Multi-Agent Systems

Christos Dimitrakakis

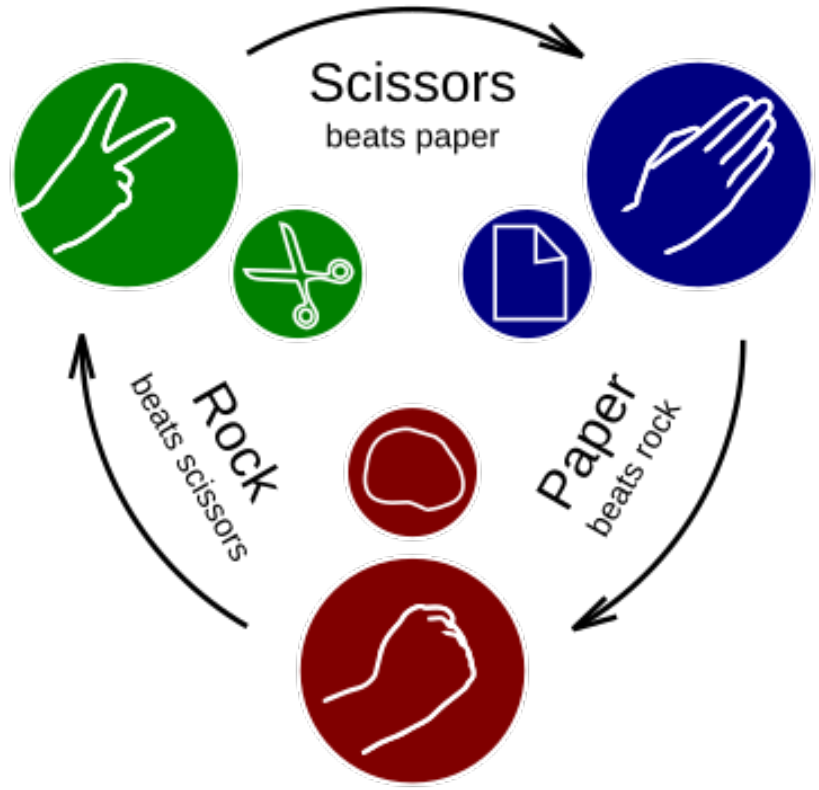
May 15, 2025

Outline

Multi-agent decision making

- Two versus n-player games
- Co-operative games
- Zero-sum games
- General-sum games
- Stochastic games
- Partial information games

Rock/Paper/Scissors



- Number of players: 2
- Zero-sum.
- Deterministic.
- Simultaneous move.

Chess/Go/Checkers/Othello

- Number of players: 2
- Zero-sum
- Deterministic
- Alternating, Full information

