Exclusion Method for Finding Nash Equilibrium in Multiplayer Games

Kimmo Berg
Department of Mathematics and Systems Analysis
Aalto University, Finland

(joint with Tuomas Sandholm)
(Carnegie Mellon University)

July 28, 2016
Outline of the presentation

- Introduction and motivation
- Earlier literature
  - Classification of methods
  - Computational complexity
- Exclusion method
  - Exclusion oracle tells if Nash equilibrium is NOT in the region
  - Subdivision scheme and region selection important
- Numerical results
Introduction

Normal-form game with $n$ players and $m$ actions
Nash equilibrium $p^*$: no player can gain by deviating
$\epsilon$-equilibrium: $u_i(a, p^*_{-i}) \leq u_i(p^*) + \epsilon$, $\forall i, a \in A_i$
How do you compute an (approximative) equilibrium?
Introduction (2)

- Two-player vs. multiplayer games
- Multiplayer games – nonlinear polynomial equations
- Correlated equilibrium? Zero-sum game?
- Find one vs. all equilibria
- Root of regret $r(p) = 0$; piecewise differentiable polynomial
- Regret of action $a$: $r_i(a, p) = u_i(a, p_{-i}) - u_i(p)$
  Regret of player $i$: $r_i(p) = \max_{a \in A_i} r_i(a, p)$
  Regret in the game: $r(p) = \max_i r_i(p)$
Classification of methods

- **Homotopy (path-following) methods**: trace equilibrium from easy, artificial game to the original game. Govindan and Wilson 2003/4, Herings and Peeters 2005, Turocy 2005, Lemke and Howson 1964

- **Polynomial equation solving and support enumeration**: Porter et al. 2008, Lipton and Markakis 2004

- **Function minimization and optimization formulations**: Sandholm et al. 2005, Chatterjee 2009, Buttler and Akchurina 2013, Borycka and Juszczuk 2013

- **Simplicial subdivision methods**: van der Laan, Talman, van der Heyden 1970-80s

