From Imitation Games to Kakutani

Andrew McLennan
University of Minnesota

and

Rabee Tourky
University of Melbourne

January 2005
Introduction

- Let $C$ be a nonempty norm-compact convex subset of Hilbert space.
Introduction

- Let $C$ be a nonempty norm-compact convex subset of Hilbert space.
- Let $f : C \rightarrow C$ be a continuous function.

Fixed points of $f$ are in one-to-one correspondence with the Nash equilibria of this game.

We will work with restrictions of this game to finite subsets of $C$. 
Introduction

- Let $C$ be a nonempty norm-compact convex subset of Hilbert space.
- Let $f : C \rightarrow C$ be a continuous function.
- Define a two person game in which:
  - For each player the set of pure strategies is $C$.
  - Agent 1’s payoff is $u(s, t) := -\|s - f(t)\|^2$.
  - Agent 2’s payoff is $v(s, t) := \delta_{st}$.

Fixed points of $f$ are in one-to-one correspondence with the Nash equilibria of this game. We will work with restrictions of this game to finite subsets of $C$. 
Introduction

- Let $C$ be a nonempty norm-compact convex subset of Hilbert space.
- Let $f : C \to C$ be a continuous function.
- Define a two person game in which:
  - For each player the set of pure strategies is $C$.
  - Agent 1’s payoff is $u(s, t) := -\|s - f(t)\|^2$.
  - Agent 2’s payoff is $v(s, t) := \delta_{st}$.
- Fixed points of $f$ are in one-to-one correspondence with the Nash equilibria of this game.
Introduction

- Let $C$ be a nonempty norm-compact convex subset of Hilbert space.
- Let $f : C \rightarrow C$ be a continuous function.
- Define a two person game in which:
  - For each player the set of pure strategies is $C$.
  - Agent 1’s payoff is $u(s, t) := -\|s - f(t)\|^2$.
  - Agent 2’s payoff is $v(s, t) := \delta_{st}$.
- Fixed points of $f$ are in one-to-one correspondence with the Nash equilibria of this game.
- We will work with restrictions of this game to finite subsets of $C$. 
Goals

This paper describes a complete proof of Kakutani’s fixed point theorem:
Goals

This paper describes a complete proof of Kakutani’s fixed point theorem:

• It is simple and elementary.
Goals

This paper describes a complete proof of Kakutani’s fixed point theorem:

- It is simple and elementary.
- It arrives at Kakutani’s theorem without an intermediate stop at Brouwer’s theorem.
Goals

This paper describes a complete proof of Kakutani’s fixed point theorem:

- It is simple and elementary.
- It arrives at Kakutani’s theorem without an intermediate stop at Brouwer’s theorem.
- It is based on game theoretic concepts and reasoning, so it is complementary to the goals of instruction in theoretical economics.
The proof yields a new algorithm for computing approximate fixed points.
The proof yields a new algorithm for computing approximate fixed points.

- In comparison with the Scarf algorithm, it has several advantages.
The proof yields a new algorithm for computing approximate fixed points.

- In comparison with the Scarf algorithm, it has several advantages.
  - It is flexible and easily programmed.
The proof yields a new algorithm for computing approximate fixed points.

- In comparison with the Scarf algorithm, it has several advantages.
  - It is flexible and easily programmed.
  - It handles the “restart” problem effectively.
The proof yields a new algorithm for computing approximate fixed points.

- In comparison with the Scarf algorithm, it has several advantages.
  - It is flexible and easily programmed.
  - It handles the “restart” problem effectively.
  - It mimics iteration of a contraction mapping under certain circumstances.
The proof yields a new algorithm for computing approximate fixed points.

- In comparison with the Scarf algorithm, it has several advantages.
  - It is flexible and easily programmed.
  - It handles the “restart” problem effectively.
  - It mimics iteration of a contraction mapping under certain circumstances.
  - Preliminary tests suggest it is very fast.
Outline of the Talk

An imitation game is a two person game in which the two players’ sets of pure strategies are “the same” and the second player wishes to play the same pure strategy as the first player.
Outline of the Talk

An imitation game is a two person game in which the two players’ sets of pure strategies are “the same” and the second player wishes to play the same pure strategy as the first player.

- We construct an imitation game whose Nash equilibria produce approximate fixed points.
Outline of the Talk

An imitation game is a two person game in which the two players’ sets of pure strategies are “the same” and the second player wishes to play the same pure strategy as the first player.

- We construct an imitation game whose Nash equilibria produce approximate fixed points.
- The second component of the proof is a variant of the Lemke-Howson algorithm, due to Lemke (1965), that computes a Nash equilibrium of an imitation game.
Outline of the Talk

An imitation game is a two person game in which the two players’ sets of pure strategies are “the same” and the second player wishes to play the same pure strategy as the first player.

- We construct an imitation game whose Nash equilibria produce approximate fixed points.
- The second component of the proof is a variant of the Lemke-Howson algorithm, due to Lemke (1965), that computes a Nash equilibrium of an imitation game.
- We conclude with some remarks on the algorithm for computing approximate fixed points that results from combining these elements.
Approximate Fixed Points

- As before, $C \neq \emptyset$ is a norm-compact convex subset of a Hilbert space.
Approximate Fixed Points

- As before, $C \neq \emptyset$ is a norm-compact convex subset of a Hilbert space.
- Now let $F : C \rightarrow C$ be an u.s.c. convex valued correspondence.
Approximate Fixed Points

- As before, $C \neq \emptyset$ is a norm-compact convex subset of a Hilbert space.
- Now let $F : C \to C$ be an u.s.c. convex valued correspondence.
  - Suppose we are given $x_1, \ldots, x_m \in C$. 

Approximate Fixed Points

- As before, $C \neq \emptyset$ is a norm-compact convex subset of a Hilbert space.
- Now let $F : C \rightarrow C$ be an u.s.c. convex valued correspondence.
  - Suppose we are given $x_1, \ldots, x_m \in C$.
  - Choose any points
    $$f(x_1) \in F(x_1), \ldots, f(x_m) \in F(x_m).$$
Approximate Fixed Points

- As before, \( C \neq \emptyset \) is a norm-compact convex subset of a Hilbert space.
- Now let \( F: C \to C \) be an u.s.c. convex valued correspondence.
  - Suppose we are given \( x_1, \ldots, x_m \in C \).
  - Choose any points
    \[ f(x_1) \in F(x_1), \ldots, f(x_m) \in F(x_m). \]
- We will construct a two player game based on this information.
Two Player Games

- Let \((A, B)\) be a two player game:
Two Player Games

- Let \((A, B)\) be a two player game:
- for given positive integers \(m\) and \(n\), \(A\) and \(B\) are \(m \times n\) payoff matrices.
Two Player Games

- Let $(A, B)$ be a two player game:
  - for given positive integers $m$ and $n$, $A$ and $B$ are $m \times n$ payoff matrices.
- For any integer $k \geq 1$ let

$$
\Delta^k := \{ \sigma \in \mathbb{R}^k_\geq : \sum_{h=1}^{k} \sigma_h = 1 \}.
$$
Two Player Games

- Let \((A, B)\) be a two player game:
  - for given positive integers \(m\) and \(n\), \(A\) and \(B\) are \(m \times n\) payoff matrices.
- For any integer \(k \geq 1\) let
  \[
  \Delta^k := \{ \sigma \in \mathbb{R}^k_\geq : \sum_{h=1}^{k} \sigma_h = 1 \}.
  \]
- A **Nash equilibrium** of the game \((A, B)\) is a pair \((\sigma, \tau)\) \(\in \Delta^m \times \Delta^n\) such that:
Two Player Games

- Let \((A, B)\) be a two player game:
  - for given positive integers \(m\) and \(n\), \(A\) and \(B\) are \(m \times n\) payoff matrices.

