition 3. *A mixed strategy for player i is a set of probabilities $(p_1, ..., p_{|A_i|})$ denoting the probability with which player i chooses action a_i for each $a_i \in A_i$.*

Now, it is conceivable that a player i chooses to never opt for an action $a_i \in A_i$. In this case, we would like a way to distinguish the non-influential actions from actions which contribute to the player's strategy profile.

Definition 4. *The support of a mixed strategy s_i for a player i is the set $\{a_i \in A_i | s_i(a_j) > 0\}$, so in the strategy, the action $a_i \in A_i$ is selected with non-zero probability.*

The notion of utility is important for finding equilibria, since players want to maximize some objective function associated with their actions.

Definition 5. *Expected utility u_i of player i where the mixed strategy profile of the game (all players) is $s = (s_1, ..., s_n)$ can be written as:*

$$u_i(s) = \sum_{a \in A} u_i(a) \Pi_{j=1}^{n} s_j(a_j)$$

We also define s_{-i} to denote the strategy profile of all players except i.

Definition 6. *The Best Response of player i to strategy s_{-i} is s'_i, such that $u_i(s'_i, s_{-i}) \geq u_i(s_i, s_{-i}) \forall s_i \in S_i$, the possible set of strategies of player i.*

We make two observations:

- A mixed strategy is a best response iff all pure strategies in the support are best responses for player i. This is obvious, because if any pure strategy "dominated" another, then the player would obviously prefer playing the dominating strategy with higher probability. Hence, the original mixed strategy would not be a best response. So, each player combines pure best response strategies.

- Any mixture of two pure strategy best responses is also a best response.

With this intuition in mind, we can finally define mixed strategy Nash equilibrium:

Definition 7. *A strategy profile $s = (s_1, ..., s_n)$ is considered to be a Nash equilibrium if $\forall i$, s_i is a best response to s_{-i}.*

Note that a pure-strategy Nash equilibrium is not always guaranteed to exist (example: 2 player tic-tac-toe). However, Nash [9] proved that a mixed-strategy NE always exists in an n-player, finite game. We will now introduce the necessary mathematical tools to prove this theorem. Note that this equilibrium does not have to be unique, but at least one is guaranteed to exist.

1.2 Brouwer, Sperner and Nash

We begin by introducing simplices and the notion of affine independence.

Definition 8. *A set of vectors $(x_0, ..., x_n)$, $x_i \in \mathbb{R}^k$ is affinely independent if:*

$$\sum_{i=0}^{n} \lambda_i x_i = 0, \sum_{i=0}^{n} \lambda_i \implies \lambda_i = 0 \forall i \in \{0, 1, ..., n\}$$

We also need the notion of a simplex before we introduce Sperner's lemma.

Definition 9. *An n-simplex is the set of all convex combinations of a set of affinely independent vectors* $(x_0,...,x_n)$*. It can be described as:*

$$x_0...x_n = \{\sum_{i=0}^{n} \lambda_i x_i, \lambda_i \geq 0, \sum_{i=0}^{n} \lambda_i = 1\}$$

For $n = 2$, the definition reduces to the standard notion of a triangle in $\mathbb{R}^2$. An n-simplex is simply a triangle's generalization to n dimensions. We refer to each x_i as a vertex of the simplex and a set of $k+1$ vertices as a k-face of the simplex.
We would like to define a "unit" simplex, which consists of convex combinations of unit vectors. This notion is defined as follows.

Definition 10. *The standard n-simplex is the set of all vector* $v \in \mathbb{R}^{n+1}$ *such that* $\sum_{i=0}^{n} v_i = 1$ *and* $v_i \geq 0 \forall i$.

Given a simplex S, say, we can divide it into multiple simplices $\{S_i\}$ such that their union $\cup_i S_i$ is S and any two "sub-simplices" S_i, S_j can only intersect along a facet. We refer to this as a "simplical subdivision". Now, if we have a point $\alpha \in x_0...x_n$, then $\alpha = \sum_{i=0}^{n} \lambda_i x_i$. We can then associate to α a set $f(\alpha)$ consisting of all x_i such that $\lambda_i > 0$. Hence, we can consider $f(\alpha)$ to be a set that contains all the vertices that "contribute" to α. Some observations and statements:

- We can call a labeling of a simplex S divided into $\{S_i\}$ "proper" if for each vertex t in the set of vertices T defined by the subsimplices, the set of "labels" (subset of indices from 0 to n) associated to t is a subset of $f(t)$.
- With this definition, we always have the different vertices of a simplex having different labels.
- We have a "complete" labeling when every label $0,1,...,n$ is used on the vertices.

We are now in a position to state Sperner's lemma/theorem.

