Efficiently searching agent-state based policies in Dec-POMDPs

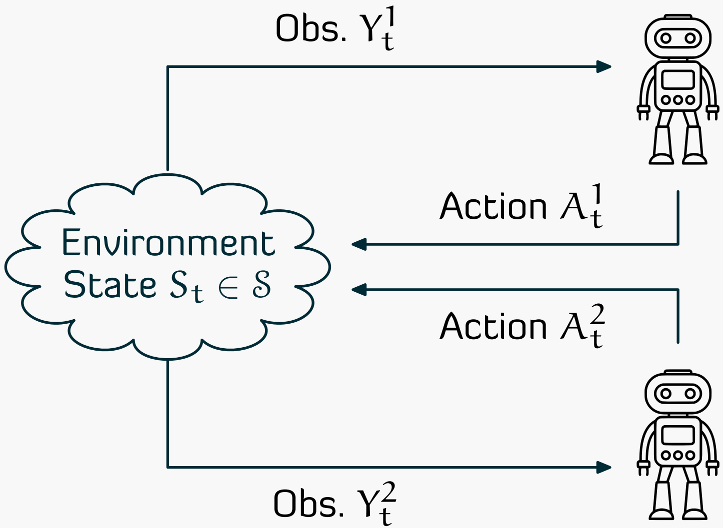
Aditya Mahajan
McGill University

Joint work with Amit Sinha and Matthieu Geist

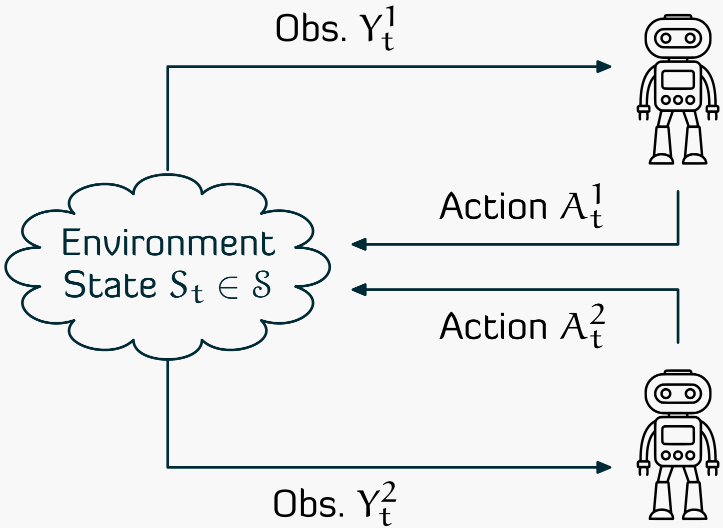
Foundations of Multi-Agent and Mean-Field Reinforcement Learning
21 May 2026

▷ email: aditya.mahajan@mcgill.ca
▷ web: https://adityam.github.io

Dec-POMDPs or multi-agent dynamic teams



Dec-POMDPs or multi-agent dynamic teams

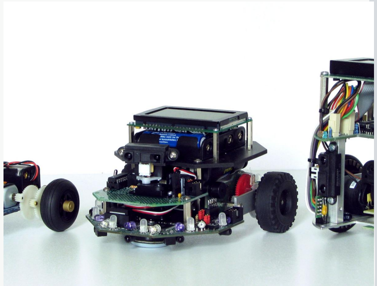


System model

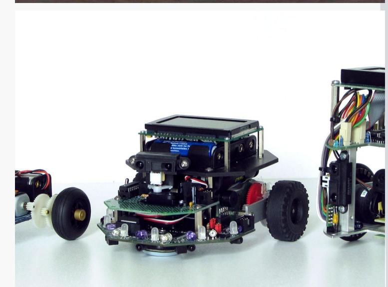
- Multiple agents
- Each sees only local observations
- must coordinate to maximize common reward
- Key challenge: Decentralized information


Common theme: multi-stage multi-agent
decision making under uncertainty

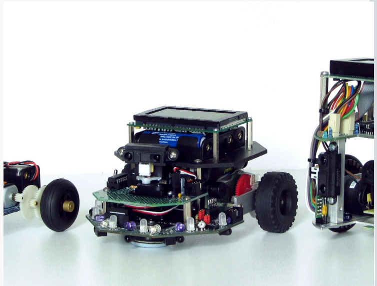
Networked control systems



Networked control systems



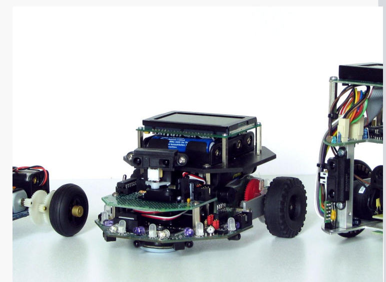
Networked control systems



Challenges

- Signals sent over wireless channels (packet drops)

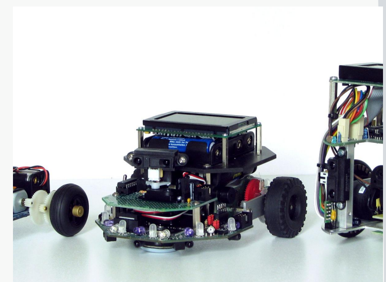
Networked control systems



Challenges

- Signals sent over wireless channels (packet drops)
- Different vehicles have different information

Networked control systems



Challenges

- Signals sent over wireless channels (packet drops)
- Different vehicles have different information
  - Decentralized control
  - Decentralized estimation
  - Decentralized learning

Salient Features of Modern Engineering Systems

Salient Features of Modern Engineering Systems

Multiple agents

Agents have different partial information about the environment and each other

Salient Features of Modern Engineering Systems

Multiple agents

Agents have different partial information about the environment and each other

Decentralized Coordination

All agents must coordinate to achieve a system-wide objective

Salient Features of Modern Engineering Systems

Multiple agents

Agents have different partial information about the environment and each other

Decentralized Coordination

All agents must coordinate to achieve a system-wide objective

Communication and Signaling

Possible to explicitly or implicitly communicate information

Salient Features of Modern Engineering Systems

Multiple agents

Agents have different partial information about the environment and each other

Decentralized Coordination

All agents must coordinate to achieve a system-wide objective

Communication and Signaling

Possible to explicitly or implicitly communicate information

Learning

Dynamics may not be completely known or may change over time

Salient Features of Modern Engineering Systems

Multiple agents

Agents have different partial information about the environment and each other

Decentralized Coordination

All agents must coordinate to achieve a system-wide objective

Communication and Signaling

Possible to explicitly or implicitly communicate information

Learning

Dynamics may not be completely known or may change over time

Conceptual difficulties in Dec-POMDPs/team problems

Witsenhausen Counterexample

- A two-step dynamical system with two controllers
- Linear dynamics, quadratic cost, and Gaussian disturbance
- Non-linear controllers outperform linear control strategies . . .

  . . . cannot use Kalman filtering + Riccati equations

Witsenhausen, "A counterexample in stochastic optimum control," SICON 1968.

Conceptual difficulties in Dec-POMDPs/team problems

Witsenhausen Counterexample

- A two-step dynamical system with two controllers
- Linear dynamics, quadratic cost, and Gaussian disturbance
- Non-linear controllers outperform linear control strategies . . .

  . . . cannot use Kalman filtering + Riccati equations

Whittle and Rudge Example

- Infinite horizon dynamical system with two symmetric controllers
- Linear dynamics, quadratic cost, and Gaussian disturbance
- A priori restrict attention to linear controllers
- Best linear ctrls **don't** have finite dimensional representation

Witsenhausen, "A counterexample in stochastic optimum control," SICON 1968.

Whittle and Rudge, "The optimal linear solution of a symmetric team control problem," App. Prob. 1974.

Conceptual difficulties in Dec-POMDPs/team problems

Witsenhausen Counterexample

- A two-step dynamical system with two controllers
- Linear dynamics, quadratic cost, and Gaussian disturbance
- Non-linear controllers outperform linear control strategies . . .

  . . . cannot use Kalman filtering + Riccati equations

Whittle and Rudge Example

- Infinite horizon dynamical system with two symmetric controllers
- Linear dynamics, quadratic cost, and Gaussian disturbance
- A priori restrict attention to linear controllers
- Best linear ctrls **don't** have finite dimensional representation

Complexity analysis

- All random variables are finite valued
- Finite horizon setup
- The problem of finding the best control strategy is in **NEXP**

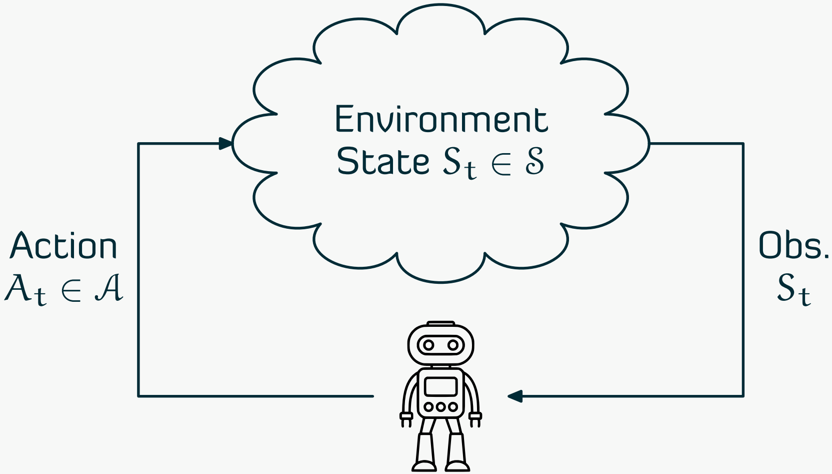
Witsenhausen, "A counterexample in stochastic optimum control," SICON 1968.
Whittle and Rudge, "The optimal linear solution of a symmetric team control problem," App. Prob. 1974.
Bernstein, et al., "The complexity of decentralized control of Markov decision processes," MOR 2002.

Why are multi-agent problems hard?

Why are multi-agent problems hard?

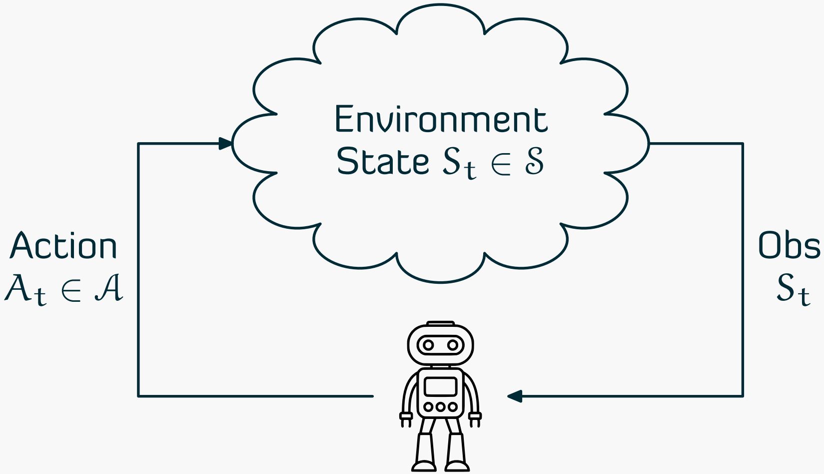
Why are single-agent problems easy?

Review: Markov decision processes (MDPs)



MDP: MARKOV DECISION PROCESS

Review: Markov decision processes (MDPs)



MDP: MARKOV DECISION PROCESS

Dynamics: $\mathbb{P}(S_{t+1} \mid S_t, A_t)$

Observations: S_t

Reward $R_t = r(S_t, A_t)$.

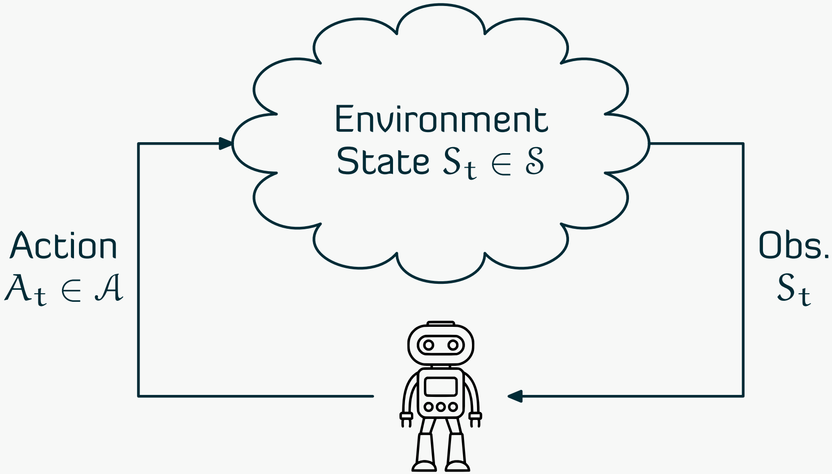
Action: $A_t \sim \pi_t(S_{1:t}, A_{1:t-1})$.

$\pi = (\pi_1, \dots, \pi_T)$ is called a policy.

Objective choose a policy π to maximize:

$$J(\pi) := \mathbb{E}^{\pi}\left[\sum_{t=1}^{T} R_t\right]$$

Review: Markov decision processes (MDPs)



MDP: MARKOV DECISION PROCESS

Dynamics: $\mathbb{P}(S_{t+1} \mid S_t, A_t)$

Observations: S_t

Reward $R_t = r(S_t, A_t)$.

Action: $A_t \sim \pi_t(S_{1:t}, A_{1:t-1})$.

$\pi = (\pi_1, \dots, \pi_T)$ is called a policy.

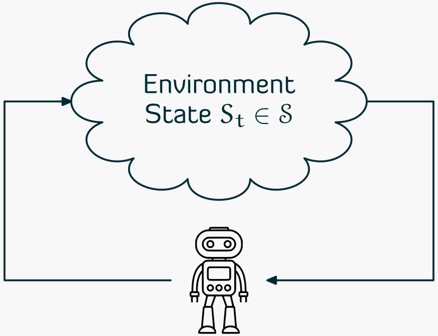
Objective choose a policy π to maximize:

$$J(\pi) := \mathbb{E}^{\pi}\left[\sum_{t=1}^{T} R_t\right]$$

Conceptual challenge

- Brute force search has an exponential complexity in time horizon.
- How to efficiently search an optimal policy?

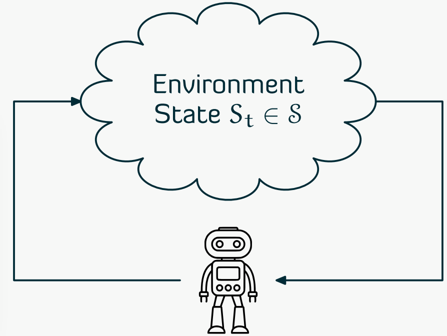
Review: Key simplifying ideas



Principle of Irrelevant information	No loss of optimality in choosing the action A_t as a function of the current state S_t

Blackwell, "Memoryless strategies in finite-stage dynamic prog.," Annals Math. Stats, 1964.

Review: Key simplifying ideas



Principle of Irrelevant information

No loss of optimality in choosing the action A_t as a function of the current state S_t

Blackwell, "Memoryless strategies in finite-stage dynamic prog.," Annals Math. Stats, 1964.

