Imitation Games and Computation

Andrew McLennan and Rabee Tourky, School of Economics Discussion Paper No. 359, March 2008, School of Economics, The University of Queensland. Australia.

Full text available as:
PDF- Requires Adobe Acrobat Reader or other PDF viewer

Abstract

An imitation game is a finite two person normal form game in which the two players have the same set of pure strategies and the goal of the second player is to choose the same pure strategy as the first player. Gale et al. (1950) gave a way of passing from a given two person game to a symmetric game whose symmetric Nash equilibria are in oneto-one correspondence with the Nash equilibria of the given game. We give a way of passing from a given symmetric two person game to an imitation game whose Nash equilibria are in one-to-one correspondence with the symmetric Nash equilibria of the given symmetric game. Lemke (1965) portrayed the Lemke-Howson algorithm as a special case of the Lemke paths algorithm. Using imitation games, we show how Lemke paths may be obtained by projecting Lemke-Howson paths.

EPrint Type:	Departmental Technical Report
Keywords:	Computational economics, Symmetric games, Nash equilibrium, Computational complexity, Two person games, 2-Nash, PPAD, Imitation games, Lemke-Howson algorithm, Lemke paths algorithm.
Subjects:	340000 Economics;
ID Code:	JEL Classification D01, I18, P36
Deposited By:	

Andrew McLennan
School of Economics, University of Queensland, Level 6, Colin Clark Building, St Lucia, QLD
4072 Australia.

Rabee Tourky
School of Economics, University of Queensland, Level 6, Colin Clark Building, St Lucia, QLD
4072 Australia.

Imitation Games and Computation

Andrew McLennan[*] and Rabee Tourky[†]

December 31, 2007

Abstract: An imitation game is a finite two person normal form game in which the two players have the same set of pure strategies and the goal of the second player is to choose the same pure strategy as the first player. Gale et al. (1950) gave a way of passing from a given two person game to a symmetric game whose symmetric Nash equilibria are in one-to-one correspondence with the Nash equilibria of the given game. We give a way of passing from a given symmetric two person game to an imitation game whose Nash equilibria are in one-to-one correspondence with the symmetric Nash equilibria of the given symmetric game. Lemke (1965) portrayed the Lemke-Howson algorithm as a special case of the Lemke paths algorithm. Using imitation games, we show how Lemke paths may be obtained by projecting Lemke-Howson paths.

Keywords: Computational economics, Symmetric games, Nash equilibrium, Computational complexity, Two person games, 2-Nash, PPAD, Imitation games, Lemke-Howson algorithm, Lemke paths algorithm.

1 Introduction

An *imitation game* is a finite two person normal form game in which the two players have the "same" sets of pure strategies. That is, they have the same number of pure strategies, and there is a given bijection between the two sets of pure strategies that identifies each pure strategy of the first player with a particular pure strategy of the second player. The first player is called the *mover*; she can have any payoff matrix. The second player, who is called the *imitator*, receives a payoff of 1 if she

[*]School of Economics, University of Queensland, Level 6, Colin Clark Building, St Lucia, QLD 4072 Australia.

[†]School of Economics, University of Queensland, Level 6, Colin Clark Building, St Lucia, QLD 4072 Australia.

plays "the same" pure strategy as the mover, and otherwise her payoff is 0, so her payoff matrix is the identity matrix. Imitation games are evidently a special, and seemingly rather simple, type of two person game.

It turns out that from the point of view of computation, imitation games, and their Nash equilibria, are as complex as general two person games and their Nash equilibria. This paper shows how, in two different settings, observations along these lines complete a circle of ideas, with the result that phenomena that had for many years seemed to be distinct are actually superficially different manifestations of a single mathematical structure. In the remainder of the introduction we give very brief and informal descriptions of our findings.

A general two person game is given by a pair (A, B) of $m \times n$ payoff matrices. Gale et al. (1950) showed that if all the entries of A and B are positive and

$$C := \begin{bmatrix} 0 & A \\ B^T & 0 \end{bmatrix},$$

then the symmetric equilibria of the two person game (C, C^T) are in one-to-one correspondence with the Nash equilibria of the given game. The Nash equilibria of (A, B) are unaffected by the addition of a constant to all the entries of A and B, so this gives a sense (described precisely in Section 3) in which any computational problem related to the Nash equilibria of two person games is "easier" than the corresponding problem for symmetric equilibria of symmetric games: given an algorithm $\mathcal{A}$ for the corresponding problem, there is an algorithm for the given problem consisting of adding a sufficiently large constant to all entries of A and B, forming C as above, and then applying $\mathcal{A}$ to (C, C^T).

However, as we will see in more detail shortly, for any square matrix C the symmetric equilibria of (C, C^T) are in natural one-to-one correspondence with the Nash equilibria of the imitation game (C, I). Given an algorithm $\mathcal{B}$ for the given problem related to Nash equilibrium, there is an algorithm for the corresponding problem related to symmetric equilibria of symmetric games consisting of applying $\mathcal{B}$ to (C, I).

Lemke and Howson (1964) gave a pivoting algorithm for computing a Nash equilibrium of (A, B) that came to be known as the *Lemke-Howson algorithm.* Generalizing slightly, it will be helpful to think of the input as consisting of a pair (A, B) together with pair (p, q) whose components are an m-vector and an n-vector. Subsequently Lemke (1965) described a family of computational procedures that take as input a pair (C, r) consisting of an $p \times p$ matrix C and an p-vector r. This procedure, which came to be known as the *Lemke paths algorithm*, includes the Lemke-Howson algorithm as the special case that arises when $p = m + n$ and C is derived from (A, B) as above. For each integer $k \geq 1$ let $\mathbf{e}^k := (1, \ldots, 1) \in \mathbb{R}^k$. Our contribution here is to show that the Lemke paths algorithm can by obtained by applying the Lemke-Howson algorithm to (C, I) and $(r, \mathbf{e}^p) \in \mathbb{R}^{2p}$.

This observation has an interesting application to the theory of the worst case running times of these algorithms. Both for Lemke-Howson and Lemke paths, there are multiple possible starting points, so that in order to show an exponential worst case running time it is necessary to find a sequence of examples in which the minimum—over all possible starting points—running time grows exponentially. Morris (1994) gave a sequence of pairs $(C_p, \mathbf{e}^p)$ of this sort for the Lemke paths algorithm. Elaborating on his methods, Savani and von Stengel (2006) construct a sequence of examples of games for which the worst case running time of the Lemke-Howson algorithm grows exponentially. As they mention (citing an earlier version of this work) one can also obtain such a sequence directly from Morris' examples, by applying the Lemke-Howson algorithm to the two person game (C_p, I).

2 Symmetric Games and Linear Complementarity Problems

For each integer $k \geq 1$ let Δ^k be the standard unit simplex in $\mathbb{R}^k$, i.e., the set of vectors whose components are nonnegative and sum to one. A *Nash equilibrium* of the game (A, B) is a pair $(\sigma, \tau) \in \Delta^m \times \Delta^n$ such that $\sigma^T A \tau \geq \tilde{\sigma}^T A \tau$ for all $\tilde{\sigma} \in \Delta^m$ and $\sigma^T B \tau \geq \sigma^T B \tilde{\tau}$ for all $\tilde{\tau} \in \Delta^n$. A *symmetric game* is a game (C, C^T) where C is a square matrix, and $\rho \in \mathbb{R}^m$ is a *symmetric equilibrium* of (C, C^T) if (ρ, ρ) is a Nash equilibrium.

