An Approximate Subgame-Perfect Equilibrium Computation Technique for Repeated Games

Andriy Burkov

Université Laval, Canada

July 15, 2010
Plan

Motivation

Game Theory Background

Problem and Approach

Conclusion and Future Work
Plan

Motivation

Game Theory Background

Problem and Approach

Conclusion and Future Work
Motivation

- Discover an algorithmic way for:
  - Finding equilibrium solutions for dynamic games
  - Computing equilibrium strategies for dynamic game players
**Motivation: Example**

- **Prisoner’s Dilemma**

<table>
<thead>
<tr>
<th>Player 1</th>
<th>Player 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>C</strong></td>
<td>2, 2</td>
</tr>
<tr>
<td><strong>D</strong></td>
<td>4, −1</td>
</tr>
</tbody>
</table>

- When the discount factor is close enough to 1, the long-term average payoff profile (2, 2) is an equilibrium point and there is a strategy, which each player can adopt for generating that point: Tit-For-Tat

- For an arbitrary discount factor, we don’t usually know:
  - What is the set of equilibrium points?
  - What are the strategies of players that generate those equilibrium points?
Plan

Motivation

Game Theory Background

Problem and Approach

Conclusion and Future Work
A stage-game is a tuple \((N, \{A_i\}_{i \in N}, \{r_i\}_{i \in N})\):

- \(N\) is a finite set of players
- \(A_i\) is a finite set of pure actions of player \(i \in N\)
- \(r_i\) is the payoff function of player \(i\): \(r_i : A \mapsto \mathbb{R}\)
  - where \(A \equiv \times_{i \in N} A_i\) defines the set of action profiles

Example: Prisoner’s Dilemma
A stage-game is a tuple \((N, \{A_i\}_{i \in N}, \{r_i\}_{i \in N})\):
- \(N\) is a finite set of players
- \(A_i\) is a finite set of pure actions of player \(i \in N\)
- \(r_i\) is the payoff function of player \(i: r_i : A \mapsto \mathbb{R}\)
  - where \(A \equiv \times_{i \in N} A_i\) defines the set of action profiles

Example: Prisoner’s Dilemma

\[
\begin{array}{c|cc}
 & C & D \\
\hline
C & 2, 2 & -1, 4 \\
D & 4, -1 & 0, 0 \\
\end{array}
\]

\(N = \{1, 2\},\)
\(A_1 = A_2 = \{C, D\},\)
\(r_1(C, C) = 2, r_1(C, D) = -1, r_1(D, C) = 4, \ldots\)
Repeated games

- In an infinitely repeated game, a certain stage-game is repeatedly played by the same set of players during an \textit{a priori} unknown number of time-steps.
- There is a probability of $\gamma$ that the repeated game will continue after the current stage-game.
Repeated games

- In an infinitely repeated game, a certain stage-game is repeatedly played by the same set of players during an *a priori* unknown number of time-steps.
- There is a probability of $\gamma$ that the repeated game will continue after the current stage-game.

```
t=0  t=1  ...
```

...
Strategies

- The set of histories up to time-step $t$ of the repeated game is given by $H^t \equiv \times_t A$.
- The set of all possible histories is given by $H \equiv \bigcup_{t=0}^{\infty} H^t$ with $h \in H$ being a particular history.
- A mixed strategy of player $i$ is a mapping $\sigma_i : H \mapsto \Delta(A_i)$ with $\alpha_i \in \Delta(A_i)$ being a mixed action of player $i$. 

Andriy Burkov, Université Laval, Canada
An Approximate Subgame-Perfect Equilibrium Computation Technique for Repeated Games
Let $\sigma_i \in \Sigma_i$ be a strategy of player $i$

Let $\sigma \in \Sigma \equiv \times_i \Sigma_i$ be a strategy profile

An outcome path is a possibly infinite sequence $\vec{a} \equiv (a^0, a^1, \ldots)$ of action profiles

The discounted average payoff of $\sigma$ for player $i$ is defined as

$$u_i^\gamma(\sigma) \equiv (1 - \gamma) \mathbb{E}_{\vec{a} \sim \sigma} \sum_{t=0}^\infty \gamma^t r_i(a^t),$$

The discount factor can be seen as a patience of players: higher it is, more important are future payoffs

A Nash equilibrium is defined as strategy profile $\sigma \equiv (\sigma_i, \sigma_{-i})$ such that for each player $i$ and for every $\sigma'_i \in \Sigma_i$:

$$u_i^\gamma(\sigma) \geq u_i^\gamma(\sigma'_i, \sigma_{-i})$$
A subgame is a repeated game which continues after a certain history.