Theorem 1. *If we have a simplical subdivision of simplex* $S = x_0...x_n$ *and a proper labeling of the simplex, then there are an odd number of "completely labeled" subsimplices.*

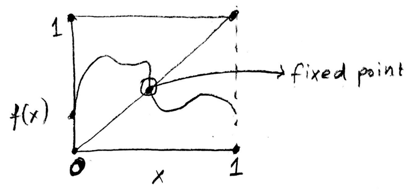
The proof relies on a careful induction on n, which we won't do here. But for intuition, note the following:

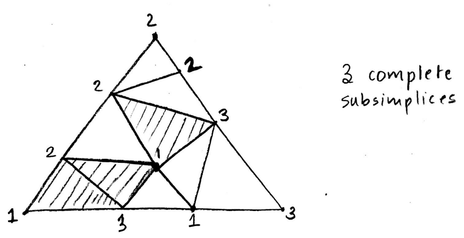
- In dimension 1, Sperner's lemma is a restatement of the Mean Value Theorem. It says that if we have a segment with endpoints colored red and blue (different colors), and if we divide this segment into many subsegments, choosing colors randomly for the endpoints, then when we write out the order of colors going from one side of the segment to the other, say: R,B,B,B,R,R,B, then the number of times we "flip" between different colors is odd. Indeed, we flip three times in the above sequence.

- The case of $d = 2$ is a little more counter intuitive. We start with a triangle and label the vertices differently, say $1, 2, 3$. Now, we divide the triangle into sub-triangles/sub-simplices such that any vertex on the edge of the simplex (big triangle) must use one of the two labels found on the edge of the large triangle (the simplex we began with). Sperner's lemma then tells us that the number of subsimplices, or smaller triangles, that use all three $1-2-3$ labels (complete labeling) is odd.

Now where do we go from here? It seems that we have a theorem about discrete mathematical objects, but Nash's theorem is on continuous utility functions. A limit argument is due. Indeed, we have another theorem that makes the connection a little clearer:

Theorem 2. *Brouwer's Fixed Point Theorem says that if we have a continuous mapping $f : S_n \to S_n$ from a subdivision of a simplex to itself, then there exists a point $t \in S_n$ such that $f(t) = t$.*





In the diagram above, we see Brouwer's Fixed point theorem for $f : [0, 1] \to [0, 1]$ and Sperner's lemma for a 2-simplex subdivision. In the statement of the theorem, we look at simplices as convex bodies. It is clear that a simplex is compact in Euclidean space since it is closed and bounded (and Heine-Borel gives the equivalence between these conditions). Hence, every infinite sequence of points contains a converging subsequence. The non-constructive, "existence" proof of Brouwer uses this very fact and it is here that the complexity troubles with finding Nash equilibria begin.

After giving a proper labeling of the simplex and introducing a subdivision such that the "distance" between (the centroids of) any two adjacent subsimplices is $\leq \epsilon$, we have the existence of a sequence of completely labeled subsimplices that converge to a fixed point.
With Brouwer's Fixed Point theorem for continuous mappings of simplices to themselves, we generalize to the continuous mapping of cartesian products of simplices to itself. These are referred to as "simplotopes".
Nash's insight at this point was to construct a function that acted as a continuous mapping between strategy profiles of players in a finite game. This function would necessarily have a fixed point if we guarantee "compactness" of sets and if we can show that the fixed points correspond to Nash equilibria. We end this part of the document with Nash's theorem and the construction of his function used to guarantee that fixed points and NE correspond to each other.

Theorem 3. *Nash - Every finite n-player game has at least one mixed strategy Nash Equilibrium.*

Proof. We are given strategy profile $s \in S$ for the set of players and action profiles $a_i \in A_i \forall i \in [N]$.
Let:

$$\phi_{i,a_i}(s) = max\{u_i(a_i, s_{-i}) - u_i(s), 0\}$$

and

$$f : S \to S, f(s) = s', s'_i(a_i) = \frac{s_i(a_i) + \phi_{i,a_i}(s)}{\sum_{\alpha_i \in A_i} s_i(\alpha_i) + \phi_{i,\alpha_i}(s)} = \frac{s_i(a_i) + \phi_{i,a_i}(s)}{1 + \sum_{\alpha_i \in A_i} \phi_{i,\alpha_i}(s)}$$

Note in the denominator, it is $\alpha_i \in A_i$, not a_i. The idea of the proof is as follows:

- f takes a strategy profile s to another strategy profile s.
- The use of f is to "reward", or redistribute probability mass to actions a_i that are better (as judged according to ϕ).
- If s is a NE, then $\phi_{i,a_i} = 0 \forall i, a_i$, so $s'_i(a_i) = s_i(a_i)$, hence s is a fixed point of f.
- On the other hand, note that for a fixed point $f(s) = s \implies \phi_{i,a_i}(s) = s_i(a_i) \sum_{\alpha_i \in A_i} \phi_{i,\alpha_i}(s)$.
- Note that $\phi_{i,a_i}(s) = 0$ if there exists some action a_i^* in the support of s_i (played with positive probability) such that $u_i(s) \geq u_{i,a_i^*}(s)$. Hence, in this case, $\sum_{\alpha_i \in A_i} \phi_{i,a_i}(s) = 0$, which implies that no player has any incentive to switch from s_i unilaterally to another pure strategy profile. Note that a_i^* satisfying the above condition must always exist since if $u_i(s) < u_{i,a_i^*}(s)$ for all possible a_i^*, then s would not be a fixed point.
- And finally, note that f, ϕ are continuous functions, so a fixed point always exists. Also note that profiles S take the structure of simplotopes and hence are compact sets.

