

# ROB311 Quiz 3

Hanhee Lee

March 30, 2025

## Contents

<b>1</b>	<b>Multi-Agent Problems</b>	<b>2</b>
1.1	Policy Equilibria . . . . .	2
1.2	Single Action Games . . . . .	3
1.3	Actions (Deterministic) . . . . .	3
1.4	Strategies (Probabilistic) . . . . .	3
1.4.1	Simplifying Games . . . . .	3
1.5	Examples . . . . .	4
1.5.1	Finding Action Equilibria . . . . .	4
1.5.2	Optimal Action Profiles . . . . .	6
1.5.3	Finding Strategy Equilibria . . . . .	7

# One-Shot Multi-Agent Decision Problems

## 1 Multi-Agent Problems

**Summary:** In a **Multi-Agent problem**, we assume that:

- Set of states for environment is  $\mathcal{S}$
- $P$  agents within environment.
- For each state  $s \in \mathcal{S}$ :
  - possible actions for agent  $i$  is  $\mathcal{A}_i(s)$
  - set of action profiles is  $\mathcal{A}(s) = \prod_{i=1}^P \mathcal{A}_i(s)$
- possible state-action pairs are  $\mathcal{T} = \{(s, a) \text{ s.t. } s \in \mathcal{S}, a \in \mathcal{A}(s)\}$
- environment in some origin state,  $s_0$
- environment destroyed after  $N$  transitions
- agent  $j$  wants to find policy  $\pi_j(a_j | s)$  so that  $\mathbb{E}[r_j(p)]$  is maximized
- agents act independently given the environment's state:  $\pi(a | s) = \prod_{j \in [P]} \pi_j(a_j | s)$

Name	Function:
State transition given state-action pair defined by $\text{tr} : \mathcal{T} \rightarrow \mathcal{S}$	$\text{tr}(s, a)$ = state transition from $s$ under $a$
Reward to each agent, $i$ defined by $r_i : \mathcal{Q} \times \mathcal{S} \rightarrow \mathbb{R}_+$	$r_i(s, a, \text{tr}(s, a))$ = rwd to agent $i$ for $(s, a, \text{tr}(s, a))$
State evolution of environment after $N$ transitions	$p = \langle (s_0, a^{(1)}, s_1), \dots, (s_{N-1}, a^{(N)}, s_N) \rangle$
<ul style="list-style-type: none"> <li>• Given sequence of actions: <math>p.a = \langle a^{(1)}, \dots, a^{(n)} \rangle</math></li> <li>• <math>s_N = \tau(s_{n-1}, a^{(n)})</math></li> </ul>	
reward to agent $i$	$r_i(p) = \sum_{n=1}^N r_i(s_{n-1}, a^{(n)}, s_n)$
expected-reward (value) of playing $a$ from $s$ for agent $j$	$q_j(s, a) = r_j(s, a, \tau(s, a)) + \sum_{a' \in \mathcal{A}(\tau(s, a))} \pi(a'   \tau(s, a)) q_j(\tau(s, a), a')$
<ul style="list-style-type: none"> <li>• <math>\mathcal{A}(s) = \emptyset</math> if <math>s \in \mathcal{G}</math></li> </ul>	

### 1.1 Policy Equilibria

**Notes:**

- **No Regret:**  $\pi$  is no-regret if  $\pi_j$  maximizes  $q_j$  when  $\pi_{-j}$  is fixed.
- If all agents play perfectly, then we expect

$$\pi(a | s) = \begin{cases} 1 & \text{if } a = a^*(s) \\ 0 & \text{otherwise} \end{cases}$$

- $a_j^*(s) = \arg \max_{a_j \in \mathcal{A}_j(s)} q_j(s, a_j, a_{-j}^*)$  is the best action for agent  $j$  given the other agents' policies.

