

## Solving Normal-Form Games

Branislav Bošanský

Artificial Intelligence Center,  
Department of Computer Science,  
Faculty of Electrical Engineering,  
Czech Technical University in Prague  
*branislav.bosansky@agents.fel.cvut.cz*

October 12, 2021

Previously ... on computational game theory.

- 1 Formal definition of a game  $\mathcal{G} = (\mathcal{N}, \mathcal{A}, u)$ 
  - $\mathcal{N}$  – a set of players
  - $\mathcal{A}$  – a set of actions
  - $u$  – outcome for each combination of actions
- 2 Pure/Mixed strategies
- 3 Dominance of strategies
- 4 Nash equilibrium

Today, you should learn ...

- 1 How to solve (compute a NE in) a normal form game (NFG)?
- 2 How to compute other equilibria in NFGs?
- 3 Key differences between different solution concepts.

## Existence of Nash equilibria?

	<b>C</b>	<b>D</b>
<b>C</b>	$(-1, -1)$	$(-5, 0)$
<b>D</b>	$(0, -5)$	$(-3, -3)$

	<b>R</b>	<b>P</b>	<b>S</b>
<b>R</b>	$(0, 0)$	$(-1, 1)$	$(1, -1)$
<b>P</b>	$(1, -1)$	$(0, 0)$	$(-1, 1)$
<b>S</b>	$(-1, 1)$	$(1, -1)$	$(0, 0)$

### Theorem (Nash)

*Every game with a finite number of players and action profiles has at least one Nash equilibrium in mixed strategies.*

# Support of Nash Equilibria

## Definition (Support)

The *support* of a mixed strategy  $s_i$  for a player  $i$  is the set of pure strategies  $\text{Supp}(s_i) = \{a_i | s_i(a_i) > 0\}$ .

## Question

Assume Nash equilibrium  $(s_i, s_{-i})$  and let  $a_i \in \text{Supp}(s_i)$  be an (arbitrary) pure strategy from the support of  $s_i$ . Which of the following possibilities can hold?

- 1  $u_i(a_i, s_{-i}) < u_i(s_i, s_{-i})$
- 2  $u_i(a_i, s_{-i}) = u_i(s_i, s_{-i})$
- 3  $u_i(a_i, s_{-i}) > u_i(s_i, s_{-i})$

# Support of Nash Equilibria

## Corollary

*Let  $s \in \mathcal{S}$  be a Nash equilibrium and  $a_i, a'_i \in \mathcal{A}_i$  are actions from the support of  $s_i$ . Now,  $u_i(a_i, s_{-i}) = u_i(a'_i, s_{-i})$ .*

Can we exploit this fact to find a Nash equilibrium?

# Finding Nash Equilibria

	<b>L</b>	<b>R</b>
<b>U</b>	(2, 1)	(0, 0)
<b>D</b>	(0, 0)	(1, 2)

Column player (player 2) plays **L** with probability  $p$  and **R** with probability  $(1 - p)$ . In NE it holds

$$\begin{aligned}\mathbb{E}u_1(\mathbf{U}) &= \mathbb{E}u_1(\mathbf{D}) \\ 2p + 0(1 - p) &= 0p + 1(1 - p) \\ p &= \frac{1}{3}\end{aligned}$$

Similarly, we can compute the strategy for player 1 arriving at  $(\frac{2}{3}, \frac{1}{3}), (\frac{1}{3}, \frac{2}{3})$  as Nash equilibrium.

# Finding Nash Equilibria

Can we use the same approach here?

	<b>L</b>	<b>C</b>	<b>R</b>
<b>U</b>	(2, 1)	(0, 0)	(0, 0)
<b>M</b>	(0, 0)	(1, 2)	(0, 0)
<b>D</b>	(0, 0)	(0, 0)	(-1, -1)

Not really... No strategy  $s_i$  of the row player ensures  $u_{-i}(s_i, L) = u_{-i}(s_i, C) = u_{-i}(s_i, R) :-$

**Can something help us?**

Iterated removal of dominated strategies.

Search for a possible support (enumeration of all possibilities).



# Maxmin

	<b>L</b>	<b>R</b>
<b>U</b>	(2, 1)	(0, 0)
<b>D</b>	(0, 0)	(1, 2)

Recall that there are multiple Nash equilibria in this game. Which one should a player play? This is a known equilibrium-selection problem.