- For any integer \(k \geq 1\) let

\[
\Delta^k := \{ \sigma \in \mathbb{R}^k_{\geq} : \sum_{h=1}^{k} \sigma_h = 1 \}.
\]

- A **Nash equilibrium** of the game \((A, B)\) is a pair \((\sigma, \tau)\) \(\in \Delta^m \times \Delta^n\) such that:
  - \(\sigma^T A \tau \geq \sigma^T A \tau\) for all \(\tilde{\sigma} \in \Delta^r\), and
Two Player Games

- Let \((A, B)\) be a two player game:
  - for given positive integers \(m\) and \(n\), \(A\) and \(B\) are \(m \times n\) payoff matrices.

- For any integer \(k \geq 1\) let

\[
\Delta^k := \{ \sigma \in \mathbb{R}^k_{\geq} : \sum_{h=1}^{k} \sigma_h = 1 \}.
\]

- 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 \(\tilde{\sigma} \in \Delta^r\), and
  - \(\sigma^T B \tau \geq \sigma^T B \tilde{\tau}\) for all \(\tilde{\tau} \in \Delta^r\).
A Derived Imitation Game

- A game \((A, B)\) is an imitation game if \(n = m\) and \(B\) is the \(m \times m\) identity matrix.

We call agent 1 the mover.

We call agent 2 the imitator.

We define a particular imitation game by letting the entries of the \(m \times m\) matrix \(A\) be \(a_{ij} = x_i - f(x_j)^2\):

- The mover wants \(x_i\) to be close to \(f(x_j)\);
- The imitator wants \(j\) to be the same as \(i\).
A Derived Imitation Game

- A game $(A, B)$ is an imitation game if $n = m$ and $B$ is the $m \times m$ identity matrix.
- We call agent 1 the mover.
A Derived Imitation Game

- A game \((A, B)\) is an imitation game if \(n = m\) and \(B\) is the \(m \times m\) identity matrix.
  - We call agent 1 the mover.
  - We call agent 2 the imitator.
A Derived Imitation Game

• A game \((A, B)\) is an imitation game if \(n = m\) and \(B\) is the \(m \times m\) identity matrix.
  • We call agent 1 the mover.
  • We call agent 2 the imitator.
• We define a particular imitation game by letting the entries of the \(m \times m\) matrix \(A\) be
  \[
a_{ij} := -\|x_i - f(x_j)\|^2.
  \]
A Derived Imitation Game

• A game \((A, B)\) is an **imitation game** if \(n = m\) and \(B\) is the \(m \times m\) identity matrix.
  
  • We call agent 1 the **mover**.
  • We call agent 2 the **imitator**.

• We define a particular imitation game by letting the entries of the \(m \times m\) matrix \(A\) be

\[
a_{ij} := -\|x_i - f(x_j)\|^2.
\]

• The mover wants \(x_i\) to be close to \(f(x_j)\);
A Derived Imitation Game

- A game \((A, B)\) is an imitation game if \(n = m\) and \(B\) is the \(m \times m\) identity matrix.
  - We call agent 1 the mover.
  - We call agent 2 the imitator.
- We define a particular imitation game by letting the entries of the \(m \times m\) matrix \(A\) be
  \[
  a_{ij} := -\|x_i - f(x_j)\|^2.
  \]
- The mover wants \(x_i\) to be close to \(f(x_j)\);
- The imitator wants \(j\) to be the same as \(i\).
• Let \((\nu, \rho)\) be a Nash equilibrium of \((A, I)\).
• Let $(\nu, \rho)$ be a Nash equilibrium of $(A, I)$.
• Let $x_{m+1} := \sum_{j=1}^{m} \rho_j f(x_j)$. 
• Let \((\nu, \rho)\) be a Nash equilibrium of \((A, I)\).
• Let \(x_{m+1} := \sum_{j=1}^{m} \rho_j f(x_j)\).
• A simple calculation shows that for each \(i\),

\[
\sum_{j=1}^{m} a_{ij} \rho_j = -\|x_i - x_{m+1}\|^2 - \sum_{j=1}^{m} \rho_j \|x_{m+1} - f(x_j)\|.
\]
• Let \((\nu, \rho)\) be a Nash equilibrium of \((A, I)\).
• Let \(x_{m+1} := \sum_{j=1}^{m} \rho_j f(x_j)\).
• A simple calculation shows that for each \(i\),

\[
\sum_{j=1}^{m} a_{ij} \rho_j = -\|x_i - x_{m+1}\|^2 - \sum_{j=1}^{m} \rho_j \|x_{m+1} - f(x_j)\|.
\]

• The second term does not depend on \(i\), so the mover’s set of pure best responses is \(\text{argmin}_{i=1,...,m} \|x_i - x_{m+1}\|\).
• Let $(\nu, \rho)$ be a Nash equilibrium of $(A, I)$.
• Let $x_{m+1} := \sum_{j=1}^{m} \rho_j f(x_j)$.
• A simple calculation shows that for each $i$,

$$\sum_{j=1}^{m} a_{ij} \rho_j = -\|x_i - x_{m+1}\|^2 - \sum_{j=1}^{m} \rho_j \|x_{m+1} - f(x_j)\|^2.$$

• The second term does not depend on $i$, so the mover’s set of pure best responses is $\operatorname{argmin}_{i=1,\ldots,m} \|x_i - x_{m+1}\|$.
• We also have $\operatorname{supp} \rho \subseteq \operatorname{supp} \nu$, so $\operatorname{supp} \rho \subseteq \operatorname{argmin}_{i=1,\ldots,m} \|x_i - x_{m+1}\|$.
Generating a Sequence of Points

- We generate a sequence of points $x_1, x_2, \ldots$ by repeating the procedure above:
Generating a Sequence of Points

- We generate a sequence of points $x_1, x_2, \ldots$ by repeating the procedure above:
  - Choose an arbitrary $x_1 \in C$. 

Let $x$ be an accumulation point of $x_m$. 
Generating a Sequence of Points

• We generate a sequence of points $x_1, x_2, \ldots$ by repeating the procedure above:
  • Choose an arbitrary $x_1 \in C$.
  • Choose $f(x_1) \in F(x_1)$. 
Generating a Sequence of Points

- We generate a sequence of points $x_1, x_2, \ldots$ by repeating the procedure above:
  - Choose an arbitrary $x_1 \in C$.
  - Choose $f(x_1) \in F(x_1)$.
  - If $x_1, \ldots, x_m$ and $f(x_1), \ldots, f(x_m)$ are given, find $x_{m+1} \in C$ and $\rho^m \in \Delta^m$ such that: 
    - $x_{m+1} = \sum_{j=1}^{m+1} \frac{1}{\rho^m_{x_j}} f(x_j)$
    - $\text{supp} \rho^m = \text{argmin}_{i=1}^{m+1} |x_i - x_{m+1}|$
Generating a Sequence of Points

- We generate a sequence of points $x_1, x_2, \ldots$ by repeating the procedure above:
  - Choose an arbitrary $x_1 \in C$.
  - Choose $f(x_1) \in F(x_1)$.
  - If $x_1, \ldots, x_m$ and $f(x_1), \ldots, f(x_m)$ are given, find $x_{m+1} \in C$ and $\rho^m \in \Delta^m$ such that:
    - $x_{m+1} = \sum_{j=1}^{m} \rho_j^m f(x_j)$;
Generating a Sequence of Points

- We generate a sequence of points $x_1, x_2, \ldots$ by repeating the procedure above:
  - Choose an arbitrary $x_1 \in C$.
  - Choose $f(x_1) \in F(x_1)$.
  - If $x_1, \ldots, x_m$ and $f(x_1), \ldots, f(x_m)$ are given, find $x_{m+1} \in C$ and $\rho^m \in \Delta^m$ such that:
    - $x_{m+1} = \sum_{j=1}^{m} \rho^m_j f(x_j)$;
    - $\text{supp } \rho^m \subset \text{argmin}_{i=1,\ldots,m} \|x_i - x_{m+1}\|$.
Generating a Sequence of Points

• We generate a sequence of points $x_1, x_2, \ldots$ by repeating the procedure above:
  
  • Choose an arbitrary $x_1 \in C$.
  • Choose $f(x_1) \in F(x_1)$.
  • If $x_1, \ldots, x_m$ and $f(x_1), \ldots, f(x_m)$ are given, find $x_{m+1} \in C$ and $\rho^m \in \Delta^m$ such that:
    • $x_{m+1} = \sum_{j=1}^{m} \rho_j^m f(x_j)$;
    • $\text{supp} \rho^m \subset \text{argmin}_{i=1,\ldots,m} \| x_i - x_{m+1} \|$.  

