# Chapter 10: Multi-agent RL (Continued)

# Recap: Normal-form Game

A normal-form game is a tuple $(n, \mathcal{A}_{1\ldots n}, R_{1\ldots n})$,

- $n$ is the number of players,
- $\mathcal{A}_i$ is the set of actions available to player $i$
  - $\mathcal{A}$ is the joint action space $\mathcal{A}_1 \times \ldots \times \mathcal{A}_n$,
- $R_i$ is player $i$'s payoff function $\mathcal{A} \rightarrow \Re$.

$$R_1 = \begin{array}{cc} & a_2 \\ a_1 & \begin{pmatrix} & \vdots & \\ \cdots & R_1(a) & \cdots \\ & \vdots & \end{pmatrix} \end{array} \qquad R_2 = \begin{array}{cc} & a_2 \\ a_1 & \begin{pmatrix} & \vdots & \\ \cdots & R_2(a) & \cdots \\ & \vdots & \end{pmatrix} \end{array}$$

# Minimax Optimal Solution

- Play strategy with the best worst-case outcome.

$$\underset{\sigma_i \in \Delta(\mathcal{A}_i)}{\text{argmax}} \; \min_{a_{-i} \in \mathcal{A}_{-i}} R_i(\langle \sigma_i, \sigma_{-i} \rangle)$$

# Nash Equilibria

- A best response set is the set of all strategies that are optimal given the strategies of the other players.

$$\mathrm{BR}_i(\sigma_{-i}) = \{\sigma_i \quad | \quad \forall \sigma'_i \quad R_i(\langle \sigma_i, \sigma_{-i} \rangle) \geq R_i(\langle \sigma'_i, \sigma_{-i} \rangle)\}$$

- A Nash equilibrium is a joint strategy, where all players are playing best responses to each other.

$$\forall i \in \{1 \ldots n\} \qquad \sigma_i \in \mathrm{BR}_i(\sigma_{-i})$$

- Nash = Minimax in Two-Player Zero-sum games, but not always.

# Existence of Nash Equilibria

- All finite normal-form games have at least one Nash equilibrium. (Nash, 1950)

- In zero-sum games…

  - Equilibria all have the same value and are interchangeable.

$$\langle \sigma_1, \sigma_2 \rangle, \langle \sigma_1', \sigma_2' \rangle \text{ are Nash } \Rightarrow \langle \sigma_1, \sigma_2' \rangle \text{ is Nash.}$$

  - Equilibria correspond to minimax optimal strategies.

# Computation of Nash Equilibria

- The exact complexity of computing a Nash equilibrium is an open problem. (Papadimitriou, 2001)

**The Complexity of Computing a Nash Equilibrium***


Constantinos Daskalakis
Computer Science Division,
UC Berkeley
costis@cs.berkeley.edu

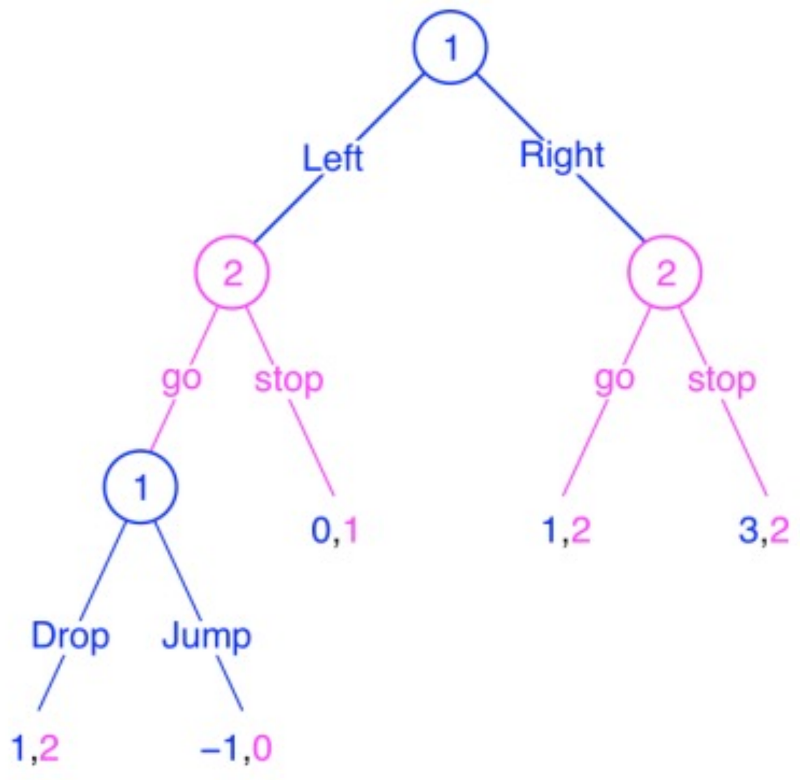
Paul W. Goldberg
Dept. of Computer Science,
University of Liverpool
P.W.Goldberg@liver-pool.ac.uk