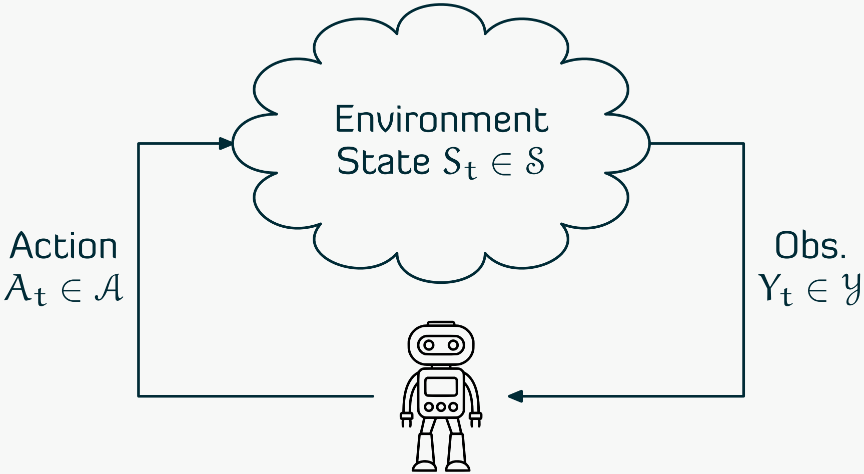
Principle of Optimality

The optimal control policy is given by a DP with state S_t:

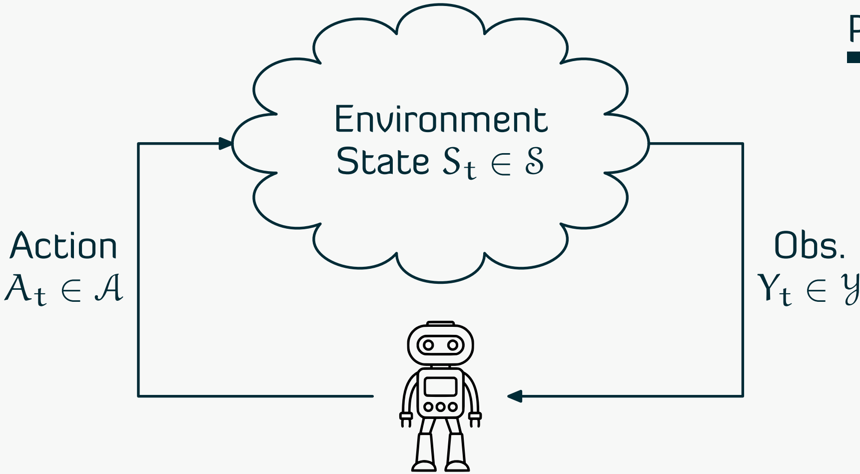
$$V_{T+1}(s) = 0, \qquad V_t(s) = \max_{a \in \mathcal{A}} \Big\{ r(s,a) + \int V_{t+1}(s') P(ds'|s,a) \Big\}$$

Bellman, "Dynamic Programming," 1957.

Review: Partially observable MDPs (POMDPs)



Review: Partially observable MDPs (POMDPs)



POMDP: PARTIALLY OBSERVABLE MDPS

$$\mathbb{P}(S_{t+1}, Y_{t+1} | S_{1:t}, Y_{1:t}, A_{1:t}) = \mathbb{P}(S_{t+1}, Y_{t+1} | S_t, A_t)$$

Reward: $R_t = r(S_t, A_t)$.

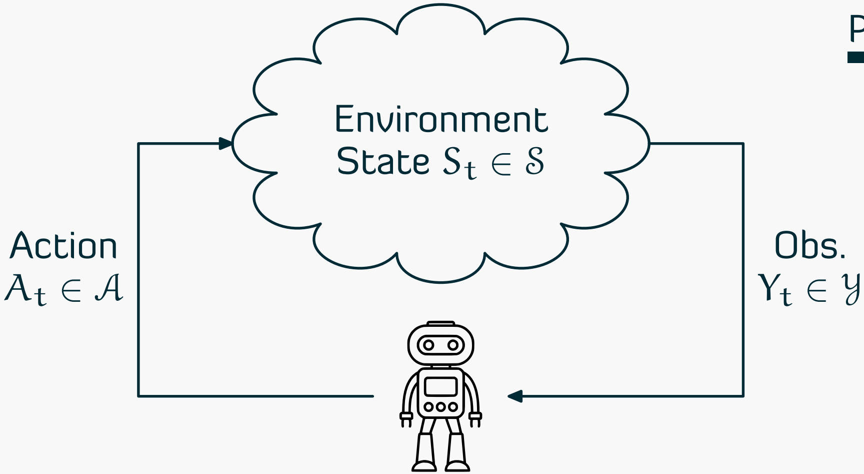
Action: $A_t \sim \pi_t(Y_{1:t}, A_{1:t-1})$.

$\pi = (\pi_1, \dots, \pi_T)$ is called a policy.

Objective choose a policy π to maximize:

$$J(\pi) := \mathbb{E}^{\pi}\Bigg[\sum_{t=1}^{T} R_t\Bigg]$$

Review: Partially observable MDPs (POMDPs)



POMDP: PARTIALLY OBSERVABLE MDPS

$$\mathbb{P}(S_{t+1}, Y_{t+1} | S_{1:t}, Y_{1:t}, A_{1:t}) = \mathbb{P}(S_{t+1}, Y_{t+1} | S_t, A_t)$$

Reward: $R_t = r(S_t, A_t)$.

Action: $A_t \sim \pi_t(Y_{1:t}, A_{1:t-1})$.

$\pi = (\pi_1, \dots, \pi_T)$ is called a policy.

Objective choose a policy π to maximize:

$$J(\pi) := \mathbb{E}^{\pi}\left[\sum_{t=1}^{T} R_t\right]$$

Conceptual challenge

- Brute force search has an exponential complexity in time horizon.
- How to efficiently search for good policies?

Review: Belief-state based planning

Key simplifying idea

Define **belief state** $B_t \in \Delta(\mathcal{S})$ as $B_t(s) = \mathbb{P}(S_t = s \mid Y_{1:t}, A_{1:t-1})$.

- Belief state updates in a state-like manner: $B_{t+1} = \text{function}(B_t, Y_{t+1}, A_t)$.
- Belief state is sufficient to evaluate rewards: $\mathbb{E}[R_t \mid Y_{1:t}, A_{1:t}] = \hat{r}(B_t, A_t)$.

Thus, $\{B_t\}_{t=1}^{T}$ is a perfectly observed controlled Markov process.

Astrom, "Optimal control of Markov processes with incomplete information," JMAA 1965.

Stratonovich, "Conditional Markov Processes," TVP 1960.

Review: Belief-state based planning

Key simplifying idea

Define **belief state** $B_t \in \Delta(\mathcal{S})$ as $B_t(s) = \mathbb{P}(S_t = s \mid Y_{1:t}, A_{1:t-1})$.

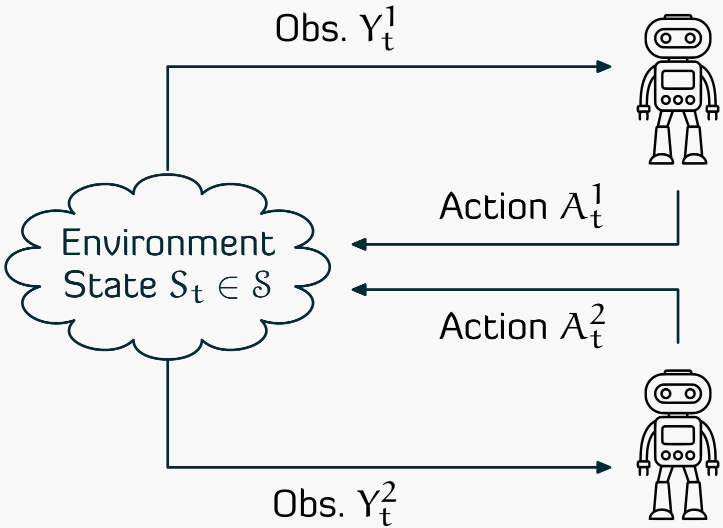
- Belief state updates in a state-like manner: $B_{t+1} = \text{function}(B_t, Y_{t+1}, A_t)$.
- Belief state is sufficient to evaluate rewards: $\mathbb{E}[R_t \mid Y_{1:t}, A_{1:t}] = \hat{r}(B_t, A_t)$.

Thus, $\{B_t\}_{t=1}^{T}$ is a perfectly observed controlled Markov process. Therefore:

Structure of optimal policy	There is no loss of optimality in choosing the action A_t as a function of the belief state B_t
Dynamic Program	The optimal control policy is given by a DP with belief B_t as state.

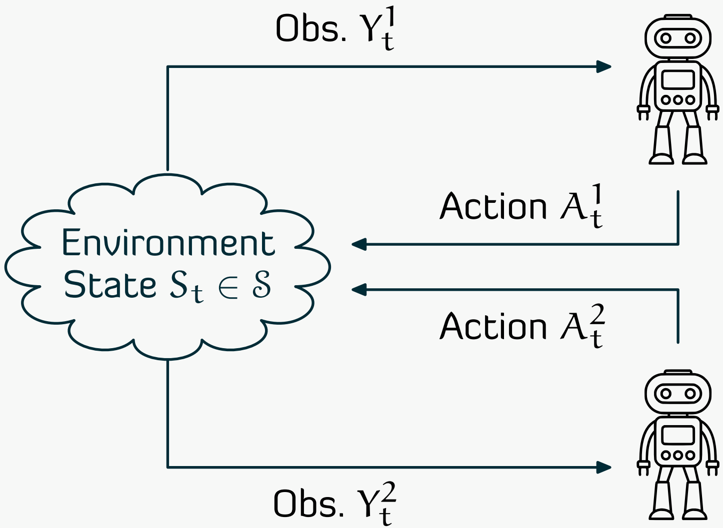
What happens in multi-agent settings?

Dec-POMDPs: Multi-agent partially observed control



Dec-POMDP: DECENTRALIZED POMDPS

Dec-POMDPs: Multi-agent partially observed control



Dec-POMDP: DECENTRALIZED POMDPS

$$\mathbb{P}(S_{t+1}, Y^1_{t+1}, Y^2_{t+1} \mid S_{1:t}, Y^1_{1:t}, Y^2_{1:t}, A^1_{1:t}, A^2_{1:t}) = \mathbb{P}(S_{t+1}, Y^1_{t+1}, Y^2_{t+1} \mid S_t, A^1_t, A^2_t)$$

Obs: Y_t^i for each agent i (private histories H_t^i).

Team reward $R_t = r(S_t, A_t^1, A_t^2)$.

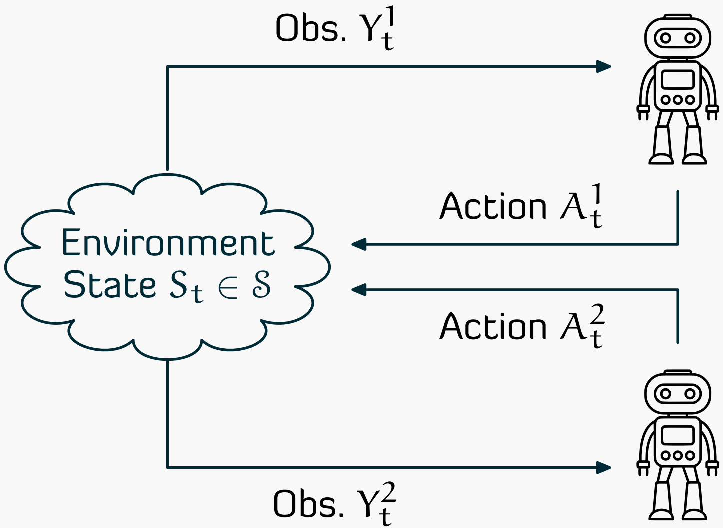
Action: $A_t^i \sim \pi_t^i(H_t^i)$, $\quad H_t^i = (Y^i_{1:t}, A^i_{1:t-1})$.

$\pi^i = (\pi^i_1, \dots, \pi^i_T)$ is agent i's policy.

Objective choose joint policies (π^1, π^2) to maximize:

$$J(\pi^1, \pi^2) := \mathbb{E}^{\pi^1, \pi^2}\left[\sum_{t=1}^{T} R_t\right]$$

Dec-POMDPs: Multi-agent partially observed control



Dec-POMDP: DECENTRALIZED POMDPS

$$\mathbb{P}(S_{t+1}, Y_{t+1}^1, Y_{t+1}^2 \mid S_{1:t}, Y_{1:t}^1, Y_{1:t}^2, A_{1:t}^1, A_{1:t}^2) = \mathbb{P}(S_{t+1}, Y_{t+1}^1, Y_{t+1}^2 \mid S_t, A_t^1, A_t^2)$$

Obs: Y_t^i for each agent i (private histories H_t^i).

Team reward $R_t = r(S_t, A_t^1, A_t^2)$.

Action: $A_t^i \sim \pi_t^i(H_t^i)$, $\quad H_t^i = (Y_{1:t}^i, A_{1:t-1}^i)$.

$\pi^i = (\pi_1^i, \dots, \pi_T^i)$ is agent i's policy.

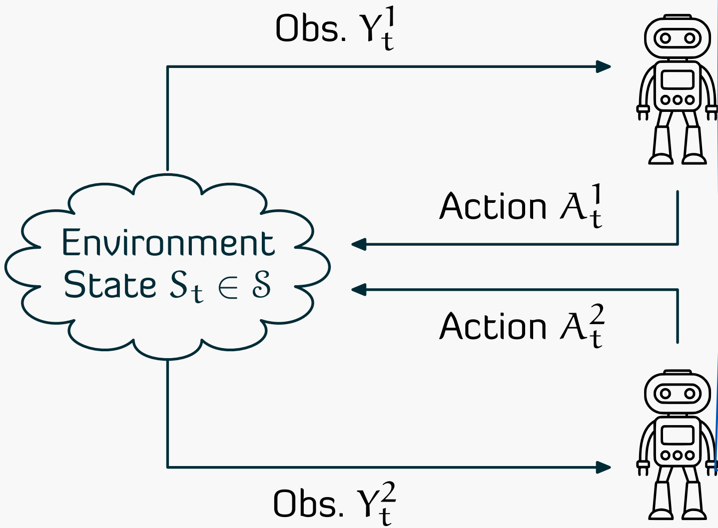
Objective choose joint policies (π^1, π^2) to maximize:

$$J(\pi^1, \pi^2) := \mathbb{E}^{\pi^1, \pi^2}\Big[\sum_{t=1}^{T} R_t\Big]$$

Conceptual challenge

- Brute force search has an exponential complexity in time horizon and number of agents.
- How to efficiently search for good policies?

Dec-POMDPs: Multi-agent partially observed control



Some remarks

- Simplest non-trivial setup for optimization over time with asymmetric information.
- no strategic considerations (cf. non cooperative games)
- no splitting of rewards (cf. cooperative games)

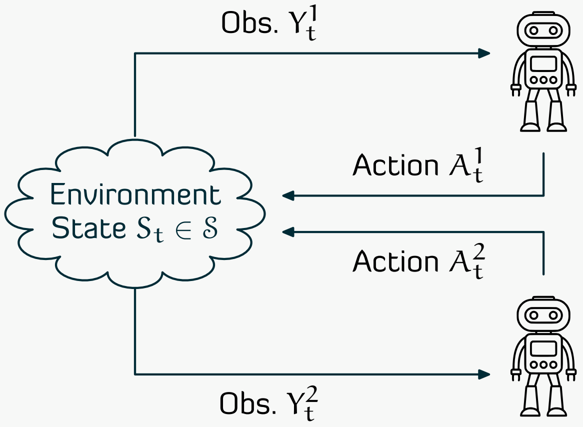
Conceptual challenge

- Brute force search has an exponential complexity in time horizon and number of agents.
- How to efficiently search for good policies?