For a pair \((\sigma, h)\), the subgame strategy profile induced by \(h\) is denoted as \(\sigma|_h\).

A strategy profile \(\sigma\) is a subgame-perfect equilibrium (SPE) in a repeated game, if for all histories \(h \in H\), the subgame strategy profile \(\sigma|_h\) is a Nash equilibrium in the subgame.
Let be a stage-game:

\[
\begin{array}{c|cc}
\text{Player 1} & C & D \\
\hline
C & r(C,C) & r(C,D) \\
D & r(D,C) & r(D,D) \\
\end{array}
\]

Given a strategy profile \( \sigma \), after any history \( h^t \), one can represent an (infinite) subgame as an augmented stage-game:

\[
\begin{array}{c|cc}
\text{Player 1} & C & D \\
\hline
C & (1-\gamma)r(C,C) + \gamma u^\gamma(\sigma|_{h^t,(C,C)}) & (1-\gamma)r(C,D) + \gamma u^\gamma(\sigma|_{h^t,(C,D)}) \\
D & (1-\gamma)r(D,C) + \gamma u^\gamma(\sigma|_{h^t,(D,C)}) & (1-\gamma)r(D,D) + \gamma u^\gamma(\sigma|_{h^t,(D,D)}) \\
\end{array}
\]

The strategy profile \( \sigma \) is called subgame perfect equilibrium if it induces a Nash equilibrium in each augmented stage-game.
Motivation

Game Theory Background

Problem and Approach

Conclusion and Future Work
Problem and Approach

- **Problem:** Given a discount factor $\gamma$ and payoff functions of players, find the set of SPE entirely or partially.

- **Previous work includes:**
  - All works on computing stage-game equilibria (ex: Lemke & Howson (1965), Porter et al. (2004))
  - Littman & Stone (2004): only for average payoff (i.e., $\gamma = 1$)
  - Judd et al. (2003): arbitrary $\gamma$ but only pure action equilibria

- **Our approach:** dynamic programming over the set of equilibrium payoff profiles
  - Permits computing SPE for an arbitrary $\gamma$, including pure and mixed action equilibria
  - Based on two ideas: *self-generating sets* and *partitioning of hypercubes*
Let \( BR_i(\alpha) \) be a best response of player \( i \) in a stage-game to the mixed action profile \( \alpha \equiv (\alpha_i, \alpha_{-i}) \):

\[
BR_i(\alpha) \equiv \max_{a_i \in A_i} r_i(a_i, \alpha_{-i}).
\]

We define the map \( B^\gamma \) on a set \( W \subset \mathbb{R}^{|N|} \) as

\[
B^\gamma(W) \equiv \bigcup_{(\alpha, w) \in \times_{i \in N} \Delta(A_i) \times W} (1 - \gamma)r(\alpha) + \gamma w,
\]

\( w \) is a \textit{continuation promise} which verifies for all \( i \in N \):

\[
(1 - \gamma)r_i(\alpha) + \gamma w_i - (1 - \gamma)r_i(BR_i(\alpha), \alpha_{-i}) - \gamma w_i \geq 0,
\]

\( w_i \equiv \inf_{w \in W} w_i \)

The largest fixed point of \( B^\gamma(W) \) is the set of all SPE in the repeated game (Abreu, 1990)
Recall the two self-generation equations:

\[ B^\gamma(W) \equiv \bigcup_{(\alpha,w) \in \times_{i \in N} \Delta(A_i) \times W} (1 - \gamma)r(\alpha) + \gamma w \]  

\[ (1 - \gamma)r_i(\alpha) + \gamma w_i - (1 - \gamma)r_i(BR_i(\alpha), \alpha_{-i}) - \gamma w_i \geq 0 \ \forall i \]  

Equation (1) promises to player \( i \in N \) a better payoff tomorrow to compensate a possible today’s loss if player \( i \) follows a given strategy.

Equation (2) guarantees to player \( i \) a sufficient punishment imposed by the other players if player \( i \) deviates from the given strategy.
Our algorithm starts with an initial approximation $W$ of the set of SPE payoff profiles.

The set $W$, in turn, is represented by a union of disjoint hypercubes belonging to the set $C$.

Initially, the set $C$, contains only one hypercube that contains all possible payoff profiles.