Proposition 2.1 (Gale et al. (1950)). *Suppose A and B are $m \times n$ matrices whose entries are all positive, and let*

$$C := \begin{bmatrix} 0 & A \\ B^T & 0 \end{bmatrix}.$$

For $\rho \in \Delta^{m+n}$ the following are equivalent:

(a) ρ *is a symmetric equilibrium of* (C, C^T)*;*

(b) there are $\sigma \in \Delta^m$*,* $\tau \in \Delta^n$*, and* $0 < \alpha < 1$ *such that:*

(i) $\rho = ((1-\alpha)\sigma, \alpha\tau)$*,*

(ii) (σ, τ) *is a Nash equilibrium of* (A, B)*, and*

(iii) $(1-\alpha)\sigma^T A\tau = \alpha\sigma^T B\tau$*.*

Proof. First suppose that ρ is a symmetric equilibrium of (C, C^T). We have $\rho = ((1-\alpha)\sigma, \alpha\tau)$ for some $\sigma \in \Delta^m$, $\tau \in \Delta^n$, and $0 \leq \alpha \leq 1$. In view of the identity $(\sigma, 0)^T C(\sigma, 0) = 0 = (0, \tau)^T C(0, \tau)$ and the fact that entries of B and C are all positive, it cannot be the case that $\alpha = 0$ or $\alpha = 1$. Because $\alpha < 1$, in the game (B, C) the strategy σ is a best response for agent 1 to τ, and similarly τ is a best response for agent 2 to σ. In addition, $(\sigma, 0)$ and $(0, \tau)$ are both best responses to ρ, so $(1-\alpha)\sigma^T B\tau = \alpha\sigma^T C\tau$.

Now suppose that (b) holds. It is easily verified that $(\sigma, 0)$ and $(0, \tau)$ are best responses to $\rho := ((1-\alpha)\sigma, \alpha\tau)$ in (C, C^T), so any convex combination of $(\sigma, 0)$ and $(0, \tau)$, such as ρ, is also a best response to ρ. ■

An *imitation game* is a game (C, I) in which C is a square matrix and I is the identity matrix. Whenever g and h are integers with $g \leq h$ we let $[g, h] := \{g, \ldots, h\}$. For any integer $k \geq 1$ the *support* of $\mu \in \Delta^k$ is

$$\operatorname{supp} \mu := \{ i \in [1, k] : \mu_i > 0 \}.$$

Let p be the number of rows and columns of C. An I*-equilibrium* of an imitation game (C, I) is a mixed strategy $\rho \in \Delta^p$ for the imitator such that

$$\operatorname{supp} \rho \subset \operatorname*{argmax}_{i \in \mathcal{I}} (C\rho)_i.$$

Proposition 2.2. *A mixed strategy* $\rho \in \Delta^p$ *is an* I*-equilibrium of* (C, I) *if and only if there is an* $\iota \in \Delta^p$ *such that* (ι, ρ) *is a Nash equilibrium of* (C, I)*.*

Proof. If (ι, ρ) is a Nash equilibrium of (C, I), then the support of ρ is contained in the support of ι, because ρ is a best response to ι for the imitator, and the support of ι is contained $\operatorname{argmax}_{i \in \mathcal{I}} (C\rho)_i$, because ι is a best response to ρ for the mover. Thus, the support of ρ is contained in $\operatorname{argmax}_{i \in \mathcal{I}} (C\rho)_i$.

Now suppose ρ is an I-equilibrium of (C, I). Because the set of best responses to ρ contains the support of ρ, we may choose an $\iota \in \Delta^p$ that assigns all probability to best responses to ρ (so ι is a best response to ρ) and maximal probability to elements of the support of ρ (so ρ is a best response to ι). ■

In addition to symmetric equilibria of symmetric games, I-equilibria of imitation games, and Nash equilibria of imitation games, a fourth formulation of the problem will be important in what follows. A *linear complementarity problem* (LCP) is a problem of the form

$$w + Cz \leq r, \; z \geq 0, \; w \geq 0, \; \langle z, w \rangle = 0$$

where the $p \times p$ matrix C and the vector $r \in \mathbb{R}^p$ are given and vectors $z, w \in \mathbb{R}^p$ are sought. The LCP is said to be *monotone* if all the entries of C are positive. The extensive literature on the linear complementarity problem is surveyed in Murty (1988) and Cottle et al. (1992).

Proposition 2.3. *For an $p \times p$ matrix C with positive entries and $\rho \in \Delta^p$ the following are equivalent:*

(a) ρ is a symmetric equilibrium of (C, C^T);

(b) ρ is an I-equilibrium of (C, I);

(c) there is $\iota \in \Delta^m$ such that (ι, ρ) is a Nash equilibrium of (C, I);

(d) $\rho := z / \sum_{i=1}^p z_i$ where (w, z) with $z \neq 0$ is a solution of the LCP

$$w + Cz \leq \mathbf{e}^p, \; z \geq 0, \; w \geq 0, \; \langle z, w \rangle = 0.$$

Proof. The equivalence of (a) and (b) is immediate. The equivalence of (b) and (c) is Proposition 2.2. To complete the proof we show that (b) and (d) are equivalent. If (w, z) is a solution of the LCP with $z \neq 0$, then $\rho := z / \sum_{i=1}^m z_i$ is an I-equilibrium of the imitation game (C, I), because the complementarity condition $\langle z, w \rangle = 0$ means precisely that each pure strategy for the first agent is either unused (that is, $\rho_i = 0$) or gives the maximal expected payoff. Conversely, if ρ is an I-equilibrium, we can obtain a solution of the LCP by setting $z := \alpha \rho$ where $\alpha > 0$ is chosen to make the smallest component of $w := \mathbf{e}^p - Cz$ zero. ■

The route by which two person games have traditionally been understood to give rise to linear complementarity problems is as follows. Adding a constant to all entries of either of the matrices A and B does not change the set of Nash equilibria of a game (A, B), and in this sense requiring all of their entries to be positive is without loss of generality. Let A and B be $m \times n$ matrices with positive entries, let $p := m + n$, and for any natural number k let $\mathbf{e}^k := (1, \ldots, 1) \in \mathbb{R}^k$. Consider the LCP

$$\begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} 0 & A \\ B^T & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \mathbf{e}^m \\ \mathbf{e}^n \end{bmatrix}, \quad u, v, x, y \geq 0, \quad u \cdot x = 0, \quad v \cdot y = 0.$$

There is a "trivial" solution $(u, v, x, y) = (\mathbf{e}^m, \mathbf{e}^n, 0, 0)$. Any other solution necessarily has both $x \neq 0$ and $y \neq 0$, and $\sigma := x / \sum x_i$ and $\tau := y / \sum y_j$ constitute a Nash equilibrium. Conversely, if (σ, τ) is a Nash equilibrium, and α and β are the smallest positive numbers such that all components of $A(\beta\tau)$ and $B^T(\alpha\sigma)$ are not greater than 1, then $u := \mathbf{e}^m - Ay$, $v := \mathbf{e}^n - B^T x$, $x := \alpha\sigma$, and $y := \beta\tau$ constitute a solution to the LCP above.