Playing a Nash strategy does not give any guarantees for the expected payoff. If we want guarantees, we can use a different concept – maxmin strategies.

## Definition (Maxmin)

The *maxmin strategy* for player  $i$  is  $\arg \max_{s_i} \min_{s_{-i}} u_i(s_i, s_{-i})$  and the *maxmin value* for player  $i$  is  $\max_{s_i} \min_{s_{-i}} u_i(s_i, s_{-i})$ .

# Maxmin and Minmax

## Definition (Maxmin)

The *maxmin strategy* for player  $i$  is  $\arg \max_{s_i} \min_{s_{-i}} u_i(s_i, s_{-i})$  and the *maxmin value* for player  $i$  is  $\max_{s_i} \min_{s_{-i}} u_i(s_i, s_{-i})$ .

## Definition (Minmax, two-player)

In a two-player game, the *minmax strategy* for player  $i$  against player  $-i$  is  $\arg \min_{s_i} \max_{s_{-i}} u_{-i}(s_i, s_{-i})$  and the *minmax value* for player  $-i$  is  $\min_{s_i} \max_{s_{-i}} u_{-i}(s_i, s_{-i})$ .

Maxmin strategies are conservative strategies against a worst-case opponent.

Minmax strategies represent punishment strategies for player  $-i$ .

# Zero-sum case

What about zero-sum case? How do

- player  $i$ 's maxmin,  $\max_{s_i} \min_{s_{-i}} u_i(s_i, s_{-i})$ , and
- player  $i$ 's minmax,  $\min_{s_i} \max_{s_{-i}} u_{-i}(s_i, s_{-i})$

relate?

---

$$\max_{s_i} \min_{s_{-i}} u_i(s_i, s_{-i}) = - \min_{s_i} \max_{s_{-i}} u_{-i}(s_i, s_{-i})$$

... but we can prove something stronger ...

# Maxmin and Von Neumann's Minimax Theorem

## Theorem (Minimax Theorem (von Neumann, 1928))

*In any finite, two-player zero-sum game, in any Nash equilibrium each player receives a payoff that is equal to both their maxmin value and the minmax value of the opponent.*



Consequences:

- 1  $\max_{s_i} \min_{s_{-i}} u_i(s_i, s_{-i}) = \min_{s_{-i}} \max_{s_i} u_i(s_i, s_{-i})$
- 2 we can safely play Nash strategies in zero-sum games
- 3 all Nash equilibria have the same payoff (by convention, the maxmin value for player 1 is called *value of the game*).

# Computing NE in Zero-Sum Games

We can now compute Nash equilibrium for two-player, zero-sum games using a linear programming:

$$\max_{s,U} U \quad (1)$$

$$\text{s.t.} \quad \sum_{a_1 \in \mathcal{A}_1} s(a_1) u_1(a_1, a_2) \geq U \quad \forall a_2 \in \mathcal{A}_2 \quad (2)$$

$$\sum_{a_1 \in \mathcal{A}_1} s(a_1) = 1 \quad (3)$$

$$s(a_1) \geq 0 \quad \forall a_1 \in \mathcal{A}_1 \quad (4)$$

Computing a Nash equilibrium in zero-sum normal-form games can be done in polynomial time.

# Computing NE in General-Sum Games

The problem is more complex for general-sum games (LCP program):

$$\sum_{a_2 \in \mathcal{A}_2} u_1(a_1, a_2) s_2(a_2) + q(a_1) = U_1 \quad \forall a_1 \in \mathcal{A}_1$$

$$\sum_{a_1 \in \mathcal{A}_1} u_2(a_1, a_2) s_1(a_1) + w(a_2) = U_2 \quad \forall a_2 \in \mathcal{A}_2$$

$$\sum_{a_1 \in \mathcal{A}_1} s_1(a_1) = 1 \quad \sum_{a_2 \in \mathcal{A}_2} s_2(a_2) = 1$$

$$q(a_1) \geq 0, w(a_2) \geq 0, s_1(a_1) \geq 0, s_2(a_2) \geq 0 \quad \forall a_1 \in \mathcal{A}_1, \forall a_2 \in \mathcal{A}_2$$

$$s_1(a_1) \cdot q(a_1) = 0, s_2(a_2) \cdot w(a_2) = 0 \quad \forall a_1 \in \mathcal{A}_1, \forall a_2 \in \mathcal{A}_2$$

Computing a Nash equilibrium in two-player general-sum normal-form game is a PPAD-complete problem. The problem gets even more complex (FIXP-hard) when moving to  $n \geq 3$  players.

