ECE700.07: Game Theory with Engineering Applications

Lecture 4: Computing Solution Concepts of Normal Form Games

Seyed Majid Zahedi
Outline

• Brief overview of (mixed integer) linear programs

• Solving for
  • Dominated strategies
  • Minimax and maximin strategies
  • Nash equilibrium
  • Correlated NE

• Readings:
  • MAS Appendix B, and Sec. 4
Linear Program Example: Reproduction of Two Paintings

- Painting 1 sells for $30
- Painting 2 sells for $20
- We have 16 units of blue, 8 green, 5 red
- Painting 1 requires 4 blue, 1 green, 1 red
- Painting 2 requires 2 blue, 2 green, 1 red

Maximize: \[ 3x + 2y \]
Subject to: \[ 4x + 2y \leq 16 \]
\[ x + 2y \leq 8 \]
\[ x + y \leq 5 \]
\[ x \geq 0 \]
\[ y \geq 0 \]
Solving Linear Program Graphically

\[
\begin{align*}
\text{max.} & \quad 3x + 2y \\
\text{s.t.} & \quad 4x + 2y \leq 16 \\
& \quad x + 2y \leq 8 \\
& \quad x + y \leq 5 \\
& \quad x \geq 0 \\
& \quad y \geq 0
\end{align*}
\]

Optimal solution: \( x = 3, y = 2 \) (objective: 13)
Modified LP

\[
\begin{align*}
\text{max.} & \quad 3x + 2y \\
\text{s.t.} & \quad 4x + 2y \leq 15 \\
& \quad x + 2y \leq 8 \\
& \quad x + y \leq 5 \\
& \quad x \geq 0 \\
& \quad y \geq 0
\end{align*}
\]

- Optimal solution: \( x = 2.5, y = 2.5 \)
- Objective = \( 7.5 + 5 = 12.5 \)
- Can we sell half paintings?
Integer Linear Program

\[
\begin{align*}
\text{max.} & \quad 3x + 2y \\
\text{s.t.} & \quad 4x + 2y \leq 15 \\
& \quad x + 2y \leq 8 \\
& \quad x + y \leq 5 \\
& \quad x \in \mathbb{N}_{0} \\
& \quad y \in \mathbb{N}_{0}
\end{align*}
\]

Optimal LP solution: \( x = 2.5, y = 2.5 \) (objective 12.5)

Optimal ILP solution: \( x = 2, y = 3 \) (objective 12)
Mixed Integer Linear Program

\[
\begin{align*}
\text{max.} & \quad 3x + 2y \\
\text{s.t.} & \quad 4x + 2y \leq 15 \\
& \quad x + 2y \leq 8 \\
& \quad x + y \leq 5 \\
& \quad x \geq 0 \\
& \quad y \in \mathbb{N}_0
\end{align*}
\]
Solving Mixed Linear/Integer Programs

• Linear programs can be solved efficiently
  • Simplex, ellipsoid, interior point methods, etc.

• (Mixed) integer programs are \textbf{NP-hard} to solve
  • Many standard NP-complete problems can be modelled as MILP
  • Search type algorithms such as branch and bound

• Standard packages for solving these
  • Gurobi, MOSEK, GNU Linear Programming Kit, CPLEX, CVXPY, etc.

• LP relaxation of (M)ILP: remove integrality constraints
  • Gives upper bound on MILP (~admissible heuristic)
Exercise 1 in Modeling: Knapsack-type Problem

• We arrive in room full of precious objects
• Can carry only 30kg out of the room
• Can carry only 20 liters out of the room
• Want to maximize our total value
• Unit of object A: 16kg, 3 liters, sells for $11 (3 units available)
• Unit of object B: 4kg, 4 liters, sells for $4 (4 units available)
• Unit of object C: 6kg, 3 liters, sells for $9 (1 unit available)
• What should we take?
We want to have a working phone in every continent (besides Antarctica) but we want to have as few phones as possible.

- Phone A works in NA, SA, Af
- Phone B works in E, Af, As
- Phone C works in NA, Au, E
- Phone D works in SA, As, E
- Phone E works in Af, As, Au
- Phone F works in NA, E
Exercise III in Modeling: Hot-dog Stands

• We have two hot-dog stands to be placed in somewhere along beach
• We know where groups of people who like hot-dogs are
• We also know how far each group is willing to walk
• Where do we put our stands to maximize #hot-dogs sold? (price is fixed)

<table>
<thead>
<tr>
<th>Group 1</th>
<th>Group 2</th>
<th>Group 3</th>
<th>Group 4</th>
<th>Group 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>location: 1</td>
<td>location: 4</td>
<td>location: 7</td>
<td>location: 9</td>
<td>location: 15</td>
</tr>
<tr>
<td>#customers: 2</td>
<td>#customers: 1</td>
<td>#customers: 3</td>
<td>#customers: 4</td>
<td>#customers: 3</td>
</tr>
<tr>
<td>willing to walk: 4</td>
<td>willing to walk: 2</td>
<td>willing to walk: 3</td>
<td>willing to walk: 3</td>
<td>willing to walk: 2</td>
</tr>
</tbody>
</table>
Checking for Strict Dominance by Mixed Strategies