As Proposition 2.3 explains, the notion of an I-equilibrium of an imitation game provides a distinct route by which games lead to LCP's.

3 Computation

In this section we recall certain basic concepts of computer science and relate them to the results above. A *computational problem* is a correspondence from a space of allowed inputs to a space of allowed outputs, both of which are subsets of the space of finite bit strings. An algorithm for such a problem is a Turing machine that converts any element of the space of inputs into one of its allowed outputs. An algorithm is *polynomial time* if its running time is bounded above by a polynomial function of the size of the input. A computational problem is *polynomial* if it has a polynomial time algorithm.

A computational problem is a *decision problem* if its space of outputs is $\{0, 1\} = \{\text{NO}, \text{YES}\}$ and for each input there is only one allowed output. The class of polynomial decision problems is denoted by **P**. A decision problem is in **NP** if, whenever the desired output is YES, there is a "witness" and a polynomial time procedure taking the problem and the witness as inputs that verifies that the desired output is YES. For example, the problem of determining whether a graph has a k-clique is in **NP** because any k-clique is a suitable witness. Clearly $\mathbf{P} \subset \mathbf{NP}$, and it is thought to be extremely likely that the containment is strict, but whether this is actually the case is one of the most prominent open problems in mathematics.

Given two computational problems P and Q, a *reduction* from Q to P is a pair of maps, one of which takes an input x for Q to an input $r(x)$ for P, and the other of which takes an output y of P to an output $s(y)$ of Q, such that s transforms any of the desired outputs of P for $r(x)$ to one of the desired outputs of Q for x. The reduction is a *polynomial time reduction*[1] if the size of the output of P for $r(x)$ is bounded by a polynomial function of the size of x and there are polynomial time algorithms that compute the values of r and s.

Suppose $\mathbf{C}$ is a class of computational problems and $P \in \mathbf{C}$. We say that P is $\mathbf{C}$-*complete* if every problem in $\mathbf{C}$ has a polynomial time reduction to P. Gilboa and Zemel (1989) show that many decision problems related to Nash equilibria of two person games (Is there more than one Nash equilibrium?, Is there a Nash equilibrium assigning positive probability to a certain pure strategy?, etc.) are **NP**-complete. Recently several new proofs (Conitzer and Sandholm (2003); Blum and Toth (2004); Codenotti and Štefanovič (2005)) of these results have been given, including one (McLennan and Tourky (2005)) using imitation games.

The computational problem of finding a Nash equilibrium of a finite two person normal form game (A, B) is called *2-Nash*. Papadimitriou (2001) has described the problem of determining whether 2-Nash has a polynomial time algorithm as (along with factoring) "the most important concrete open question on the boundary of **P**." The most important reason for this is the fact that 2-Nash is a fixed point problem that is seemingly a small step beyond linear programming, which is in **P**.

Hirsch et al. (1989) studied a discrete version of Brouwer's fixed point theorem that is based on function evaluation. Specifically, one is given an "oracle" or "black box" that computes the value of a function $f : [0, 1]^n \to [0, 1]^n$, and the goal is to find a point x satisfying $\|f(x)-x\| \leq 2^{-p}$. They show that if $n \geq 3$ and $x \mapsto f(x)-x$ is known to be Lipschitzian with Lipschitz constant L, then any algorithm must, in the worst case, evaluate the function at at least $(c(2^p - 10)L)^{n-2}$ points, where c is a positive constant. Specifically, they construct a collection of example functions such that, when np is sufficiently large, any smaller number of function evaluations

[1] Other conditions on the reduction can also be considered (Papadimitriou, 1994a, Section 8.1).

will necessarily fail to distinguish between two examples that have different fixed points.

The fixed point problems that arise in practice typically come from applications that endow them with additional structure. In order to systematically study the possibility that such structure might allow more efficient computation, Papadimitriou (1994b) introduced the computational class **PPAD**. The search space of a problem in **PPAD** is the set $\{0,1\}^r$ of bitstrings of length r for some natural number r. An element of **PPAD** has two inputs. The first is a Turing machine $\mathcal{M}$, whose worst case running time is bounded by a polynomial function of r, with the following properties. Given an input from the search space, $\mathcal{M}$ outputs a "predecessor" in $\{0,1\}^r$ (or an indication that the bitstring is a source) and a "successor" in $\{0,1\}^r$ (or an indication that the bitstring is a sink). The input cannot coincide with either the predecessor or the successor, it is the successor of its predecessor when it is not a source, and it is the predecessor of its successor when it is not a sink. Thus the Turing machine computes a directed graph whose vertices are the bitstrings and whose maximum in-degree and maximum out-degree are both one. The second input is an element of the search space that is known to be a source. The desired output is a bitstring that is either a sink or a different source.

Given a method of computing values of a continuous function from the n-dimensional unit simplex to itself, the Scarf algorithm follows such a path in a graph to an approximate fixed point. The algorithms described in subsequent sections are also based on following paths in graphs. Papadimitriou (1994b) showed how to display the problem solved by the Scarf algorithm as a member of **PPAD** if the function evaluations are performed by a Turing machine that is sufficiently fast. Iterative search can be used to solve a problem in **PPAD**, but there is also the possibility that for certain types of problems there may be algorithms that exploit additional structure that will be embedded in $\mathcal{M}$. The Lemke-Howson algorithm, which is based on combinatoric search rather than topology, is a concrete instance of this possibility.

Recently, culminating a rapid sequence of developments (Goldberg and Papadimitriou (2005); Daskalakis et al. (2005); Daskalakis and Papadimitriou (2005); Chen and Deng (2005a)) Chen and Deng (2005b) have given a polynomial time reduction from **PPAD** to 2-Nash, so 2-Nash is **PPAD**-complete. Especially in view of the result of Hirsch et al. (1989), it seems extremely unlikely that there is a polynomial time algorithm for **PPAD**, so this finding is regarded as compelling evidence that there is no polynomial time algorithm for 2-Nash, even if it does not quite amount to a complete proof.

Proposition 2.3 implies that there is a polynomial time reduction passing between any two of the following problems.

(i) 2-Nash.

(ii) Finding an I-equilibrium of an imitation game.

(iii) Finding a symmetric equilibrium of a symmetric game.

(iv) Finding a solution of an LCP $w + Cz = r$, $z \geq 0$, $w \geq 0$, $\langle z, w \rangle = 0$ with all entries of C and r positive.

This result complements the work described above: in addition to fully embodying the complexity of **PPAD**, to a greater extent than had been known previously 2-Nash turns out to be a unifying aspect of the computational theory of games and linear complementarity problems.