Backgammon

- Number of players: 2
- Zero-sum
- Stochastic
- Alernating, Full information

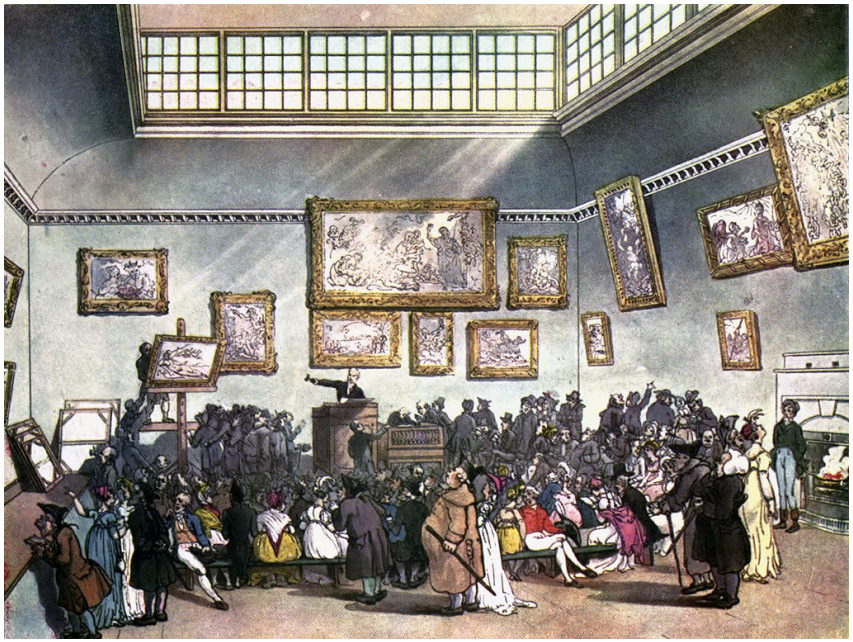
Poker/Blackjack

- Number of players: n
- Zero-sum
- Stochastic [Partially]
- Alternating, Partial information

Doom/Quake/CoD

- Number of players: n
- General sum
- Stochastic
- Simultaneous, Sequential, Partial information

Auctions



- Number of players: n
- General sum
- Deterministic
- Simultaneous move

Humans and AI

Any system involving interaction between multiple agent can be describe through game theory. One question is how to define the preferences of each agent.

Human preferences

- These are typically unknown.
- They might not be expressible in mathematical form.
- Nevertheless, we make the utility assumption.

AI preferences

- These are typically specified by humans as utilities.
- However, it is hard to fully specify them.

Normal form

In the table below, we see how much reward each player obtains for every combination of actions

ρ^1, ρ^2	b = 0	b = 1
a = 0	2, 1	4, 0
a = 1	1, 0	3, 1

Simultaneous moves

We assume that each player is playing without seeing the move of the other player.

Commitment

However, we can also look at commitment or Stackleberg games, where one player either *commits* to playing a move, or plays before the other player.

Information structure

For other types of move sequencing, we have to encode the information structure of a game as a graph.

Co-operative, adversarial and general games

More generally, we can say that every player i in the game:

- Takes an action $a^i \in A_i$.
- Obtains a reward $\rho^i(x)$ for each possible outcome/choice x.

Co-operative, adversarial and general games

More generally, we can say that every player i in the game:

- Takes an action $a^i \in A_i$.
- Obtains a reward $\rho^i(x)$ for each possible outcome/choice x.

2-player Zero-sum games

- $\rho^1 = -\rho^2$
- Can be solved effiicently.

Co-operative, adversarial and general games

More generally, we can say that every player i in the game:

- Takes an action $a^i \in A_i$.
- Obtains a reward $\rho^i(x)$ for each possible outcome/choice x.

2-player Zero-sum games

- $\rho^1 = -\rho^2$
- Can be solved effiicently.

n-player Collaborative games

- $\rho^i = \rho^j$ for all players i, j.
- If the players can co-ordinate, then it reduces to a single-agent problem with action-space $A = A_1 \times \cdots A_n$.

Co-operative, adversarial and general games

More generally, we can say that every player i in the game:

- Takes an action $a^i \in A_i$.
- Obtains a reward $\rho^i(x)$ for each possible outcome/choice x.

2-player Zero-sum games

- $\rho^1 = -\rho^2$
- Can be solved effiicently.

n-player Collaborative games

- $\rho^i = \rho^j$ for all players i, j.
- If the players can co-ordinate, then it reduces to a single-agent problem with action-space $A = A_1 \times \cdots A_n$.

n-player General-sum games

- ρ^i can be anything.
- Finding solutions for these games is harder.

Zero-Sum: Rock Paper Scissors

ρ^1, ρ^2	Rock	Paper	Scissors
Rock	0, 0	-1, 1	1, -1
Paper	1, -1	0, 0	-1, 1
Scissors	-1, 1	1, -1	0, 0

Co-operative: Party

People want to bring something to the party. Ideally, one brings food, and the other drinks. But if they do not co-ordinate, then there is only food, or only drink.

ρ^1, ρ^2	food	drink
food	2, 2	10, 10
drink	10, 10	1, 1

Here, co-ordination makes the outcomes better for everybody.