## Gambit algorithms on GAMUT games

**Computation times (sec) and instances not solved (percentage)**

<table>
<thead>
<tr>
<th>Game class</th>
<th>gnm</th>
<th>ipa</th>
<th>enumpoly</th>
<th>simpdiv</th>
<th>liap</th>
<th>logit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bertrand oligopoly</td>
<td>0.05 (30)</td>
<td>0.05 (75)</td>
<td>0.04 (50)</td>
<td>0.04 (0.4)</td>
<td>0.24 (99)</td>
<td>0.06</td>
</tr>
<tr>
<td>Bidirectional LEG</td>
<td>0.09 (0.3)</td>
<td>0.05 (58)</td>
<td>0.84 (1)</td>
<td>0.04 (2)</td>
<td>0.24 (99)</td>
<td>0.06 (0.1)</td>
</tr>
<tr>
<td>Collaboration</td>
<td>0.24 (0.1)</td>
<td>0.04</td>
<td>3.3 (50)</td>
<td>0.05</td>
<td>0.34 (99)</td>
<td>0.06 (0.3)</td>
</tr>
<tr>
<td>Congestion</td>
<td>0.05 (0.2)</td>
<td>0.05 (85)</td>
<td>0.05 (0.6)</td>
<td>0.04 (0.7)</td>
<td>0.21 (100)</td>
<td>0.05</td>
</tr>
<tr>
<td>Coordination</td>
<td>0.24 (2)</td>
<td>0.05</td>
<td>27 (8)</td>
<td>0.04</td>
<td>0.37 (99)</td>
<td>0.05 (0.3)</td>
</tr>
<tr>
<td>Covariant r=0.9</td>
<td>0.19 (1)</td>
<td>0.06 (87)</td>
<td>39</td>
<td>0.04 (20)</td>
<td>0.31 (99)</td>
<td>0.06 (0.3)</td>
</tr>
<tr>
<td>Covariant r=-0.5</td>
<td>0.13 (3)</td>
<td>0.05 (94)</td>
<td>36</td>
<td>0.04 (20)</td>
<td>0.31 (100)</td>
<td>0.05 (1)</td>
</tr>
<tr>
<td>Dispersion</td>
<td>1.18 (1)</td>
<td>0.04</td>
<td>10</td>
<td>0.04</td>
<td>0.44 (93)</td>
<td>0.05</td>
</tr>
<tr>
<td>Majority voting</td>
<td>0.77 (25)</td>
<td>0.05</td>
<td>0.32</td>
<td>0.04</td>
<td>0.24 (100)</td>
<td>0.06 (1)</td>
</tr>
<tr>
<td>Minimum effort</td>
<td>0.06</td>
<td>0.04</td>
<td>1.7</td>
<td>0.04</td>
<td>0.26 (98)</td>
<td>0.05 (0.1)</td>
</tr>
<tr>
<td>N player chicken (*)</td>
<td>0.05 (0.2)</td>
<td>0.04 (52)</td>
<td>0.04</td>
<td>0.04</td>
<td>0.07 (67)</td>
<td>0.05 (0.2)</td>
</tr>
<tr>
<td>N player PD (*)</td>
<td>0.04</td>
<td>0.04 (99)</td>
<td>0.04</td>
<td>0.04</td>
<td>0.05 (22)</td>
<td>0.04</td>
</tr>
<tr>
<td>Polymatrix</td>
<td>0.06 (1)</td>
<td>0.04 (79)</td>
<td>0.04 (50)</td>
<td>0.06 (0.4)</td>
<td>0.3 (92)</td>
<td>0.05 (0.4)</td>
</tr>
<tr>
<td>Random compound (*)</td>
<td>0.04</td>
<td>0.04 (37)</td>
<td>0.05</td>
<td>0.04</td>
<td>0.08 (56)</td>
<td>0.05</td>
</tr>
<tr>
<td>Random LEG</td>
<td>0.05 (1)</td>
<td>0.04 (59)</td>
<td>8.1 (2)</td>
<td>0.05 (4)</td>
<td>0.24 (99)</td>
<td>0.06</td>
</tr>
<tr>
<td>Random graphical</td>
<td>0.08 (3)</td>
<td>0.04 (96)</td>
<td>6.3 (6)</td>
<td>0.10 (17)</td>
<td>0.31 (99)</td>
<td>0.06 (0.3)</td>
</tr>
<tr>
<td>Traveler’s dilemma</td>
<td>0.04</td>
<td>0.06</td>
<td>1.1</td>
<td>0.04</td>
<td>0.33 (98)</td>
<td>0.06</td>
</tr>
<tr>
<td>Uniform LEG</td>
<td>0.07 (0.4)</td>
<td>0.05 (55)</td>
<td>0.04 (17)</td>
<td>0.04 (10)</td>
<td>0.23 (99)</td>
<td>0.06</td>
</tr>
</tbody>
</table>

**NO Gambit algorithm can solve ALL instances.**
Earlier results

- Computing Nash is PPAD-complete in two-player general-sum games (Chen, Deng, Teng 2006/9)
- So is approximative equilibrium (Daskalakis 2013, Rubinstein 2016)
- Polynomial Parity Arguments on Directed graphs (Papadimitriou 1991)
- PPAD is believed to be hard
- Computing (approximative) Nash is FIXP-complete in multiplayer games (Etessami and Yannakakis 2010)
Earlier results: uniform strategies

- \(\epsilon\)-equilibrium in “small” supports using \(k\)-uniform strategies
- \(k\)-uniform: probabilities are all with denominator \(k\)
- Babichenko et al. 2014: \(k = O((\log m + \log n - \log \epsilon)/\epsilon^2)\)
- Number of profiles: \(m^{nk}\) and \((k+1)^{nm}\)
- If \(n = m = 3\), \(\epsilon = 10^{-3}\), \(k > 10^7\) and \(10^{42}\) points
- \(O(m^{\log m}), O((\log n)^n), O(((\log 1/\epsilon)/\epsilon^2)^c)\) (best in \(m\))
- We improve \(n\) and \(\epsilon\): \(O(c^n), O(1/\epsilon^c), c\) constant (best in \(n\))
Earlier results: solving algebraic equations

- Lipton and Markakis 2004: algebraic numbers and finite representation
- Not only approximative but close to actual Nash equilibrium
- Polynomial in $\log 1/\epsilon$, $n^{nm}$, $L$ (best in $\epsilon$)
- $L$ is maximum bit size of payoff data
Main idea behind our method: exclusion of regions

- For any point with positive regret, the solution cannot be near this point
- Based on the function being continuous and having maximum value of derivative
How to determine the maximum derivative ($M$) of piecewise polynomial?