# Correlated Equilibrium

Consider again the following game:

	<b>L</b>	<b>R</b>
<b>U</b>	(2, 1)	(0, 0)
<b>D</b>	(0, 0)	(1, 2)

Wouldn't it be better to coordinate 50:50 between the outcomes (U,L) and (D,R)? Can we achieve this coordination? We can use a *correlation device*—a coin, a streetlight, commonly observed signal—and use this signal to avoid unwanted outcomes.



Robert Aumann

# Correlated Equilibrium

## Definition (Correlated Equilibrium (simplified))

Let  $\mathcal{G} = (\mathcal{N}, \mathcal{A}, u)$  be a normal-form game and let  $\sigma$  be a probability distribution over joint pure strategy profiles  $\sigma \in \Delta(\mathcal{A})$ . We say that  $\sigma$  is a correlated equilibrium if for every player  $i$ , every signal  $a_i \in \mathcal{A}_i$  and every possible action  $a'_i \in \mathcal{A}_i$  it holds

$$\sum_{a_{-i} \in \mathcal{A}_{-i}} \sigma(a_i, a_{-i}) u_i(a_i, a_{-i}) \geq \sum_{a_{-i} \in \mathcal{A}_{-i}} \sigma(a_i, a_{-i}) u_i(a'_i, a_{-i})$$

## Corollary

*For every Nash equilibrium there exists a corresponding Correlated Equilibrium.*



# Computing Correlated Equilibrium

Computing a Correlated equilibrium is easier compared to Nash and can be found by linear programming even in general-sum case:

$$\sum_{a_{-i} \in \mathcal{A}_{-i}} \sigma(a_i, a_{-i}) u_i(a_i, a_{-i}) \geq \sum_{a_{-i} \in \mathcal{A}_{-i}} \sigma(a_i, a_{-i}) u_i(a'_i, a_{-i}) \quad \forall i \in \mathcal{N}, \forall a_i, a'_i \in \mathcal{A}_i$$

$$\sum_{a \in \mathcal{A}} \sigma(a) = 1 \quad \sigma(a) \geq 0 \quad \forall a \in \mathcal{A}$$

This is a feasibility LP without any objective function.

Adding an objective function allows us to find some specific CE (e.g., the one that maximizes social welfare).

# Stackelberg Equilibrium

Finally, consider a situation where an agent is a central public authority (police, government, etc.) that needs to design and publish a policy that will be observed and reacted to by other agents.



itemize

- *the leader* – publicly commits to a strategy
- *the follower(s)* – play a Nash equilibrium with respect to the commitment of the leader

Stackelberg equilibrium is a strategy profile that satisfies the above conditions and maximizes the expected utility value of the leader:

$$\arg \max_{s \in \mathcal{S}; \forall i \in \mathcal{N} \setminus \{1\} s_i \in BR_i(s_{-i})} u_1(s)$$

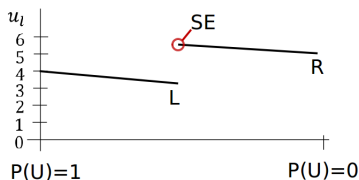
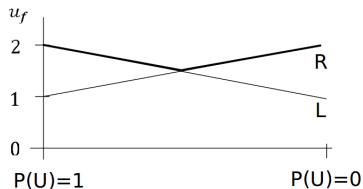
# Stackelberg Equilibrium

Consider the following game:

	<b>L</b>	<b>R</b>
<b>U</b>	(4, 2)	(6, 1)
<b>D</b>	(3, 1)	(5, 2)

(**U**, **L**) is a Nash equilibrium.

What happens when the row player commits to play strategy **D** with probability 1? Can the row player get even more?



## There may be Multiple Nash Equilibria

The followers need to break ties in case there are multiple NE:

- arbitrary but fixed tie breaking rule
- *Strong SE* – the followers select such NE that maximizes the outcome of the leader (when the tie-breaking is not specified we mean SSE),
- *Weak SE* – the followers select such NE that minimizes the outcome of the leader.