4 The Lemke Paths Algorithm

In the remainder we study specific algorithms that have been applied to 2-Nash and linear complementarity problems. In this section we study the Lemke paths algorithm in its application to the LCP

$$w + Cz = r,\ z \geq 0,\ w \geq 0,\ \langle z, w \rangle = 0,$$

where C is a $p \times p$ matrix and $r \in \mathbb{R}^p$. (Except in special settings described later, we impose no restrictions on the signs of the entries of C and r.) For $z \in \mathbb{R}^p$ let $w(z) := r - Cz$. Let

$$Z := \{\, z \in \mathbb{R}^p : z \geq 0 \text{ and } w(z) \geq 0 \,\}.$$

If Z is nonempty it is a convex polyhedron.

A *label* is a pair $\lambda = (\lambda_z, \lambda_w)$ in which λ_z and λ_w are subsets of $[1, p]$. For $z \in Z$ let $\ell(z) = (\ell_z(z), \ell_w(z))$ where

$$\ell_z(z) := \{\, i = 1, \ldots, p : z_i > 0 \,\} \quad \text{and} \quad \ell_w(z) := \{\, j = 1, \ldots, p : w_j(z) > 0 \,\}.$$

For a label λ let

$$Z_\lambda := \{ z \in Z : \ell(z) = \lambda \}.$$

We say that λ is *feasible* if Z_λ is nonempty. If λ is feasible, then Z_λ is the relative interior of a convex polyhedron.

We will only consider nondegenerate problems. Let $|\lambda| := |\lambda_z| + |\lambda_w|$ where the summands are the cardinalities of λ_z and λ_w. The pair (C, r) is *nondegenerate* if $Z_\lambda = \emptyset$ whenever $|\lambda| < p$ and has the "expected" dimension $|\lambda| - p$ whenever it is nonempty. Various techniques are known for extending the Lemke paths and Lemke-Howson algorithms to degenerate problems. For the most part we expect the main point made here to extend to those techniques, but we will not consider the issue explicitly.

A *basis* is a label β such that $|\beta| = p$. If β is feasible, then nondegeneracy implies that Z_β is a singleton whose unique element will be denoted by z_β, and we will usually write w_β in place of $w(z_\beta)$. The basis is *complementary* if $\beta_z \cup \beta_w = [1, p]$ or, equivalently, $\beta_z \cap \beta_w = \emptyset$. When the LCP given by C and r is nondegenerate its solutions are precisely the pairs (z_β, w_β) associated with feasible complementary bases β.

An *edge label* is a label ε such that $|\varepsilon| = p+1$. If ε is feasible, then Z_ε is an open line segment or an open ray (Z does not contain any lines because is contained in the positive orthant of $\mathbb{R}^p$) and its endpoints are points of the form z_β where β is a feasible basis obtained by dropping one element of ε_z or ε_w.

Fix an $\ell = 1, \dots, p$. A basis β is *ℓ-almost complementary* if $\beta_z \cap \beta_w \subset \{\ell\}$. Note that if β is complementary, it is ℓ-almost complementary for any ℓ. When β is ℓ-almost complementary but not complementary there is a *missing label* h such that $\beta_z \cup \beta_w = [1, p] \setminus \{h\}$. An edge label ε is *ℓ-almost complementary* if $\varepsilon_z \cap \varepsilon_w = \{\ell\}$, in which case $\varepsilon_z \cup \varepsilon_w = [1, p]$ because $|\varepsilon| = p + 1$. Let Z^ℓ be the union of the sets Z_ε for all ℓ-almost complementary edge labels ε and the sets Z_β for all ℓ-almost complementary bases β. The Lemke paths algorithm is derived from the geometric properties of Z^ℓ.

If ε is an ℓ-almost complementary edge label and β is obtained from ε by dropping one element of ε_z or ε_w, then β is ℓ-almost complementary. In particular, β is feasible and ℓ-almost complementary whenever ε is a feasible ℓ-almost complementary edge label and z_β is an endpoint of Z_ε.

Suppose that β is a feasible basis and ε is obtained from β by adding one element, say i^*, to β_z. Then ε is feasible. In detail, the solution set of the the system of $p-1$ equations that require that $z_i = 0$ if $i \notin \varepsilon_z$ and $w_j = 0$ if $j \notin \varepsilon_w$ consists of z_β and two rays, on one of which z_{i^*} is positive, and at points on this ray near z_β the variables z_i for $i \in \varepsilon_z$ and w_j for $j \in \varepsilon_w$ are all positive. A similar argument shows that ε is feasible when it is obtained from β by adding one element to β_w.

If β is feasible and complementary, there is precisely one way to obtain an ℓ-complementary edge label by adding an element to either β_z or β_w, namely adding ℓ to whichever set does not already contain it. Therefore z_β is an endpoint of Z_ε for precisely one ℓ-almost complementary edge label ε. If β is feasible and ℓ-almost complementary, but not complementary, then there are precisely two ways to add an element to β_z or β_w to obtain an ℓ-almost complementary edge label, namely adding the missing label to either of these sets. The two edges obtained in this way are said to be *adjacent* in Z^ℓ.

The analysis above yields the following picture of Z^ℓ. It is a closed subset of $\mathbb{R}^p$ that is the union of finitely many points, open line segments, and open half lines. It is a one dimensional manifold with boundary that has finitely many connected components, each of which is homeomorphic to the circle, $[0,1]$, $[0,1)$, or $(0,1)$. The points z_β for complementary feasible bases β are the boundary points of this manifold.

The general idea of the Lemke paths algorithm is to begin at either an open half line in Z^ℓ or a point z_β associated with a complementary feasible basis. It follows the connected component of Z^ℓ containing the starting point, "pivoting" from one vertex to the next until it arrives at either a boundary point or a half line. The computation is considered successful if it terminates at some z_β, and termination at a half line is regarded as failure.

The phrase "Lemke paths algorithm" is a bit imprecise insofar as additional conditions are required before one has an algorithm in the sense of a well defined computational procedure that is guaranteed to halt in finite time. Specifically, there must be a way to find a half line in Z^ℓ or a complementary feasible basis β. The connected component of Z^ℓ containing the starting point cannot be homeomorphic to a circle, so the pivoting procedure beginning there cannot cycle and must eventually terminate at z_β for some complementary feasible β or at a half line.

As is explained in Lemke (1965) and other sources, a variety of assumptions on the given data allow one to find starting points of the process for which success is guaranteed. Consider, for example, the possibility that all the entries of C are negative. Then $\varepsilon = (\{\ell\}, [1,p])$ is an ℓ-almost complementary edge label for which Z_ε is a ray. Moreover, this is the only ℓ-almost complementary edge label that can be associated with a ray: if $\{\, z_0 + \alpha z : \alpha > 0 \,\}$ is Z_ε for some ℓ-almost complementary edge label ε, then $z \geq 0$ and $z \neq 0$, so $\varepsilon_w = [1,p]$ because every component of Cz is negative. Thus this case provides both easily located starting points and a guarantee of success. Another such case is when all the entries of C and r are positive: $([0,p], \emptyset)$ is feasible and complementary, hence a starting point of a path in each Z^ℓ. Since Z is bounded, the algorithm cannot terminate at a half line.