General-Sum: Prisoner's dilemma

ρ^1, ρ^2	cooperate	defect
cooperate	-1, -1	-5, 0
defect	0, -5	-3, -3

Basic concepts in normal form games

ρ^1, ρ^2	b = 0	b = 1
a = 0	2, 1	4, 0
a = 1	1, 0	3, 1

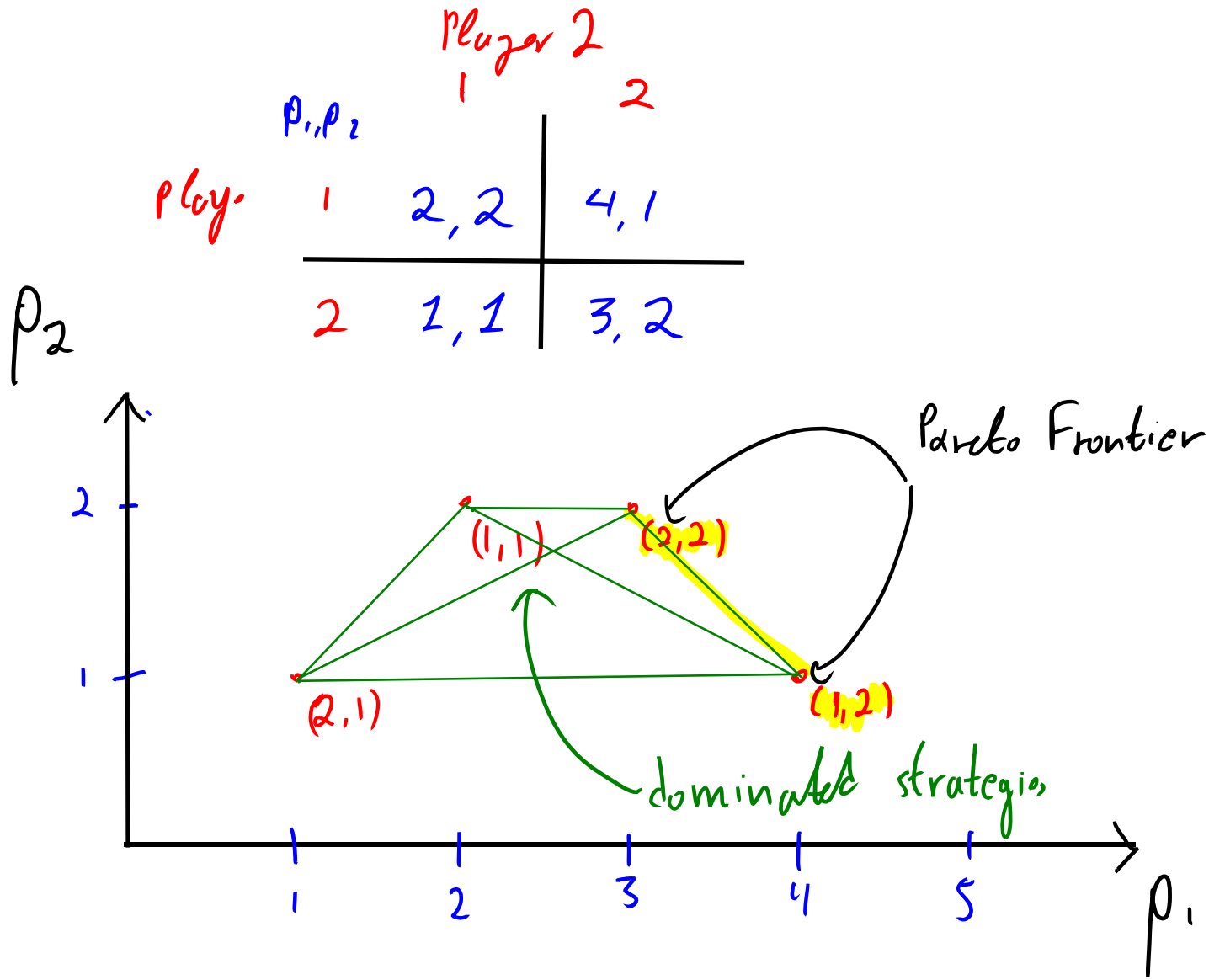
Domination and best response

- $b = 1$ is a best response to $a = 1$, i.e. $\rho^2(1,1) > \rho^2(1,0)$
- $a = 0$ is a strictly dominant strategy. Given any b, it is strictly better to play $a = 0$, i.e. $\rho^1(0,b) > \rho^1(1,b)$.
- If a pair (a, b) is *not dominated*, then it is Pareto-efficient.