Each iteration of the algorithm consists of verifying, for each hypercube $c \in C$, whether it has to be withdrawn.
Updates by hypercubes: Example

Payoffs of Player 1
Payoffs of Player 2

Andriy Burkov, Université Laval, Canada

An Approximate Subgame-Perfect Equilibrium Computation Technique for Repeated Games
Updates by hypercubes: Example
Updates by hypercubes: Example
Updates by hypercubes: Example
$w = (1 - \gamma)r(\alpha) + \gamma w'$

![Diagram showing payoffs for Player 1 and Player 2 with the equation $w = (1 - \gamma)r(\alpha) + \gamma w'$]
Updates by hypercubes: Example

\[ w = (1 - \gamma) r(\alpha) + \gamma w' \]
Updates by hypercubes: Example

\[ w = (1 - \gamma)r(\alpha) + \gamma w' \]
\[ w_i - (1 - \gamma)r_i(BR_i(\alpha), \alpha_{-i}) - \gamma w_i \geq 0, \ \forall i \]
Updates by hypercubes: Example
Updates by hypercubes: Example

Payoffs of Player 1

Payoffs of Player 2
Updates by hypercubes: Example
If, after having tested all hypercubes in $C$, we haven’t withdrawn any hypercube, we partition each remaining hypercube on a number of smaller hypercubes

- We retest the remaining hypercubes the same way
- This permits improving the precision of approximation of the set of equilibria

The algorithm terminates when the required precision is achieved
Partitioning the hypercubes: Example
**Partitioning the hypercubes: Example**

<table>
<thead>
<tr>
<th>Payoffs of Player 1</th>
<th>Payoffs of Player 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Partitioning the hypercubes: Example
The Main Theorem

**Theorem**

For any repeated game, any discount factor $\gamma$ and for any level of approximation, (i) Our algorithm terminates in finite time, (ii) the set of hypercubes $C$, at any moment, contains at least one hypercube, (iii) for any input $v \in W$, the algorithm returns a strategy profile (represented by a finite automaton) that satisfies the required approximation properties.
Example: The Prisoner’s Dilemma

Player 1

<table>
<thead>
<tr>
<th></th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>2, 2</td>
<td>-1, 4</td>
</tr>
<tr>
<td>D</td>
<td>4, -1</td>
<td>0, 0</td>
</tr>
</tbody>
</table>

\[ \gamma = 0.7 \]
The Prisoner’s Dilemma

Iteration 1
The Prisoner’s Dilemma

Iteration 4
The Prisoner’s Dilemma

Iteration 8
The Prisoner’s Dilemma

Iteration 12
Iteration 20
The Prisoner’s Dilemma

Diagram showing the payoffs for the Prisoner’s Dilemma with the four possible outcomes:
- \(r(D, D)\)
- \(r(D, C)\)
- \(r(C, D)\)
- \(r(C, C)\)

Iteration 30
The Prisoner’s Dilemma

Iteration 50
Plan

Motivation

Game Theory Background

Problem and Approach

Conclusion and Future Work
Conclusion and Future Work

- We proposed an algorithmic approach for approximating the set of subgame-perfect equilibrium payoff profiles in repeated games.
- Our algorithm is capable of computing a profile of player strategies that approximately induces any given SPE point.
- Future work will aim at extending the proposed approach for solving more complex dynamic games such as Markov chain games and stochastic games.
Thank you!
Another Example: Battle of the Sexes ($\gamma = 0.45$)

Stage-game equilibrium payoff profiles:

- (1, 2)
- (2, 1)
- (2/3, 2/3)
Example: Repeated Battle of the Sexes
Example: Repeated Battle of the Sexes
Example: Repeated Battle of the Sexes
Example: Repeated Battle of the Sexes
Example: Repeated Battle of the Sexes
Example: Repeated Battle of the Sexes
Example: Repeated Battle of the Sexes
Example: Repeated Battle of the Sexes
Example: Repeated Battle of the Sexes
Example: Repeated Battle of the Sexes
Example: Repeated Battle of the Sexes
Example: Repeated Battle of the Sexes
Example: Repeated Battle of the Sexes
Let $M \equiv (Q, q^0, f, \tau)$ be an automaton implementation of a strategy profile $\sigma$ where

- $Q$, set of automaton states with $q^0 \in Q$ being the initial state
- $f \equiv (f_i)_{i \in N}$, where $f_i : Q \mapsto \Delta(A_i)$, la fonction de décision du joueur $i$
- $\tau : Q \times A \mapsto Q$, une fonction de transition

**Theorem (Kalai and Stanford, 1988)**

Any SPE can be approximated by a finite automaton