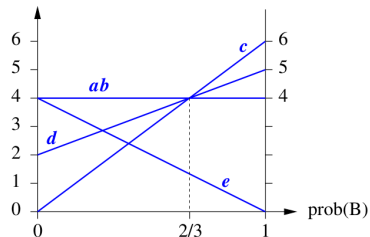
Exact Weak Stackelberg equilibrium does not have to exist.

# Different Stackelberg Equilibria

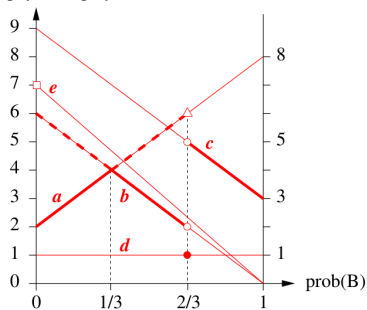
Exact Weak Stackelberg equilibrium does not have to exist.

1 \ 2	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>
<i>T</i>	(2, 4)	(6, 4)	(9, 0)	(1, 2)	(7, 4)
<i>B</i>	(8, 4)	(0, 4)	(3, 6)	(1, 5)	(0, 0)

payoff to player 2



payoff to player 1



# Computing a Stackelberg equilibrium in NFGs

The problem is polynomial for two-players normal-form games; 1 is the leader, 2 is the follower.

Baseline polynomial algorithm requires solving  $|\mathcal{A}_2|$  linear programs:

$$\begin{aligned} & \max_{s_1 \in \mathcal{S}_1} \sum_{a_1 \in \mathcal{A}_1} s_1(a_1) u_1(a_1, a_2) \\ & \sum_{a_1 \in \mathcal{A}_1} s_1(a_1) u_2(a_1, a_2) \geq \sum_{a_1 \in \mathcal{A}_1} s_1(a_1) u_2(a_1, a'_2) \quad \forall a'_2 \in \mathcal{A}_2 \\ & \sum_{a_1 \in \mathcal{A}_1} s_1(a_1) = 1 \end{aligned}$$

one for each  $a_2 \in \mathcal{A}_2$  assuming  $a_2$  is the best response of the follower.

# Comparison of Different Equilibria

The expected values of these solution concepts coincide in zero-sum games.

For two-players general-sum games, the solution concepts are fundamentally different:

- CE is a probability distribution over possible outcomes, desired outcome is sampled and corresponding actions are sent to players as recommendations; following the recommendations is best response for the players.
- NE is a pair of mixed strategies (probability distributions over pure strategies)—one for each player—such that these mixed strategies are best responses to the strategy of the opponent.
- SE is a pair of strategies where leader's mixed strategy is such a public commitment that maximizes the outcome of the leader while the follower plays the best response.

# Comparison of Different Equilibria

For two-players general-sum games, the solution concepts are fundamentally different:

- CE can be computed in polynomial time with a single LP.  
Even finding some specific CE is polynomial.
- NE can be computed in exponential time with a single LCP.  
Finding some specific NE is NP-complete.
- SSE can be computed in polynomial time with multiple LPs.



# Regret

The concept of regret is useful when the other players are not completely malicious.

	<b>L</b>	<b>R</b>
<b>U</b>	$(100, a)$	$(1 - \varepsilon, b)$
<b>D</b>	$(2, c)$	$(1, d)$

## Definition (Regret)

A player  $i$ 's *regret* for playing an action  $a_i$  if the other agents adopt action profile  $a_{-i}$  is defined as

$$\left[ \max_{a'_i \in \mathcal{A}_i} u_i(a'_i, a_{-i}) \right] - u_i(a_i, a_{-i})$$

## Definition (MaxRegret)

A player  $i$ 's *maximum regret* for playing an action  $a_i$  is defined as

$$\max_{a_{-i} \in \mathcal{A}_{-i}} \left( \left[ \max_{a'_i \in \mathcal{A}_i} u_i(a'_i, a_{-i}) \right] - u_i(a_i, a_{-i}) \right)$$

## Definition (MinimaxRegret)

Minimax regret actions for player  $i$  are defined as

$$\arg \min_{a_i \in \mathcal{A}_i} \max_{a_{-i} \in \mathcal{A}_{-i}} \left( \left[ \max_{a'_i \in \mathcal{A}_i} u_i(a'_i, a_{-i}) \right] - u_i(a_i, a_{-i}) \right)$$