The numerical implementation of the pivoting procedure of the Lemke paths algorithm, and related algorithms below, is usually described in terms of updating a "tableau," and may be thought of as a matter of using the sorts of pivots involved in Gaussian elimination to update a coordinate system that, at each step, has as its coordinates the variables z_i for $i \in \beta_z$ and w_j for $j \in \beta_w$ for some ℓ-almost complementary β, translated to place the origin at z_β. Detailed descriptions of this numerical procedure can be found in many sources, but our analysis will not refer to them.

5 Lemke Howson from Lemke Paths

In order to display the Lemke-Howson algorithm as a special case of the Lemke paths algorithm we now specialize to the case of an LCP in which the matrix has the block structure that occurs in the application to two person games. Let $p = m + n$ where m and n are both positive integers, and let A and B be $m \times n$ matrices. We study the LCP

$$\begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} 0 & A \\ B^T & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} p \\ q \end{bmatrix}, \quad u, v, x, y \geq 0, \quad u \cdot x = 0, \quad v \cdot y = 0.$$

For $x \in \mathbb{R}^m$ and $y \in \mathbb{R}^n$ let

$$v(x) := q - B^T x \in \mathbb{R}^n \quad \text{and} \quad u(y) := p - Ay \in \mathbb{R}^m.$$

Let

$$Z := \{\, (x,y) \in \mathbb{R}^m \times \mathbb{R}^n : x \geq 0,\ y \geq 0,\ u(y) \geq 0, \text{ and } v(x) \geq 0 \,\}.$$

Then $Z := X \times Y$ where

$$X := \{\, x \in \mathbb{R}^m : x \geq 0 \text{ and } v(x) \geq 0 \,\} \text{ and } Y := \{\, y \in \mathbb{R}^n : y \geq 0 \text{ and } u(y) \geq 0 \,\}.$$

Under the identifications $r = (p, q)$, $z = (x, y)$, and $w = (u, v)$ the components of p, x, and u have indices in $[1, m]$, and the components of s, y, and v have indices in $[m+1, m+n]$. A label is now

$$\lambda = (\lambda_z, \lambda_w) = \big((\lambda_x, \lambda_y), (\lambda_u, \lambda_v)\big)$$

where $\lambda_x, \lambda_u \subset [1, m]$ and $\lambda_y, \lambda_v \subset [m+1, m+n]$. For $x \in X$ let

$$\ell_x(x) := \{\, i \in [1, m] : x_i > 0 \,\} \text{ and } \ell_v(x) := \{\, j \in [m+1, m+n] : v_j(x) > 0 \,\},$$

and for $y \in Y$ let

$$\ell_y(y) := \{\, j \in [m+1, m+n] : y_j > 0 \,\} \text{ and } \ell_u(y) := \{\, i \in [1, m] : u_i(y) > 0 \,\}.$$

We now have $\ell_z(x, y) = \ell_x(x) \cup \ell_y(y)$ and $\ell_w(x, y) = \ell_u(y) \cup \ell_v(x)$. For a label λ let

$$X_\lambda := \{\, x \in X : \ell_x(x) = \lambda_x \text{ and } \ell_v(x) = \lambda_v \,\}$$

and

$$Y_\lambda := \{\, y \in Y : \ell_y(y) = \lambda_y \text{ and } \ell_u(y) = \lambda_u \,\}.$$

Clearly $Z_\lambda = X_\lambda \times Y_\lambda$.

If λ is feasible, then linear algebra implies that the dimension of X_λ is at least $|\lambda_x| + |\lambda_v| - n$ and the dimension of Y_λ is at least $|\lambda_y| + |\lambda_u| - m$, so nondegeneracy implies that the dimension of X_λ is exactly $|\lambda_x| + |\lambda_v| - n$ and the dimension of Y_λ is exactly $|\lambda_y| + |\lambda_u| - m$. As before, a basis is a label β with

$$|\beta_x| + |\beta_y| + |\beta_u| + |\beta_v| = m + n.$$

If β is feasible, then $|\beta_x| + |\beta_v| = n$ and $|\beta_y| + |\beta_u| = m$. An edge label ε is an X*-edge label* if $|\varepsilon_x| + |\varepsilon_v| = n + 1$ and $|\varepsilon_y| + |\varepsilon_u| = m$, and it is a Y*-edge label* if $|\varepsilon_x| + |\varepsilon_v| = n$ and $|\varepsilon_y| + |\varepsilon_u| = m + 1$. When ε is feasible one of these two cases holds, and we say that Z_ε is either an X-edge or a Y-edge as the case may be.

Suppose that β is a feasible basis that is ℓ-almost complementary but not complementary, and let h be the missing label of β. If $1 \le h \le m$ then the ℓ-almost complementary edge label obtained by adding h to β_x is an X-edge label and the ℓ-almost complementary edge label obtained by adding h to β_u is a Y-edge label. If $m+1 \le h \le m+n$ then the ℓ-almost complementary edge label obtained by adding h to β_v is an X-edge label and the ℓ-almost complementary edge label obtained by adding h to β_y is a Y-edge label. In either case, of the two edges in Z^ℓ that have z_β as an endpoint, one is an X-edge and the other is a Y-edge, so that we may think of any path in Z^ℓ as an alternation between moves in X and moves in Y. But without further assumptions the projections of a path in Z^ℓ onto X and Y need not be one dimensional manifolds.

The derivation of the Lemke-Howson algorithm per se is as follows. Suppose that all the entries of A and B are positive, and that $p = \mathbf{e}^m$ and $q = \mathbf{e}^n$. Then $z = 0 \in \mathbb{R}^p$ is complementary, but does not correspond to a Nash equilibrium. For any ℓ there is a path in Z^ℓ starting at 0 that leads to z_{β_ℓ} for some feasible complementary basis β_ℓ. Necessarily $z_{\beta_\ell} \neq 0$, and complementarity, together with the block structure of C, imply that $x_{\beta_\ell} \neq 0$ and $y_{\beta_\ell} \neq 0$, so that rescaling these vectors by dividing by the sum of components gives a Nash equilibrium. If more than one equilibrium is sought then, in addition to finding each z_{β_ℓ}, one can follow the paths in $Z^{\ell'}$ beginning at z_{β_ℓ} when $\ell' \neq \ell$, then continue in this fashion recursively at any new equilibria that are found. Wilson gave an example (reported in Shapley (1974)) of a game with equilibria that are *inaccessible* in the sense that this procedure does not find them.

6 From Lemke-Howson to Lemke Paths

For the most part the material in the last two section has been well known since Lemke (1965). We now present this paper's contribution to the understanding of these algorithms, which is a derivation of the Lemke paths, in a special case of the framework above, as projections of the Lemke-Howson paths onto Y.