• LP for checking if strategy \( t_i \) is strictly dominated by any mixed strategy

\[
\begin{align*}
\text{max.} \quad & \epsilon \\
\text{s.t.} \quad & \sum_{s_i \in S_i} p_{s_i} u_i(s_i, s_{-i}) \geq u_i(t_i, s_{-i}) + \epsilon, \quad \forall s_{-i} \in S_{-i} \\
& \sum_{s_i \in S_i} p_{s_i} = 1 \\
& p_{s_i} \geq 0, \quad \forall s_i \in S_i
\end{align*}
\]
Checking for Weak Dominance by Mixed Strategies

• LP for checking if strategy $t_i$ is weakly dominated by any mixed strategy

\[
\begin{align*}
\text{max.} \quad & \sum_{s_{-i} \in S_{-i}} \left( \left( \sum_{s_i \in S_i} p_{s_i} u_i(s_i, s_{-i}) \right) - u_i(t_i, s_{-i}) \right) \\
\text{s.t.} \quad & \sum_{s_i \in S_i} p_{s_i} u_i(s_i, s_{-i}) \geq u_i(t_i, s_{-i}), \quad \forall s_{-i} \in S_{-i} \\
\quad & \sum_{s_i \in S_i} p_{s_i} = 1 \\
\quad & p_{s_i} \geq 0, \quad \forall s_i \in S_i
\end{align*}
\]
Path Dependency of Iterated Dominance

• Iterated weak dominance is path-dependent
  • Sequence of eliminations may determine which solution we get (if any)

• Iterated strict dominance is path-independent:
  • Elimination process will always terminate at the same point
Two Computational Questions for Iterated Dominance

• 1. Can any given strategy be eliminated using iterated dominance?

• 2. Is there some path of elimination by iterated dominance such that only one strategy per player remains?

• For strict dominance (with or without dominance by mixed strategies), both can be solved in polynomial time due to path-independence
  - Check if any strategy is dominated, remove it, repeat

• For weak dominance, both questions are NP-hard (even when all utilities are 0 or 1), with or without dominance by mixed strategies
  [Conitzer, Sandholm 05], and weaker version proved by [Gilboa, Kalai, Zemel 93]
Minimax and Maximin Values

• Maximin strategy for agent $i$  (leading to maximin value for agent $i$)

\[
\arg\max_{\sigma_i} \min_{s_i} u_i(\sigma_i, s_i)
\]

• Minimax strategy of other agents  (leading to minimax value for agent $i$)

\[
\arg\min_{\sigma_i} \max_{s_i} u_i(s_i, \sigma_i)
\]
LP for Calculating Maximin Strategy and Value

\[
\begin{align*}
\text{max.} & \quad u \\
\text{s.t.} & \quad \sum_{s_i \in S_i} p_{s_i} u_i(s_i, s_{-i}) \geq u, \quad \forall s_{-i} \in S_{-i} \\
& \quad \sum_{s_i \in S_i} p_{s_i} = 1 \\
& \quad p_{s_i} \geq 0, \quad \forall s_i \in S_i
\end{align*}
\]

- Objective of this LP, \( u \), is maximin value of agent \( i \)
- Given \( p_{s_i} \), first constraint ensures that \( u \) is less than any achievable expected utility for any pure strategies of opponents
Minimax Theorem [von Neumann 1928]

• Each player’s NE utility in any finite, two-player, zero-sum game is equal to her maximin value and minimax value

\[
\max_{\sigma_i} \min_{s_i} u_i(\sigma_i, s_i) = \min_{\sigma_i} \max_{s_i} u_i(s_i, \sigma_i)
\]

• Minimax theorem does not hold with pure strategies only (example?)
Example

- What is maximin value of agent 1 with and without mixed strategies?
- What is minimax value of agent 1 with and without mixed strategies?
- What is NE of this game?
Solving NE of Two-Player, Zero-Sum Games

• Minimax value of agent 1

\[
\begin{align*}
\text{min.} & \quad u_1 \\
\text{s.t.} & \quad \sum_{s_2 \in S_2} p_{s_2} u_1(s_1, s_2) \leq u_1 \quad \forall s_1 \in S_1 \\
& \quad \sum_{s_2 \in S_2} p_{s_2} = 1 \\
& \quad p_{s_2} \geq 0, \quad \forall s_2 \in S_2
\end{align*}
\]

• Maximin value of agent 1

\[
\begin{align*}
\text{max.} & \quad u_1 \\
\text{s.t.} & \quad \sum_{s_1 \in S_1} p_{s_1} u_1(s_1, s_2) \geq u_1 \quad \forall s_2 \in S_2 \\
& \quad \sum_{s_1 \in S_1} p_{s_1} = 1 \\
& \quad p_{s_1} \geq 0, \quad \forall s_1 \in S_1
\end{align*}
\]