## 1.2 Single Action Games

**Summary:** In a **Single Action Game**, we assume that:

- $N = 1$  (one-shot game)
- Initial state is  $s_0 \in \mathcal{S}$
- Agent  $j$  wants to find policy,  $\pi_i(a_i \mid s_0)$  so  $\mathbb{E}[r_i(p)]$  is maximized

## 1.3 Actions (Deterministic)

**Summary:**

Name	Function:
Action $j$ for agent $i$	$[0 \cdots 0 \ 1 \ 0 \cdots 0]$
<ul style="list-style-type: none"> <li>• One-hot vector of <math>M_i</math> components, <math>\mathbf{e}_{i,j}</math></li> </ul>	
Agent $i$ 's set of possible actions	$\mathcal{A}_i = \{a_i \in \{0, 1\}^{M_i} \mid \sum_{j \in [M_i]} a_{i,j} = 1\}$
<ul style="list-style-type: none"> <li>• Agent <math>i</math>'s chosen action with <math>a_i \in \mathcal{A}_i</math></li> </ul>	
Action profile is a tuple of actions	$a = (a_1, \dots, a_P)$
<ul style="list-style-type: none"> <li>• <b>Notational Convenience:</b> <math>a_{-i} = (a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_P)</math> so that <math>a = (a_i, a_{-i})</math>.</li> </ul>	
Optimal action profile	$a^+$ s.t. $\forall a \exists i$ s.t. $r_i(a) < r_i(a^+)$
<ul style="list-style-type: none"> <li>• An action profile w.r.t. which any other action profile leaves at least one player worse off.</li> </ul>	
Set of optimal action profiles	$\text{aOpt} = \{a^+ \mid \forall a \exists i : r_i(a) < r_i(a^+)\}$
Best-action mapping, $\text{ba}_i: \mathcal{A}_{-i} \rightarrow \mathcal{A}_i$	$\text{ba}_i(a_{-i}) = \arg \max_{a_i \in \mathcal{A}_i} r_i(a_i, a_{-i})$ $= \{a_i \in \mathcal{A}_i \mid r_i(a_i, a_{-i}) = \max_{a'_i \in \mathcal{A}_i} r_i(a'_i, a_{-i})\}$
Agent $i$ will <b>not regret</b> playing $a_i^*$ when others play $a_{-i}^*$ if	$r_i(a_i^*, a_{-i}^*) \geq r_i(a_i, a_{-i}^*) \ \forall a_i \in \mathcal{A}_i$ or $a_i^* \in \text{ba}_i(a_{-i}^*)$
Action equilibria is any action, $a^*$ in which no agent regrets	$a_i^* \in \text{ba}_i(a_{-i}^*) \ \forall i \in [P]$
Existence of action equilibria	May not always exist, i.e., it may be that $\text{aEq} = \emptyset$

## 1.4 Strategies (Probabilistic)

**Summary:**

### 1.4.1 Simplifying Games

**Notes:**

## 1.5 Examples

### 1.5.1 Finding Action Equilibria

**Process:** To find action equilibria:

1. For each  $i$ , compute  $\text{ba}_i(a_{-i})$  for all  $a_{-i}$
2. Define  $\text{bap}_i$  so that  $\text{bap}_i = \{(a'_i, a_{-i}), \forall a'_i \in \text{ba}_i(a_{-i}), \forall a_{-i} \in \mathcal{A}_{-i}\}$
3. Action equilibria are then  $\text{aEq} = \bigcap_{i \in [P]} \text{bap}_i$ .

**Example:**

1. **Given:** Suppose lion and cavemen both want meat. Each must decide whether to fight for the food or share it.



			
		<b>Fight</b>	<b>Share</b>
	<b>Fight</b>	(1, 1)	(4, 0)
	<b>Share</b>	(0, 4)	(2, 2)

Figure 1

2. **Problem:** Find the action equilibria

3. **Solution:**

(a) **Best Action Profiles:**

- Cavemen:  $\text{bap}_{\text{cavemen}} = \{(\text{Fight}, \text{Fight}), (\text{Fight}, \text{Share})\}$ . Cavemen fights no matter what.
  - If lion fights, then cavemen fights to get maximum reward in this scenario of +1.
  - If lion shares, then caveman fights to get maximum reward in this scenario of +4.
- Lion:  $\text{bap}_{\text{lion}} = \{(\text{Fight}, \text{Fight}), (\text{Share}, \text{Fight})\}$ . Lion fights no matter what.
  - If caveman fights, then lion fights to get maximum reward in this scenario of +1.
  - If caveman shares, then lion fights to get maximum reward in this scenario of +4.

(b) **Best Action Equilibria:** Intersection of the best action profiles.

- $\text{aEq} = \text{bap}_{\text{cavemen}} \cap \text{bap}_{\text{lion}} = \{(\text{Fight}, \text{Fight})\}$

**Example:**

1. **Given:** Suppose lion and cavemen both want meat. Each must decide whether to fight for the food or share it.