Christos H. Papadimitriou
Computer Science Division,
UC Berkeley
christos@cs.berkeley.edu

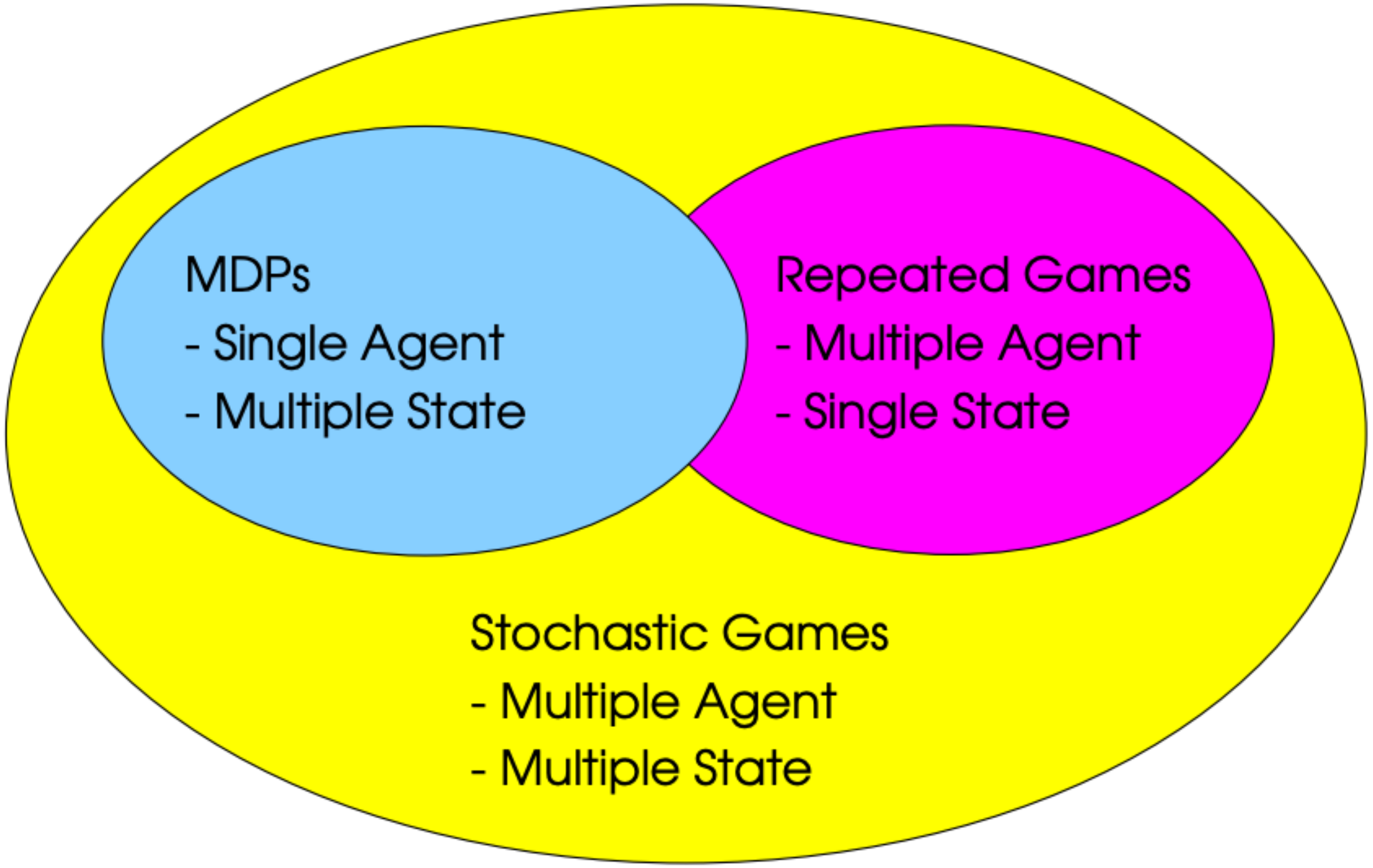

- Nash-equilibrium is PPAD-hard [2008].