• Let $x^*$ be an accumulation point of $\{x_m\}$.  

Completing the Proof

We claim that $x^*$ is a fixed point of $F$. 
Completing the Proof

We claim that \( x^* \) is a fixed point of \( F \).

- Let \( V \) be a neighborhood of \( F(x^*) \).
Completing the Proof

We claim that \( x^* \) is a fixed point of \( F \).

- Let \( V \) be a neighborhood of \( F(x^*) \).
- Since \( F \) is upper semicontinuous, there is an \( \varepsilon > 0 \) such that \( F(x) \subset V \) for all \( x \in B_{\varepsilon}(x^*) \).
Completing the Proof

We claim that $x^*$ is a fixed point of $F$.

- Let $V$ be a neighborhood of $F(x^*)$.
- Since $F$ is upper semicontinuous, there is an $\varepsilon > 0$ such that $F(x) \subset V$ for all $x \in B_\varepsilon(x^*)$.
- For any $\delta > 0$, $x_{m+1} \in B_{\delta/2}(x^*)$ for large $m$. 

Completing the Proof

We claim that $x^*$ is a fixed point of $F$.

- Let $V$ be a neighborhood of $F(x^*)$.
- Since $F$ is upper semicontinuous, there is an $\varepsilon > 0$ such that $F(x) \subseteq V$ for all $x \in B_\varepsilon(x^*)$.
- For any $\delta > 0$, $x_{m+1} \in B_{\delta/2}(x^*)$ for large $m$.
- Simply because $C$ is compact,

  \[
  \lim_{m \to \infty} \min_{j=1,\ldots,m} \|x_j - x_{m+1}\| = 0.
  \]
Completing the Proof

We claim that \( x^* \) is a fixed point of \( F \).

- Let \( V \) be a neighborhood of \( F(x^*) \).
- Since \( F \) is upper semicontinuous, there is an \( \varepsilon > 0 \) such that \( F(x) \subset V \) for all \( x \in B_\varepsilon(x^*) \).
- For any \( \delta > 0 \), \( x_{m+1} \in B_{\delta/2}(x^*) \) for large \( m \).
- Simply because \( C \) is compact,
  \[
  \lim_{m \to \infty} \min_{j=1,\ldots,m} \| x_j - x_{m+1} \| = 0.
  \]

- Therefore, for arbitrarily large \( m \) we have
  \[
  \{ x_j : \rho_j^m > 0 \} \subset B_{\delta/2}(x_{m+1}) \subset B_\delta(x^*).\]
• If $\delta < \varepsilon$, then

$$f(\{x_j : \rho_j^m > 0\}) \subset \bigcup_{\rho_j^m > 0} F(x_j) \subset V.$$
• If $\delta < \varepsilon$, then

$$f(\{x_j : \rho_j^m > 0\}) \subset \bigcup_{\rho_j^m > 0} F(x_j) \subset V.$$  

• If $V$ is convex, then $x_{m+1} \in V$ and $x^* \in B_\delta(V)$. 

- p. 12/34
• If \( \delta < \varepsilon \), then

\[
F(\{x_j : \rho_j^m > 0\}) \subseteq \bigcup_{\rho_j^m > 0} F(x_j) \subset V.
\]

• If \( V \) is convex, then \( x_{m+1} \in V \) and \( x^* \in B_\delta(V) \).

• Since this is the case for all \( \delta > 0 \), if \( V \) is also closed, then \( x^* \in V \).
• If $\delta < \varepsilon$, then

$$f(\{x_j : \rho_j^m > 0\}) \subset \bigcup_{\rho_j^m > 0} F(x_j) \subset V.$$ 

• If $V$ is convex, then $x_{m+1} \in V$ and $x^* \in B_\delta(V)$.
• Since this is the case for all $\delta > 0$, if $V$ is also closed, then $x^* \in V$.

• The intersection of all of the closed convex neighborhoods of $F(x^*)$ is $F(x^*)$ itself. Therefore $x^* \in F(x^*)$. 

– p. 12/34
• If $\delta < \varepsilon$, then

$$f\left(\{x_j : \rho_j^m > 0\}\right) \subset \bigcup_{\rho_j^m > 0} F(x_j) \subset V.$$ 

• If $V$ is convex, then $x_{m+1} \in V$ and $x^* \in B_\delta(V)$.

• Since this is the case for all $\delta > 0$, if $V$ is also closed, then $x^* \in V$.

• The intersection of all of the closed convex neighborhoods of $F(x^*)$ is $F(x^*)$ itself.

Therefore $x^* \in F(x^*)$.

• The remaining gap in our proof of Kakutani’s theorem is to show that the imitation game defined above has a Nash equilibrium.
The Lemke-Howson Algorithm

- The Nash equilibrium conditions can be reexpressed as follows:
The Lemke-Howson Algorithm

- The Nash equilibrium conditions can be reexpressed as follows:
  - for all $i = 1, \ldots, m$, either:
    - $\sigma_i = 0$, or
    - $e_i^T A\tau \geq e_h^T A\tau$ for all $h = 1, \ldots, m$;
The Lemke-Howson Algorithm

- The Nash equilibrium conditions can be reexpressed as follows:
  - for all $i = 1, \ldots, m$, either:
    - $\sigma_i = 0$, or
    - $e_i^T A \tau \geq e_h^T A \tau$ for all $h = 1, \ldots, m$;
  - for all $j = 1, \ldots, n$, either:
    - $\tau_j = 0$, or
    - $\sigma^T B e_j \geq \sigma^T B e_k$ for all $k = 1, \ldots, n$. 
The Lemke-Howson Algorithm

• The Nash equilibrium conditions can be reexpressed as follows:
  • for all $i = 1, \ldots, m$, either:
    • $\sigma_i = 0$, or
    • $e_i^T A \tau \geq e_h^T A \tau$ for all $h = 1, \ldots, m$;
  • for all $j = 1, \ldots, n$, either:
    • $\tau_j = 0$, or
    • $\sigma^T B e_j \geq \sigma^T B e_k$ for all $k = 1, \ldots, n$.