■

CHAPTER 2

PPAD AND NASH-EQUILIBRIA

Let us begin by defining NASH as the problem of finding a Nash Equilibrium given a game matrix. NP-completeness is not a very useful measure of complexity for NASH since every game has a guaranteed mixed nash equilibrium. This is the reason complexity theorists looked at the class PPAD to classify NASH.
The proof of the existence of NE in the previous chapter was non-constructive. We were essentially able to show that there is a reduction:

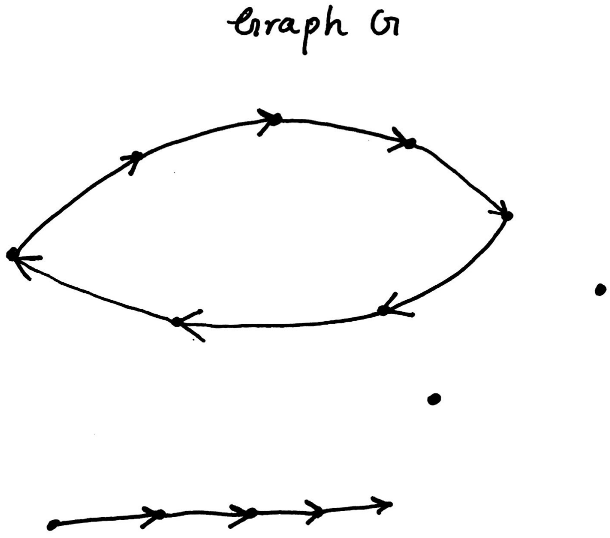
$$NASH \implies BROUWER$$

In other words, solving BROUWER efficiently, the problem of finding fixed points for a continuous mapping between homeomorphic compact spaces, would give us efficient solutions to NASH. It turns out that:

$$BROUWER \implies NASH$$

as well, hence finding a NE is just as hard as finding a fixed point. To capture the difficulty of finding these fixed points, Christos Papadimitriou introduced the complexity class PPAD ("Polynomial Parity Argument on Directed Graphs") in 1994. The goal of defining this class of problems was to have a benchmark for talking about "search problems" where the existence of an object is guaranteed.
This guarantee is preserved in the description of a directed graph with an unbalanced node. The idea is that a directed graph with an unbalanced node implies the existence of another unbalanced node. Searching for the other unbalance node is the problem in PPAD.
We begin by describing the EOTL, or End of The Line problem, which was shown to be PPAD-complete.
Suppose we have a directed graph G such that $|G| = 2^n$. We also impose the condition that every

vertex $v \in G$ has at most one incoming edge and at most one outgoing edge. This implies that G can be decomposed into a set of paths and cycles. So, for example, it must necessarily look something like the figure below.



Each vertex in the graph G is given a unique n-length binary string. We also have predecessor and successor functions p, s respectively, such that if $v - v'$ is an edge in the graph, then $p(v') = v$ and $s(v) = v'$. The problem is then: If we are given a starting vertex e_0 that is unbalanced, how do we find vertex e_1 that is unbalanced as well?
Once it was shown that EOTL was PPAD complete, multiple efforts were taken to reduce NASH and BROUWER to EOTL. The efforts of Daskalakis, Chen [1], etc. eventually showed that NASH is PPAD-complete, giving us a clue for why it might be quite difficult to find a NE after all.
It is surprising that even the problem of ϵ-approximating is difficult.

2.1 Approximating NE

Definition 11. *A Brouwer function F is a continuous function from a convex, compact domain D to itself.*

By Brouwer's theorem, there exists a fixed point $x \in D$ such that $F(x) = x$.

Definition 12. *If F is a Brouwer function, then an ϵ-approximate fixed point is a point x such that $|F(x) - x| \leq \epsilon$.*

Now, Browder's theorem from the previous chapter implies that there exists for any F_0, F_1, a finite sequence $x_0, x_{t_1}, \ldots, x_{t_k}, x_1$ of "$\epsilon-$approximate" fixed points of $F_0, F_{t_1}, \ldots, F_{t_k}, F_1$ such that

$|x_{t_i} - x_{t_{i+1}}| \leq \epsilon$ for any i in the sequence. Since we know that finding a fixed point corresponds to the existence of a NE for appropriately defined functions, we can see intuitively that there must exist ϵ-NE approximations as well. The question then is, how hard is it to find ϵ approximations for NE, such that the utilities of each player at the approximate NE are no more than ϵ smaller than if they played a different strategy.

In [3], the authors prove that it is PPAD-hard to find an ϵ-approximate NE if $\epsilon = \Theta(\frac{1}{2^{|S|}})$, where $|S|$ is the size of the game instance under consideration. In [2], it was further proved that finding an $\epsilon-$ NE for 2-NASH where each player has n strategies is PPAD-complete even when $\epsilon = \Theta(\frac{1}{poly(n)})$. This shows that there isn't a FPTAS for finding NE, but the problem of finding a PTAS is still open.

Chen and co. in [2] were able to show their result by generalizing the arguments of [3] to finding Brouwer fixed points on $n-$dimensional hypercubes. We do not go through the proof here, but the interested reader is encouraged to compare the proofs of [3] and [2] for insight into the hardness of approximating NE.