Fundamental challenge in multi-agent systems

Strategic coupling

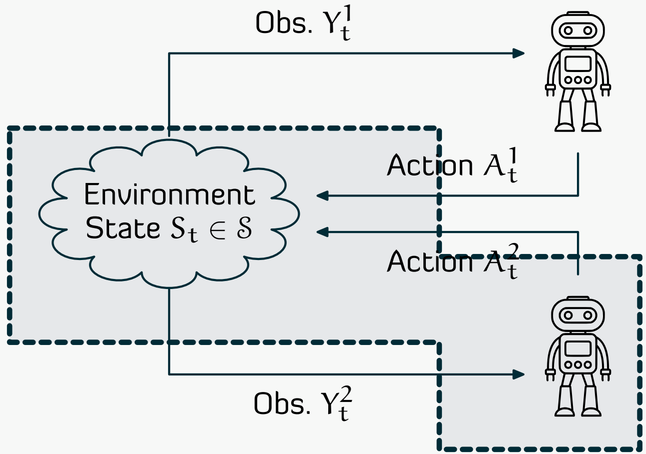
- The non-stationarity problem
- Second-guessing problem
- Signaling aspect of control



Fundamental challenge in multi-agent systems

Strategic coupling

- The non-stationarity problem
- Second-guessing problem
- Signaling aspect of control



From the pov of agent 1

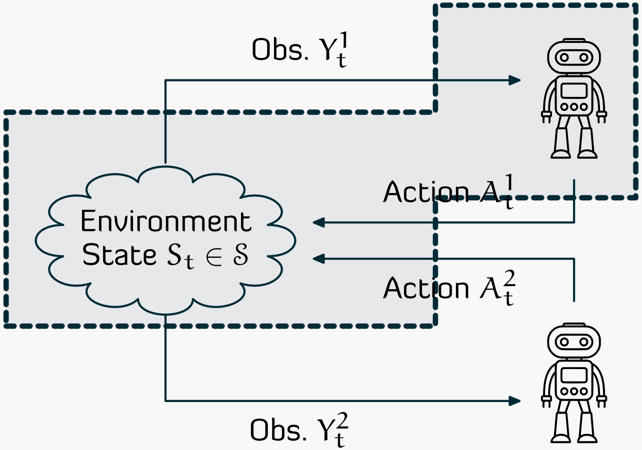
Sufficient statistic

- $b_t^{1,1} = \mathbb{P}(S_t, H_t^2 \mid H_t^1)$

Fundamental challenge in multi-agent systems

Strategic coupling

- ▷ The non-stationarity problem
- ▷ Second-guessing problem
- ▷ Signaling aspect of control



From the pov of agent 1

Sufficient statistic

▷ $b_t^{1,1} = \mathbb{P}(S_t, H_t^2 \mid H_t^1)$

From the pov of agent 2

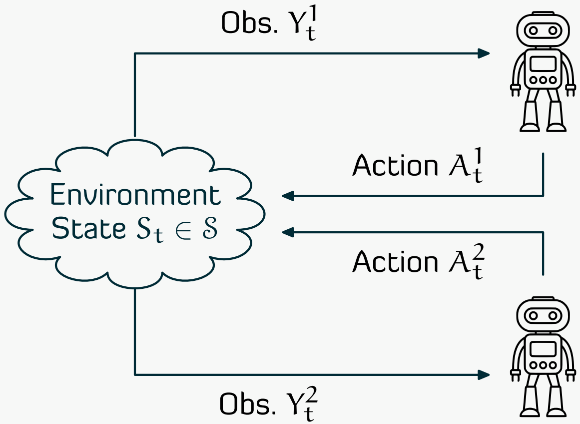
Sufficient statistic

▷ $b_t^{2,1} = \mathbb{P}(S_t, H_t^1 \mid H_t^2)$

Fundamental challenge in multi-agent systems

Strategic coupling

- The non-stationarity problem
- Second-guessing problem
- Signaling aspect of control



From the pov of agent 1

Sufficient statistic

- $b_t^{1,1} = \mathbb{P}(S_t, H_t^2 \mid H_t^1)$
- $b_t^{1,2} = \mathbb{P}(S_t, b_t^{2,1} \mid H_t^1)$

From the pov of agent 2

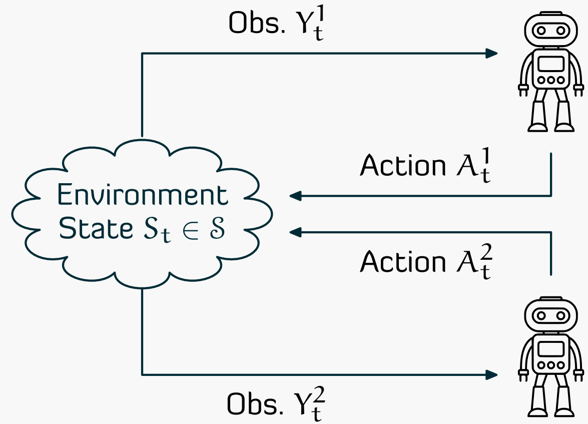
Sufficient statistic

- $b_t^{2,1} = \mathbb{P}(S_t, H_t^1 \mid H_t^2)$
- $b_t^{2,2} = \mathbb{P}(S_t, b_t^{1,1} \mid H_t^2)$

Fundamental challenge in multi-agent systems

Strategic coupling

- The non-stationarity problem
- Second-guessing problem
- Signaling aspect of control



From the pov of agent 1

Sufficient statistic

- $b_t^{1,1} = \mathbb{P}(S_t, H_t^2 \mid H_t^1)$
- $b_t^{1,2} = \mathbb{P}(S_t, b_t^{2,1} \mid H_t^1)$
- ...

From the pov of agent 2

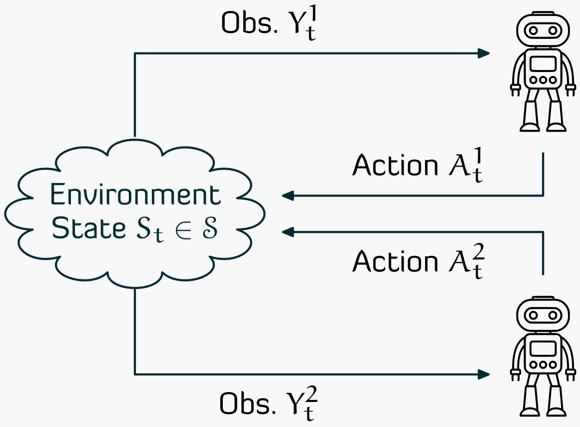
Sufficient statistic

- $b_t^{2,1} = \mathbb{P}(S_t, H_t^1 \mid H_t^2)$
- $b_t^{2,2} = \mathbb{P}(S_t, b_t^{1,1} \mid H_t^2)$
- ...

Fundamental challenge in multi-agent systems

Strategic coupling

- The non-stationarity problem
- Second-guessing problem
- Signaling aspect of control



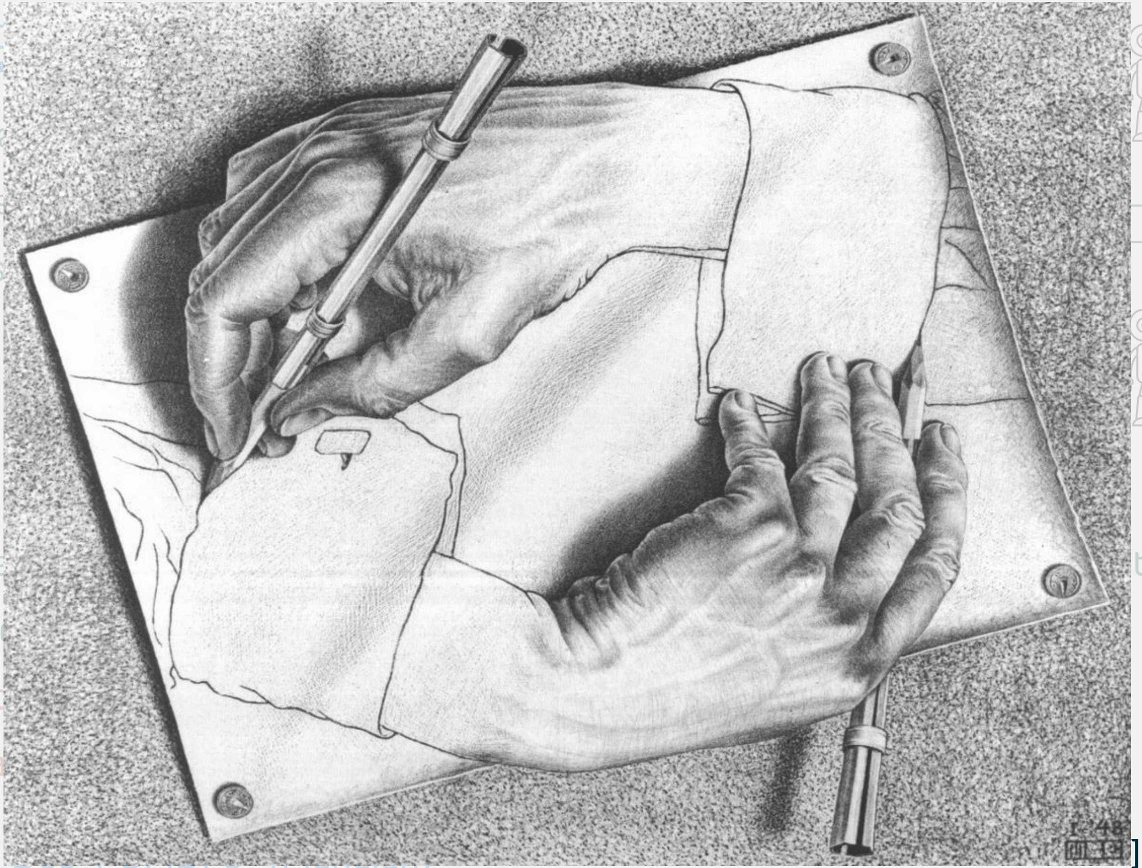
Challenge

- An agent needs to form a belief on the belief of the belief of the ... other agent!
- Gives rise to a nested hierarchy of beliefs.
- How to find "joint" sufficient statistics?
- How to search for optimal policies?

Fundamental challenge in multi-agent systems

Strategic co

- The non-sta
- Second-gue
- Signaling as



Challenge

- An agent ne ... t!
- Gives rise t
- How to find
- How to sea

Revisit the modeling assumptions

Why assume that agents have perfect recall?

Why do we make this assumption?

- Perfect recall is an idealized assumption. No real system can have infinite memory.
- In single-agent systems (POMDPs), this assumption simplifies the analysis and leads to a policy that can be implemented with fixed memory.
- But this assumption leads to conceptual headaches in multi-agent systems!

Why assume that agents have perfect recall?

Why do we make this assumption?

- Perfect recall is an idealized assumption. No real system can have infinite memory.
- In single-agent systems (POMDPs), this assumption simplifies the analysis and leads t[...] mented with fixed memory.
- But this assumption leads to [...] multi-agent systems!

Obvious fix: Don't make the assumption!

- Simply assume that all agents have fixed memory:
  - finite window of past observations
  - finite state machines
  - RNNs
- Similar assumption made in many recent RL papers
- **Key challenge:** decision process is no longer Markov!

How to search for optimal agent-state policies

How to search for optimal agent-state policies

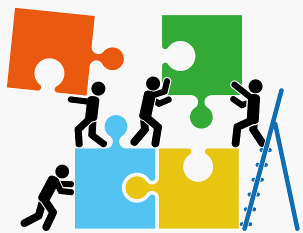
Searching for optimal policies

- Designer's approach
- Simple idea, hard to scale

How to search for optimal agent-state policies

Searching for optimal policies

- Designer's approach
- Simple idea, hard to scale



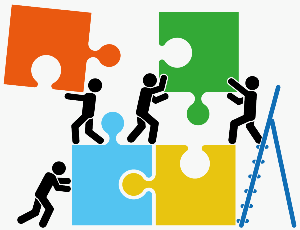
Searching for approximately optimal policies

- Approximate information state
- Works for POMDPs, doesn't work for Dec-POMDPs

How to search for optimal agent-state policies

Searching for optimal policies

- Designer's approach
- Simple idea, hard to scale



Searching for approximately optimal policies

- Approximate information state
- Works for POMDPs, doesn't work for Dec-POMDPs

Searching for team sequential equilibrium

- . . . with some additional ingredients
- Works remarkably well for POMDPs and Dec-POMDPs

How to search for optimal agent-state policies

Searching for optimal policies

- Designer's approach
- Simple idea, hard to scale

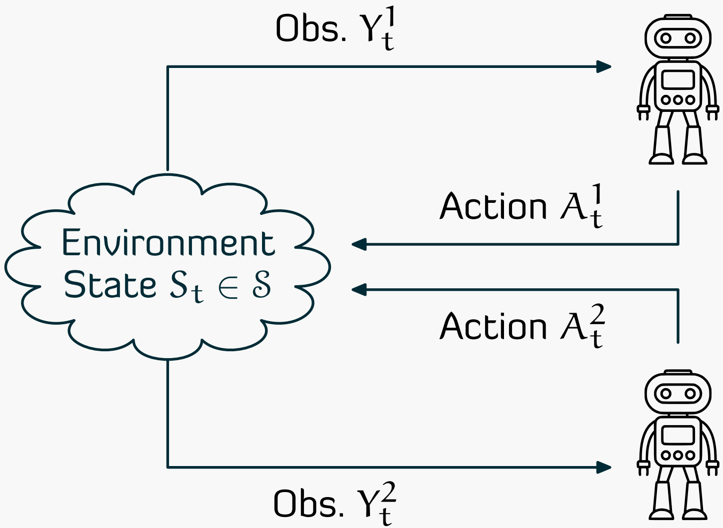
Searching for approximately optimal policies

- Approximate information state
- Works for POMDPs, doesn't work for Dec-POMDPs

Searching for team sequential equilibrium

- . . . with some additional ingredients
- Works remarkably well for POMDPs and Dec-POMDPs

Dec-POMDPs with agent-state based policies

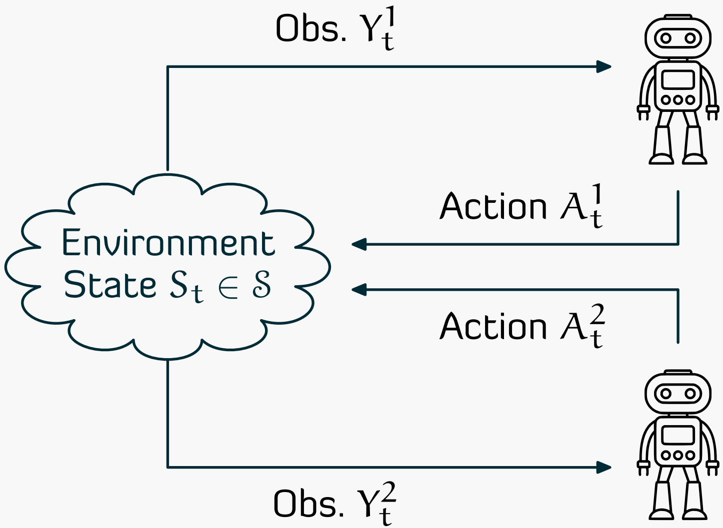


Agent state: $Z_t^i \in \mathcal{Z}^i$, where

$$(A_t^i, Z_{t+1}^i) = \pi_t^i(Y_t^i, Z_t^i)$$

- Can fix agent-state update function, e.g., $Z_t^i = (Y_{t-k:t}^i, A_{t-k:t-1}^i)$
- Agent state may be real-valued (e.g., RNNs)

Dec-POMDPs with agent-state based policies



Agent state: $Z_t^i \in \mathcal{Z}^i$, where

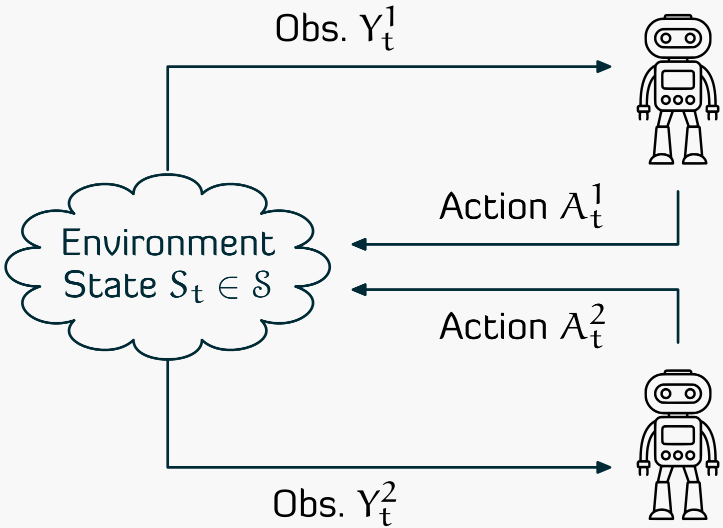
$$(A_t^i, Z_{t+1}^i) = \pi_t^i(Y_t^i, Z_t^i)$$

- Can fix agent-state update function, e.g., $Z_t^i = (Y_{t-k:t}^i, A_{t-k:t-1}^i)$
- Agent state may be real-valued (e.g., RNNs)

Objective choose joint (π^1, π^2) to maximize:

$$J(\pi^1, \pi^2) := \mathbb{E}^{\pi^1, \pi^2}\left[\sum_{t=1}^{T} R_t\right]$$

Dec-POMDPs with agent-state based policies



Agent state: $Z^i_t \in \mathcal{Z}^i$, where

$$(A^i_t, Z^i_{t+1}) = \pi^i_t(Y^i_t, Z^i_t)$$

- Can fix agent-state update function, e.g., $Z^i_t = (Y^i_{t-k:t}, A^i_{t-k:t-1})$
- Agent state may be real-valued (e.g., RNNs)

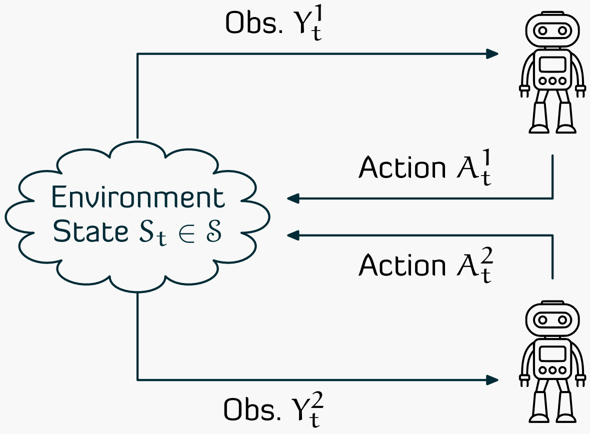
Objective choose joint (π^1, π^2) to maximize:

$$J(\pi^1, \pi^2) := \mathbb{E}^{\pi^1, \pi^2}\left[\sum_{t=1}^{T} R_t\right]$$

Conceptual challenge

- Brute force search has an exponential complexity in time horizon and number of agents.
- How to efficiently search for good agent-state policies?