Questions

- How much reward can a obtain?
- Does b have a dominant strategy?
- Does this take into account what b likes?

Pareto-Optimality



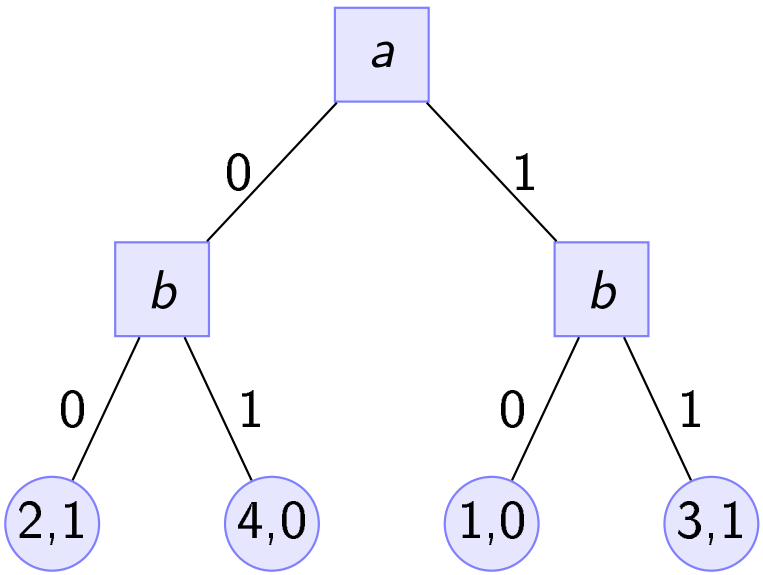
Commitment

Let us see what happens when one player commits to a move

ρ^1, ρ^2	b = 0	b = 1
a = 0	2, 1	4, 0
a = 1	1, 0	3, 1

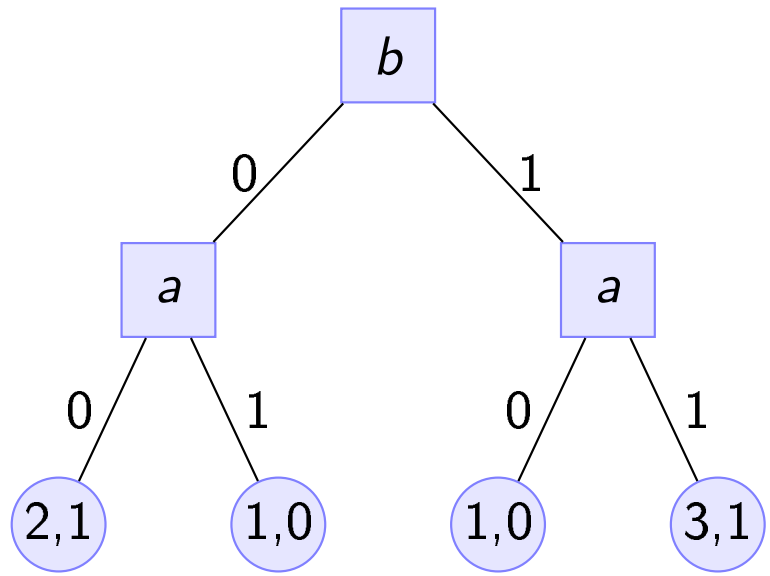
Player *a* is first

- What should *b* play?
- What is *a*'s best move?



Player *b* is first

- What should *a* play in each case?



