An Introduction to Game Theory
Part III: Strictly Competitive Games
Bernhard Nebel
Strictly Competitive Games

- A *strictly competitive* or zero-sum game is a 2-player strategic game such that for each $a \in A$, we have $u_1(a) + u_2(a) = 0$.
  - What is good for me, is bad for my opponent and *vice versa*

- **Note:** Any game where the sum is a constant $c$ can be transformed into a zero-sum game with the same set of equilibria:
  - $u'_1(a) = u_1(a)$
  - $u'_2(a) = u_2(a) - c$
How to Play Zero-Sum Games?

- Assume that only pure strategies are allowed
  - Dominating strategy?
  - Nash equilibrium?
- Be paranoid: Try to minimize your loss by assuming the worst!
- Player 1 takes minimum over row values:
  - T: -6, M: -1, B: -6
- then maximizes:
  - M: -1

<table>
<thead>
<tr>
<th></th>
<th>L</th>
<th>M</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>8,-8</td>
<td>3,-3</td>
<td>-6,6</td>
</tr>
<tr>
<td>M</td>
<td>2,-2</td>
<td>-1,1</td>
<td>3,-3</td>
</tr>
<tr>
<td>B</td>
<td>-6,6</td>
<td>4,-4</td>
<td>8,-8</td>
</tr>
</tbody>
</table>
Maximinimizer

• An action $x^*$ is called *maximinimizer* for player 1, if
  
  \[ \min_{y \in A_2} u_1(x^*, y) \geq \min_{y \in A_2} u_1(x, y) \quad \text{for all } x \in A_1 \]

• Similar for player 2

• Maximinimizer try to minimize the loss, but do not necessarily lead to a Nash equilibrium.

• However, if a NE exists, then the action profile is a pair of maximiminizers!
Maximinimizer Theorem

In strictly competitive games:

1. If \((x^*, y^*)\) is a Nash equilibrium of \(G\) then \(x^*\) is a maximinimizer for player 1 and \(y^*\) is a maximinimizer for player 2.

2. If \((x^*, y^*)\) is a Nash equilibrium of \(G\) then
   \[
   \max_x \min_y u_1(x,y) = \min_y \max_x u_1(x,y) = u_1(x^*, y^*). 
   \]

3. If \(\max_x \min_y u_1(x,y) = \min_y \max_x u_1(x,y)\) and \(x^*\) is a maximinimizer for player 1 and \(y^*\) is a maximinimizer for player 2, then \((x^*, y^*)\) is a Nash equilibrium.
Some Consequences

- Because of (2): if \((x^*, y^*)\) is a NE then \(\max_x \min_y u_1(x, y) = u_1(x^*, y^*)\), all NE yield the same payoff
  - it is irrelevant which we choose.

- Because of (2), if \((x^*, y^*)\) and \((x', y')\) are a NEs then \(x^*, x'\) are maximinimizers for player 1 and \(y^*, y'\) are maximinimizers for player 2. Because of (3), then \((x^*, y')\) and \((x', y^*)\) are NEs as well!
  - it is not necessary to coordinate in order to play in a NE!
Example

- Minimum in rows (for player 1):
  - T: -6, M: -1, B: -6
- Maximinimizer:
  - M: -1
- Maximum over columns (for player 1)
  - L: 8, M: -1, R: 8
- Minimaximizer:
  - M: -1
- Also NE, apparently

<table>
<thead>
<tr>
<th></th>
<th>L</th>
<th>M</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>8,-8</td>
<td>-3,3</td>
<td>-6,6</td>
</tr>
<tr>
<td>M</td>
<td>2,-2</td>
<td>-1,1</td>
<td>3,-3</td>
</tr>
<tr>
<td>B</td>
<td>-6,6</td>
<td>-4,4</td>
<td>8,-8</td>
</tr>
</tbody>
</table>
How to Find NEs in Mixed Strategies?

- While it is non-trivial to find NEs for general sum games, *zero-sum* games are “easy.”
- Let’s test all mixed strategies of player 1 $\alpha_1$ against all mixed strategies of player 2 $\alpha_2$. Then use only those that are maximinimizers.
- Since all mixed strategies are linear combinations of pure strategies, it is enough to check against the pure strategies of player 2 (support theorem).
- We just have to optimize, i.e., find the best mixed strategy
  - Use linear programming
Linear Programming: The Idea

• The article-mix problem:
  – article 1 needs: 25 min of cutting, 60 min of assembly, 68 min of postprocessing
    • results in 30 Euro profit per article
  – article 2 needs: 75 min of cutting, 60 min of assembly, and 34 min of postprocessing
    • results in 40 Euro profit per article
  – per day: 450 min of cutting, 480 min of assembly and 476 min of postprocessing

• Try to maximize profit
Resulting Constraints & Optimization Goals

- \( x: \#\text{article1}, y: \#\text{article2} \)
- \( x \geq 0, y \geq 0 \)
- \( 25x + 75y \leq 450 \) (cutting)
  - \( y \leq 6 - (1/3 \cdot x) \)
- \( 60x + 60y \leq 480 \) (assembly)
  - \( y \leq 8 - x \)
- \( 68x + 34y \leq 476 \) (postprocessing)
  - \( y \leq 14 - 2x \)
- **Maximize** \( z = 30x + 40y \)
The inequalities describe convex sets in $R^2$.
The intersection of all convex sets represents the set of feasible solutions.
Each point in the set of feasible solutions could get a quality measure according to the objective function.
Consider lines of equal quality and then do hill climbing!
Linear Programming: The Standard Form

- $n$ real-valued variables $x_i \geq 0$
- $n$ coefficients $b_i$ and $m$ constants $c_j$
- $m \cdot n$ coefficients $a_{ij}$
- $m$ equations $\sum_i a_{ij} x_i = c_j$
- Objective function: $\sum_i b_i x_i$ is to be minimized
- Can be solved by the simplex method
  - lpsolve for example
Other Forms

- **Maximization** instead of minimization:
  - set $b'_i = -b_i$

- **Inequalities**
  - introduce slack (non-negative) variables $z_i$:
    - $\sum a_{ij} x_i \leq c_j \iff \sum a_{ij} x_i + z_i = c_j$

- **Larger or equal**
  - Multiply both sides with -1
Solving Zero-Sum Games

• Let $A_1 = \{a_{11}, \ldots, a_{1n}\}$, $A_2 = \{a_{21}, \ldots, a_{2m}\}$,

• Player 1 looks for a mixed strategy $\alpha_1$
  - $\sum_j \alpha_1(a_{1j}) = 1$
  - $\alpha_1(a_{1j}) \geq 0$
  - $\sum_j \alpha_1(a_{1j}) \cdot u_1(a_{1j}, a_{2i}) \geq u$ for all $i \in \{1, \ldots, m\}$
  - Maximize $u$!

• Similarly for player 2.
Conclusion

- **Zero-sum** games are particularly simple.
- Playing a pure **maximinizing** strategy minimizes loss (for pure strategies).
- If **NE exists**, it is a pair of **maximinizers**.
- **NEs** can be freely "**mixed**".
- In mixed strategies, **NEs** always exist.
- Can be determined by **linear programming**.