Continuing with the setup of the last section, we now specialize further, assuming that $n = m$, that $B = I$ is the $m \times m$ identity matrix, and that $q = \mathbf{e}^m$. We now have $X = [0,1]^m$. Let $\rho : [m+1, 2m] \to [1, m]$ be the map $\rho(j) := j - m$. For any label λ, $X_\lambda \neq \emptyset$ if and only if

$$\lambda_x \cup \rho(\lambda_v) = [1, m]. \tag{$*$}$$

A Y-*label* is a pair $\mu = (\mu_y, \mu_u)$ where $\mu_y \subset [m+1, 2m]$ and $\mu_u \subset [1, m]$. For such a label let

$$Y_\mu := \{\, y \in Y : \ell_y(y) = \mu_y \text{ and } \ell_u(y) = \mu_u \,\}.$$

If Y_μ is nonempty we way that μ is *feasible.* For a label λ let $\pi(\lambda) := (\lambda_y, \lambda_u)$ be the derived Y-label. Then $Y_{\pi(\lambda)} = Y_\lambda$, and $|\beta_x| + |\beta_v| = m = |\beta_y| + |\beta_u|$ when β is a feasible basis, so:

Lemma 6.1. *A label λ is feasible if and only if $\pi(\lambda)$ is feasible and $(*)$ holds. In particular, if $|\lambda_x| + |\lambda_v| = m$ (as is the case when β is a feasible basis) then*

$$\rho(\lambda_v) = [1, m] \setminus \lambda_x. \tag{$**$}$$

A Y-*basis* is a Y-label α with $|\alpha_y| + |\alpha_u| = m$. A Y-basis α is Y-*complementary* if $\rho(\alpha_y) \cup \alpha_u = [1, m]$. If β is a feasible basis, then $\pi(\beta)$ is feasible and $\rho(\beta_v) = [1, m] \setminus \beta_x$, and if β is also complementary, then $\beta_u = [1, m] \setminus \beta_x$ and $\rho(\beta_v) = [1, m] \setminus \rho(\beta_y)$, so that $\beta_u = [1, m] \setminus \rho(\beta_y)$, i.e., $\pi(\beta)$ is complementary. Thus:

Lemma 6.2. *If a basis β is feasible and complementary, then $\pi(\beta)$ is feasible and complementary.*

Conversely:

Lemma 6.3. *If α is a complementary Y-basis, then*

$$\beta := \big((\rho(\alpha_y), \alpha_y), (\alpha_u, \rho^{-1}(\alpha_u))\big)$$

is the unique complementary basis with $\pi(\beta) = \alpha$, and α is feasible, then so is β.

Proof. Suppose that β is a basis such that $\pi(\beta) = \alpha$, i.e., $\beta_y = \alpha_y$ and $\beta_u = \alpha_u$. If β is complementary, then $\beta_x = [1, m] \setminus \alpha_u$ and $\beta_v = [m+1, 2m] \setminus \alpha_y$, and since α is complementary it follows that $\beta_x = [1, m] \setminus \rho(\beta_v)$. That is, $(*)$ holds, so β is feasible is α is. ■

An *edge* Y-*label* (not to be confused with a Y-edge label) is a Y-label δ with $|\delta_y| + |\delta_u| = m + 1$. Fix an integer $\ell \in [1, m]$. An edge Y-label δ is ℓ-*almost complementary* if $\rho(\delta_y) \cap \delta_u = \{\ell\}$, in which case $\rho(\delta_y) \cup \delta_u = [1, m]$.

Lemma 6.4. *If a Y-edge label ε is feasible and ℓ-almost complementary or $(\ell + m)$-almost complementary, then the edge Y-label $\pi(\varepsilon)$ is ℓ-almost complementary. Conversely, if δ is a feasible ℓ-almost complementary edge Y-label, then there is exactly one feasible ℓ-almost complementary Y-edge label ε such that $\pi(\varepsilon) = \delta$ and exactly one feasible $(\ell + m)$-almost complementary Y-edge label ε' such that $\pi(\varepsilon') = \delta$.*

Proof. Let ε be a feasible Y-edge label. If ε is ℓ-almost complementary, then

$$\rho(\varepsilon_y) = \rho([m + 1, 2m] \setminus \varepsilon_v) = [1, m] \setminus \rho(\varepsilon_v) = \varepsilon_x = ([1, m] \setminus \varepsilon_u) \cup \{\ell\},$$

(the third equality is $(**)$) so $\pi(\varepsilon)$ is ℓ-almost complementary. If ε is $(\ell + m)$-almost complementary, then

$$\rho(\varepsilon_y) = \rho([m + 1, 2m] \setminus \varepsilon_v \cup \{\ell + m\}) = [1, m] \setminus \rho(\varepsilon_v) \cup \{\ell\}$$

$$= \varepsilon_x \cup \{\ell\} = ([1, m] \setminus \varepsilon_u) \cup \{\ell\},$$

so $\pi(\varepsilon)$ is ℓ-almost complementary.

Suppose that δ is a feasible ℓ-almost complementary edge Y-label and ε is a feasible Y-edge label with $\pi(\varepsilon) = \delta$. If ε ℓ-almost complementary, then $\varepsilon_v = [m + 1, 2m] \setminus \delta_y$, because ε is ℓ-almost complementary, and feasibility implies that $\varepsilon_x = [1, m] \setminus \rho(\varepsilon_v) = \rho(\delta_y)$. It is easy to see that $\big((\rho(\delta_y), \delta_y), (\delta_u, [m + 1, 2m] \setminus \delta_y)\big)$ is, in fact, ℓ-almost complementary. If ε is $(\ell + m)$-almost complementary, then $\varepsilon_x = [1, m] \setminus \delta_u$, because ε is $(\ell + m)$-almost complementary, and feasibility implies that $\varepsilon_v = [m + 1, 2m] \setminus \rho^{-1}(\varepsilon_x) = \rho^{-1}(\delta_u)$. Clearly $\big(([1, m] \setminus \delta_u, \delta_y), (\delta_u, \rho^{-1}(\delta_u))\big)$ is, in fact, ℓ-almost complementary. ■

A Y-basis α is *ℓ-almost complementary* if $\rho(\alpha_y) \cap \alpha_u \subset \{\ell\}$.

Lemma 6.5. *If a basis β is feasible and ℓ-almost complementary or $(\ell + m)$-almost complementary, then $\pi(\beta)$ is ℓ-almost complementary.*

Proof. Above we saw that $\pi(\beta)$ is complementary if β is complementary, so suppose otherwise, and let h be the missing label: $\beta_z \cup \beta_w = [1, 2m] \setminus \{h\}$. There are four cases according to whether β is ℓ-almost complementary or $(\ell + m)$-almost complementary and whether $1 \leq h \leq m$ or $m + 1 \leq h \leq 2m$.