Chapter 3

LH, Homotopies and a Worst Case Example

In [11], Rahul Savani and his advisor, Bernhard Von Stengel, presented the first class of games for which the Lemke-Howson algorithm takes an exponential number of steps to find any Nash equilibria in the game. Previously, Morris [8] had used “dual cyclic polytopes” to produce exponentially long “Lemke paths”, finding symmetric equilibria in a symmetric game. However the games that Morris initially constructed also had nonsymmetric equilibria for which LH terminates very quickly. Savani sought to introduce games for which LH computation to find any equilibrium would take exponentially long paths. In fact, the square-matrix 2-player games constructed in [12] have exactly one nash equilibria.
Lemke Howson (LH) was initially constructed to work for 2-player games. It is perhaps the most widely known algorithm for finding NE for 2-NASH. We will describe the workings of LH and introduce the idea of a “polytope”. We describe the construction in [11] in this chapter and see how the worst case examples for LH are constructed. Before we do that, we discuss a larger class of methods for finding NE which generalizes the properties of LH.

3.1 Homotopic description of LH

The LH algorithm can be thought of topologically as finding a path from an initial starting set of strategies to the final equilibrium, making specific changes called “pivots” at every iteration. For an in-depth discussion on homotopy methods, refer to [5].

Definition 13. *A homotopy between continuous functions f,g from a topological space X to another space Y is a continuous function $H:[0,1]\times X \to Y$ such that $H(0,x)=f(x)\forall x\in X$ and $H(1,x)=g(x)\forall x\in X$.*

What exactly does this mean? Well, if we think of a gradual shift from the function f to g in time t, then at time 0, we can imagine $H(0,x) = f(x) + 0 * g(x)$. At an intermediate time $t \in (0,1)$, we can imagine something of the form $H(t,x) = (1-t)f(x) + tg(x)$ such that at time $t = 1$, we get $H(1,x) = g(x)$. This is a continuous deformation process.
Browder's (not to be confused with Brouwer) Theorem gives a way to relate notions of homotopies and fixed points.

Theorem 4. *Let $S \in \mathbb{R}^n$ be a non-empty compact, convex set. Let $H : [0,1] \times S \to S$ be a continuous function. The set of fixed points is denoted by $F_H = \{(\lambda, s) \in [0,1] \times S | H(\lambda, s) = s\}$.*
Then, F_H contains a connected set F_H^c satisfying the following properties:

- $(\{0\} \times S) \cap F_H^c \neq \phi$
- $(\{1\} \times S) \cap F_H^c \neq \phi$

What this means is that a fixed point of f is in the same connected set as a fixed point of g. So, we can formulate a general strategy for finding fixed points:

- Given a final problem, formulate it in terms of finding a fixed point for a function $H_1 : \{1\} \times S \to S$. Finding a fixed point would give a solution to the "final problem".
- Phrase the "starting problem" as a fixed point problem as well for the function $H_0 : \{0\} \times S \to S$ such that the fixed point here is computable.
- Ensure that by gluing together H_0 and H_1, we have a continuous function on $[0,1] \times S$.
- Browder's theorem tells us that the fixed point $(0, s_0)$ for H_0 (or just H) is connected to the fixed point $(1, s_1)$ for H_1 (they belong to the same connected set).
- By formulating effective ways of going through the connected set F_H^c from one fixed point to another, we can compute the fixed point for the given problem.

Lemke-Howson,van den Elzen-Talman and McKelvey-palfrey are all well known algorithms for computing NE that fall under the general category of "homotopic methods".
We quickly introduce the "homotopic description" for LH. Hopefully it gives us a better understanding of the underlying behavior and geometry of LH. We talk exclusively of 2-player "bimatrix" games from here on out, unless otherwise mentioned.

- Let Γ be a bimatrix game.
- Let $\alpha \in \mathbb{R}^+$ be a positive value large enough such that any pure strategy can be made a dominant strategy if added to the utility for this strategy, say for a_i of player i.
- We can let $\alpha = max_{i,s} u_i(s) - min_{i,s} u_i(s) + \epsilon$, where $\epsilon > 0$ is a small positive constant.

- For $t \in [0,1]$, we can define the bimatrix game $\Gamma(t) =< [N], A, \{v_i(t,\cdot)\} >$, where $[N]$ is the set of players, A is the set of all possible action vectors and v_i gives the new payoff/utility function for player i defined by $v_i(t,s) = u_i(s)+(1-t)\alpha$ if $s_i = a_i$ and $v_i(t,s) = u_i(s)$ otherwise.

- $v_{-i}(t,s) = u_{-i}(s)$ for all other players.

- Then, $H(t,s)$ is defined to be $\Pi_{h\in[N]}\beta_h(t,s)$ where the function $\beta_h(t,s)$ is defined as: $\text{argmax}_{s'_h \in S_h} v_h(t,s^{-1},s')$, the best response correspondence of player h in the new constructed game $\Gamma(t)$ with utility function v_h for player h instead of the original utility.

- For the game $\Gamma(1)$, we have no contribution of α, hence a strategy and a best response correspond to NE directly. A fixed point of $\Gamma(1)$ is what we would like to obtain.

- For the game $\Gamma(0)$, note that the strategy a_i and a best response gives an NE.

- By gradually varying t from 0 to 1, we reduce the weight of α.

- It turns out that the set of strategy profiles of the homotopy path as t goes from 0 to 1 are precisely the same strategy profiles generated by LH!

- By Browder's theorem, we know that a NE of $\Gamma(0)$ is connected to an NE of $\Gamma(1) = \Gamma$, the game we want to find an equilibrium for.