# Extensive-form Game

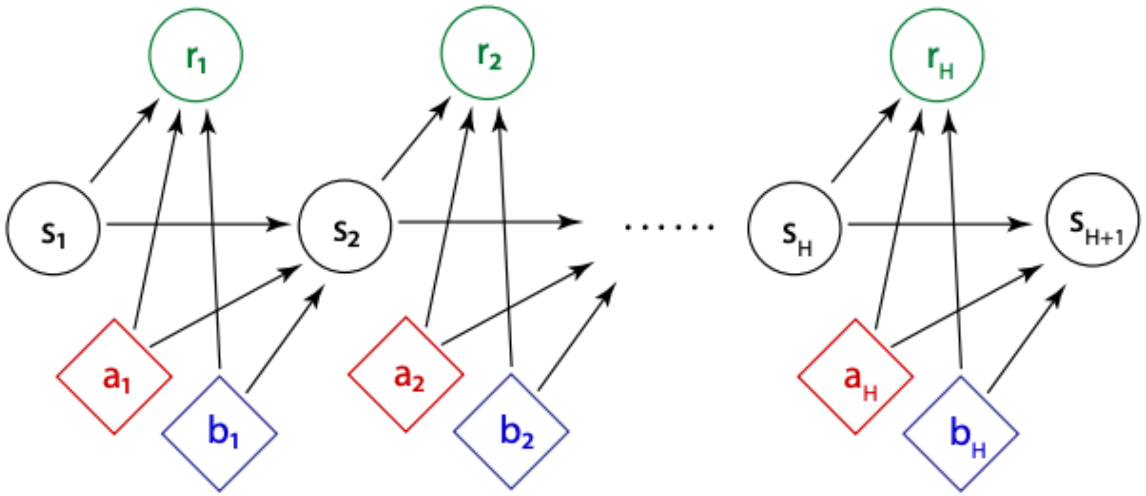
- Example: any full-observation turn-based games, e.g. Chess, Go.



# Stochastic/Markov Games



# Stochastic/Markov Games




Two-player zero-sum Markov Game $(\mathcal{S}, \mathcal{A}, \mathcal{B}, \mathbb{P}, r, H)$ [Shapley 1953].

- $\mathcal{S}$: set of states; $\mathcal{A}, \mathcal{B}$: set of actions for the max-player/the min-player.
- $\mathbb{P}_h(s_{h+1}|s_h, a_h, b_h)$: transition probability.
- $r_h(s_h, a_h, b_h) \in [0, 1]$: reward for the max-player (loss for the min-player).
- $H$: horizon/the length of the game.

# Our Setup

- Fully observable: joint actions and states are revealed to both agents.
- Tabular: the size of $\mathcal{S}, \mathcal{A}, \mathcal{B}$ is finite and small.

# Policy and Value

- General policy for the max-player (depends on the entire history):

$$\pi_{1,h} : (\mathcal{S} \times \mathcal{A} \times \mathcal{B})^{h-1} \times \mathcal{S} \to \Delta_{\mathcal{A}}$$

- Markov policy for the max-player (depends on the current state):

$$\pi_{1,h} : \mathcal{S} \to \Delta_{\mathcal{A}}$$

Policy of the min-player can be defined by symmetry.

- Value $V^\pi$ for joint policy $\pi = (\pi_1, \pi_2)$: the expected cumulative reward received by the max-player if both agents follow the joint policy $\pi$:

$$V^\pi = \mathbb{E}_\pi \left[ \sum_{h=1}^{H} r_h(s_h, a_h, b_h) \right]$$

# Nash Equilibria

**Nash Equilibria**

The policies $(\pi_1^\star, \pi_2^\star)$ is a Nash equilibrium if no player has incentive to deviate from her current policy. That is, for any $\pi_1, \pi_2$

$$V^{\pi_1, \pi_2^\star} \le V^{\pi_1^\star, \pi_2^\star} \le V^{\pi_1^\star, \pi_2}$$

In two-player zero-sum Markov games, minimax theorem holds:

$$\max_{\pi_1} \min_{\pi_2} V^{\pi_1, \pi_2} = \min_{\pi_2} \max_{\pi_1} V^{\pi_1, \pi_2}$$

# Nash Equilibria

The optimal strategy if always facing best responses.

"We may not win by a large margin, but no one beats us."

**Objective**: find $\epsilon$-approximate Nash equilibria $(\hat{\pi}_1, \hat{\pi}_2)$ using a small number of samples with mild dependency on $S, A_1, A_2, \epsilon, H$.

$$\max_{\pi_1} V^{\pi_1, \hat{\pi}_2} - \min_{\pi_2} V^{\hat{\pi}_1, \pi_2} \leq \epsilon.$$

# Technical Challenges

To name a few:

- Large size of policy space:

$$\Omega((1/\epsilon)^{HSA}) \text{ Markov policies in the tabular setting}$$

- Nash equilibrium policy is Markov, but the best response may not be.