First, suppose that β is ℓ-almost complementary. If $1 \le h \le m$, then

$$\rho(\beta_y) = \rho([m+1,2m] \setminus \beta_v) = [1,m] \setminus \rho(\beta_v) = \beta_x = ([1,m] \setminus \beta_u) \cup \{\ell\} \setminus \{h\}.$$

If $m+1 \le h \le 2m$, then

$$\rho(\beta_y) = \rho([m+1,2m] \setminus (\beta_v \cup \{h\})) = [1,m] \setminus (\rho(\beta_v) \cup \{h-m\})$$

$$= \beta_x \setminus \{h-m\} = ([1,m] \setminus \beta_u) \cup \{\ell\} \setminus \{h-m\}.$$

Now suppose β is $(\ell+m)$-almost complementary. If $1 \le h \le m$, then

$$\rho(\beta_y) = \rho([m+1,2m] \setminus \beta_v \cup \{\ell+m\}) = [1,m] \setminus \rho(\beta_v) \cup \{\ell\}$$

$$= \beta_x \cup \{\ell\} = ([1,m] \setminus \beta_u) \cup \{\ell\} \setminus \{h\},$$

and if $m+1 \le h \le 2m$, then

$$\rho(\beta_y) = \rho([m+1,2m] \setminus \beta_v \cup \{\ell+m\} \setminus \{h\}) = ([1,m] \setminus \rho(\beta_v)) \cup \{\ell\} \setminus \{h-m\})$$

$$= \beta_x \cup \{\ell\} \setminus \{h-m\} = ([1,m] \setminus \beta_u) \cup \{\ell\} \setminus \{h-m\}.$$

In all four cases $\pi(\beta)$ is ℓ-almost complementary. ■

We now analyze the preimages of a feasible ℓ-almost complementary Y-basis. There are two cases, according to whether α is complementary.

Lemma 6.6. *Let α be a complementary feasible Y-basis, and let*

$$\beta := \big((\rho(\alpha_y), \alpha_y), (\alpha_u, \rho^{-1}(\alpha_u))\big).$$

If $\ell \notin \alpha_u$, then there is no other feasible ℓ-almost complementary basis β' with $\pi(\beta') = \alpha$, and when $\ell \in \alpha_u$ there is exactly one such β', and z_β and $z_{\beta'}$ are the endpoints of a feasible ℓ-almost complementary X-edge. If $\ell+m \notin \alpha_y$, then there is no other feasible $(\ell+m)$-almost complementary basis β' with $\pi(\beta') = \alpha$, and when $\ell+m \in \alpha_u$ there is exactly one such β', in which case z_β and $z_{\beta'}$ are the endpoints of a feasible $(\ell+m)$-almost complementary X-edge.

Proof. Let β' be a feasible basis that is not complementary with $\pi(\beta') = \alpha$.

Suppose that β' is ℓ-almost complementary with $\beta'_z \cap \beta'_w = \{\ell\}$ and $\beta'_z \cup \beta'_w = [1, 2m] \setminus \{h\}$. Then $\ell \in \beta'_u = \alpha_u$, so $\ell + m \notin \alpha_y = \beta'_y$, and $\ell \in \beta'_x = [1, m] \setminus \rho(\beta'_v)$, so $\ell + m \notin \beta'_v$, which means that $h = \ell + m$. That is, when $\ell \in \alpha_u$ it must be the case that

$$\beta' = \big((\rho(\alpha_y) \cup \{\ell\}, \alpha_y), (\alpha_u, \rho^{-1}(\alpha_u \setminus \{\ell\}))\big).$$

Note that $\big((\rho(\alpha_y) \cup \{\ell\}, \alpha_y), (\alpha_u, \rho^{-1}(\alpha_u))\big)$ is a feasible ℓ-almost complementary X-edge label.

Now suppose that β' is $(\ell + m)$-almost complementary, with $\beta'_z \cap \beta'_w = \{\ell + m\}$ and $\beta'_z \cup \beta'_w = [1, 2m] \setminus \{h\}$ for some h. Then $\ell + m \in \alpha_y$, so $\ell \notin \alpha_u$, and $\ell + m \in \alpha_v$, so that $\ell \notin \beta'_x = [1, m] \setminus \rho(\beta'_v)$, which means that $h = \ell$. That is, when $\ell + m \in \alpha_y$ we have

$$\beta' = \big((\rho(\alpha_y) \setminus \{\ell\}, \alpha_y), (\alpha_u, \rho^{-1}(\alpha_u \cup \{\ell\}))\big).$$

In this case $\big((\rho(\alpha_y), \alpha_y), (\alpha_u, \rho^{-1}(\alpha_u \cup \{\ell\}))\big)$ is a feasible $(\ell + m)$-almost complementary X-edge label. ■

Lemma 6.7. *Suppose that α is a feasible Y-basis that is ℓ-almost complementary, but not complementary, with $\rho(\alpha_y) \cup \alpha_u = [1, m] \setminus \{h\}$. Then there are precisely two ℓ-almost complementary bases that project to α, say β and β', and z_β and $z_{\beta'}$ are the endpoints of a feasible ℓ-almost complementary X-edge. There are also precisely two $(\ell + m)$-almost complementary bases that project to α, say β and β', and z_β and $z_{\beta'}$ are the endpoints of a feasible $(\ell + m)$-almost complementary X-edge.*

Proof. Let β be a feasible basis with $\pi(\beta) = \alpha$. Since $\pi(\beta)$ is not complementary, β cannot be complementary.

First suppose that β is ℓ-almost complementary. Since $\emptyset = \beta_y \cap \beta_v = \alpha_y \cap \beta_v$, we have $\rho(\alpha_y) \cap ([1, m] \setminus \beta_x) = \emptyset$, i.e., $\rho(\alpha_y) \subset \beta_x$. In addition, $\alpha_u = \beta_u$, so $\beta_x \subset ([1, m] \setminus \alpha_u) \cup \{\ell\} = \rho(\alpha_y) \cup \{h\}$. Thus the only possibilities for β are

$$\big((\rho(\alpha_y), \alpha_y), (\alpha_u, [m + 1, 2m] \setminus \alpha_y)\big)$$

and

$$\big((\rho(\alpha_y) \cup \{h\}, \alpha_y), (\alpha_u, [m + 1, 2m] \setminus (\alpha_y \cup \{h + m\}))\big),$$

and in fact both are feasible and ℓ-almost complementary. Note that

$$\big((\rho(\alpha_y) \cup \{h\}, \alpha_y), (\alpha_u, [m + 1, 2m] \setminus \alpha_y)\big)$$

is a feasible ℓ-almost complementary X-edge label.

Now assume that β is $(\ell+m)$-almost complementary. We have $\emptyset = \alpha_u \cap \beta_x = \alpha_u \cap ([1,m] \setminus \rho(\beta_v))$, so $\rho^{-1}(\alpha_u) \subset \beta_v$. In addition, $\alpha_y = \rho^{-1}([1,m] \setminus \alpha_u \cup \{\ell\} \setminus \{h\})$ and $\beta_y = \alpha_y$, so

$$\beta_v \subset [m+1, 2m] \setminus \alpha_y \cup \{\ell + m\} = \rho^{-1}(\alpha_u) \cup \{h+m\}.$$

Therefore it is only possible that β is either

$$\big(([1,m] \setminus \alpha_u, \alpha_y), (\alpha_u, \rho^{-1}(\alpha_u))\big)$$

or

$$\big(([1,m] \setminus (\alpha_u \cup \{h\}), \alpha_y), (\alpha_u, \rho^{-1}(\alpha_u \cup \{h\}))\big),$$

and in fact both are feasible and $(\ell+m)$-almost complementary. Note that

$$\big(([1,m] \setminus \alpha_u, \alpha_y), (\alpha_u, \rho^{-1}(\alpha_u \cup \{h\}))\big)$$

is a feasible $(\ell+m)$-almost complementary X-edge label. $\blacksquare$

Let Y^ℓ be the union of the sets Y_δ for all ℓ-almost complementary edge Y-labels δ and the sets Y_α for all ℓ-almost complementary bases α. Of course these are just the concepts introduced in Section 4, now applied to the LCP

$$u + Ay = p, \ y \geq 0, \ u \geq 0, \ \langle y, u \rangle = 0,$$

with the added twist that the components of y are indexed by the elements of $[m+1, 2m]$. All the results from that section apply.