From the above description, there is clearly a natural "homotopic" way of looking at LH. We won't go into further discussion of homotopic methods, but it is clear that if we know how to apply homotopic methods for functions $H(t,s)$ that we defined above, then we can effectively calculate the complexity of LH and its behavior on certain games. The interested reader should consult [5] for a further discussion.

3.2 Lemke Howson

3.2.1 Payoffs, Polytopes, Best-Response

We consider bimatrix (2-player) games in normal form from here on out. In [7], the authors introduce the algorithm as an "algebraic proof" for the existence of mixed strategy NE in finite two player games. Indeed, the authors prove that there must exist an odd number of equilibrium points in any non-degenerate 2-player game. This implies that there must exist at least one equilibrium point, as required. We begin by introducing some terminology, definitions, and eventually the algorithm itself. We refer to [13] and the original paper [7] for a clear exposition of the material. David Pritchard's lecture notes also provided solid intuition for the approach below. A bimatrix game consists of two players, with two associated payoff/utility matrices:

$$A = \{a_{ij}\}, B = \{b_{ij}\}$$

where $i, j \in I, J$ respectively.

$$I = \{1, ..., m\}, J = \{m+1, ..., m+n\}$$

denote the set of actions/pure-strategies possible for players A and B (or 1 and 2), respectively. The mixed strategy profile for the players can be represented as (x^T, y), where x, y are column vectors indicating the mixed strategies of players A,B respectively.
Few observations:

- Payoff for player 1 = $x^T Ay$ and Payoff for player 2 = $x^T By$.
- All entries of A,B are assumed nonnegative. Adding a positive number to every single element in matrices A and B does not change the NE structure, so this is admissible.
- We can brute force the problem by guessing the supports of the profiles at equilibrium for each player and then solving a simple LP (which can be done in poly-time). However, there are $O(2^m)$,$O(2^m)$ support sets for the players, so trying every single one until an equilibrium is found would be inefficient and take exponential time.

Definition 14. *A polytope in $\mathbb{R}^n$ is the convex hull of a finite set of points and a facet is a maximal dimension face.*

A polytope is described in $\mathbb{R}$ by a system of linear inequalities. We can define subdivisions and formulate a generalization of Sperner's lemma for polytopes as well, although we won't need it here. Essentially, we can think of a polytope as a generalization of a simplex (which was an abstraction of a triangle in two dimensions).
For our purposes, we need to consider two polytopes:

$$P_1 = \{x \in \mathbb{R}^m | (\forall i \in I : x_i \geq 0) \& (\forall j \in J : x^T B_j \leq 1)\}$$

and

$$P_2 = \{y \in \mathbb{R}^n | (\forall j \in J : y_j \geq 0) \& (\forall i \in I : A_i y \leq 1)\}$$

We also normalize vectors x, y to be:

$$x' = \frac{x}{\sum_i x_i}$$

and

$$y' = \frac{y}{\sum_j y_j}$$

just so they satisfy the probability distribution constraint (sum of probabilities = 1).

Lemma 1. *(Best response condition) If x, y are mixed strategies, then x is a best response to y iff $\forall i \in I$, we have:*

$$x_i > 0 \implies (Ay)_i = u = max\{(Ay)_k | k \in M\}$$

Proof.

$$x^T A y = \sum_{i \in I} (Ay)_i = \sum_{i \in M} x_i (u - (u - (Ay)_i)) = u - \sum_{i \in I} x_i (u - (Ay)_i)$$

Note that $u - (Ay)_i \geq 0$ by definition, so $\sum_{i \in I} x_i (u - (Ay)_i) \geq 0$. Hence, $x^T Ay \leq u$. Equality can only arise when $u = (Ay)_i \forall i \in I$, as required. ∎

As described in [10], Ch.3, the interpretation of the best response condition is that if player 1's payoff is maximized on the face of a simplex, then it is maximized on all the vertices as well (and vice-versa). In the original paper [7], the LH algorithm is described to deal with non-degenerate 2-player games.

Definition 15. *A bimatrix game is non-degenerate if no mixed strategy of support size k has more than k pure best responses.*

Assuming we only deal with nondegenerate bimatrix games, we immediately have a corollary to the best response condition.

Corollary 1. *If (x, y) is a Nash Equilibrium for a bimatrix game, then the supports of x, y have equal size.*

Note the following:

- If $x \in P_1$ and $x_i = 0$, then i is not in the support of strategies for player 1.
- If $x \in P_1$ and $x^T B_j = 1$, then index j denoting an action of player 2, is a best response to x'.