Extensive-form alternating-move game

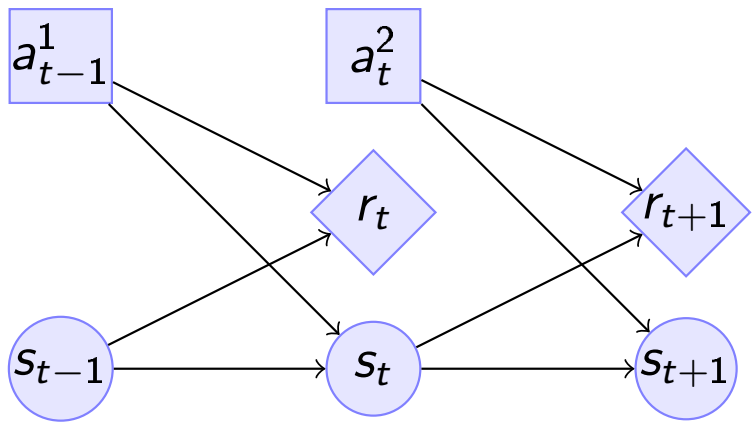
- The state $s_t \in S$.
- The actions $a_t^i \in A$.
- The rewards $r_t^i \in \mathbb{R}$, $r_t = (r_t^1, r_t^2)$.
- The transition probabilities

  $$\mathbb{P}(s_{t+1} \mid s_t, a_{t-1}^i)$$

Extensive-form alternating-move game

- The state $s_t \in S$.
- The actions $a_t^i \in A$.
- The rewards $r_t^i \in \mathbb{R}$, $r_t = (r_t^1, r_t^2)$.
- The transition probabilities

  $$\mathbb{P}(s_{t+1} \mid s_t, a_{t-1}^i)$$



Extensive-form alternating-move zero sum games

- At time t:

Extensive-form alternating-move zero sum games

- At time t:
- The state is s_t, players receive rewards $r_t^1 = \rho(s_t), r_t^2 = -\rho(s_t)$

Extensive-form alternating-move zero sum games

- At time t:
- The state is s_t, players receive rewards $r_t^1 = \rho(s_t), r_t^2 = -\rho(s_t)$
- Player i chooses action a_t^i, which is revealed.

Extensive-form alternating-move zero sum games

- At time t:
- The state is s_t, players receive rewards $r_t^1 = \rho(s_t), r_t^2 = -\rho(s_t)$
- Player i chooses action a_t^i, which is revealed.
- The state changes to s_{t+1}, and is revealed.

Extensive-form alternating-move zero sum games

- At time t:
- The state is s_t, players receive rewards $r_t^1 = \rho(s_t), r_t^2 = -\rho(s_t)$
- Player i chooses action a_t^i, which is revealed.
- The state changes to s_{t+1}, and is revealed.
- Players receive reward $\rho(s_{t+1}), -\rho(s_{t+1})$

Extensive-form alternating-move zero sum games

- At time t:
- The state is s_t, players receive rewards $r_t^1 = \rho(s_t), r_t^2 = -\rho(s_t)$
- Player i chooses action a_t^i, which is revealed.
- The state changes to s_{t+1}, and is revealed.
- Players receive reward $\rho(s_{t+1}), -\rho(s_{t+1})$
- Player $j = 1 - i$ chooses action a_{t+1}^j.

Extensive-form alternating-move zero sum games

- At time t:
- The state is s_t, players receive rewards $r_t^1 = \rho(s_t), r_t^2 = -\rho(s_t)$
- Player i chooses action a_t^i, which is revealed.
- The state changes to s_{t+1}, and is revealed.
- Players receive reward $\rho(s_{t+1}), -\rho(s_{t+1})$
- Player $j = 1 - i$ chooses action a_{t+1}^j.
- The state changes to s_{t+2}.

Extensive-form alternating-move zero sum games

- At time t:
- The state is s_t, players receive rewards $r_t^1 = \rho(s_t), r_t^2 = -\rho(s_t)$
- Player i chooses action a_t^i, which is revealed.
- The state changes to s_{t+1}, and is revealed.
- Players receive reward $\rho(s_{t+1}), -\rho(s_{t+1})$
- Player $j = 1 - i$ chooses action a_{t+1}^j.
- The state changes to s_{t+2}.
- Player 1 receives $\rho(s_t)$ and 2 receives $-\rho(s_t)$.