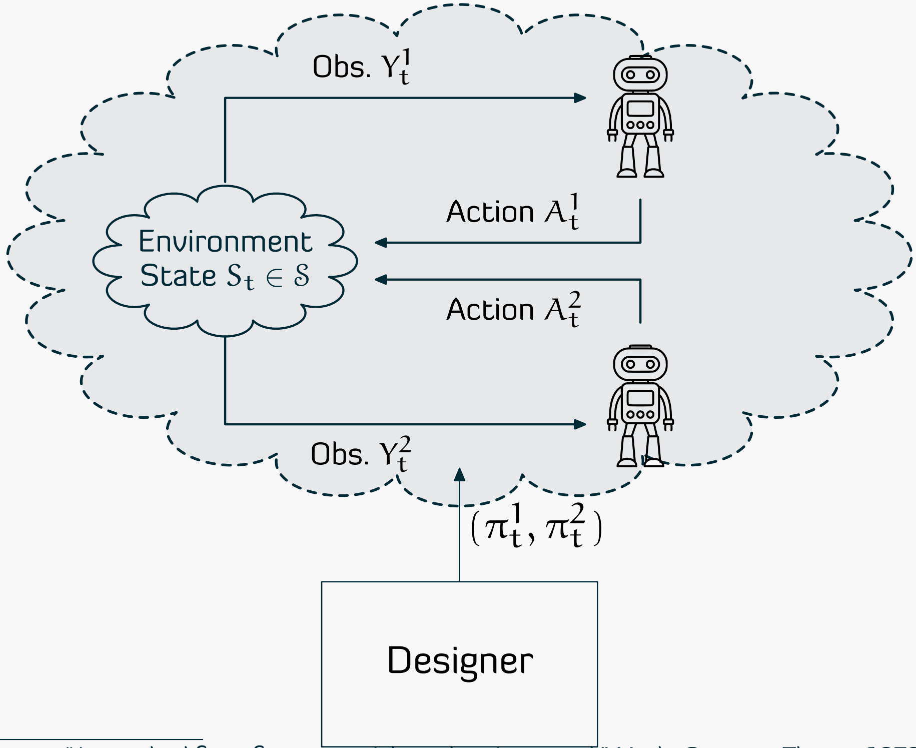
Designer's approach to find optimal policy



Witsenhausen, "A standard form for sequential stochastic control," Math. Systems Theory, 1973.

Mahajan, "Sequential decomposition of sequential teams", PhD thesis, 2008.

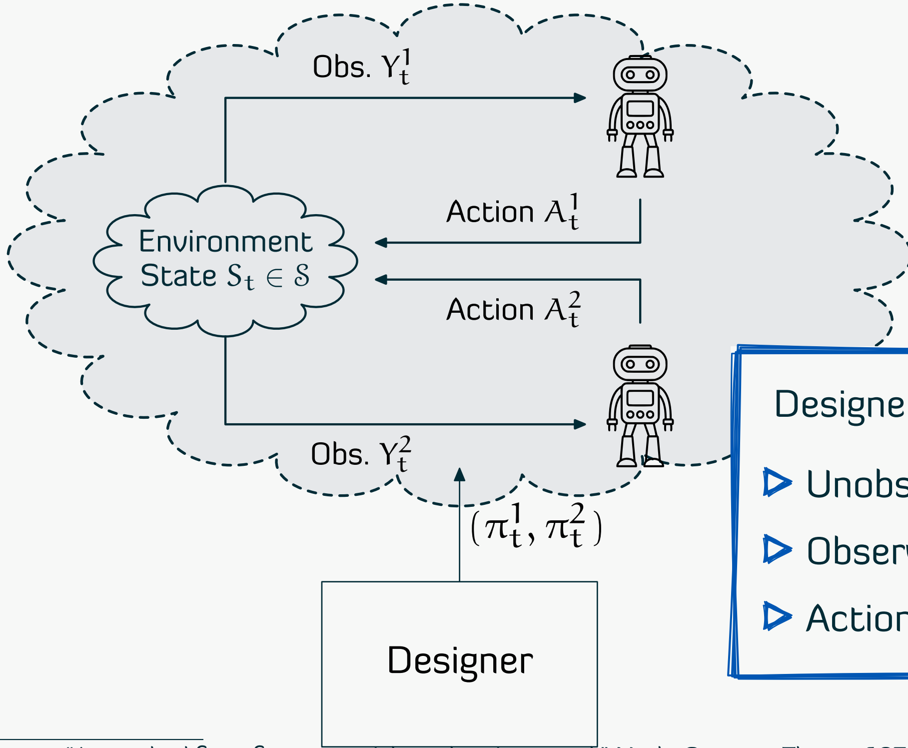
Designer's approach to find optimal policy



Witsenhausen, "A standard form for sequential stochastic control," Math. Systems Theory, 1973.

Mahajan, "Sequential decomposition of sequential teams", PhD thesis, 2008.

Designer's approach to find optimal policy



Designer's problem is a POMDP with:

- Unobserved state: (S_t, Z_t^1, Z_t^2)
- Observations: $\emptyset$
- Action: (π_t^1, π_t^2)

Witsenhausen, "A standard form for sequential stochastic control," Math. Systems Theory, 1973.

Mahajan, "Sequential decomposition of sequential teams", PhD thesis, 2008.

Designer's approach to find optimal policy

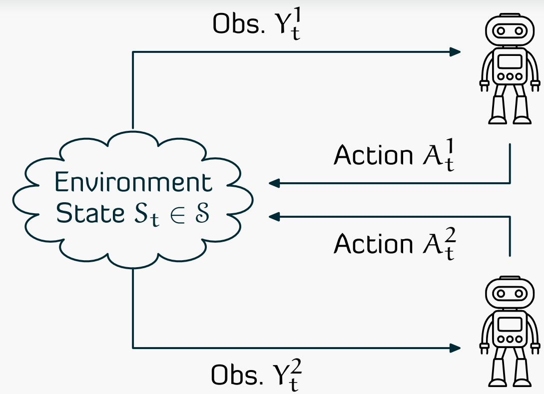
Joint distribution of env and agent states

For any $\boldsymbol{\pi} = (\boldsymbol{\pi}^1, \boldsymbol{\pi}^2)$, define

$$\xi^{\boldsymbol{\pi}}_t(s, z^1, z^2) := \mathbb{P}^{\boldsymbol{\pi}}(S_t = s, Z^1_t = z^1, Z^2_t = z^2)$$

Then:

- $\xi^{\boldsymbol{\pi}}_{t+1} = \phi_{\mathsf{DES}}(\xi^{\boldsymbol{\pi}}_t, \boldsymbol{\pi}_t)$.
- $\mathbb{E}^{\boldsymbol{\pi}}[R_t] = r_{\mathsf{DES}}(\xi^{\boldsymbol{\pi}}_t, \boldsymbol{\pi}_t)$.



Mahajan, "Sequential decomposition of sequential teams", PhD thesis, 2008.

Designer's approach to find optimal policy

Joint distribution of env and agent states

For any $\boldsymbol{\pi} = (\boldsymbol{\pi}^1, \boldsymbol{\pi}^2)$, define

$$\xi_t^{\boldsymbol{\pi}}(s, z^1, z^2) := \mathbb{P}^{\boldsymbol{\pi}}(S_t = s, Z_t^1 = z^1, Z_t^2 = z^2)$$

Then:

- $\xi_{t+1}^{\boldsymbol{\pi}} = \phi_{\mathsf{DES}}(\xi_t^{\boldsymbol{\pi}}, \boldsymbol{\pi}_t)$.
- $\mathbb{E}^{\boldsymbol{\pi}}[R_t] = r_{\mathsf{DES}}(\xi_t^{\boldsymbol{\pi}}, \boldsymbol{\pi}_t)$.

DP using designer's approach

Consider the following DP:

$$V_{\mathsf{DES},t}(\xi) = \max_{\boldsymbol{\pi}} \big\{ r_{\mathsf{DES}}(\boldsymbol{\pi}, \xi) + V_{\mathsf{DES},t+1}(\phi_{\mathsf{DES}}(\boldsymbol{\pi}, \xi)) \big\}.$$

Let $\psi_{\mathsf{DES},t}(\xi)$ denote any arg max of the RHS. Let $\xi_1^* = \xi_1$ and recursively define

$$\boldsymbol{\pi}_t^* = \psi_{\mathsf{DES},t}(\xi_t^*) \quad \text{and} \quad \xi_{t+1}^* = \phi_{\mathsf{DES},t}(\boldsymbol{\pi}_t^*, \xi_t^*).$$

Then, the policy $\boldsymbol{\pi}^* = (\boldsymbol{\pi}_1^*, \dots, \boldsymbol{\pi}_T^*)$ is optimal.

Mahajan, "Sequential decomposition of sequential teams", PhD thesis, 2008.

Some comments

Historical review

- The idea goes back to Witsenhausen's standard form (1973).
- Used for POMDPs in Sandell (1974) and general finite state Dec-POMDPs in Mahajan (2008).
- Related to Observational MDP approach of Dibangoye et al (2016).

Some comments

Historical review

- The idea goes back to Witsenhausen's standard form (1973).
- Used for POMDPs in Sandell (1974) and general finite state Dec-POMDPs in Mahajan (2008).
- Related to Observational MDP approach of Dibangoye et al (2016).

Implications

- Provides a DP to find the optimal policy.
- Using DP, we can show that the optimal policy is deterministic.

Some comments

Historical review

- The idea goes back to Witsenhausen's standard form (1973).
- Used for POMDPs in Sandell (1974) and general finite state Dec-POMDPs in Mahajan (2008).
- Related to Observational MDP approach of Dibangoye et al (2016).

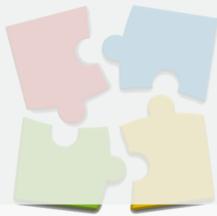
Implications

- Provides a DP to find the optimal policy.
- Using DP, we can show that the optimal policy is deterministic.

Limitations

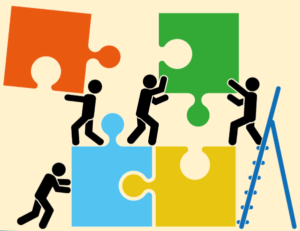
- The action space of the DP is the space of all policies (π^1_t, π^2_t).
- Difficult to scale using standard sampling-based methods.

How to search for optimal agent-state policies



Searching for optimal policies

- Designer's approach
- Simple idea, hard to scale



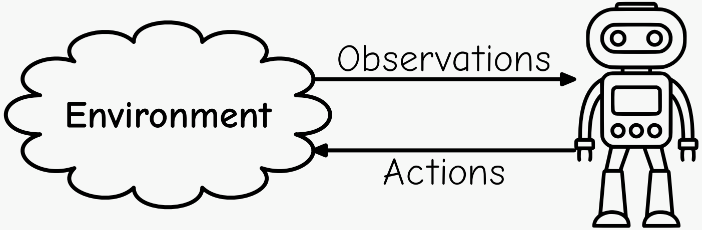
Searching for approximately optimal policies

- Approximate information state
- Works for POMDPs, doesn't work for Dec-POMDPs

Searching for team sequential equilibrium

- . . . with some additional ingredients
- Works remarkably well for POMDPs and Dec-POMDPs

Simplify: Consider POMDPs with agent-state policies



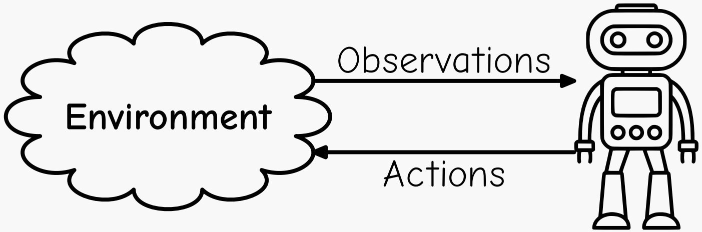
Further simplification: Fix agent state update

$$Z_{t+1} = \varphi_t(Z_t, Y_{t+1}, A_t)$$

By unrolling this assumption, we can say

$$Z_t = \sigma_t(H_t)$$

Simplify: Consider POMDPs with agent-state policies



Further simplification: Fix agent state update

$$Z_{t+1} = \varphi_t(Z_t, Y_{t+1}, A_t)$$

By unrolling this assumption, we can say

$$Z_t = \sigma_t(H_t)$$

Approximate information state (AIS)

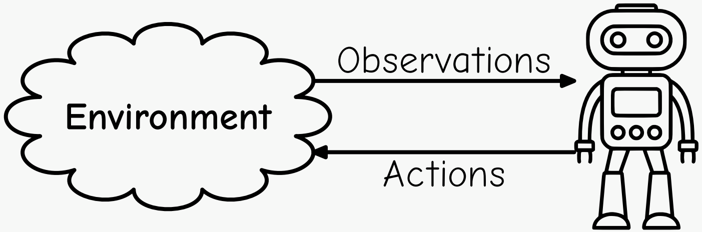
$\{Z_t\}_{t=1}^{T}$ is an (ε, δ)-AIS, where $\varepsilon = (\varepsilon_1, \dots, \varepsilon_T)$ and $\delta = (\delta_1, \dots, \delta_T)$, if there exist

$$\underbrace{r_{\mathrm{AIS},t} \colon \mathcal{Z} \times \mathcal{A} \to \mathbb{R}}_{\text{approx reward estimator}} \quad \text{and} \quad \underbrace{P_{\mathrm{AIS},t} \colon \mathcal{Z} \times \mathcal{A} \to \Delta(\mathcal{Z})}_{\text{approx dynamic pred}}$$

such that

Subramanian, Sinha, Seraj, and Mahajan, “Approximate information state for . . . partially observed systems”, JMLR 2022.