			
		Fight	Share
	Fight	(3, 0)	(0, 3)
	Share	(0, 1)	(1, 0)

Figure 2

2. **Problem:** Find the action equilibria

3. **Solution:**

- (a) **Best Action Profiles:**

- Cavemen:  $\text{bap}_{\text{cavemen}} = \{(\text{Fight}, \text{Fight}), (\text{Share}, \text{Share})\}$ . Cavemen fights no matter what.
  - If lion fights, then cavemen fights to get maximum reward in this scenario of +3.
  - If lion shares, then caveman shares to get maximum reward in this scenario of +1.
- Lion:  $\text{bap}_{\text{lion}} = \{(\text{Fight}, \text{Share}), (\text{Share}, \text{Fight})\}$ . Lion fights no matter what.
  - If caveman fights, then lion shares to get maximum reward in this scenario of +3.
  - If caveman shares, then lion fights to get maximum reward in this scenario of +1.

- (b) **Best Action Equilibria:** Intersection of the best action profiles.

- $\text{aEq} = \text{bap}_{\text{cavemen}} \cap \text{bap}_{\text{lion}} = \emptyset$

### 1.5.2 Optimal Action Profiles

**Example:**

1. **Given:** Suppose lion and cavemen both want meat. Each must decide whether to fight for the food or share it.



			
		Fight	Share
	Fight	(1, 1)	(4, 0)
	Share	(0, 4)	(2, 2)

Figure 3

2. **Problem:** Find the optimal action profiles:
3. **Solution:**
  - (a)  $a_{\text{Opt}} = \{(\text{Share}, \text{Share}), (\text{Fight}, \text{Share}), (\text{Share}, \text{Fight})\}$ 
    - FINISH

## 1.5.3 Finding Strategy Equilibria

**Example:**

1. **Given/Problem:** Find all equilibria of the following one-shot game or state that none exist.

	B1 (y)	B2 (1-y)
A1 (x)	(5, 3)	(1, 0)
A2 (1-x)	(0, 1)	(2, 4)

- (#,#) is the payoff to P1 and P2 respectively for a given action profile.

2. **Solution:**

(a) **Define Probabilities:**

- Let  $y$  be the probability that B1 plays action B1 so  $1 - y$  is the probability that B1 plays action B2.
- Let  $x$  be the probability that A1 plays action A1 so  $1 - x$  is the probability that A1 plays action A2.

(b) **Expected Rewards:**

- P1:

$$\begin{aligned}
 E[x] &= 5xy + 1x(1 - y) + 0(1 - x)y + 2(1 - x)(1 - y) = 5xy + x - xy + 2 - 2x - 2y + 2xy \\
 &= 5xy - xy + 2xy + x - 2x - 2y + 2 \\
 &= 6xy - x - 2y + 2 \quad \text{simplify} \\
 &= \underbrace{(6y - 1)}_c x + 2 - 2y \quad \text{linear in } x
 \end{aligned}$$

- P2:

$$\begin{aligned}
 E[y] &= 3xy + 0x(1 - y) + 1(1 - x)y + 4(1 - x)(1 - y) = 3xy + 0 + y - xy + 4 - 4x - 4y + 4xy \\
 &= 3xy - xy + 4xy + y - 4x - 4y + 4 \\
 &= 6xy - 4x - 3y + 4 \quad \text{simplify} \\
 &= \underbrace{(6x - 3)}_c y + 4 - 4x \quad \text{linear in } y
 \end{aligned}$$

- **Note:**  $E[x]$  is linear in  $x$  and  $E[y]$  is linear in  $y$ .

(c) **Constrained Argmax Expected Rewards w.r.t  $x \in [0, 1]$  (since P1):** If it was cost, then minimize. Also don't care about constant term in  $y$  since we are derivating w.r.t  $x$ .

- P1:

$$x = \begin{cases} 1 & \text{if } y > \frac{1}{6} \text{ i.e. } c > 0 \text{ since positive want maximum positive} \\ [0, 1] & \text{if } y = \frac{1}{6} \text{ i.e. } c = 0 \text{ doesn't matter since } 0 \\ 0 & \text{if } y < \frac{1}{6} \text{ i.e. } c < 0 \text{ since negative want maximum negative} \end{cases}$$

- P2:

$$y = \begin{cases} 1 & \text{if } x > \frac{3}{6} \text{ i.e. } c > 0 \text{ since positive want maximum positive} \\ [0, 1] & \text{if } x = \frac{3}{6} \text{ i.e. } c = 0 \text{ doesn't matter since } 0 \\ 0 & \text{if } x < \frac{3}{6} \text{ i.e. } c < 0 \text{ since negative want maximum negative} \end{cases}$$

(d) **Finding all equilibrium:** Lines on the graph represents where your reward is maximized.