Extensive-form alternating-move zero sum games

- At time t:
- The state is s_t, players receive rewards $r_t^1 = \rho(s_t), r_t^2 = -\rho(s_t)$
- Player i chooses action a_t^i, which is revealed.
- The state changes to s_{t+1}, and is revealed.
- Players receive reward $\rho(s_{t+1}), -\rho(s_{t+1})$
- Player $j = 1 - i$ chooses action a_{t+1}^j.
- The state changes to s_{t+2}.
- Player 1 receives $\rho(s_t)$ and 2 receives $-\rho(s_t)$.

Extensive-form alternating-move zero sum games

- At time t:
- The state is s_t, players receive rewards $r_t^1 = \rho(s_t), r_t^2 = -\rho(s_t)$
- Player i chooses action a_t^i, which is revealed.
- The state changes to s_{t+1}, and is revealed.
- Players receive reward $\rho(s_{t+1}), -\rho(s_{t+1})$
- Player $j = 1 - i$ chooses action a_{t+1}^j.
- The state changes to s_{t+2}.
- Player 1 receives $\rho(s_t)$ and 2 receives $-\rho(s_t)$.

The utility for player 1 is

$$U^1 = \sum_t \rho(s_t),$$

while for 2 it is

$$U^2 = -\sum_t \rho(s_t)$$

Backwards induction for Alternating Zero Sum Games

Let π_1 and π_2 be the policies of each player and π the joint policy.

Backwards induction for Alternating Zero Sum Games

Let π_1 and π_2 be the policies of each player and π the joint policy.

The value function of a policy $\pi = (\pi_1, \pi_2)$

For the utility of player 1, we get:

$$V_t^{1,\pi}(s) \triangleq \mathbb{E}_\pi[U_t^1 \mid s_t = s] = \rho(s) + \mathbb{E}[U_{t+1}^1 \mid s_t = s] \tag{3}$$

Backwards induction for Alternating Zero Sum Games

Let π_1 and π_2 be the policies of each player and π the joint policy.

The value function of a policy $\pi = (\pi_1, \pi_2)$

For the utility of player 1, we get:

$$V_t^{1,\pi}(s) \triangleq \mathbb{E}_\pi[U_t^1 \mid s_t = s] = \rho(s) + \mathbb{E}[U_{t+1}^1 \mid s_t = s] \tag{1}$$

$$= \rho(s) + \sum_{a^1} \pi(a^1 \mid s) \sum_j V_{t+1}^{1,\pi}(j) P(j \mid s, a^1)$$

(3)

Backwards induction for Alternating Zero Sum Games

Let π_1 and π_2 be the policies of each player and π the joint policy.

The value function of a policy $\pi = (\pi_1, \pi_2)$

For the utility of player 1, we get:

$$V_t^{1,\pi}(s) \triangleq \mathbb{E}_\pi[U_t^1 \mid s_t = s] = \rho(s) + \mathbb{E}[U_{t+1}^1 \mid s_t = s] \tag{1}$$

$$= \rho(s) + \sum_{a^1} \pi(a^1 \mid s) \sum_j V_{t+1}^{1,\pi}(j) P(j \mid s, a^1) \tag{2}$$

$$V_{t+1}^{1,\pi}(j) = \rho(j) + \sum_{a^2} \pi(a^2 \mid j) \sum_j V_{t+2}^{1,\pi}(j) P(k \mid j, a^2) \tag{3}$$

The optimal value function

We can define the optimal value function analogously to MDPs, but player 2 is minimising.
The value for player 1, together with the recursion is given below:

The optimal value function

We can define the optimal value function analogously to MDPs, but player 2 is minimising.
The value for player 1, together with the recursion is given below:

$$V_t^{1,*}(s) = \max_{\pi_1} \min_{\pi_2} \mathbb{E}_{\pi}[U_t^1 \mid s_t = s] \tag{6}$$