Simplify: Consider POMDPs with agent-state policies



Further simplification: Fix agent state update

$$Z_{t+1} = \varphi_t(Z_t, Y_{t+1}, A_t)$$

By unrolling this assumption, we can say

$$Z_t = \sigma_t(H_t)$$

Approximate information state (AIS)

$\{Z_t\}_{t=1}^{T}$ is an (ε, δ)-AIS, where $\varepsilon = (\varepsilon_1, \dots, \varepsilon_T)$ and $\delta = (\delta_1, \dots, \delta_T)$, if there exist

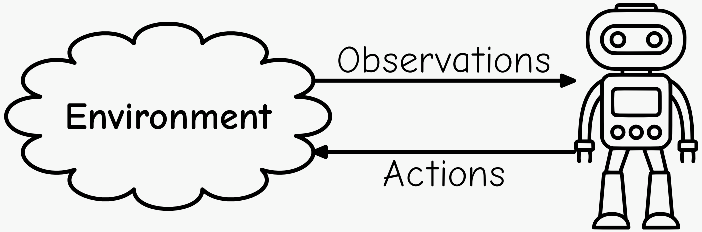
$$\underbrace{r_{\mathrm{AIS},t} \colon \mathcal{Z} \times \mathcal{A} \to \mathbb{R}}_{\text{approx reward estimator}} \quad \text{and} \quad \underbrace{P_{\mathrm{AIS},t} \colon \mathcal{Z} \times \mathcal{A} \to \Delta(\mathcal{Z})}_{\text{approx dynamic pred}}$$

such that

- reward est. is approx sufficient stat : $\big|\mathbb{E}[R_t \mid h_t, a_t] - r_{\mathrm{AIS},t}(\sigma_t(h_t), a_t)\big| \leqslant \varepsilon_t$
- dynamics are approx self predictive : $d(\mathbb{P}(Z_{t+1} \mid h_t, a_t), P_{\mathrm{AIS},t}(Z_{t+1} \mid \sigma_t(h_t), a_t)) \leqslant \delta_t$

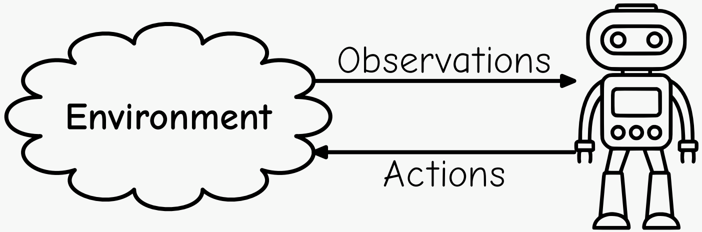
Subramanian, Sinha, Seraj, and Mahajan, "Approximate information state for . . . partially observed systems", JMLR 2022.

Approx solutions for POMDPs with agent-state policies



- The pair $(P_{\text{AIS}}, r_{\text{AIS}})$ defines an MDP.
- Let π_{AIS} be the optimal policy of this MDP
- Then $\pi_t(h_t) = \pi_{\text{AIS},t}(\sigma_t(h_t))$ is a feasible agent-state policy.

Approx solutions for POMDPs with agent-state policies



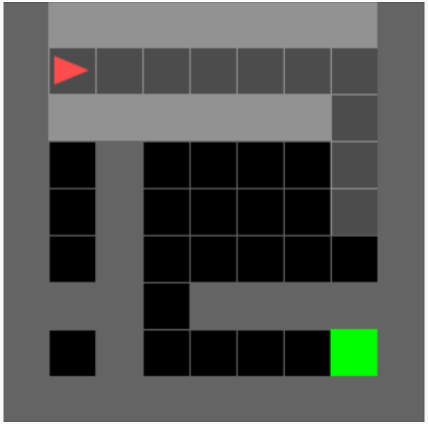
- The pair $(P_{\text{AIS}}, r_{\text{AIS}})$ defines an MDP.
- Let π_{AIS} be the optimal policy of this MDP
- Then $\pi_t(h_t) = \pi_{\text{AIS},t}(\sigma_t(h_t))$ is a feasible agent-state policy.

Main result

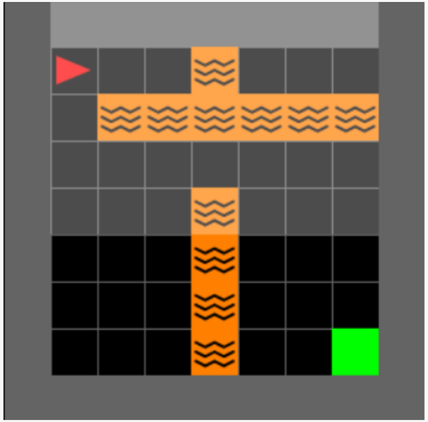
- The agent-state policy π is α-optimal, where α depends on (ε, δ) and properties of V_{AIS} (depending on the choice of metric d)
- Can use $\lambda\varepsilon^2 + (1-\lambda)\delta^2$ as an auxiliary loss for RL algorithms.
- Works beautifully for Q-learning and actor-critic algorithms

Subramanian, Sinha, Seraj, and Mahajan, "Approximate information state for . . . partially observed systems", JMLR 2022.
Ni, et al., "Bridging State and History Representations: Understanding self-predictive RL", ICLR 2024.
Sinha, Geist, Mahajan, "Periodic agent-state based Q-learning for POMDPs", NeurIPS 2024.

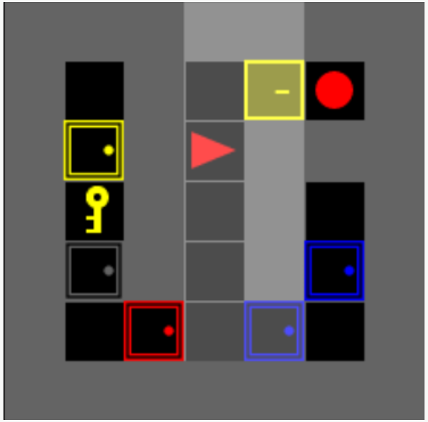
MiniGrid Environments



Simple Crossing

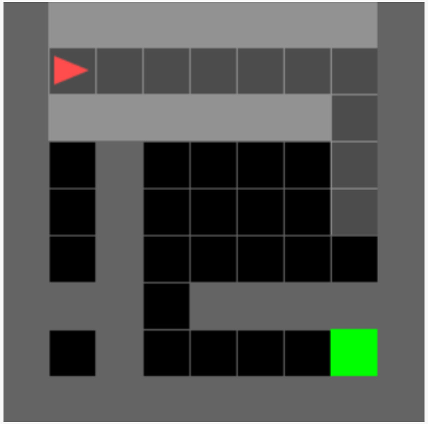


Lava Crossing

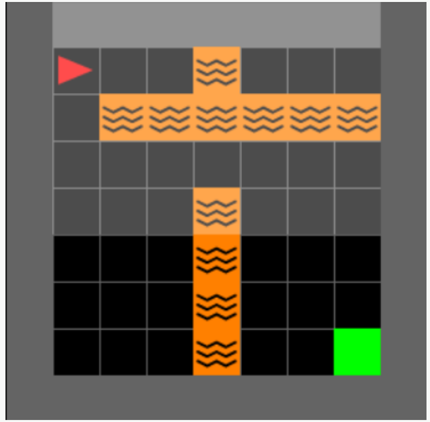


Key Corridor

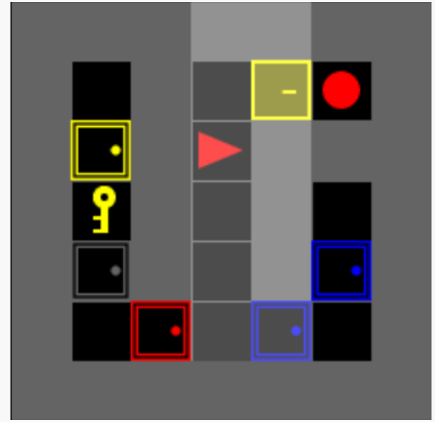
MiniGrid Environments



Simple Crossing



Lava Crossing

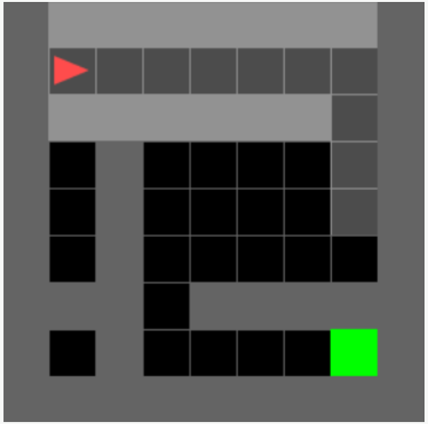


Key Corridor

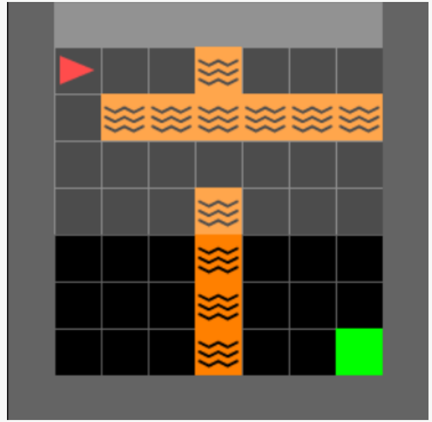
Features

- Partially observable 2D grids. Agent has a view of a 7×7 field in front of it. Observations are obstructed by walls.
- Multiple entities (agents, walls, lava, boxes, doors, and keys)
- Multiple actions (Move Forward, Turn Left, Turn Right, Open Door/Box, . . .)

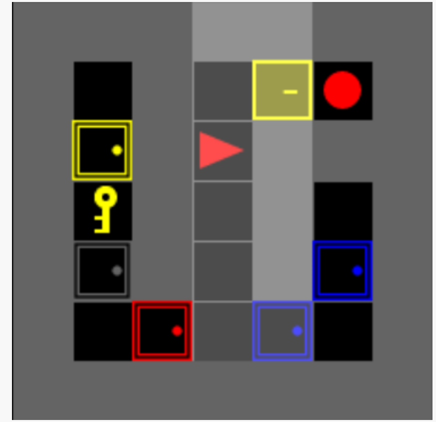
MiniGrid Environments



Simple Crossing



Lava Crossing



Key Corridor

Algorithms

AIS + MMD

AIS with MMD as IPM

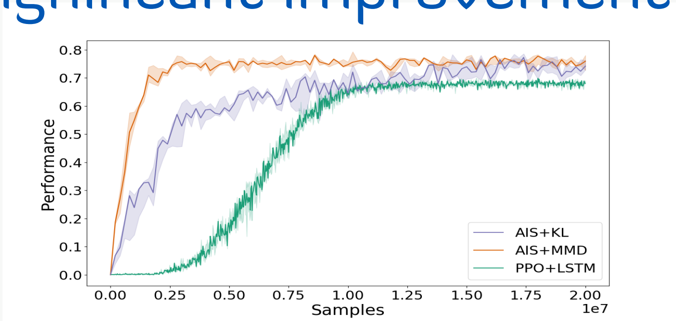
AIS + KL

AIS with KL as upper bound of Wasserstein distance

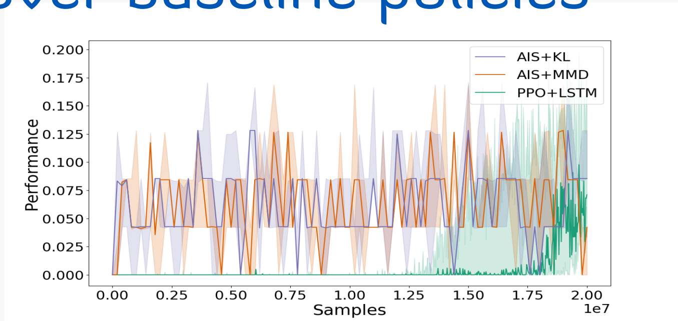
PPO + LSTM

Baseline proposed in paper introducing minigrid envs

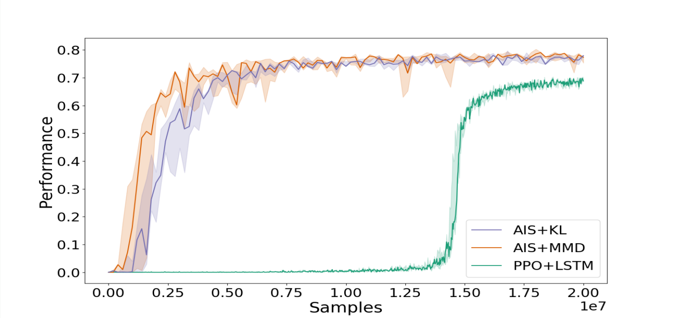
Significant improvement over baseline policies



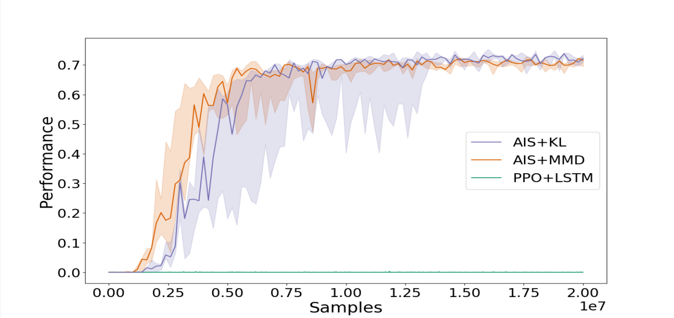
Simple Crossing S9N3



Lava Crossing S9N2



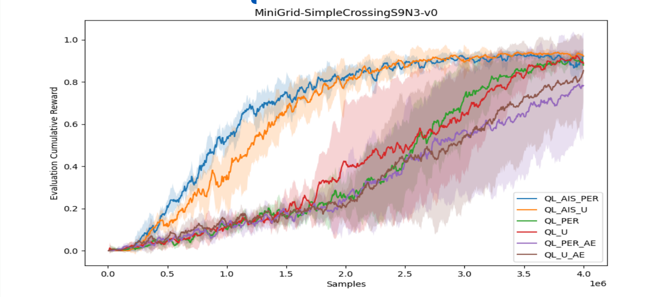
Doorkey 8x8



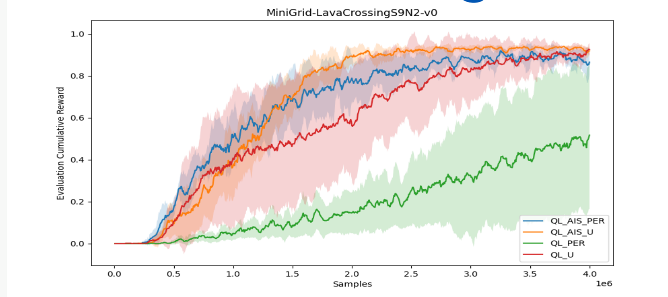
Key Corridor S3R2

Subramanian, Sinha, Seraj, and Mahajan, “Approximate information state for . . . partially observed systems”, JMLR 2022.

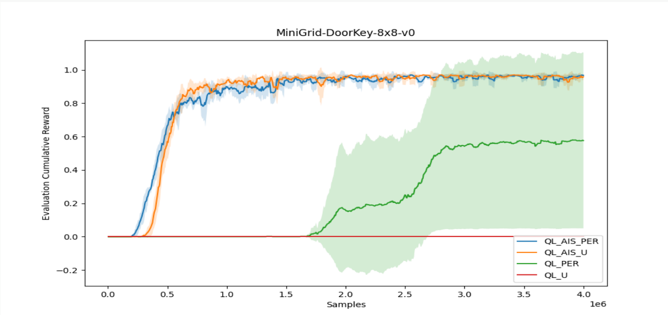
Similar improvements for AIS-enhanced Q-learning



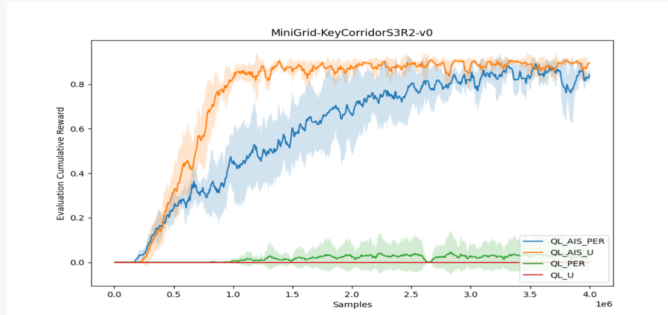
Simple Crossing S9N3



Lava Crossing S9N2



Doorkey 8x8

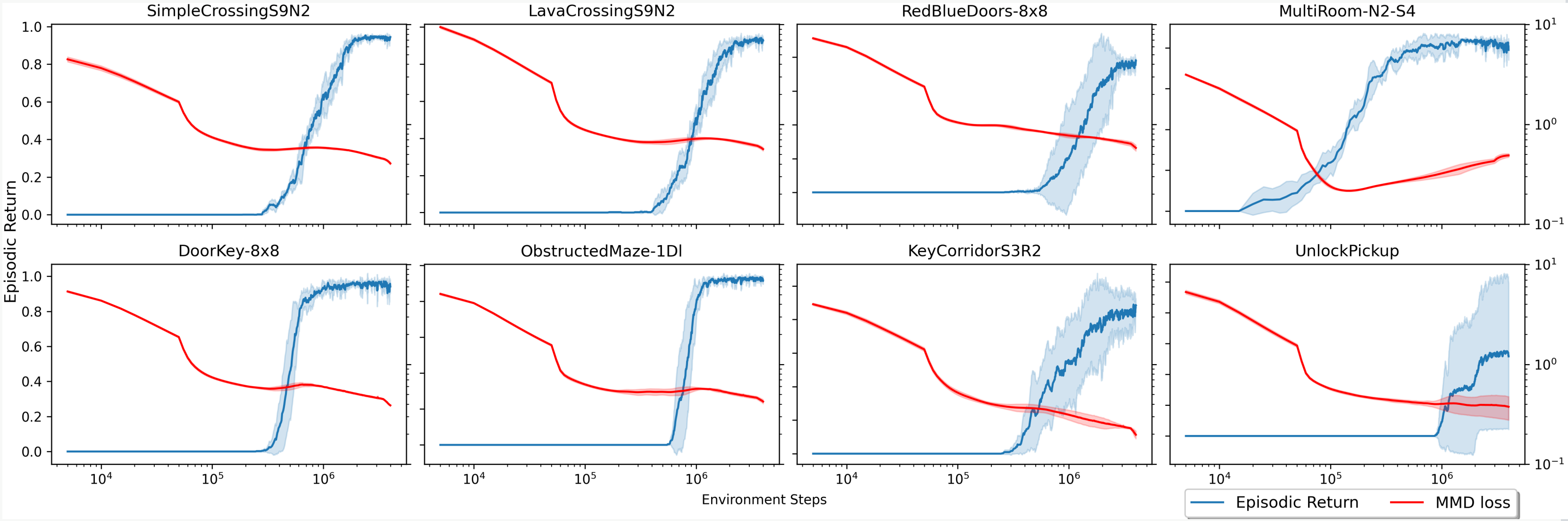


Key Corridor S3R2

Ni, et al., “Bridging State and History Representations: Understanding self-predictive RL”, ICLR 2024.