We can now introduce the idea of "labeling". For a vector belonging to a polytope, the vector carries a label according to whether it satisfies equality in the inequality conditions described above.
For example, $x \in P_1$ has label $k \in I \cup J$ if ($k \in I$ and $x_k = 0$) or ($k \in J$ and $x^T B_k = 1$). Similar conditions hold for the labels of y.
We now state a simple theorem that relates the Nash Equilibria to the set of labels:

Theorem 5. *If $x \in P_1$ and $y \in P_2$, with $x, y \neq 0$ (the zero vector), then x, y have all labels in $\{1, ...m + n\}$ iff (x', y') is a NE.*

Definition 16. *A pair $(x, y) \in P_1 \times P_2$ is called k-almost completely labeled if every label in $I \cup J - \{k\}$ appears as a label of either x or y.*

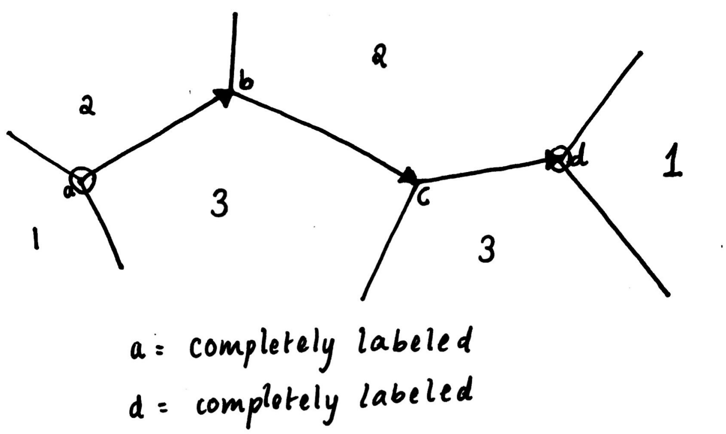
At this stage, we can begin our description of the Lemke-Howson Algorithm.

3.2.2 Description of LH

The LH algorithm finds a single Nash equilibrium and hence provides an elementary proof of the existence of NE. The path followed by the algorithm is called an "LH path". The algorithm works as follows:

- Start at $(0,0) \in P_1 \times P_2$.
- Note that $(0,0)$ is a completely labeled pair since every pure strategy has probability 0, so the coordinates have m, n labels respectively. Since $m + n$, we consider it a "completely labeled" pair.
- We arbitrarily pick an initial free choice, which is a strategy $k \in I \cup J$, say $k \in I$. This label is known as the "missing" label.
- Then, the label k is dropped and we move along an edge connecting it to a vertex that picks up a new label in P_1. This label must be duplicate in J, so we drop it and move along an edge in P_2, picking up a new label.
- This process keeps going on until the missing label is encountered again (picked up), in which case we have a Nash equilibrium.
- The reasoning for this is that when we pick up the missing label at some stage, we are dropping a duplicate label and adding the missing label to the set of labels for (x, y). This is a completely labeled vertex pair in $P_1 \times P_2$, and hence we have a Nash Equilibrium!

The LH algorithm as described above must always terminate, since the set of vertex pairs in $P_1 \times P_2$ is finite. As Stengel [10] describes it, we must think of the LH algorithm as following a directed path on the product polytope $P_1 \times P_2$ on pairs that are $k-$almost completely labeled. The following figure (from Savani) is a simple example of LH in the third dimension. LH is a "pivoting" algorithm that pivots along the dropped/duplicated label at every stage.



$a \to$ drops (1) $\to$ drops (2), picks up (3) $\to$ drops (3), picks up (1) $\to$ end at d = NE (complete labeling). Some further observations on LH:

- For the case of nondegenerate bimatrix games with simple polytopes, the path from a starting NE is unique.
- We do not have to start with the artificial NE at $(0,0)$. We could start at any NE and reach another NE, although if we start at an NE that is not $(0,0)$, we might end up at $(0,0)$.
- LH does not guarantee that it will find all possible NE. In fact, games can be constructed where certain NE cannot be reached. However, the guarantee is that there is always one that is reached.
- Finally, by looking at the graph structure of the $k-$almost completely labeled vertices, we can conclude that a nondegenerate bimatrix game has an odd number of NE.

For a more detailed explanation, refer to [10], Ch.3, [7] and [13]. Because LH performs computationally well on average, (only heuristics are given, there isn't a well structured theory explaining this phenomenon), it was of great interest initially to produce examples for which LH was guaranteed to perform very badly (exponential time). Savani came up with the first such example. We will look at an overview of his construction below.

3.3 Savani's Construction

When trying to come up with a worst-case example for an algorithm, we would like to guarantee that no matter how the algorithm proceeds, it will always reach its solution only after a certain number of guaranteed steps.
The best way to accomplish this for LH would be to construct a non-degenerate bimatrix game in which there exists exactly one non-trivial NE. LH would then guarantee that there is a unique path from $(0,0)$ to the NE.
If we can bound the length of this path and show it must be exponentially long in the size of the matrix, then we are done. In [12], the authors do precisely this.
We want to construct square games, which are simpler to analyze. So, $m = n = d$. We construct the games by first looking at objects called "dual cyclic polytopes".

Definition 17. *A cyclic polytope is a convex polytope represented as a convex hull of n distinct points on a rational normal curve in $\mathbb{R}^d$, with $n > d$. In particular, we can consider the moment curve $\{m(t) = (t^1, ..., t^d)^T\}$*

We won't go into too much detail at this point. The interested reader should look at lecture notes on cyclic polytopes for combinatorial properties and intuition.