The optimal value function

We can define the optimal value function analogously to MDPs, but player 2 is minimising.
The value for player 1, together with the recursion is given below:

$$V_t^{1,*}(s) = \max_{\pi_1} \min_{\pi_2} \mathbb{E}_\pi[U_t^1 \mid s_t = s] \tag{4}$$

$$= \rho(s) + \max_{a^1} \sum_j V_{t+1}^{1,*}(j) P(j \mid s, a^1)$$

(6)

The optimal value function

We can define the optimal value function analogously to MDPs, but player 2 is minimising.
The value for player 1, together with the recursion is given below:

$$V_t^{1,*}(s) = \max_{\pi_1} \min_{\pi_2} \mathbb{E}_\pi[U_t^1 \mid s_t = s] \tag{4}$$

$$= \rho(s) + \max_{a^1} \sum_j V_{t+1}^{1,*}(j) P(j \mid s, a^1) \tag{5}$$

$$V_{t+1}^{1,*}(j) = \rho(j) + \min_{a^2} \sum_j V_{t+1}^{1,*}(j) P(k \mid j, a^2) \tag{6}$$

Normal-form simultaneous-move zero-sum games

(Also called minimax games)

- Player a chooses action a in secret.
- Player b chooses action b in secret.
- Players observe both actions
- Player a receives $\rho(a, b)$, and b receives $-\rho(a, b)$.

Mixed strategies

Each player chooses an action randomly, independently of one another:

$$\pi(a, b) = \pi_1(a)\pi_2(b)$$

π_i is called a mixed strategy.

Optimal strategies for zero-sum games

The value of a strategy pair

The expected value of the game for the first player is

$$V(\pi_1, \pi_2) \triangleq \sum_{a,b} \pi_1(a)\rho(a,b)\pi_2(b) = \boldsymbol{\pi}_1^\top \boldsymbol{R}\boldsymbol{\pi}_2,$$

where $\boldsymbol{\pi}_i$ is the vector form representation of i's strategy.

The value of the game

Any zero-sum game has at least one solution π^* over mixed strategies so that

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

The problem can be solved through linear programming
The idea is to set find a policy corresponding to the greatest lower bound (or lowest upper bound) on the value.

Linear programming solution for ZSG

linear programming problem

This is an optimisation problem with linear objective and constraints. In canonical form it is written as:

$$\min_x \ \theta^\top x, \qquad \text{s.t. } c^\top x \geq 0.$$

Primal formulation

Find the higest lower bound for player 1

$$\max_v \ v, \qquad \text{s.t. } (\boldsymbol{R}\boldsymbol{\pi}_2)_j \geq v \ \forall j, \ \sum_j \pi_2(j) = 1, \pi_2(j) \geq 0$$

Dual formulation

Find the lowest upper bound for player 2

$$\min_v \ v, \qquad \text{s.t. } (\boldsymbol{\pi}_1^\top \boldsymbol{R})_j \leq v \ \forall j, \ \sum_j \pi_1(j) = 1, \pi_1(j) \geq 0$$

Normal-form general sum games

Game structure

- Each player i chooses action $a^i \in A_i$ in secret.
- The joint action is $\boldsymbol{a} = (a^1, \ldots, a^n)$.
- Player i receives a reward $\rho^i(\boldsymbol{a})$

Mixed strategies

Players independently draw actions a^i from strategies $\pi^i(a^i)$, i.e.

$$\pi(\boldsymbol{a}) = \prod_i \pi^i(a^i).$$

(This means they cannot coordinate).
The expected utility of the strategy π is then

$$\mathbb{E}[\rho^i] = \sum_{\boldsymbol{a}} \pi(\boldsymbol{a})\rho^i(\boldsymbol{a}).$$

Example: penalty shot

ρ^1, ρ^2	kick left	kick right
dive left	1, -1	-1, 1
dive right	-1 1	1, -1

Example: Chicken

ρ^1, ρ^2	turn	dare
turn	0, 0	-1, +1
dare	+1, -1	-10, -10

Example: Prisoner's dilemma

ρ^1, ρ^2	cooperate	defect
cooperate	-1, -1	-5, 0
defect	0, -5	-3, -3

Solution concept: Nash equilibrium

- n players
- Actions $\boldsymbol{a} = (a^1, \ldots, a^n)$.
- Payoffs $\rho_i(a)$