Sinha, Geist, Mahajan, “Periodic agent-state based Q-learning for POMDPs”, NeurIPS 2024.

AIS loss vs. performance



Sayedsalehi , Akbarzadeh, Sinha , and Mahajan, “AIS based convergence analysis of recurrent Q-learning”, EWRL 2023.

How do we extend the notion of AIS to multi-agent settings?

How do we extend the notion of AIS to multi-agent settings?

Some theoretical results using common-information approach, but difficult to scale to solve common benchmarks.

How to search for optimal agent-state policies

Searching for optimal policies

- Designer's approach
- Simple idea, hard to scale

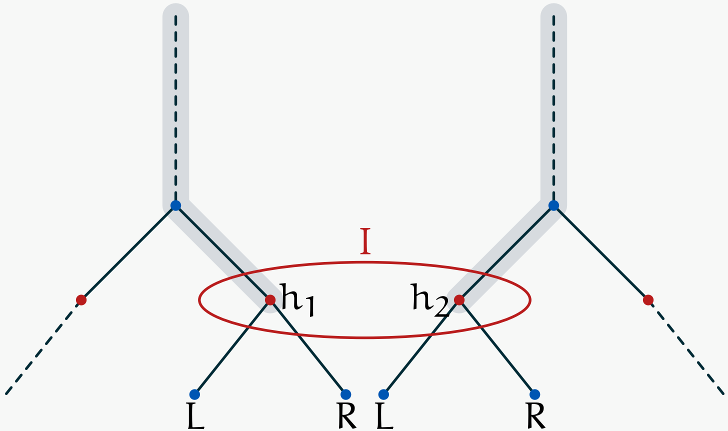
Searching for approximately optimal policies

- Approximate information state
- Works for POMDPs, doesn't work for Dec-POMDPs

Searching for team sequential equilibrium

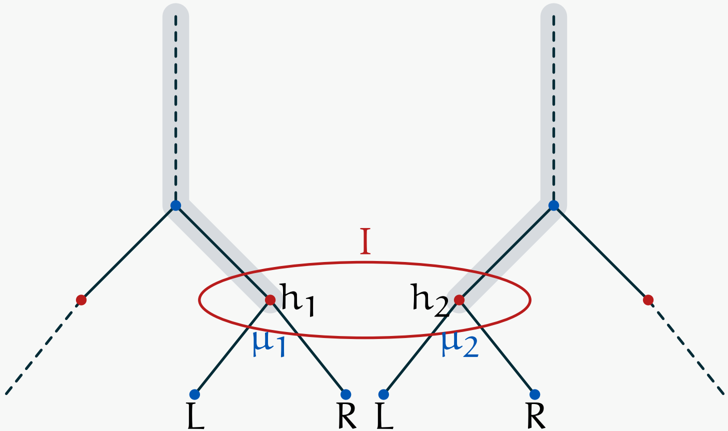
- . . . with some additional ingredients
- Works remarkably well for POMDPs and Dec-POMDPs

Dealing with information decentralization in Game Theory



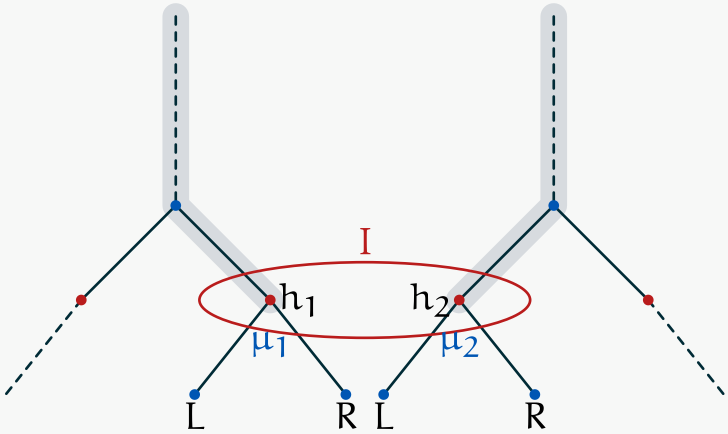
- $Q(I, L) = ?$
- $Q(I, R) = ?$

Dealing with information decentralization in Game Theory



- $Q(I, L) = \mu_1 u(h_1, L) + \mu_2 u(h_2, L)$
- $Q(I, R) = \mu_1 u(h_1, R) + \mu_2 u(h_2, R)$

Dealing with information decentralization in Game Theory

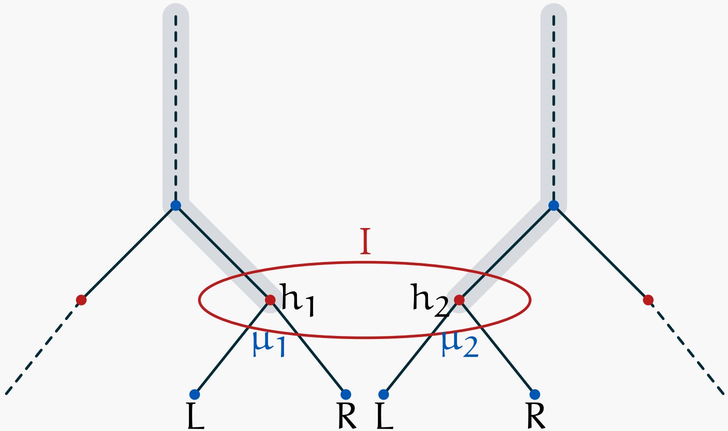


Games with asymmetric information

- Equilibrium is no longer just a strategy profile
- (strategy profile, **belief system**)
- Beliefs need to be consistent with strategy
- Strategies are best response to beliefs, where agents average at info sets using beliefs

- $Q(I, L) = \mu_1 u(h_1, L) + \mu_2 u(h_2, L)$
- $Q(I, R) = \mu_1 u(h_1, R) + \mu_2 u(h_2, R)$

Dealing with information decentralization in Game Theory



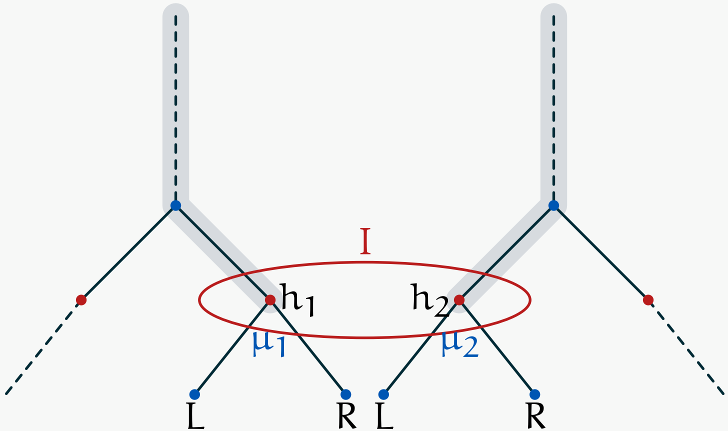
Games with asymmetric information

- Equilibrium is no longer just a strategy profile
- (strategy profile, **belief system**)
- Beliefs need to be consistent with strategy
- Strategies are best response to beliefs, where agents average at info sets using beliefs

Solution concepts

- Perfect Bayesian Equilibrium
- Sequential Equilibrium
- ...

Dealing with information decentralization in Game Theory



Games with asymmetric information

- Equilibrium is no longer just a strategy profile
- (strategy profile, **belief system**)
- Beliefs need to be consistent with strategy
- Strategies are best response to beliefs, where agents average at info sets using beliefs

Solution concepts

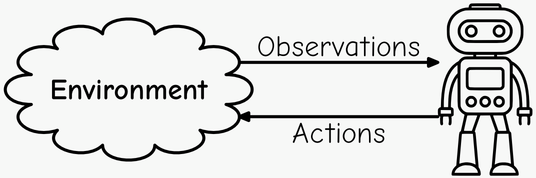
- Perfect Bayesian Equilibrium
- Sequential Equilibrium
- . . .

Omitting several details

- Beliefs are updated according to Bayes' rule
- It is assumed that agents must have perfect recall
- Need delicate limit arguments when info sets have zero probability

Team sequential equilibrium for POMDPs

Team sequential equilibrium



A team sequential equilibrium is a collection of

- a policy $\boldsymbol{\pi} = (\pi_1, \dots, \pi_T)$
- a belief system $\boldsymbol{\xi} = (\xi_1, \dots, \xi_T)$

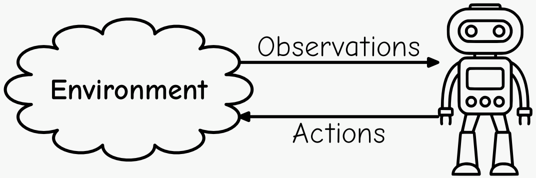
such that

- Sequential rationality The policy π_t is best response when averaged using beliefs.
- Consistency **How do we define consistency?**

  What does Bayesian update mean for a model without perfect recall?

Team sequential equilibrium for POMDPs

Team sequential equilibrium



A team sequential equilibrium is a collection of

- a policy $\boldsymbol{\pi} = (\pi_1, \dots, \pi_T)$
- a belief system $\boldsymbol{\xi} = (\xi_1, \dots, \xi_T)$

such that

- Sequential rationality The policy π_t is best response when averaged using beliefs.
- Consistency **How do we define consistency?**

  What does Bayesian update mean for a model without perfect recall?

Main idea: Track unconditional "beliefs"

How to find team sequential equilibrium for POMDPs

Notation

- $X_t = (S_t, Y_t, Z_t)$
- $U_t = (A_t, Z_{t+1})$

How to find team sequential equilibrium for POMDPs

Notation

- $X_t = (S_t, Y_t, Z_t)$
- $U_t = (A_t, Z_{t+1})$

Observation

- $\{X_t\}_{t \geqslant 1}$ is a controlled Markov process, controlled by $\{U_t\}_{t \geqslant 1}$
- However, $\pi_t \colon X_t \mapsto U_t$ is **not a valid policy**
- $\pi_t \colon (Y_t, Z_t) \mapsto U_t$ can be evaluated using standard formulas

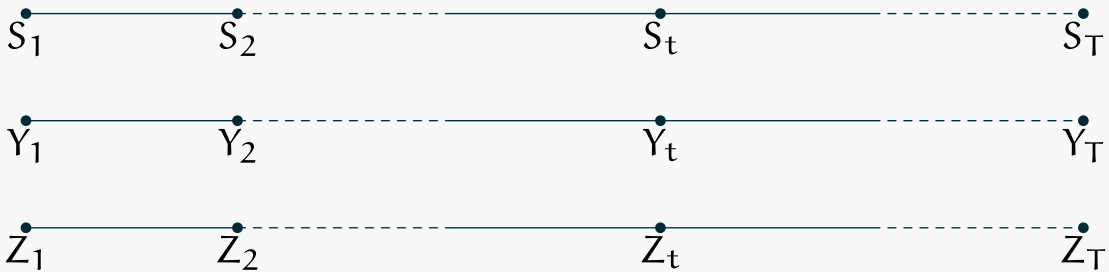
How to find team sequential equilibrium for POMDPs

Notation

- $X_t = (S_t, Y_t, Z_t)$
- $U_t = (A_t, Z_{t+1})$

Observation

- $\{X_t\}_{t \geqslant 1}$ is a controlled Markov process, controlled by $\{U_t\}_{t \geqslant 1}$
- However, $\pi_t \colon X_t \mapsto U_t$ is **not a valid policy**
- $\pi_t \colon (Y_t, Z_t) \mapsto U_t$ can be evaluated using standard formulas



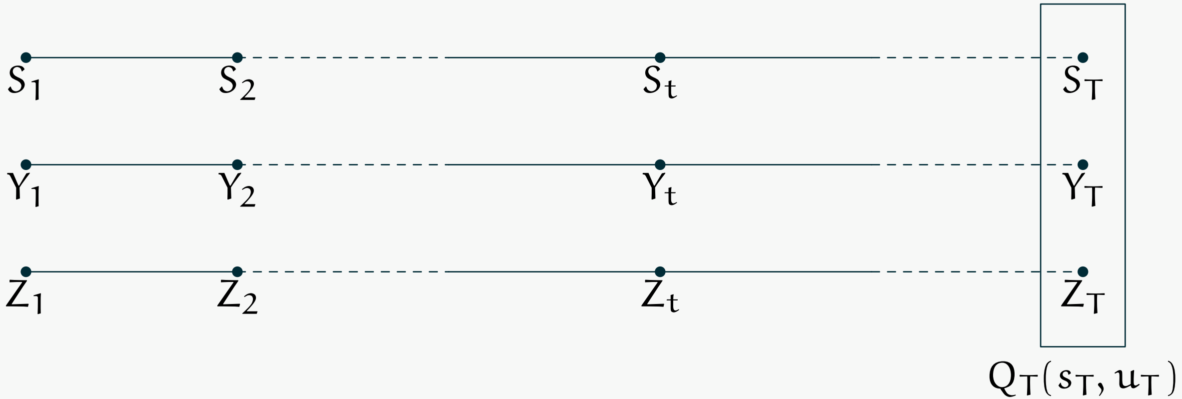
How to find team sequential equilibrium for POMDPs

Notation

- $X_t = (S_t, Y_t, Z_t)$
- $U_t = (A_t, Z_{t+1})$

Step 1: Backward pass

- Compute policy evaluation Q functions $Q_t(x_t, u_t)$



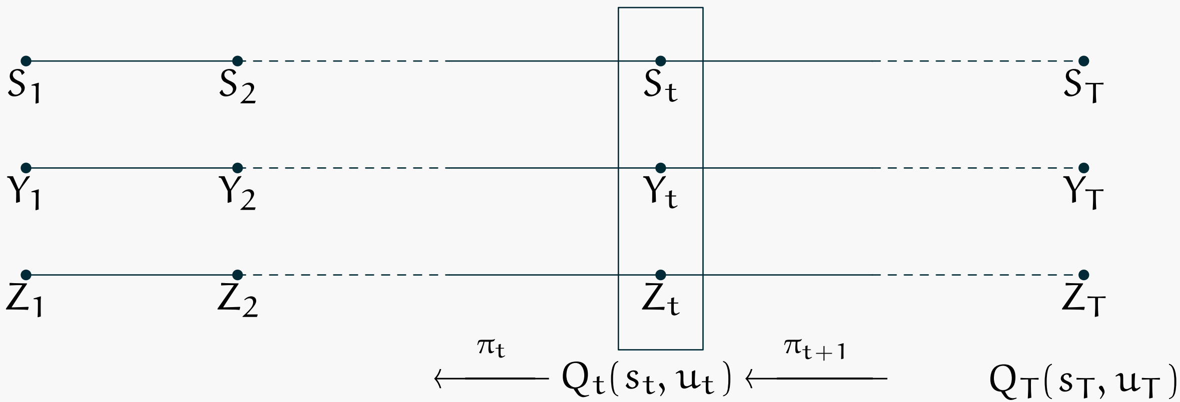
How to find team sequential equilibrium for POMDPs