• For generic $A$ and $B$, the Lemke-Howson algorithm traces a path in the one dimensional set of points in $\Delta^m \times \Delta^n$ satisfying all but a particular one of these $m + n$ conditions.
Imitation Games

• The game \((A, B)\) is an imitation game if \(n = m\) and \(B\) is the \(m \times m\) identity matrix.
Imitation Games

- The game \((A, B)\) is an imitation game if \(n = m\) and \(B\) is the \(m \times m\) identity matrix.
- If \((\nu, \rho)\) is a Nash equilibrium of the imitation game \((A, I)\), then

\[
\text{supp} \ \rho \subseteq \text{supp} \ \nu \subseteq \arg\max_{h=1,\ldots,m} e_h^T A \rho.
\]
Imitation Games

- The game \((A, B)\) is an imitation game if \(n = m\) and \(B\) is the \(m \times m\) identity matrix.
- If \((\nu, \rho)\) is a Nash equilibrium of the imitation game \((A, I)\), then

\[
\text{supp } \rho \subset \text{supp } \nu \subset \arg\max_{h=1,\ldots,m} e_h^T A \rho.
\]

- Conversely, if \(\text{supp } \rho \subset \arg\max_{h=1,\ldots,m} e_h^T A \rho\), then \((\beta_{\text{supp } \rho}, \rho)\) is a Nash equilibrium. (Here \(\beta_{\text{supp } \rho}\) is the uniform distribution on \(\text{supp } \rho\).)
Therefore we will have established the required existence result if we show that there is $\rho \in \Delta^m$ such that for each $j = 1, \ldots, m$, either:

- $\rho_j = 0$, or
- $e_j^T A \rho = \max_{h=1,\ldots,m} e_h^T A \rho$.

Such a $\rho$ is called an $I$-equilibrium of the imitation game $(A, I)$.
The Lemke Path Algorithm

• Our second main contribution is to show that the Lemke-Howson algorithm follows the path of the Lemke (1965) algorithm when applied to an imitation game.

In view of a result of Morris (1994), this gives a new proof of a recent result of Savani and von Stengel (2004): there is a sequence of two player games for which the length of the shortest Lemke-Howson path grows exponentially with the size of the game.

The Lemke (1965) algorithm has a simpler description than the Lemke-Howson algorithm, which improves the "package" used to prove Kakutani.
Our second main contribution is to show that the Lemke-Howson algorithm follows the path of the Lemke (1965) algorithm when applied to an imitation game. In view of a result of Morris (1994), this gives a new proof of a recent result of Savani and von Stengel (2004): there is a sequence of two player games for which the length of the shortest Lemke-Howson path grows exponentially with the size of the game.
The Lemke Path Algorithm

- Our second main contribution is to show that the Lemke-Howson algorithm follows the path of the Lemke (1965) algorithm when applied to an imitation game.

- In view of a result of Morris (1994), this gives a new proof of a recent result of Savani and von Stengel (2004): there is a sequence of two player games for which the length of the shortest Lemke-Howson path grows exponentially with the size of the game.

- The Lemke (1965) algorithm has a simpler description than the Lemke-Howson algorithm, which improves the "package" used to prove Kakutani.
Labels

- Let the set of labels be

\[ L := \{ (\iota, \kappa) : \iota = 1, 2, \kappa = 1, \ldots, m \}. \]
Labels

- Let the set of labels be

\[ L := \{ (\nu, \kappa) : \nu = 1, 2, \kappa = 1, \ldots, m \} . \]

- Let \( H^m := \{ \rho \in \mathbb{R}^m : \sum_{j=1}^{m} \rho_j = 1 \} . \)
Labels

- Let the set of labels be

\[ L := \{ (\iota, \kappa) : \iota = 1, 2, \kappa = 1, \ldots, m \}. \]

- Let \( H^m := \{ \rho \in \mathbb{R}^m : \sum_{j=1}^{m} \rho_j = 1 \} \).

- For \( \kappa = 1, \ldots, m \) define:
Labels

Let the set of labels be

\[ L := \{ (\iota, \kappa) : \iota = 1, 2, \kappa = 1, \ldots, m \} \].

Let \( H^m := \{ \rho \in \mathbb{R}^m : \sum_{j=1}^{m} \rho_j = 1 \} \).

For \( \kappa = 1, \ldots, m \) define:

- \( \lambda_{1\kappa} : H^m \times \mathbb{R} \rightarrow \mathbb{R} \) by
  \[
  \lambda_{1\kappa}(\rho, u) = \rho_{\kappa};
  \]
Labels

- Let the set of labels be
  \[ L := \{ (\iota, \kappa) : \iota = 1, 2, \kappa = 1, \ldots, m \}. \]
- Let \( H^m := \{ \rho \in \mathbb{R}^m : \sum_{j=1}^{m} \rho_j = 1 \} \).
- For \( \kappa = 1, \ldots, m \) define:
  - \( \lambda_{1\kappa} : H^m \times \mathbb{R} \rightarrow \mathbb{R} \) by
    \[ \lambda_{1\kappa}(\rho, u) = \rho_\kappa; \]
  - \( \lambda_{2\kappa} : H^m \times \mathbb{R} \rightarrow \mathbb{R} \) by
    \[ \lambda_{2\kappa}(\rho, u) = u - \sum_{j=1}^{m} a_{\kappa j} \rho_j. \]
The Polyhedron

- Let $P$ be the set of $(\rho, u) \in H^m \times \mathbb{R}$ such that $\lambda_{\iota \kappa}(\rho, u) \geq 0$ for all $(\iota, \kappa) \in L$. 


The Polyhedron

- Let $P$ be the set of $(\rho, u) \in H^m \times \mathbb{R}$ such that $\lambda_{\iota \kappa}(\rho, u) \geq 0$ for all $(\iota, \kappa) \in L$.

- For $\alpha \subset L$ let $F_\alpha$ be the set of $(\rho, u) \in P$ such that $\lambda_{\iota \kappa}(\rho, u) = 0$ for all $(\iota, \kappa) \in \alpha$. \\

The Polyhedron

- Let $P$ be the set of $(\rho, u) \in H^m \times \mathbb{R}$ such that $\lambda_{\kappa}(\rho, u) \geq 0$ for all $(\iota, \kappa) \in L$.
- For $\alpha \subset L$ let $F_\alpha$ be the set of $(\rho, u) \in P$ such that $\lambda_{\kappa}(\rho, u) = 0$ for all $(\iota, \kappa) \in \alpha$.
- The matrix $A$ is in general position if every $F_\alpha$ is either empty or $(m - |\alpha|)$-dimensional.
The Polyhedron

- Let $P$ be the set of $(\rho, u) \in H^m \times \mathbb{R}$ such that $\lambda_{\iota\kappa}(\rho, u) \geq 0$ for all $(\iota, \kappa) \in L$.
- For $\alpha \subset L$ let $F_{\alpha}$ be the set of $(\rho, u) \in P$ such that $\lambda_{\iota\kappa}(\rho, u) = 0$ for all $(\iota, \kappa) \in \alpha$.
- The matrix $A$ is in general position if every $F_{\alpha}$ is either empty or $(m - |\alpha|)$-dimensional.
  - It suffices to establish that there is an $I$-equilibrium when $A$ is in general position, as we shall assume for the remainder, because general position matrices are dense in the set of all $m \times m$ matrices.
Feasible Bases

- A basis is an $m$-element set $\beta \subset L$ such that \( \{ \lambda_{\nu\kappa} : (\nu, \kappa) \in \beta \} \) is linearly independent.
Feasible Bases

- A basis is an \( m \)-element set \( \beta \subset L \) such that \( \{ \lambda_{\mu\kappa} : (\mu, \kappa) \in \beta \} \) is linearly independent.
- For example, \( \lambda_{11}, \lambda_{12}, \ldots, \lambda_{1m} \) are not linearly independent.
Feasible Bases

- A basis is an $m$-element set $\beta \subset L$ such that $\{\lambda_{\kappa} : (\nu, \kappa) \in \beta\}$ is linearly independent.
- For example, $\lambda_{11}, \lambda_{12}, \ldots, \lambda_{1m}$ are not linearly independent.
- A basis $\beta$ is feasible if $F_{\beta}$ is nonempty.
Feasible Bases

- A basis is an $m$-element set $\beta \subset L$ such that $\{ \lambda_{\nu\kappa} : (\nu, \kappa) \in \beta \}$ is linearly independent.

- For example, $\lambda_{11}, \lambda_{12}, \ldots, \lambda_{1m}$ are not linearly independent.