Continuing with Savani's construction, first we take cyclic polytope P' with $2d$ vertices, translated and reconstructed as its dual. This results in its "dual cyclic polytope":

$$P'' = \{z \in \mathbb{R}^d | {c_i}^T z \leq 1, 1 \leq i \leq 2d\}$$

Properties of the "dual cyclic polytope":

- Vertices in the original polytope are normal vectors of facets (recall: max dimension faces) in the dual.
- Vertices of P'' are $2d$-length bit strings $u = u_1 ... u_{2d}$, each bit being an "indicator" for whether u is on the k'th facet.
- Polytope is simple (Think non-degenerate matrix. So, exactly d of the bits are 1, corresponding to max size of the support).
- Each ${c_i}^T z$ condition is made binding.

At this point, the author uses the following condition.

Theorem 6. *(Gale evenness condition) [4]*
Let $T = \{t_1, ..., t_n\}$. Then a size d subset $T_d \subseteq T$ forms a facet of $C(n,d)$, where $C(n,d)$ denotes the cyclic polytope formed by n points on a moment curve in $\mathbb{R}$, iff any two elements belonging to $T - T_d$ are separated by an even number of elements from T_d in the sequence $\{t_1, ..., t_n\}$.

The statement of the theorem is a little opaque without a better understanding of the combinatorial behavior of cyclic polytopes. We can still apply it to our particular case and conclude that: *A vertex of polytope P" is represented by u iff any substring of the form* 01...10 *has even length.*
We define $G(d)$ to be the set of bitstrings satisfying the Gale evenness condition, with length $2d$ and d ones. With affine transformations, we can go between P_1,P_1'' and P_2,P_2''.
The LH algorithm then proceeds depending on which facet a vertex belongs to. This information is encoded by the bitstrings in $G(d)$. The relationship is as follows:

- For vertex $u \in P_1$, the labels of its bitstring are $l(k)$, where $u_k = 1$. Similarly, the labels for vertex $v \in P_2$ are $l'(k)$, where $v_k = 1$. Here, $k \in [2d]$.
- Note that $l(k) = k$, so l fixes all indices. It is the identity permutation.
- On the other hand, $l'(1) = 1$, $l'(d) = d$ and otherwise,

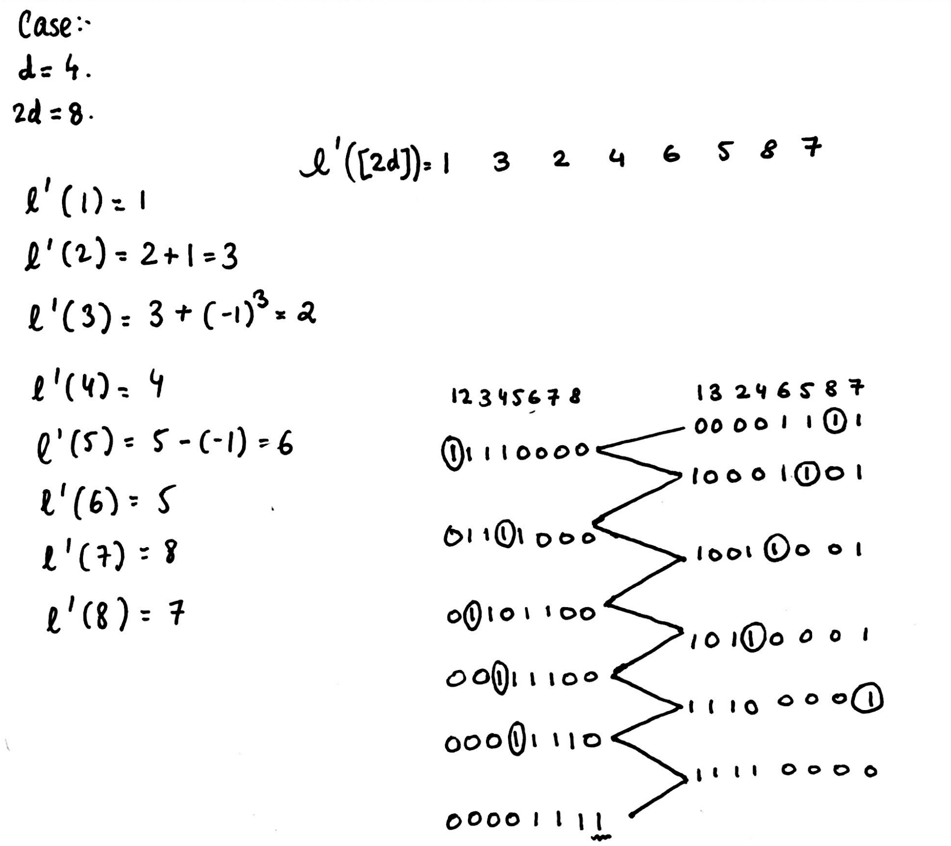
$$l'(k) = \begin{cases} k + (-1)^k & 2 \leq k \leq d-1 \\ k - (-1)^k & d+1 \leq k \leq 2d \end{cases}$$

- The "artificial equilibrium", which is the starting point of the LH computation, is $e_0 = (u,v)$ characterized by $(u,v) = (1^d 0^d, 0^d 1^d)$. Note that u,v satisfy the Gale evenness condition since there are no substrings of the form 01...10 in either one.