The results above combine to give a simple picture of the relationship between Y^ℓ, Z^ℓ, and $Z^{\ell+m}$. Let $\pi_Y : Z \to Y$ be the projection $\pi_Y(x,y) := y$, and for each $k = 1, \ldots, 2m$ let π_Y^k be the restriction of π_Y to Z^k. Then π_Y^ℓ maps Z^ℓ surjectively onto Y^ℓ, and $\pi_Y^{\ell+m}$ maps $Z^{\ell+m}$ surjectively onto Y^ℓ. Since X is bounded, any X-edge in Z^ℓ or $Z^{\ell+m}$ is bounded and has compact closure, and each such closed X-edge is mapped to a single point that has the closed X-edge as its preimage. Outside of these X-edges π_Y^ℓ and $\pi_Y^{\ell+m}$ are injective. Except for the compressions of X-edges, π_Y^ℓ and $\pi_Y^{\ell+m}$ are "topologically faithful": the topology of Y^ℓ is the quotient topology induced by each of these maps, and the preimage under π_Y^ℓ ($\pi_Y^{\ell+m}$)

of a connected component of Y^ℓ is a single connected component of Z^ℓ ($Z^{\ell+m}$) that is homeomorphic to it. In particular, any Lemke-Howson path in Z^ℓ or in $Z^{\ell+m}$ projects onto a Lemke path in Y^ℓ with half as many steps, after allowance for "off by one" adjustments at the endpoints, and any Lemke path in Y^ℓ has well defined "lifts" to Lemke-Howson paths in Z^ℓ and $Z^{\ell+m}$.

7 Concluding Remarks

We have shown that the solutions of several problems are in one-to-one correspondence. This implies that the computational problem of finding a solution of one of these problems is the same as the problem of finding a solution to any other. That is, these problems are all avatars of 2-Nash. We have also shown that the Lemke-Howson algorithm for 2-Nash and the Lemke paths algorithm for LCP's are essentially the same, thereby achieving a unified understanding of their exponential worst case complexity.

In linear programming worst case complexity presents a relatively coarse and somewhat misleading view of the subject: although the simplex algorithm has exponential worst case complexity, various investigations of its mean time complexity show it to be quite fast. The results described in Section 3 show that from the point of view of worst case complexity, any problem in **PPAD** is equivalent to 2-Nash, but it may still be the case that different problems have very different degrees of difficulty in practice. The Lemke-Howson algorithm routinely solves two person games with hundreds of pure strategies, and recently Porter et al. (2004) and Sun et al. (2006) have presented experimental evidence suggesting that simple search methods can result in faster mean performance. We hope that a deeper understanding of the complexity of 2-Nash will emerge from analysis of the mean time complexity of Lemke-Howson, and perhaps other algorithms, for natural and tractable distributions on the space of games.

Bibliography

A. Blum and B. Toth. NP-hardness involving Nash equilibria in a non-degenerate setting. unpublished, 2004.

Xi Chen and Xiaotie Deng. 3-NASH is PPAD-complete. *Electronic Colloquium on Computational Complexity (ECCC)*, (134), 2005a.

Xi Chen and Xiaotie Deng. Settling the complexity of 2-player Nash-equilibrium. *Electronic Colloquium on Computational Complexity (ECCC)*, (140), 2005b.

B. Codenotti and D. Štefanovič. On the computational complexity of Nash equilibria for $(0, 1)$ bimatrix games. forthcoming in *Information Processing Letters*, 2005.

V. Conitzer and T. Sandholm. Complexity results about Nash equilibria. *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pages 765–771, 2003.

R. W. Cottle, J. Pang, and R. E. Stone. *The linear complementarity problem*. Computer Science and Scientific Computing. Academic Press Inc., Boston, MA, 1992.

K. Daskalakis, P. W. Goldberg, and C. H. Papadimitriou. The complexity of computing a Nash equilibrium. *Electronic Colloquium on Computational Complexity (ECCC)*, (115), 2005.

K. Daskalakis and C. H. Papadimitriou. Three-player games are hard. *Electronic Colloquium on Computational Complexity (ECCC)*, (139), 2005.

D. Gale, H. Kuhn, and A. W. Tucker. On symmetric games. *Annals of Math. Studies*, 24:81–87, 1950.

I. Gilboa and E. Zemel. Nash and correlated equilibria: some complexity considerations. *Games and Economic Behavior*, 1(1):80–93, 1989.

P. W. Goldberg and C.H. Papadimitriou. Reducibility among equilibrium problems. *Electronic Colloquium on Computational Complexity (ECCC)*, (090), 2005.

M. Hirsch, C.H. Papadimitriou, and S. Vavasis. Exponential lower bounds for finding Brouwer fixed points. *Journal of Complexity*, 5:379–416, 1989.

C. E. Lemke. Bimatrix equilibrium points and mathematical programming. *Management Sci.*, 11:681–689, 1965.

C. E. Lemke and J. T. Howson. Equilibrium points of bimatrix games. *J. Soc. Indust. Appl. Math.*, 12:413–423, 1964.

A. McLennan and R. Tourky. Simple complexity from imitation games. forthcoming in *Games and Economic Behavior*, 2005.

W. D. Morris, Jr. Lemke paths on simple polytopes. *Math. Oper. Res.*, 19(4): 780–789, 1994.

K. G. Murty. *Linear complementarity, linear and nonlinear programming.* Heldermann, Berlin, 1988. Sigma serties in applied mathematics; 3.

C. H. Papadimitriou. *Computational Complexity.* Addison Wesley Longman, New York, 1994a.

C. H. Papadimitriou. Algorithms, games and the internet. In *Annual ACM Symposium on the Theory of Computing*, pages 749–253, 2001.

C.H. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *Journal of Computer and System Science*, 48:498–532, 1994b.

R. W. Porter, E. Nudelman, and Y. Shoham. Simple search methods for finding a Nash equilibrium. forthcoming in *Games and Economic Behavior*, 2004.

R. Savani and B. von Stengel. Hard-to-solve bimatrix games. *Econometrica*, 74: 397–429, 2006.

L. S. Shapley. A note on the Lemke-Howson algorithm. *Math. Programming Stud.*, 1:175–189, 1974. Pivoting and extensions.

S.-Y. Sun, D.-Y. Liu, and C.-H. Sun. Finding a simple Nash equilibrium. *International Journal of computer Science and Network Secutiry*, 6(2A):153–155, 2006.