- A basis $\beta$ is feasible if $F_\beta$ is nonempty.

- In this case $F_\beta$ is 0-dimensional, by nondegeneracy, and convex. That is, the unique element $(\rho_\beta, u_\beta)$ of $F_\beta$ is a vertex of $P$. 
Pivoting

• If \( \beta \) is a feasible basis and \((\nu, \kappa) \in \beta\), then 
\( F_{\beta \setminus \{(\nu, \kappa)\}} \) is nonempty (it contains \( F_{\beta} \)) so (by general position) it is an edge of \( P \).

• When is an edge \( F_{\nu} \) of \( P \) unbounded?

The projection of \( F_{\nu} \) onto \( H_{m} \) is contained in \( \mathcal{C}_{m} \), so an unbounded edge is a vertical ray.

If this is the case, then \( \nu = 1 \) for all \((\nu, \kappa) \in \beta\), and for some \( 1 \leq \nu \leq m \) we have
\( \nu = f(1; 1); \ldots; (1; m) \setminus f(1; \nu) \).
Pivoting

- If $\beta$ is a feasible basis and $(\iota, \kappa) \in \beta$, then $F_{\beta \setminus \{(\iota, \kappa)\}}$ is nonempty (it contains $F_{\beta}$) so (by general position) it is an edge of $P$.
- If $F_{\beta \setminus \{(\iota, \kappa)\}}$ is bounded, and its other endpoint is $F_{\beta'}$, then we say that $\beta'$ is reached from $\beta$ via the pivot that drops $(\iota, \kappa)$.
Pivoting

• If $\beta$ is a feasible basis and $(\iota, \kappa) \in \beta$, then $F_{\beta \setminus \{(\iota, \kappa)\}}$ is nonempty (it contains $F_\beta$) so (by general position) it is an edge of $P$.

• If $F_{\beta \setminus \{(\iota, \kappa)\}}$ is bounded, and its other endpoint is $F_{\beta'}$, then we say that $\beta'$ is reached from $\beta$ via the pivot that drops $(\iota, \kappa)$.

• When is an edge $F_\alpha$ of $P$ unbounded?
Pivoting

- If $\beta$ is a feasible basis and $(\iota, \kappa) \in \beta$, then $F_{\beta \setminus \{(\iota, \kappa)\}}$ is nonempty (it contains $F_{\beta}$) so (by general position) it is an edge of $P$.
- If $F_{\beta \setminus \{(\iota, \kappa)\}}$ is bounded, and its other endpoint is $F_{\beta'}$, then we say that $\beta'$ is reached from $\beta$ via the pivot that drops $(\iota, \kappa)$.
- When is an edge $F_{\alpha}$ of $P$ unbounded?
  - The projection of $F_{\alpha}$ onto $H^m$ is contained in $\Delta^m$, so an unbounded edge is a vertical ray.
Pivoting

- If $\beta$ is a feasible basis and $(\iota, \kappa) \in \beta$, then $F_{\beta \setminus \{(\iota, \kappa)\}}$ is nonempty (it contains $F_{\beta}$) so (by general position) it is an edge of $P$.
- If $F_{\beta \setminus \{(\iota, \kappa)\}}$ is bounded, and its other endpoint is $F_{\beta'}$, then we say that $\beta'$ is reached from $\beta$ via the pivot that drops $(\iota, \kappa)$.
- When is an edge $F_\alpha$ of $P$ unbounded?
  - The projection of $F_\alpha$ onto $H^m$ is contained in $\Delta^m$, so an unbounded edge is a vertical ray.
  - If this is the case, then $\iota = 1$ for all $(\iota, \kappa) \in \alpha$, and for some $1 \leq \mu \leq m$ we have
    \[
    \alpha = \{(1, 1), \ldots, (1, m)\} \setminus \{(1, \mu)\}.
    \]
• Let \( \pi : L \rightarrow \{1, \ldots, m\} \) be the projection
\[ \pi(\iota, \kappa) := \kappa. \]
• Let $\pi : L \to \{1, \ldots, m\}$ be the projection $\pi(\iota, \kappa) := \kappa$.

• A feasible basis $\beta$ is complementary if

$$\pi(\beta) = \{1, \ldots, m\}.$$
• Let $\pi : L \rightarrow \{1, \ldots, m\}$ be the projection $\pi(\iota, \kappa) := \kappa$.

• A feasible basis $\beta$ is **complementary** if $\pi(\beta) = \{1, \ldots, m\}$.

• Our goal is to find a complementary basis.
• Let \( \pi : L \to \{1, \ldots, m\} \) be the projection \( \pi(\nu, \kappa) := \kappa \).

• A feasible basis \( \beta \) is **complementary** if

\[
\pi(\beta) = \{1, \ldots, m\}.
\]

• Our goal is to find a complementary basis.

• Fix an integer \( 1 \leq \mu \leq m \). A feasible basis \( \beta \) is \( \mu \)-almost complementary if

\[
\{1, \ldots, \mu - 1, \mu + 1, \ldots, m\} \subset \pi(\beta).
\]
• Let $\pi : L \to \{1, \ldots, m\}$ be the projection $\pi(\iota, \kappa) := \kappa$.

• A feasible basis $\beta$ is complementary if

$$\pi(\beta) = \{1, \ldots, m\}.$$ 

• Our goal is to find a complementary basis.

• Fix an integer $1 \leq \mu \leq m$. A feasible basis $\beta$ is $\mu$-almost complementary if

$$\{1, \ldots, \mu - 1, \mu + 1, \ldots, m\} \subset \pi(\beta).$$

• The Lemke path algorithm pivots through the set of $\mu$-almost complementary bases until it reaches a complementary basis.
Graph Terminology

• A (simple, undirected) graph is a pair $G = (V, E)$ in which:

- $V$ is a finite set of vertices,
- $E$ is a finite set of edges,
- each edge has two distinct endpoints in $V$, and
- for any two distinct vertices there is at most one edge that has them as its endpoints.

Two vertices are neighbors if they are the endpoints of an edge, and the degree of a vertex is the number of neighbors it has.

Fact: If no vertex has degree greater than 2, then $G$ is a union of paths, loops, and isolated points.
A (simple, undirected) graph is a pair $G = (V, E)$ in which:

- $V$ is a finite set of vertices,
- $E$ is a finite set of edges,
- each edge has two distinct endpoints in $V$,
- for any two distinct vertices there is at most one edge that has them as its endpoints.

Two vertices are neighbors if they are the endpoints of an edge, and the degree of a vertex is the number of neighbors it has.

Fact: If no vertex has degree greater than 2, then $G$ is a union of paths, loops, and isolated points.
Graph Terminology

- A (simple, undirected) graph is a pair \( G = (V, E) \) in which:
  - \( V \) is a finite set of vertices,
  - \( E \) is a finite set of edges,
  - each edge has two distinct endpoints in \( V \), and
  - for any two distinct vertices there is at most one edge that has them as its endpoints.

Two vertices are neighbors if they are the endpoints of an edge, and the degree of a vertex is the number of neighbors it has.

Fact: If no vertex has degree greater than 2, then \( G \) is a union of paths, loops, and isolated points.
Graph Terminology

• A (simple, undirected) graph is a pair $G = (V, E)$ in which:
  • $V$ is a finite set of vertices,
  • $E$ is a finite set of edges,
  • each edge has two distinct endpoints in $V$, and
Graph Terminology

- A (simple, undirected) graph is a pair $G = (V, E)$ in which:
  - $V$ is a finite set of vertices,
  - $E$ is a finite set of edges,
  - each edge has two distinct endpoints in $V$, and
  - for any two distinct vertices there is at most one edge that has them as its endpoints.
A (simple, undirected) graph is a pair $G = (V, E)$ in which:

- $V$ is a finite set of vertices,
- $E$ is a finite set of edges,
- each edge has two distinct endpoints in $V$, and
- for any two distinct vertices there is at most one edge that has them as its endpoints.