The remaining computation follows the steps:
Step 1: Find another NE in $P_1 \times P_2$.
Step 2: Show that this is the only NE apart from e_0.
Step 3: Use the properties of l, l' to show that the path from e_0 to e_1(say), is of length that grows exponentially in d.
Savani and Stengel [11],[12] prove the following first:

Lemma 2. *$e_1 = (0^d 1^d, 1^d 0^d)$ is a NE and in fact is the only NE of the game.*

Hence, there is a unique path from e_0 to e_1 once a label is dropped and this is the only possible path taken by LH. It does not matter which label is dropped initially.The authors use the notation $\pi(d,k)$ to denote the path from e_0 to e_1 when label k is dropped initially in dimension d. Below is an example for $\pi(4,8)$. The variables that are dropped (and picked up) are circled.



The remainder of the proof in [12] deals with finding properties of $\pi(d,k)$ and deducing combinatorial properties of $\pi(d,k)$ from the definition of l, l'. This stage of the proof is simply noticing patterns and mostly combinatorial calculations. If we let $L(d,k)$ denote the path length of $\pi(d,k)$, then the authors prove a theorem on the lengths of the paths.

Theorem 7. *The LH paths follow relations given by:*

- $L(d,k) = L(d,d-k+1)$, $L(d,d+k) = L(d,2d-k+1))$ *for* $1 \le k \le d$
- $L(d,k) = L(d,k+1)$ *for* $2 \le k \le d-2$ *and even* k*, and* $d+1 \le k \le 2d-1$ *and odd* k*.*
- $L(d,k) = L(k,1) + L(d-k,1)$ *for* $2 \le k \le d-2$ *and even* k
- $L(d,d+k)) = L(k,2k) + L(d-k+2,2(d-k+2)) - 4$ *for* $4 \le k \le d-2$ *and* k *even*

Analyzing these relations gives a simple connection to the fibonacci series. We can prove that the series of shortest path lengths $L(2n,3n)$ for $n = d/2 = 1,2,3,..$ is $4,10,16,42,68,178$, which is the Fibonacci sequence with every third number omitted and every term multiplied by two. To be explicit, $2,5,8,21,34,89,...$ is the fibonacci sequence, where we drop terms $3,13,55,...$
This finally gives the result that there exist $d \times d$ bimatrix (non-degenerate) games such that for even d, all LH paths have length $\Omega(\phi^{3d/4})$, where ϕ is the golden ratio.
The authors later generalize their construction to non-square matrices, but the idea sketched out above remains the same.

3.4 Future Work

The initial goal of the investigation in the thesis was to gain a better understanding of how Lemke-Howson performs in the “average” case. Indeed, even though there are worst-case examples as given by Savani, where every LH path is exponentially long, it is possible that “perturbing" the game slightly would break the example, and create LH paths which are short (poly length). Indeed, ever since Spielman applied “smoothed analysis” to prove results on average-case complexity of the Simplex Algorithm, there have been efforts to translate the techniques to LH. However, this has proved to be a difficult task since the inequalities for polytopes are not linear and the pivoting rule is based on picking a neighbor rather than maximizing an objective function, as in the simplex.
With a better understanding of dual-cyclic polytopes, it might be possible to check the robustness of Savani’s examples. This is the immediate goal. Analyzing this case would illuminate ways to approach other “worst-case” examples for LH, perhaps eventually paving a path for average-case complexity analysis.

Bibliography

[1] X. Chen and X. Deng, *Settling the complexity of two-player nash equilibrium*, in Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on, IEEE, 2006, pp. 261–272.

[2] X. Chen, X. Deng, and S.-H. Teng, *Computing nash equilibria: Approximation and smoothed complexity*, in Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on, IEEE, 2006, pp. 603–612.

[3] C. Daskalakis, P. W. Goldberg, and C. H. Papadimitriou, *The complexity of computing a nash equilibrium*, SIAM Journal on Computing, 39 (2009), pp. 195–259.

[4] D. Gale, *Neighborly and cyclic polytopes*, in Proc. Sympos. Pure Math, vol. 7, 1963, pp. 225–232.

[5] P. J.-J. Herings and R. Peeters, *Homotopy methods to compute equilibria in game theory*, Economic Theory, 42 (2010), pp. 119–156.

[6] A. X. Jiang and K. Leyton-Brown, *A tutorial on the proof of the existence of nash equilibria*, University of British Columbia Technical Report TR-2007-25. pdf, (2009).

[7] C. E. Lemke and J. T. Howson, Jr, *Equilibrium points of bimatrix games*, Journal of the Society for Industrial and Applied Mathematics, 12 (1964), pp. 413–423.

[8] W. D. Morris Jr, *Lemke paths on simple polytopes*, Mathematics of Operations Research, 19 (1994), pp. 780–789.

[9] J. F. Nash et al., *Equilibrium points in n-person games*, Proceedings of the national academy of sciences, 36 (1950), pp. 48–49.

[10] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, *Algorithmic game theory*, vol. 1, Cambridge University Press Cambridge, 2007.

[11] R. Savani and B. Stengel, *Hard-to-solve bimatrix games*, Econometrica, 74 (2006), pp. 397–429.

[12] R. SAVANI AND B. VON STENGEL, *Exponentially many steps for finding a nash equilibrium in a bimatrix game*, in Foundations of Computer Science, 2004. Proceedings. 45th Annual IEEE Symposium on, IEEE, 2004, pp. 258–267.

[13] L. S. SHAPLEY, *A note on the lemke-howson algorithm*, Pivoting and Extension, (1974), pp. 175–189.