Notation

- $X_t = (S_t, Y_t, Z_t)$
- $U_t = (A_t, Z_{t+1})$

Step 1: Backward pass

- Compute policy evaluation Q functions $Q_t(x_t, u_t)$



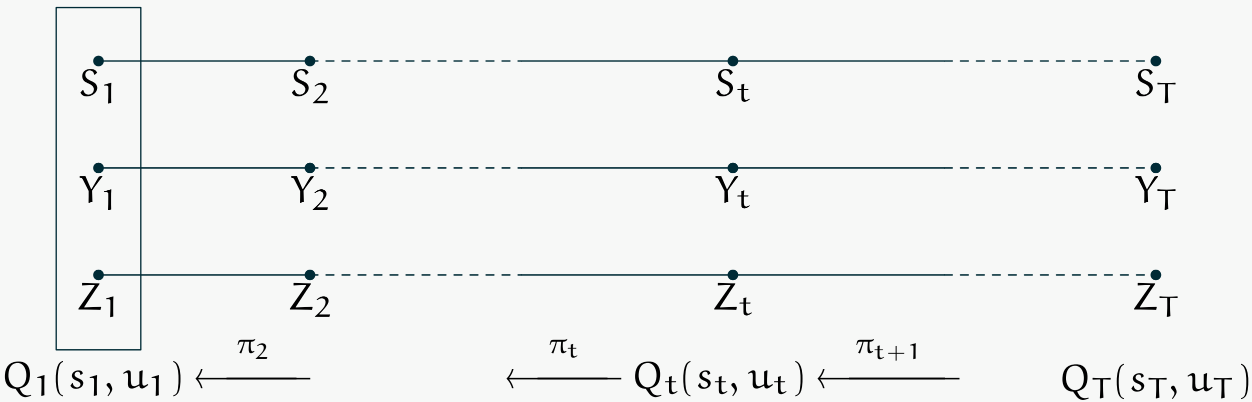
How to find team sequential equilibrium for POMDPs

Notation

- $X_t = (S_t, Y_t, Z_t)$
- $U_t = (A_t, Z_{t+1})$

Step 1: Backward pass

- Compute policy evaluation Q functions $Q_t(x_t, u_t)$



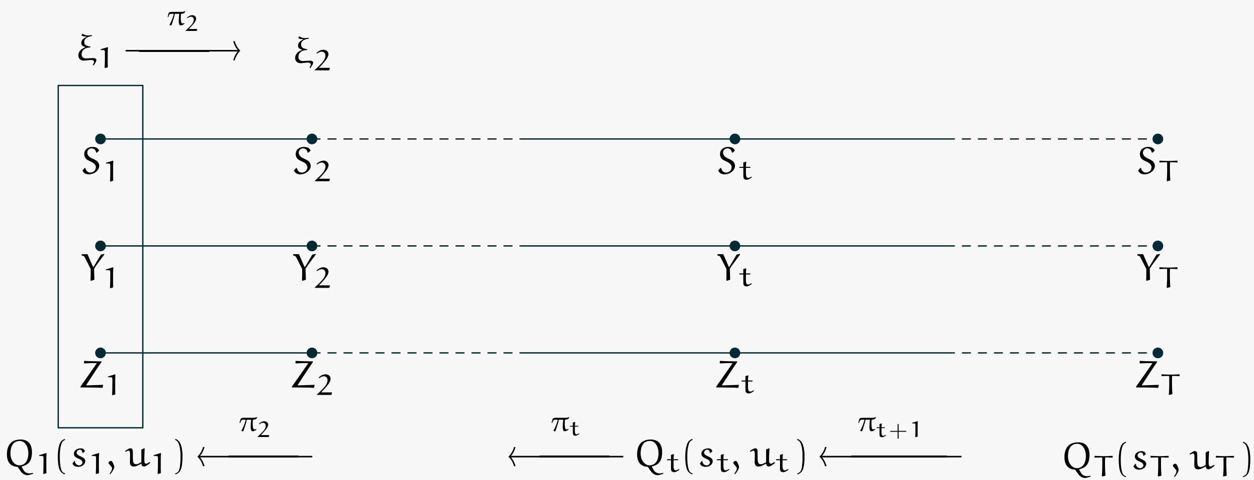
How to find team sequential equilibrium for POMDPs

Notation

- $X_t = (S_t, Y_t, Z_t)$
- $U_t = (A_t, Z_{t+1})$

Step 2: Forward pass

- Compute marginal distribution $\xi_t(x)$ of Markov chain $\{X_t\}_{t \geqslant 1}$
- Use $\xi_t(s_t | y_t, z_t)$ to denote conditional beliefs wrt ξ_t



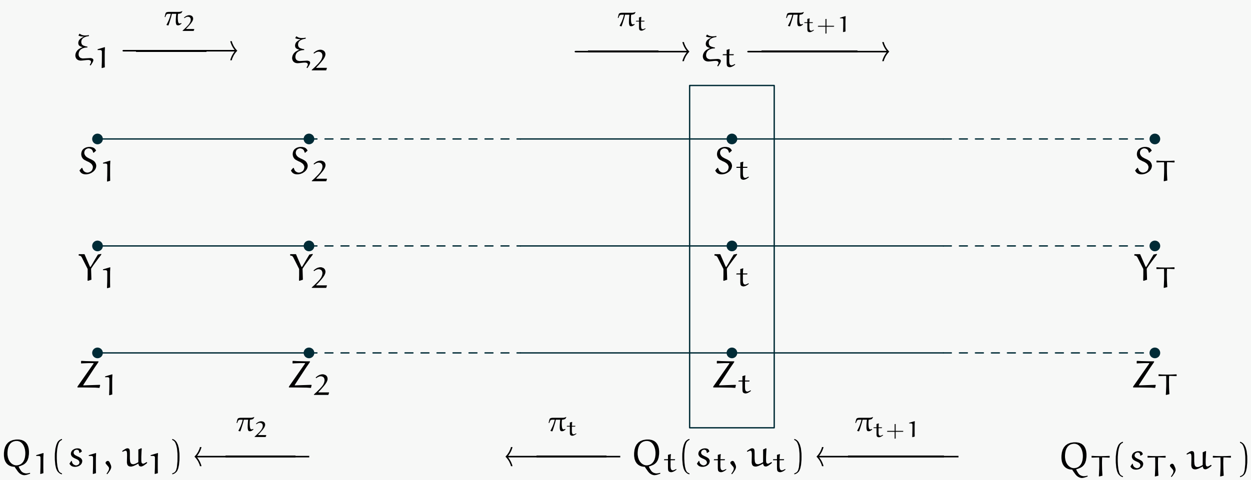
How to find team sequential equilibrium for POMDPs

Notation

- $X_t = (S_t, Y_t, Z_t)$
- $U_t = (A_t, Z_{t+1})$

Step 2: Forward pass

- Compute marginal distribution $\xi_t(x)$ of Markov chain $\{X_t\}_{t \geqslant 1}$
- Use $\xi_t(s_t | y_t, z_t)$ to denote conditional beliefs wrt ξ_t



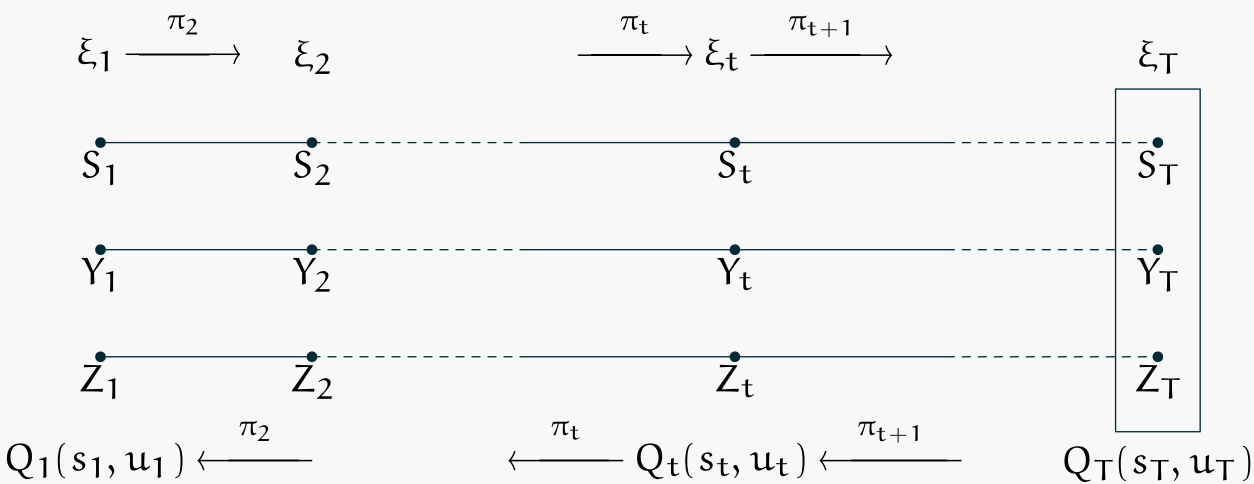
How to find team sequential equilibrium for POMDPs

Notation

- $X_t = (S_t, Y_t, Z_t)$
- $U_t = (A_t, Z_{t+1})$

Step 2: Forward pass

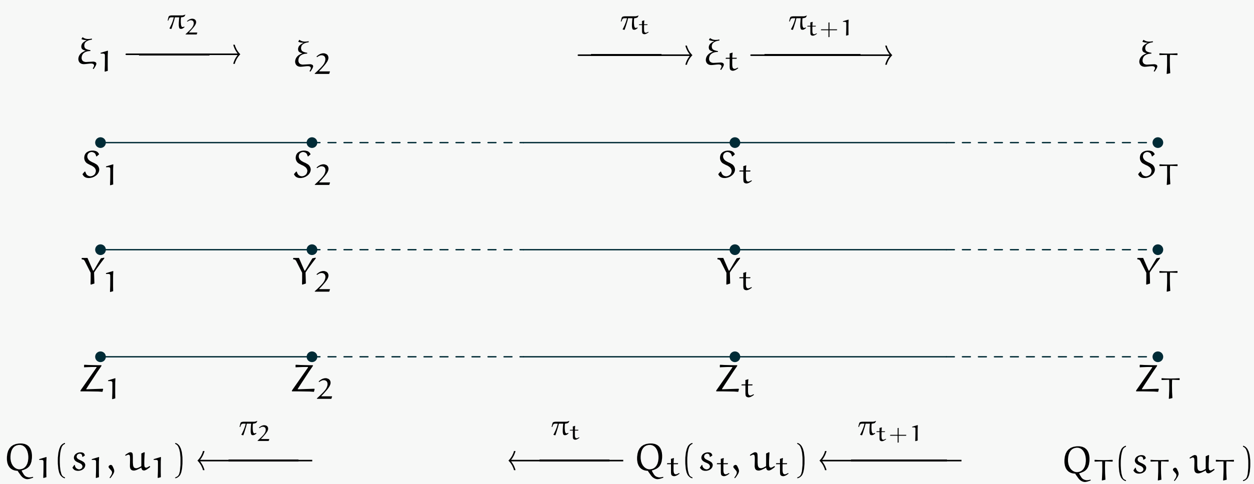
- Compute marginal distribution $\xi_t(x)$ of Markov chain $\{X_t\}_{t \geq 1}$
- Use $\xi_t(s_t | y_t, z_t)$ to denote conditional beliefs wrt ξ_t



How to find team sequential equilibrium for POMDPs

Step 3: Averaging and optimization

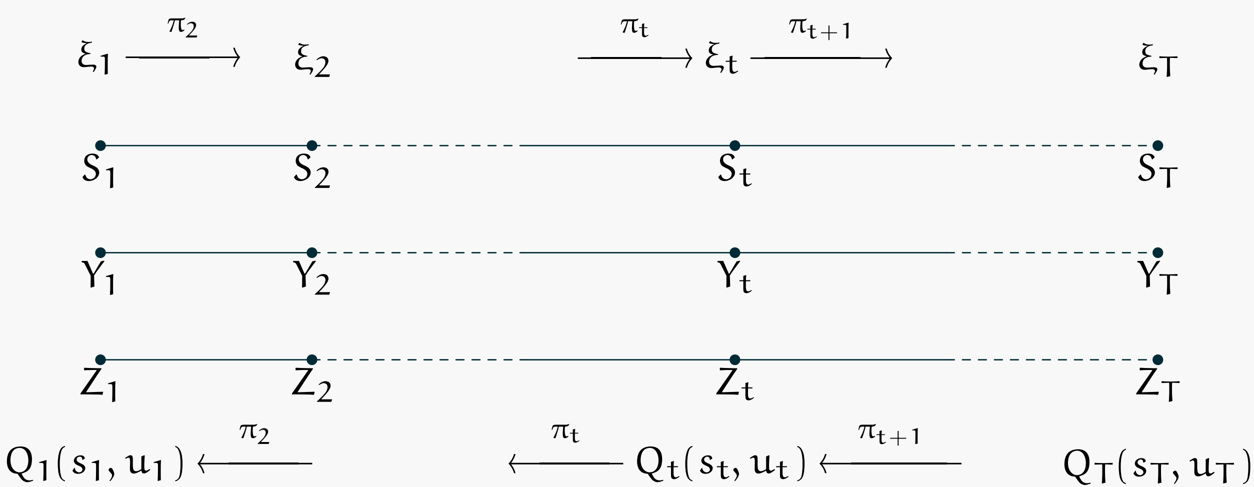
- Q_t depends on the future policies $\pi_{t+1:T}$
- ξ_t depends on the past policies $\pi_{1:t-1}$
- **Averaged Q-function:** $\bar{Q}_t((y, z), u) = \sum_{s} \xi_t(s|y, z)\, Q_t(s, u)$



How to find team sequential equilibrium for POMDPs

Step 4: Local greedy update

$$\pi_t^{\text{next}} \in \mathcal{G}(\pi_{1:t-1}, \pi_{t+1:T}) := \arg\min_{a, z} \bar{Q}_t(y, z, a, z_+)$$



How to find a team sequential equilibrium for POMDPs

Improvement guarantee

For any $\boldsymbol{\pi} = (\pi_1, \dots, \pi_T)$ and any t,
let $\pi_t^{\text{next}} \in \mathcal{G}_t(\pi_{1:t-1}, \pi_{t+1:T})$. Then

$$J(\pi_{1:t-1}, \pi_t^{\text{next}}, \pi_{t+1:T}) \geqslant J(\pi_{1:t-1}, \pi_t, \pi_{t+1:T})$$

How to find a team sequential equilibrium for POMDPs

Improvement guarantee

For any $\boldsymbol{\pi} = (\pi_1, \dots, \pi_T)$ and any t,
let $\pi_t^{\text{next}} \in \mathcal{G}_t(\pi_{1:t-1}, \pi_{t+1:T})$. Then

$$J(\pi_{1:t-1}, \pi_t^{\text{next}}, \pi_{t+1:T}) \geqslant J(\pi_{1:t-1}, \pi_t, \pi_{t+1:T})$$

Policy iteration algorithm

initialize $\boldsymbol{\pi}^{(0)} = (\pi_1^{(0)}, \dots, \pi_T^{(0)})$ and $k \leftarrow 0$
repeat
 $k \leftarrow k + 1$
 for $t \in \{T, \dots, 1\}$ **do**
  $\pi_t^{(k)} \in \mathcal{G}_t(\pi_{1:t-1}^{(k-1)}, \pi_{t+1:T}^{(k)})$
 end for
until $\pi^{(k)} \equiv \pi^{(k-1)}$

How to find a team sequential equilibrium for POMDPs

Improvement guarantee

For any $\boldsymbol{\pi} = (\pi_1, \dots, \pi_T)$ and any t,
let $\pi_t^{\text{next}} \in \mathcal{G}_t(\pi_{1:t-1}, \pi_{t+1:T})$. Then

$$J(\pi_{1:t-1}, \pi_t^{\text{next}}, \pi_{t+1:T}) \geqslant J(\pi_{1:t-1}, \pi_t, \pi_{t+1:T})$$