- MGs do not allow efficient no-regret learning [Bai, **Jin**, Yu, 2020].

$$\max_{\pi_1} \sum_{t=1}^{T} V_1^{\pi_1 \times \pi_2^t} - \sum_{t=1}^{T} V_1^{\pi_1^t \times \pi_2^t} \leq \mathrm{poly}(H, S, A, B) T^{1-\alpha}.$$

# Computing NE in Zero-sum Markov Games: “anecdotal Recipe”

Key observation: given a fixed opponent, computing best response (BR) is a single-agent RL problem.

**Nash finding algorithms with BR oracle**

self-play
fictitious play
double oracle
. . .

→

**Single-agent RL algorithms**

value-iteration
Q-learning
DQN
PPO
. . .

commonly used in practice.

# Computing NE in Zero-sum Markov Games

**Fictitious play [Brown, 1949]**

**for** $k = 1, \ldots, K$,

$$\pi_1^{k+1} = BR[(1/k) \cdot (\pi_2^1 + \ldots + \pi_2^k)].$$

$$\pi_2^{k+1} = BR[(1/(k+1)) \cdot (\pi_1^1 + \ldots + \pi_1^{k+1})].$$

$\pi_i^k$: the policy of the $i^{\text{th}}$ player at the $k^{\text{th}}$ iteration

Computing the best response to the average policy of the opponent.

makes more sense in rock-paper-scissor.

# Computing NE in Zero-sum Markov Games

**Asymptotic convergence of fictitious play [Robinson 1951]**

Ficitious play indeed converges to Nash equilibrium!

However, how **fast**?

- inspecting the proof of [Robinson 1951], it requires $(1/\epsilon)^{\Omega(A)}$ iterations to converge to $\epsilon$-Nash equilibrium for a normal-form game with $A$ actions.
- Karlin conjectured in 1959 that this rate can be improved to $\mathcal{O}(1/\epsilon^2)$.
- Daskalakis and Pan [2014] refute the conjecture, and prove that $(1/\epsilon)^{\Omega(A)}$ is real in the worst case.

# Drawbacks of Direct Combinations

- Algorithms are designed based on black-box usage of single-agent RL, which does not exploit the detailed structure of MGs.

- Converting a MG into a norm-form game gives a number of action $A = (1/\epsilon)^{HSA'}$.

- Finding BR is NOT a easy single-agent RL problem:
  - When the min-player deploys a fixed **non-Markovian** policy, the game is NOT an MDP from the perspective of the max-player.
  - Existing single-agent RL results do not apply.

# Planning in Markov Games

We start with the setting of known transition $\mathbb{P}$ and reward $r$.

A Nash equilibrium of a MG is a Markov policy.

We define $V_h^\star(s)$, $Q_h^\star(s,a,b)$ which satisfies the **Bellman optimality equation**:

$$
\begin{aligned}
Q_h^\star(s,a,b) =& r_h(s,a,b) + \mathbb{E}_{s' \sim \mathbb{P}_h(\cdot|s,a,b)} V_{h+1}^\star(s') \\
V_h^\star(s) =& \max_{\mu \in \Delta_{\mathcal{A}}} \min_{\nu \in \Delta_{\mathcal{B}}} \sum_{a,b} \mu(a)\nu(b) Q_h^\star(s,a,b) \\
:=& \text{Nash\_Value}(Q_h^\star(s,\cdot,\cdot))
\end{aligned}
$$

# Planning in Markov Games

A dynamical programming approach to find a Nash equilibrium.

**Nash Value Iteration (Nash VI)**

Initialize $V^\star_{H+1}(s) = 0$ for all $s$.

**for** $h = H, \ldots, 1$,

  **for** all $(s, a, b)$,

    $Q^\star_h(s, a, b) \leftarrow r_h(s, a, b) + \mathbb{E}_{s' \sim \mathbb{P}_h(\cdot|s,a,b)} V^\star_{h+1}(s')$

  **for** all $s$

    $(\pi^\star_{1,h}(\cdot|s), \pi^\star_{2,h}(\cdot|s)) \leftarrow \mathrm{Nash}(Q^\star_h(s, \cdot, \cdot))$

    $V^\star_h(s) \leftarrow \langle \pi^\star_{1,h}(\cdot|s) \times \pi^\star_{2,h}(\cdot|s), Q^\star_h(s, \cdot, \cdot) \rangle$

Nash VI computes the Nash equilibrium of MGs in poly$(H, S, A, B)$ steps!