Two vertices are neighbors if they are the endpoints of an edge, and the degree of a vertex is the number of neighbors it has.
A (simple, undirected) graph is a pair \( G = (V, E) \) in which:

- \( V \) is a finite set of vertices,
- \( E \) is a finite set of edges,
- each edge has two distinct endpoints in \( V \), and
- for any two distinct vertices there is at most one edge that has them as its endpoints.

Two vertices are neighbors if they are the endpoints of an edge, and the degree of a vertex is the number of neighbors it has.

Fact: If no vertex has degree greater than 2, then \( G \) is a union of paths, loops, and isolated points.
The Algorithm’s Graph

- Define a graph $G_{\mu} = (V_{\mu}, E_{\mu})$ by letting:

$V_{\mu}$ be the set of $\mu$-almost complementary bases, and

$E_{\mu}$ be the set of unordered pairs $\bar{\bar{0}}$ of distinct vertices $\bar{0} \in V_{\mu}$ such that $\left(\bar{0} \setminus \bar{0}^0\right) = \{1, \ldots, \mu, \ldots, m\}$.
The Algorithm’s Graph

• Define a graph $G_\mu = (V_\mu, E_\mu)$ by letting:
  • $V_\mu$ be the set of $\mu$-almost complementary bases, and
The Algorithm’s Graph

- Define a graph $G_\mu = (V_\mu, E_\mu)$ by letting:
  - $V_\mu$ be the set of $\mu$-almost complementary bases, and
  - $E_\mu$ be the set of unordered pairs $\beta \beta'$ of distinct vertices $\beta, \beta' \in V_\mu$ such that
    $$\pi(\beta \cap \beta') = \{1, \ldots, \mu - 1, \mu + 1, \ldots, m\}.$$
Pivoting in $G_\mu$

- If $\beta$ and $\beta'$ are neighbors in $G_\mu$, then $\beta'$ is reached from $\beta$ by the pivot that drops the unique element of $\beta \setminus (\beta \cap \beta')$. 

If $\bar{\beta}$ is almost complementary, $(\pi; \cdot) \in \bar{\beta}$, and $\bar{\mu} = (\bar{\pi} \cdot f (\pi; \cdot) g) = f_1; \ldots; \bar{\mu} - 1; \bar{\mu} + 1; \ldots; m g$, then either:

- There is a neighbor $\bar{\beta}_0$ of $\bar{\beta}$ that is reached from $\bar{\beta}$ by the pivot that drops $(\pi; \cdot)$, or
- $\bar{\beta} f (\pi; \cdot) g$ is an unbounded edge of $P$. 

- p. 24/3
Pivoting in $G_{\mu}$

- If $\beta$ and $\beta'$ are neighbors in $G_{\mu}$, then $\beta'$ is reached from $\beta$ by the pivot that drops the unique element of $\beta \setminus (\beta \cap \beta')$.

- If $\beta$ is almost complementary, $(\iota, \kappa) \in \beta$, and
  $$\pi(\beta \setminus \{(\iota, \kappa)\}) = \{1, \ldots, \mu - 1, \mu + 1, \ldots, m\},$$
  then either:
Pivoting in $G_{\mu}$

- If $\beta$ and $\beta'$ are neighbors in $G_{\mu}$, then $\beta'$ is reached from $\beta$ by the pivot that drops the unique element of $\beta \setminus (\beta \cap \beta')$.

- If $\beta$ is almost complementary, $(\iota, \kappa) \in \beta$, and

$$\pi(\beta \setminus \{(\iota, \kappa)\}) = \{1, \ldots, \mu - 1, \mu + 1, \ldots, m\},$$

then either:

  - There is a neighbor $\beta'$ of $\beta$ that is reached from $\beta$ by the pivot that drops $(\iota, \kappa)$, or
Pivoting in $G_\mu$

- If $\beta$ and $\beta'$ are neighbors in $G_\mu$, then $\beta'$ is reached from $\beta$ by the pivot that drops the unique element of $\beta \setminus (\beta \cap \beta')$.

- If $\beta$ is almost complementary, $(\iota, \kappa) \in \beta$, and
  \[ \pi(\beta \setminus \{ (\iota, \kappa) \}) = \{1, \ldots, \mu - 1, \mu + 1, \ldots, m\}, \]
  then either:
  - There is a neighbor $\beta'$ of $\beta$ that is reached from $\beta$ by the pivot that drops $(\iota, \kappa)$, or
  - $F_{\beta \setminus \{(\iota,\kappa)\}}$ is an unbounded edge of $P$. 
**Complementary Vertices**

- If $\beta \in V_\mu$ is complementary, then there is a unique $(\iota, \mu) \in \beta$ such that

  \[ \pi(\beta \setminus \{(\iota, \mu)\}) = \{1, \ldots, \mu - 1, \mu + 1, \ldots, m\}. \]
Complementary Vertices

- If $\beta \in V_\mu$ is complementary, then there is a unique $(\iota, \mu) \in \beta$ such that
  \[ \pi(\beta \setminus \{(\iota, \mu)\}) = \{1, \ldots, \mu - 1, \mu + 1, \ldots, m\}. \]

- Either:
Complementary Vertices

- If $\beta \in V_\mu$ is complementary, then there is a unique $(\iota, \mu) \in \beta$ such that

$$\pi(\beta \setminus \{(\iota, \mu)\}) = \{1, \ldots, \mu - 1, \mu + 1, \ldots, m\}.$$ 

- Either:
  - $\beta$ has degree one in $G_\mu$, or
Complementary Vertices

- If $\beta \in V_\mu$ is complementary, then there is a unique $(\iota, \mu) \in \beta$ such that
  \[ \pi(\beta \setminus \{(\iota, \mu)\}) = \{1, \ldots, \mu - 1, \mu + 1, \ldots, m\}. \]

- Either:
  - $\beta$ has degree one in $G_\mu$, or
  - $\beta$ has degree zero in $G_\mu$ because
    \[ \beta \setminus \{(\iota, \mu)\} = \{(1, 1), \ldots, (1, m)\} \setminus \{(1, \mu)\}. \]
Almost Complementary $\beta$

- If $\beta$ is almost complementary, but not complementary, then there is a unique $r$, called the **redundant label**, such that $(1, r), (2, r) \in \beta$. 

Note that for $(\bar{\gamma}; \cdot)$ we have

$$\bar{\gamma} \in f(1; \cdot) \cup \cdots \cup \{1; \cdots; \bar{\gamma}_1; \bar{\gamma} + 1; \cdots; \bar{\gamma}_m\}$$

if and only if $\cdot = r$.

 Either:

- the degree of $\bar{\gamma}$ in $G \bar{\gamma}$ is two, or
- the degree of $\bar{\gamma}$ in $G \bar{\gamma}$ is one because $\bar{\gamma} \in f(1; \cdot) \cup \cdots \cup \{1; \cdots; \bar{\gamma}_1; \bar{\gamma} + 1; \cdots; \bar{\gamma}_m\}$:
Almost Complementary $\beta$

- If $\beta$ is almost complementary, but not complementary, then there is a unique $r$, called the **redundant label**, such that $(1, r), (2, r) \in \beta$.

- Note that for $(\nu, \kappa) \in \beta$ we have

$$\pi(\beta \setminus \{(\nu, \kappa)\}) = \{1, \ldots, \mu - 1, \mu + 1, \ldots, m\}$$

if and only if $\kappa = r$. 

Almost Complementary $\beta$

- If $\beta$ is almost complementary, but not complementary, then there is a unique $r$, called the redundant label, such that $(1, r), (2, r) \in \beta$.