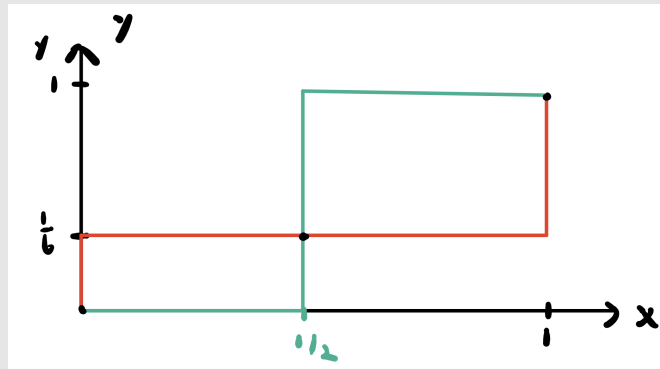


Figure 4

- **Case 1:**  $x = 0$  and  $y = 0$ 
    - $P(\text{P1 chooses A1}) = 0$
    - $P(\text{P1 chooses A2}) = 1$
    - $P(\text{P2 chooses B1}) = 0$
    - $P(\text{P2 chooses B2}) = 1$
  - **Case 2:**  $x = 1/2$  and  $y = 1/6$ 
    - $P(\text{P1 chooses A1}) = 1/2$
    - $P(\text{P1 chooses A2}) = 1/2$
    - $P(\text{P2 chooses B1}) = 1/6$
    - $P(\text{P2 chooses B2}) = 5/6$
  - **Case 3:**  $x = 1$  and  $y = 1$ 
    - $P(\text{P1 chooses A1}) = 1$
    - $P(\text{P1 chooses A2}) = 0$
    - $P(\text{P2 chooses B1}) = 1$
    - $P(\text{P2 chooses B2}) = 0$
- (e) **Unstable Equilibrium:** P1 moves left and right b/c  $x$  is associated with  $x$ -axis. P2 moves up and down b/c  $y$  is associated with  $y$ -axis.

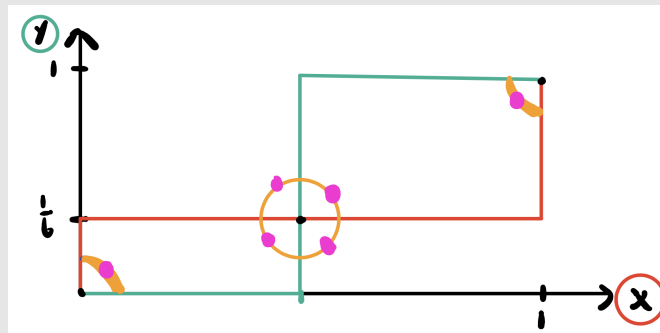


Figure 5

- Stability means that in a radius disc around the equilibrium, if you move a little bit, you will still be in the equilibrium (have to check all relevant quadrants)
  - If one quadrant is unstable, then don't need to check the other quadrants as the equilibrium point is unstable.
  - Simultaneous (both players move at the same time) and sequential (one player moves first and the other player moves second)
- **Case 1:**  $x = 0$  and  $y = 0$  is stable
  - Q1: Always converges to  $(0, 0)$  since P1 moves left to red and P2 moves down to turquoise.
- **Case 2:**  $x = 1/2$  and  $y = 1/6$  is unstable
  - Q1 (Top Left): P1 moves right to red and P2 moves up to turquoise  $\Rightarrow (1, 1)$
  - Q2 (Top Right): P1 moves right to red and P2 moves up to turquoise  $\Rightarrow (1, 1)$
  - Q3 (Bottom Left): P1 moves left to red and P2 moves down to turquoise  $\Rightarrow (0, 0)$



- Q4 (Bottom Right): P1 moves left to red and P2 moves down to turquoise  $\implies (0, 0)$
- **Case 3:**  $x = 1$  and  $y = 1$  is stable
  - Q1: Always converges to  $(1, 1)$  since P1 moves left to red and P2 moves down to turquoise.