• NE is expressed as LP, which means equilibria can be computed in polynomial time
Maximin Strategy for General-Sum Games

• Agents could still play minimax strategy in general-sum games
  • I.e., pretend that the opponent is only trying to hurt you

• But this is not rational:

<table>
<thead>
<tr>
<th></th>
<th>Left</th>
<th>Right</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Up</strong></td>
<td>(0, 0)</td>
<td>(3, 1)</td>
</tr>
<tr>
<td><strong>Down</strong></td>
<td>(1, 0)</td>
<td>(2, 1)</td>
</tr>
</tbody>
</table>

• If A2 was trying to hurt A1, she would play Left, so A1 should play Down
• In reality, A2 will play Right (strictly dominant), so A1 should play Up
Hardness of Computing NE for General-Sum Games

• Complexity was open for long time
  • “together with factoring […] the most important concrete open question on the boundary of P today” [Papadimitriou STOC’01]

• Sequence of papers showed that computing any NE is PPAD-complete (even in 2-player games) [Daskalakis, Goldberg, Papadimitriou 2006; Chen, Deng 2006]

• All known algorithms require exponential time (in worst case)
Hardness of Computing NE for General-Sum Games (cont.)

- What about computing NE with specific property?
  - NE that is not Pareto-dominated
  - NE that maximizes expected social welfare (i.e., sum of all agents’ utilities)
  - NE that maximizes expected utility of given agent
  - NE that maximizes expected utility of worst-off player
  - NE in which given pure strategy is played with positive probability
  - NE in which given pure strategy is played with zero probability
  - …

- All of these are NP-hard (and the optimization questions are inapproximable assuming P != NP), even in 2-player games
  [Gilboa, Zemel 89; Conitzer & Sandholm IJCAI-03/GEB-08]
Search-Based Approaches (for Two-Player Games)

• We can use (feasibility) LP, if we know support $X_i$ of each player $i$'s mixed strategy
  • I.e., we know which pure strategies receive positive probability

\[
\begin{align*}
\text{find} & \quad (u_1, u_2) \\
\text{s.t.} & \quad p_{s_i} \geq 0, \quad \forall i, s_i \in S_i \\
& \quad \sum_{s_i \in S_i} p_{s_i} = 1, \quad \forall i \\
& \quad p_{s_i} = 0, \quad \forall i, s_i \in S_i/X_i \\
& \quad \sum_{s_{-i} \in S_{-i}} p_{s_{-i}} u_i(s_i, s_{-i}) = u_i, \quad \forall i, s_i \in X_i \\
& \quad \sum_{s_{-i} \in S_{-i}} p_{s_{-i}} u_i(s_i, s_{-i}) \leq u_i, \quad \forall i, s_i \in S_i/X_i
\end{align*}
\]

• Thus, we can search over possible supports, which is basic idea underlying methods in [Dickhaut & Kaplan 91; Porter, Nudelman, Shoham AAAI04/GEB08]
Solving for NE using MILP (for Two-Player Games)

[Sandholm, Gilpin, Conitzer AAAI05]

\[
\begin{align*}
\text{max.} \quad & \text{whatever you like (e.g., social welfare)} \\
\text{s.t.} \quad & p_{s_i} \geq 0, \quad \forall i, s_i \in S_i \\
& \sum_{s_i \in S_i} p_{s_i} = 1, \quad \forall i \\
& \sum_{s_i \in S_i} p_{s_i} u_i(s_{-i}, s_{-i}) = u_{s_i}, \quad \forall i, s_i \in S_i \\
& u_{s_i} \leq u_i, \quad \forall i, s_i \in S_i \\
& p_{s_i} \leq b_{s_i}, \quad \forall i, s_i \in S_i \\
& u_i - u_{s_i} \leq M(1 - b_{s_i}), \quad \forall i, s_i \in S_i \\
& b_{s_i} \in \{0, 1\}, \quad \forall i, s_i \in S_i
\end{align*}
\]

* $b_{s_i}$ is binary variable indicating if $s_i$ is in support of $i$’s mixed strategy, and $M$ is large number
Solving for Correlated Equilibrium using LP (N-Player Games!)

• Variables are now \( p_s \) where \( s \) is profile of pure strategies (i.e., outcome)

\[
\begin{align*}
\text{max.} & \quad \text{whatever you like (e.g., social welfare)} \\
\text{s.t.} & \quad \sum_{s_{-i} \in S_{-i}} p_s u_i(s) \geq \sum_{s_{-i} \in S_{-i}} p_s u_i(t_i, s_{-i}) \quad \forall i, s_i \in S_i, t_i \in S_i \\
& \quad \sum_{s \in S} p_s = 1 \\
& \quad p_s \geq 0, \\
& \quad \forall s \in S
\end{align*}
\]
Questions?
Acknowledgement

• This lecture is a slightly modified version of ones prepared by
  • Vincent Conitzer [Duke CPS 590.4]