- Note that for $(\nu, \kappa) \in \beta$ we have

$$\pi(\beta \setminus \{(\nu, \kappa)\}) = \{1, \ldots, \mu - 1, \mu + 1, \ldots, m\}$$

if and only if $\kappa = r$.

- Either:
Almost Complementary $\beta$

• If $\beta$ is almost complementary, but not complementary, then there is a unique $r$, called the **redundant label**, such that $(1, r), (2, r) \in \beta$.

• Note that for $(\nu, \kappa) \in \beta$ we have

$\pi(\beta \setminus \{(\nu, \kappa)\}) = \{1, \ldots, \mu - 1, \mu + 1, \ldots, m\}$

if and only if $\kappa = r$.

• Either:
  • the degree of $\beta$ in $G_\mu$ is two, or
Almost Complementary $\beta$

- If $\beta$ is almost complementary, but not complementary, then there is a unique $r$, called the **redundant label**, such that $(1, r), (2, r) \in \beta$.

- Note that for $(\nu, \kappa) \in \beta$ we have

$$
\pi(\beta \setminus \{(\nu, \kappa)\}) = \{1, \ldots, \mu - 1, \mu + 1, \ldots, m\}
$$

if and only if $\kappa = r$.

- Either:
  - the degree of $\beta$ in $G_\mu$ is two, or
  - the degree of $\beta$ in $G_\mu$ is one because

$$
\beta \setminus \{(2, r)\} = \{(1, 1), \ldots, (1, m)\} \setminus \{(1, \mu)\}.
$$
The $\mu$-Initial Basis

- A $\mu$-almost complementary basis $\beta$ is $\mu$-initial if
  \[(1, 1), \ldots, (1, \mu - 1), (1, \mu + 1), \ldots, (1, m) \subseteq \beta.\]
The $\mu$-Initial Basis

- A $\mu$-almost complementary basis $\beta$ is $\mu$-initial if $
\{(1, 1), \ldots, (1, \mu-1), (1, \mu+1), \ldots, (1, m)\} \subset \beta.$

- For $1 \leq \mu \leq m$ let $e_\mu$ be the vertex of $\Delta^m$
corresponding to $\mu$. 

The $\mu$-Initial Basis

- A $\mu$-almost complementary basis $\beta$ is $\mu$-initial if
  \[\{(1, 1), \ldots, (1, \mu-1), (1, \mu+1), \ldots, (1, m)\} \subset \beta.\]

- For $1 \leq \mu \leq m$ let $e_\mu$ be the vertex of $\Delta^m$ corresponding to $\mu$.

- If $\beta$ is $\mu$-initial, then necessarily
  \[F_\beta = \{(e_\mu, \max_{i=1,\ldots,m} a_{i\mu})\}.\]
The \( \mu \)-Initial Basis

- A \( \mu \)-almost complementary basis \( \beta \) is \( \mu \)-initial if
  \[
  \{(1, 1), \ldots, (1, \mu - 1), (1, \mu + 1), \ldots, (1, m)\} \subset \beta.
  \]

- For \( 1 \leq \mu \leq m \) let \( e_\mu \) be the vertex of \( \Delta^m \) corresponding to \( \mu \).

- If \( \beta \) is \( \mu \)-initial, then necessarily
  \[
  F_{\beta} = \{(e_\mu, \max_{i=1,\ldots,m} a_{i\mu})\}.
  \]

- The general position assumption implies that there is a unique \( \mu \)-initial basis \( \beta_\mu \).
The Algorithm’s $\mu$-Path

- The algorithm begins at $\beta_\mu$. 
The Algorithm’s $\mu$-Path

- The algorithm begins at $\beta_\mu$.
- Either:
The Algorithm’s $\mu$-Path

- The algorithm begins at $\beta_\mu$.
- Either:
  - $\beta_\mu$ is complementary, in which case the algorithm terminates, or
The Algorithm’s $\mu$-Path

- The algorithm begins at $\beta_\mu$.
- Either:
  - $\beta_\mu$ is complementary, in which case the algorithm terminates, or
  - the degree of $\beta_\mu$ in $G_\mu$ is one, in which case the algorithm follows the path in $G_\mu$ that begins at $\beta_\mu$ until it reaches this path’s other endpoint. This endpoint:
The Algorithm’s $\mu$-Path

- The algorithm begins at $\beta_\mu$.
- Either:
  - $\beta_\mu$ is complementary, in which case the algorithm terminates, or
  - the degree of $\beta_\mu$ in $G_\mu$ is one, in which case the algorithm follows the path in $G_\mu$ that begins at $\beta_\mu$ until it reaches this path’s other endpoint. This endpoint:
    - has degree one in $G_\mu$ and
The Algorithm’s $\mu$-Path

- The algorithm begins at $\beta_\mu$.
- Either:
  - $\beta_\mu$ is complementary, in which case the algorithm terminates, or
  - the degree of $\beta_\mu$ in $G_\mu$ is one, in which case the algorithm follows the path in $G_\mu$ that begins at $\beta_\mu$ until it reaches this path’s other endpoint. This endpoint:
    - has degree one in $G_\mu$ and
    - is different from $\beta_\mu$, so
The Algorithm’s $\mu$-Path

- The algorithm begins at $\beta_\mu$.
- Either:
  - $\beta_\mu$ is complementary, in which case the algorithm terminates, or
  - the degree of $\beta_\mu$ in $G_\mu$ is one, in which case the algorithm follows the path in $G_\mu$ that begins at $\beta_\mu$ until it reaches this path’s other endpoint. This endpoint:
    - has degree one in $G_\mu$ and
    - is different from $\beta_\mu$, so
    - it must be complementary.
Illustrating Lemke Paths

• For $A \subseteq \{1, \ldots, m\}$ let $p(A) \in \Delta^m$ be the $I$-equilibrium at which every pure strategy outside $A$ is assigned zero probability and every action in $A$ is a best response for the mover.
Illustrating Lemke Paths

- For $A \subset \{1, \ldots, m\}$ let $p(A) \in \Delta^m$ be the \textit{I}-equilibrium at which every pure strategy outside $A$ is assigned zero probability and every action in $A$ is a best response for the mover.

- For $A \subset \{1, \ldots, m\}$ and $\mu, \nu \in \{1, \ldots, m\} \setminus A$, let $q^\mu_\nu(A) := \rho\beta^\mu_\nu(A)$ be the mixed strategy at which every pure strategy outside $A \cup \{\mu\}$ is assigned zero probability and every action in $A \cup \{\nu\}$ is a best response for the mover.
Illustrating Lemke Paths

- For $A \subset \{1, \ldots, m\}$ let $p(A) \in \Delta^m$ be the $I$-equilibrium at which every pure strategy outside $A$ is assigned zero probability and every action in $A$ is a best response for the mover.

- For $A \subset \{1, \ldots, m\}$ and $\mu, \nu \in \{1, \ldots, m\} \setminus A$, let $q^\mu_{\nu}(A) := \rho \beta^\mu_{\nu}(A)$ be the mixed strategy at which every pure strategy outside $A \cup \{\mu\}$ is assigned zero probability and every action in $A \cup \{\nu\}$ is a best response for the mover.