**Theorem**

$p^0$-centered ball of radius $s$ cannot contain 0-Nash if $r_i(p^0) > s \cdot M_i$ for some $i$

**Theorem**

If $r_i(p^0) \geq \epsilon$, for some $i \in N$, then region size $d < \epsilon/2M_i$ is small enough to exclude $p^0$. 
Subdivision scheme

- Exclude balls? Remaining regions difficult to keep track
- How to encode the regions? Simplexes?
- We use hyperrectangles (boxes)
- Easy to store min and max values in each dimension
- Split using bisection, divide along the longest edge
Region selection heuristic

- Select a region that is likely to contain Nash
- Compute ranking function based on available function values
  - We use \( g(R, p^0) = \max_i r_i(p^0)/(d(R) \cdot M_i(p^0)) \)
  - \( R \) region, \( d \) its diameter, \( M_i(p^0) \) maximum derivative of regret
- Favor big regions with low regret and big derivatives
Exclusion method using bisection

Repeat until $\epsilon$-Nash found

1. Select the box with minimal value of ranking function $g$

2. Compute regret $r(p^0)$. If regret small enough, $\epsilon$-Nash found. Else either exclude the box (regret is large), or bisect it along the longest edge.
Computational complexity

- $O(c^n)$, $O(c^m)$, $O(1/\epsilon^c)$
- Exponential both in $n$ and $m$

**Theorem**

*Any bisection method excludes all points with $r(p) \geq \epsilon$ within $2^{(m-1)n} \lceil \log_2 \frac{2M^*}{\epsilon} \rceil$ iterations.*
## Our method vs. enumeration of k-uniform profiles

<table>
<thead>
<tr>
<th>Game class</th>
<th>Time (sec)</th>
<th>95% bound</th>
<th>Time Alg. 2</th>
<th>NS (%)</th>
<th>NS Time</th>
<th>NS $\epsilon$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bertrand oligopoly</td>
<td>13.7</td>
<td>19.3</td>
<td>0.01</td>
<td>0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Bidirectional LEG</td>
<td>159</td>
<td>337</td>
<td>0.013</td>
<td>0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Collaboration</td>
<td>2.8</td>
<td>3.7</td>
<td>0.0009</td>
<td>0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Congestion</td>
<td>29</td>
<td>71</td>
<td>0.027</td>
<td>0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Coordination</td>
<td>1.6</td>
<td>2.3</td>
<td>0.0009</td>
<td>0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Covariant $r=0.9$</td>
<td>5.5</td>
<td>8.4</td>
<td>0.006</td>
<td>0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Covariant $r=-0.5$</td>
<td>95</td>
<td>202</td>
<td>80</td>
<td>16</td>
<td>434</td>
<td>0.003</td>
</tr>
<tr>
<td>Dispersion</td>
<td>31</td>
<td>52</td>
<td>0.01</td>
<td>0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Majority voting</td>
<td>5.6</td>
<td>15.6</td>
<td>0.0008</td>
<td>0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Minimum effort</td>
<td>0.014</td>
<td>0.015</td>
<td>0.0008</td>
<td>0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>N player chicken (*)</td>
<td>0.016</td>
<td>0.018</td>
<td>0.0008</td>
<td>0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>N player PD (*)</td>
<td>0.005</td>
<td>0.005</td>
<td>0.0008</td>
<td>0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Polymatrix</td>
<td>172</td>
<td>358</td>
<td>27.2</td>
<td>7</td>
<td>373</td>
<td>0.003</td>
</tr>
<tr>
<td>Random compound (*)</td>
<td>0.014</td>
<td>0.015</td>
<td>0.001</td>
<td>0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Random LEG</td>
<td>880</td>
<td>1970</td>
<td>0.02</td>
<td>0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Traveler's dilemma</td>
<td>0.01</td>
<td>0.01</td>
<td>0.008</td>
<td>0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Uniform LEG</td>
<td>793</td>
<td>1850</td>
<td>0.02</td>
<td>0</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Our method is the only one to solve all instances – slowly but surely.
Dependency in $\epsilon$ in random games, 3/4/5-player games.
Conclusion

- Computation of equilibrium is difficult
- Fast algorithms and complete algorithms are different
- New approach for computing equilibrium
- Best upper bound in number of players $n$
- Development of new exclusion oracles, subdivision schemes and ranking functions
- Better bounds for derivatives of polynomials (e.g., Markov inequality 1889)
- Hybrid schemes using different methods together
Remember to live without regret...

Thank you for your attention! Any questions?