Convergence Guarantee

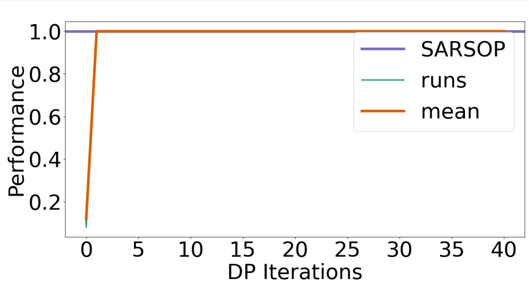
The PI algorithm converges to a team sequential equilibrium, i.e.,

- Beliefs are consistent with policy
- Policy is BR to beliefs

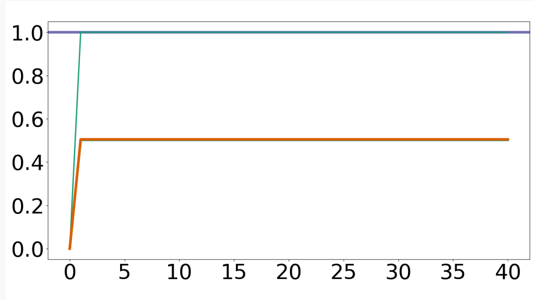
Policy iteration algorithm

initialize $\boldsymbol{\pi}^{(0)} = (\pi_1^{(0)}, \dots, \pi_T^{(0)})$ and $k \leftarrow 0$
repeat
  $k \leftarrow k + 1$
  for $t \in \{T, \dots, 1\}$ **do**
    $\pi_t^{(k)} \in \mathcal{G}_t(\pi_{1:t-1}^{(k-1)}, \pi_{t+1:T}^{(k)})$
  end for
until $\pi^{(k)} \equiv \pi^{(k-1)}$

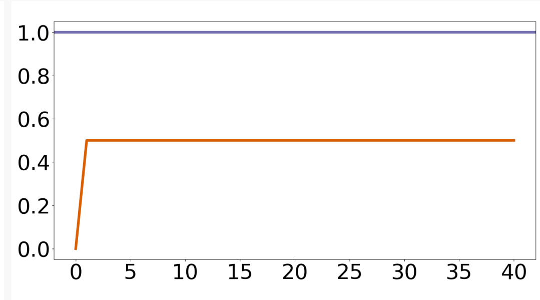
POMDP Experiments (with 1-bit memory)



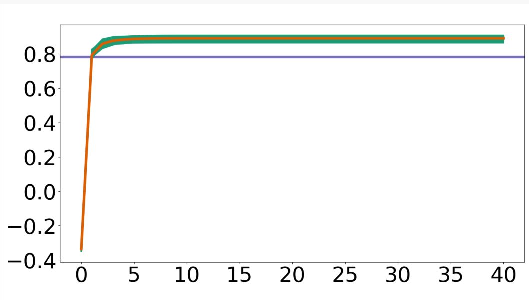
Cheesemaze



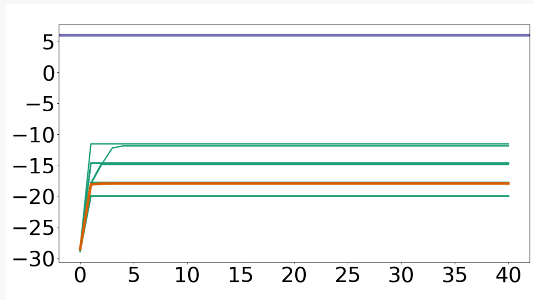
T-maze



Heaven Hell



Drone Surveillance



Shopping

- Agent-state with 1-bit memory
- Planning soln. No learning
- Different runs have different init
- Rest of the algo is deterministic

What's going wrong?

After the first iteration,
the policy is deterministic

Simple fix: Use conservatism

Conservative Policy Update

$$\mathcal{G}_t^{\alpha}(\pi_{1:t-1}, \pi_t, \pi_{t+1:T}) = \alpha\pi_t + (1-\alpha)\,\mathcal{G}_t(\pi_{1:t-1}, \pi_{t+1:T}).$$

- In MDPs, conservatism is used to stabilize policy updates under approx policy evaluation
- We are dealing with exact policy evaluation here.
- But conservatism ensures that the policy remains stochastic

Kakade and Langford, “Approximately Optimal Approximate Reinforcement Learning,” ICML 2002.

Simple fix: Use conservatism

Conservative Policy Update

$$\mathcal{G}_t^{\alpha}(\pi_{1:t-1}, \pi_t, \pi_{t+1:T}) = \alpha\pi_t + (1-\alpha)\,\mathcal{G}_t(\pi_{1:t-1}, \pi_{t+1:T}).$$

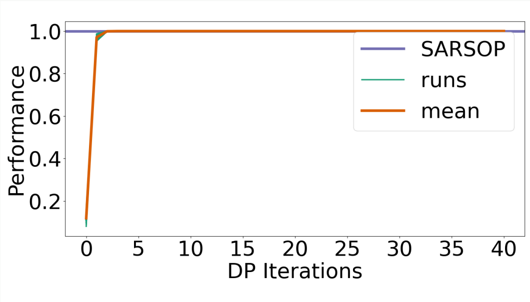
- In MDPs, conservatism is used to stabilize policy updates under approx policy evaluation
- We are dealing with exact policy evaluation here.
- But conservatism ensures that the policy remains stochastic

Choice of α

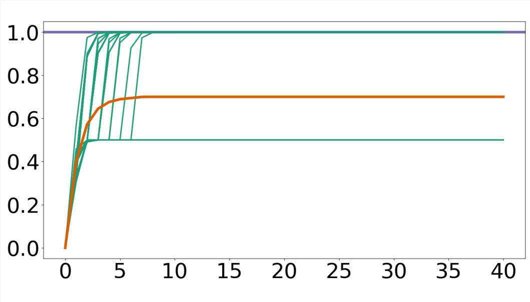
- α can change with iteration k
- Absence of conservatism: $\alpha^{(k)} \equiv 0$
- Constant conservatism: $\alpha^{(k)} \equiv \alpha$
- Policy averaging $\alpha^{(k)} = 1 - \frac{1}{k}$.

Kakade and Langford, "Approximately Optimal Approximate Reinforcement Learning," ICML 2002.

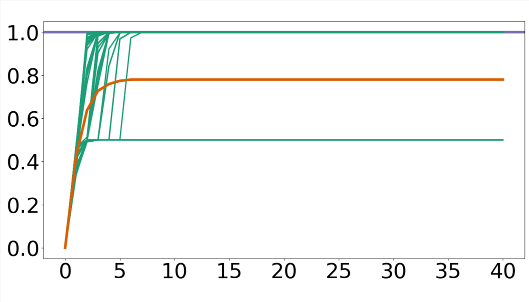
POMDP Experiments (with α=0.05)



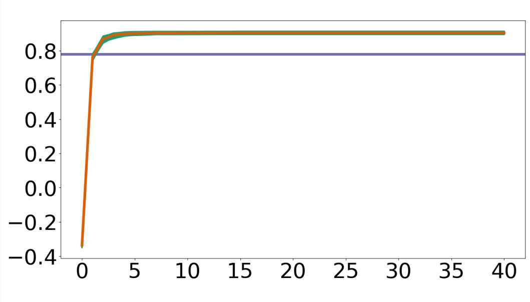
Cheesemaze



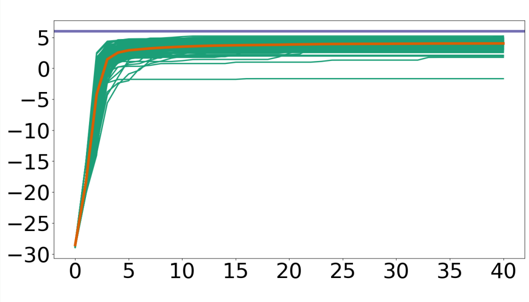
T-maze



Heaven Hell



Drone Surveillance

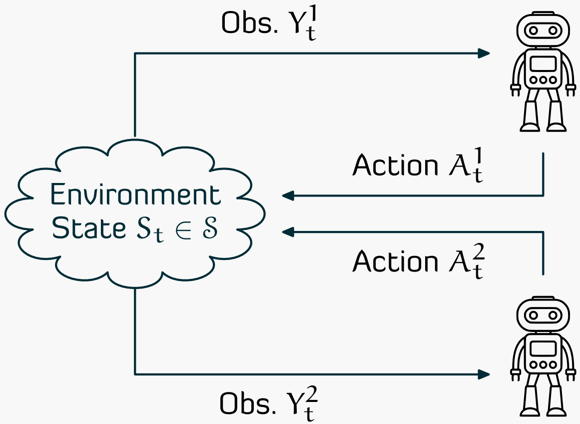


Shopping

Back to the multi-agent setting?

The same idea works!

Conservative Policy Iteration (CPI) for Dec-POMDPs



Basic idea

- Agents at time t, agents act in parallel
- Fix a total order in which the agents act, say $1 \to 2 \to 1 \to 2 \to 1 \to 2 \to \cdots$
- Use the same algorithm

Dec-POMDP Experiments (1-bit memory, $\alpha = 0.05$)

T	FB-HSVI	PF-MAA*	CPI
10	15.18	15.18	8.42
20	28.75	29.12	23.17
50	80.66	81.03	47.95
100	170.90	170.91	91.93

Tiger

T	FB-HSVI	PF-MAA*	CPI
10	223.74	224.28	55.95
20	458.10	474.97	124.03
50	1134.70	1209.80	435.48
100	——	2433.51	771.35

Box Pushing

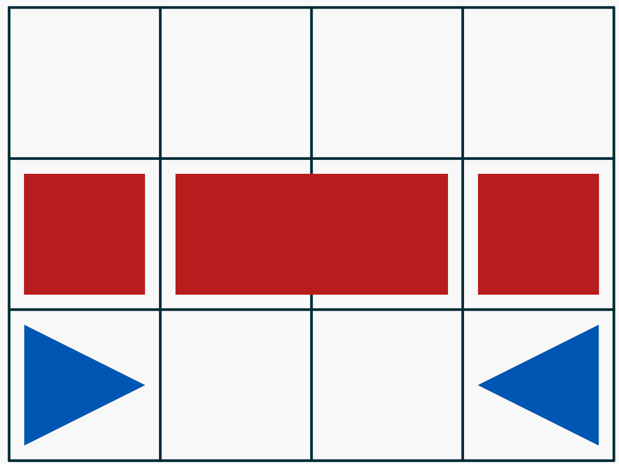
T	FB-HSVI	PF-MAA*	CPI
10	26.31	26.31	18.02
20	52.13	49.37	35.80
50	128.95	122.56	89.37
100	249.92	234.06	178.60

Mars Rover

T	FB-HSVI	PF-MAA*	CPI
100	308.78	308.79	308.40
500	——	1539.56	1536.21
1000	——	3078.02	3074.56
2000	——	6154.94	6148.05

Recycling Robots

Why did it not work?



How to encourage coordination?

Bias the rewards towards cooperation

Biasing rewards towards cooperation

Risk-seeking policy evaluation ($\lambda > 0$)

$$Q^{\pi}_{t}(s, y, z, a, z_{+}) = r(s, a) + \frac{1}{\lambda} \log \left(\sum_{s_{+}, y_{+}} P(s_{+}, y_{+} \mid s, a) \exp\big(\lambda Q^{\pi}_{t+1}(s_{+}, y_{+}, z_{+}, \pi_{t+1}(y_{+}, z_{+}))\big) \right)$$

- Rest of the algorithm remains the same!

Biasing rewards towards cooperation

Risk-seeking policy evaluation ($\lambda > 0$)

$$Q^{\pi}_t(s, y, z, a, z_+) = r(s, a) + \frac{1}{\lambda} \log \left(\sum_{s_+, y_+} P(s_+, y_+ \mid s, a) \exp\big(\lambda Q^{\pi}_{t+1}(s_+, y_+, z_+, \pi_{t+1}(y_+, z_+))\big) \right)$$

▷ Rest of the algorithm remains the same!

Improvement guarantee

For any $\boldsymbol{\pi} = (\pi_1, \dots, \pi_T)$ and any t,
let $\pi^{\text{next}}_t \in \mathcal{G}^{\alpha}_t(\pi_{1:t-1}, \pi_t, \pi_{t+1:T})$. Then

$$J_\lambda(\pi_{1:t-1}, \pi^{\text{next}}_t, \pi_{t+1:T}) \geqslant J_\lambda(\pi_{1:t-1}, \pi_t, \pi_{t+1:T})$$

Biasing rewards towards cooperation

Risk-seeking policy evaluation ($\lambda > 0$)

$$Q^{\boldsymbol{\pi}}_t(s, y, z, a, z_+) = r(s, a) + \frac{1}{\lambda} \log\left(\sum_{s_+, y_+} P(s_+, y_+ \mid s, a) \exp\left(\lambda Q^{\boldsymbol{\pi}}_{t+1}(s_+, y_+, z_+, \pi_{t+1}(y_+, z_+))\right)\right)$$

- Rest of the algorithm remains the same!

Improvement guarantee

For any $\boldsymbol{\pi} = (\pi_1, \dots, \pi_T)$ and any t, let $\pi^{\text{next}}_t \in \mathcal{G}^{\alpha}_t(\pi_{1:t-1}, \pi_t, \pi_{t+1:T})$. Then

$$J_\lambda(\pi_{1:t-1}, \pi^{\text{next}}_t, \pi_{t+1:T}) \geqslant J_\lambda(\pi_{1:t-1}, \pi_t, \pi_{t+1:T})$$

RS-CPI

- For fixed λ, converges to risk-sensitive TSE
- Start with a large λ
- Slowly anneal λ with time

Dec-POMDP Experiments (with RS-CPI)

T	FB-HSVI	PF-MAA*	CPI	RS-CPI
10	15.18	15.18	8.42	13.76
20	28.75	29.12	23.17	27.63
50	80.66	81.03	47.95	54.11
100	170.90	170.91	91.93	120.30

Tiger

T	FB-HSVI	PF-MAA*	CPI	RS-CPI
10	223.74	224.28	55.95	223.78
20	458.10	474.97	124.03	469.39
50	1134.70	1209.80	435.48	1192.46
100	——	2433.51	771.35	2352.18

Box Pushing

T	FB-HSVI	PF-MAA*	CPI	RS-CPI
10	26.31	26.31	18.02	26.32
20	52.13	49.37	35.80	50.47
50	128.95	122.56	89.37	126.43
100	249.92	234.06	178.60	252.63

Mars Rover

T	FB-HSVI	PF-MAA*	CPI	RS-CPI
100	308.78	308.79	308.40	308.79
500	——	1539.56	1536.21	1539.17
1000	——	3078.02	3074.56	3077.63
2000	——	6154.94	6148.05	6154.56

Recycling Robots

Conclusion

An algorithm to compute team sequential equilibrium for Dec-POMDPs

Conclusion

An algorithm to compute team sequential equilibrium for Dec-POMDPs

Key ideas

- For a fixed policy, $\{(S_t, Y_t, Z_t)\}_{t=1}^{T}$ is a MC.
- Easy to evaluate a policy $Q_t^{\pi}(s, a, z_+)$
- Easy to compute marginal distribution $\xi_t(s_t, y_t, z_t)$.
- Average wrt conditional belief $\xi_t(s_t \mid y_t, z_t)$

Conclusion

An algorithm to compute team sequential equilibrium for Dec-POMDPs

Key ideas

- For a fixed policy, $\{(S_t, Y_t, Z_t)\}_{t=1}^{T}$ is a MC.
- Easy to evaluate a policy $Q_t^{\pi}(s, a, z_+)$
- Easy to compute marginal distribution $\xi_t(s_t, y_t, z_t)$.
- Average wrt conditional belief $\xi_t(s_t \mid y_t, z_t)$

Ongoing work

Generalize to

- Sampling based version
- Linear function approximation

Thank you

Amit Sinha

Matthieu Geist

Jayakumar Subramanian

Demos Teneketzis

NSERC CRSNG

IVADO

IDEaS INNOVATION FOR DEFENCE EXCELLENCE AND SECURITY

Google