- For $A \subset \{1, \ldots, m\}$ and $\mu \in \{1, \ldots, m\} \setminus A$, let $e^\mu(A)$ be the set of $\rho$ at which every pure strategy outside $A \cup \{\mu\}$ is assigned zero probability and every action in $A$ is a best response for the mover.
Lemke Paths

1 2 3

1 2 3

1 2 3

1 2 3

1 2 3

1 2 3

1 2 3
Lemke Paths

\[ q_3^2(\emptyset) \]

\[ e^2(\{3\}) \]
Lemke Paths

\[ q_3^2(\emptyset) \]
\[ e^2(\{3\}) \]
\[ q_1^2(\{3\}) \]
Lemke Paths

\[ q_3^2(\emptyset) \]
\[ e^2(\{3\}) \]
\[ q_1^2(\{3\}) \]
\[ e^2(\{1, 3\}) \]
Lemke Paths

\[ q_3^2(\emptyset) \]
\[ e^2(\{3\}) \]
\[ q_3^2(\{1\}) \]
\[ e^2(\{1,3\}) \]
\[ q_1^2(\{3\}) \]
Lemke Paths

\[ q_3^2(\emptyset) \]
\[ e^2(\{3\}) \]
\[ q_3^2(\{1\}) \]
\[ e^2(\{1, 3\}) \]
\[ e^2(\{1\}) \]

\[ q_1^2(\{3\}) \]
Lemke Paths
Lemke Paths

- $q_3^2(\emptyset)$
- $e^2(\{3\})$
- $q_3^2(\{1\})$
- $q_1^2(\{3\})$
- $p(\{1, 2\})$
- $e^2(\{1, 3\})$
- $e^2(\{1\})$
- $e^3(\{1, 2\})$
- $e^3(\{1\})$
- $q_1^3(\emptyset)$
- $q_2^3(\{1\})$
Lemke Paths

\[ q_3^2(\emptyset) \quad e^2(\{3\}) \quad q_1^2(\{3\}) \]

\[ q_3^2(\{1\}) \quad e^2(\{1,3\}) \quad q_1^2(\{3\}) \]

\[ q_2^3(\{1\}) \quad e^3(\{1\}) \quad q_2^3(\{1\}) \]

\[ q_2^3(\{1\}) \quad e^3(\{1,2\}) \quad q_2^3(\emptyset) \]

\[ q_2^1(\emptyset) \quad e^1(\{2\}) \quad q_2^1(\emptyset) \]

\[ p(\{1,2\}) \quad e^1(\{2\}) \quad p(\{1,2\}) \]
Computing Approximate FP’s

• Generating a certain number of terms of the sequence \( \{x_m\} \), then stopping according to an appropriate rule, is an algorithm for computing an approximate fixed point.
Computing Approximate FP’s

- Generating a certain number of terms of the sequence \( \{x_m\} \), then stopping according to an appropriate rule, is an algorithm for computing an approximate fixed point.

- One may use the Lemke path algorithm to find an equilibrium of the game used to compute \( x_{m+1} \), but this is not essential. Any other procedure for finding an approximate Nash equilibrium, could also be used for this subroutine.
Example
Example

\[ f(x_1) \]

\[ x_1 \]
Example

Nash 1. $\rho(x_1) = 1$

- $\frac{1}{2}(x_1) = 1$
- $\frac{1}{2}(x_1) = 0$
- $\frac{1}{2}(x_2) = 0$

Diagram:

- $x_1$ to $f(x_1)$
- $x_2$
Example

Nash 1. $\rho(x_1) = 1$

\[ f(x_2) \]

\[ f(x_1) \]
Example

Nash 1. $\rho(x_1) = 1$

Nash 2. $\rho(x_1) = 0.5$, $\rho(x_2) = 0.5$
Example

Nash 1. $\rho(x_1) = 1$
Nash 2. $\rho(x_1) = 0.5$, $\rho(x_2) = 0.5$
Nash 1. \( \rho(x_1) = 1 \)
Nash 2. \( \rho(x_1) = 0.5, \rho(x_2) = 0.5 \)

\[ f(x_1) \]
\[ f(x_2) \]
\[ f(x_3) \]
Example

Nash 1. $\rho(x_1) = 1$
Nash 2. $\rho(x_1) = 0.5$, $\rho(x_2) = 0.5$
Nash 3. $\rho(x_2) = 0.3$, $\rho(x_3) = 0.7$
Example

Nash 1. $\rho(x_1) = 1$
Nash 2. $\rho(x_1) = 0.5$, $\rho(x_2) = 0.5$
Nash 3. $\rho(x_2) = 0.3$, $\rho(x_3) = 0.7$
Example

Nash 1. $\rho(x_1) = 1$
Nash 2. $\rho(x_1) = 0.5$, $\rho(x_2) = 0.5$
Nash 3. $\rho(x_2) = 0.3$, $\rho(x_3) = 0.7$
Example

Nash 1. $\rho(x_1) = 1$

Nash 2. $\rho(x_1) = 0.5$, $\rho(x_2) = 0.5$

Nash 3. $\rho(x_2) = 0.3$, $\rho(x_3) = 0.7$
Example

Nash 1. \( \rho(x_1) = 1 \)
Nash 2. \( \rho(x_1) = 0.5, \rho(x_2) = 0.5 \)
Nash 3. \( \rho(x_2) = 0.3, \rho(x_3) = 0.7 \)
The Competition

The Scarf algorithm, which is an algorithmic extension of Sperner's lemma, also computes approximate fixed points. In the Scarf algorithm: the given space is triangulated, each vertex of the triangulation is given a label, and a pivoting procedure is followed until one reaches a "completely labelled simplex."
The Competition

- The Scarf algorithm, which is an algorithmic extension of Sperner’s lemma, also computes approximate fixed points.
The Competition

• The Scarf algorithm, which is an algorithmic extension of Sperner’s lemma, also computes approximate fixed points.

• In the Scarf algorithm:
  • the given space is triangulated,
The Competition

• The **Scarf algorithm**, which is an algorithmic extension of Sperner’s lemma, also computes approximate fixed points.

• In the Scarf algorithm:
  • the given space is triangulated,
  • each vertex of the triangulation is given a label, and
The Competition

- The **Scarf algorithm**, which is an algorithmic extension of Sperner’s lemma, also computes approximate fixed points.
- In the Scarf algorithm:
  - the given space is triangulated,
  - each vertex of the triangulation is given a label, and
  - a pivoting procedure is followed until one reaches a “completely labelled simplex.”
Comparison

In comparison with the Scarf algorithm, our procedure has the following advantages:

1. Whereas the Scarf algorithm presumes that \( C \) is a simplex or a cartesian product of simplices, our procedure assumes only a compact convex set.

2. In the Scarf procedure and its variants the computer has to compute a triangulation of \( C \).

3. Our procedure automatically handles the “restart” problem by starting the search for \( x^m+1 \) at \( x^m \).

4. Our procedure sometimes mimics iteration of a local contraction.
Comparison

In comparison with the Scarf algorithm, our procedure has the following advantages:

- Whereas the Scarf algorithm presumes that \( C \) is a simplex or a cartesian product of simplices, our procedure assumes only a compact convex set.
Comparison

In comparison with the Scarf algorithm, our procedure has the following advantages:

- Whereas the Scarf algorithm presumes that $C$ is a simplex or a cartesian product of simplices, our procedure assumes only a compact convex set.
- In the Scarf procedure and its variants the computer has to compute a triangulation of $C$. 
Comparison

In comparison with the Scarf algorithm, our procedure has the following advantages:

- Whereas the Scarf algorithm presumes that $C$ is a simplex or a cartesian product of simplices, our procedure assumes only a compact convex set.
- In the Scarf procedure and its variants the computer has to compute a triangulation of $C$.
- Our procedure automatically handles the “restart” problem by starting the search for $x_{m+1}$ at $x_m$. 
Comparison

In comparison with the Scarf algorithm, our procedure has the following advantages:

- Whereas the Scarf algorithm presumes that $C$ is a simplex or a cartesian product of simplices, our procedure assumes only a compact convex set.
- In the Scarf procedure and its variants the computer has to compute a triangulation of $C$.
- Our procedure automatically handles the “restart” problem by starting the search for $x_{m+1}$ at $x_m$.
- Our procedure sometimes mimics iteration of a local contraction.