Pure Nash equilibrium

A joint action $\boldsymbol{a}$ is a pure Nash equlibrium if no player i can play some alternative action $\hat{a}^i$ and obtain a higher payoff:

$$\rho^i(\hat{a}^i, a^{-i}) \leq \rho^i(\boldsymbol{a}) \forall i, \hat{a}^i,$$

where $a^{-i} = (a_1, \ldots, a^{i-1}, a^{i+1}, \ldots, a_n)$ are the actions of all players apart from i.

Nash equilibrium

A strategy profile $\pi = (\pi^1, \ldots, \pi^n)$, is a Nash equlibrium if no player i can deviate to obtain a higher payoff:

$$\mathbb{E}[\rho^i \mid \hat{a}^i, \pi^{-i}] \leq \mathbb{E}[\rho^i \mid \pi]$$

Computing Nash equilibria

- A pure Nash equilibrium may not exist.
- A Nash equilibrium always exists (Nash, 1950)
- Nash is PPAD, with $\mathtt{P} \subseteq \mathtt{PPAD} \subseteq \mathtt{NP}$.

Pre-processing steps

- Remove all dominated actions for each player.
- Check if the remaining combinations have a pure equilibrium.
- If not, the equilibrium is mixed, with support on these actions.

Nash theory

The Brouwer problem (PPAD)

Input:

- a function $F : [0,1]^m \to [0,1]^m$
- $L \in (0,1)$ is a Lipschitz constant such that $\|F(x) - F(x')\| \leq L\|x - x'\|$
- An $\epsilon > 0$

Output:

- x^* such that $\|F(x^*) - x^*\| \leq \epsilon$.

The connection with Nash

- Given by Nash himself in his 1950 proof.
- The fixed point of F is the Nash equlibrium

The Linear Complementarity Problem

- $\sum_b \rho^1(a,b)\pi_2(b) + s_1(a) = v_1$ for all a
- $\sum_a \rho^2(j,b)\pi_1(a) + s_2(b) = v_1$ for all b
- $\|\pi_i\|_1 = 1$, $\pi_i \geq 0$
- $s \geq 0$
- $\pi_i \cdot s_i = 0$: assigns zero to slack variables corresponding to actions with probability > 0

Optimistic hedge

Hedge

$$w_{t+1} \propto w_t * exp(\eta r_t)$$

Optimistic hedge

$$x_{t+1} \propto x_t * \exp(\eta r_{t-1} - 2r_t)$$

Extensive-form general sum games

- At time t:
- The state is s_t, players receive rewards $\rho^i(s_t)$.
- Player $i = I(s_t)$ chooses an action.
- The state changes to s_{t+1}, and is revealed.

The utility for each player is

$$U^i = \sum_t \rho^i(s_t)$$

Backwards induction for Alternating General Sum Games

Let π_i be the policy of the i-th player and π the joint policy.

The value function of a policy $\pi = (\pi_i)_{i=1}^n$

For any player i, we can define their value at time t as:

$$V_t^{i,\pi}(s) \triangleq \mathbb{E}_\pi[U_t^i \mid s_t = s] \tag{7}$$

$$= \rho^i(s) + \sum_{a \in A} \pi_{I(s)}(a \mid s) \sum_j V_{t+1}^{1,\pi}(j) P(j \mid s, a) \tag{8}$$

Optimal policies

For perfect information games, we can use this recursion:

$$a_t^*(s) = \arg\max_{a \in A} \sum_j V_{t+1}^{I(s),*}(j) P(j \mid s, a) \tag{9}$$

$$V_t^{i,*} = \rho^i(s) + \sum_j V_{t+1}^{i,\pi}(j) P(j \mid s, a_t^*(s)) \qquad \forall i \tag{10}$$

This ensures that we update the values of all players at each step.