Robust Partially Observable Markov Decision Processes:

Decision-Making Under State Uncertainty and Imprecise Probabilities

Marnix Suilen

May 6, 2026

University of Antwerp
Belgium

Outline

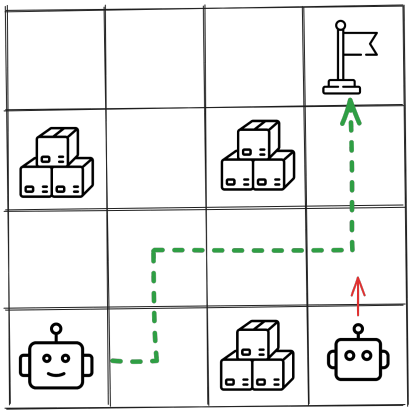
1. Introduction: Decision-Making Under Uncertainty
2. Robust partially observable Markov decision processes
3. Algorithms for RPOMDPs

Decision-Making Under Uncertainty

Decision-Making Under Uncertainty

Applications in

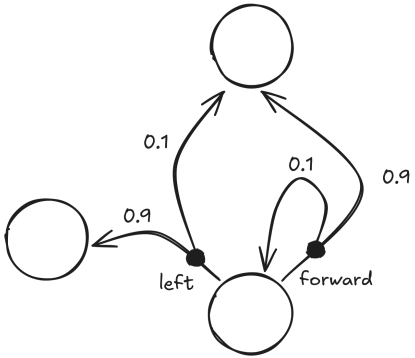
- Healthcare
  Epidemiology, treatment planning
- Finance
  Portfolio optimization, recommendation systems
- Ecology
  Preservation of endangered species
- Robotics
  Navigation planning



Markov Decision Processes

Markov decision process (MDP):

- S states
- A actions
- $P\colon S \times A \to \mathcal{D}(S)$ transition function
- $R\colon S \times A \to \mathbb{R}$ reward function
- γ discount factor



Solving MDPs

Policies select actions: $\pi \colon (S \times A)^* \times S \to \mathcal{D}(A)$

Objective: find a policy that maximizes the expected discounted reward

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\pi,P} \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \mid s_0 = \iota \right]$$

Other typical objectives: finite-horizon reward, reachability probability, stochastic shortest path, ...

Solving MDPs

Memoryless deterministic policies $\pi\colon S \to A$ suffice for discounted reward

Given an MDP M and policy π its value V_M^π is uniquely defined through its associated Markov chain

Algorithms: value iteration, policy iteration, linear programming

Bellman equation

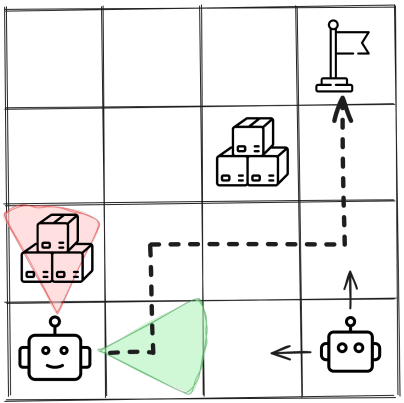
$$V^{(t+1)}(s) = \max_{a \in A} R(s,a) + \gamma \sum_{s' \in S} P(s,a)(s') V^{(t)}(s')$$

has a unique least fixed point V^*

Partial observability

MDPs are fully observable:

- Agent always knows the precise state
- Real-world problems often have incomplete state information (e.g., due to sensors)

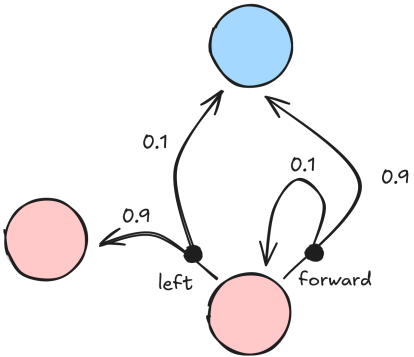


Partially observable MDPs

POMDP:

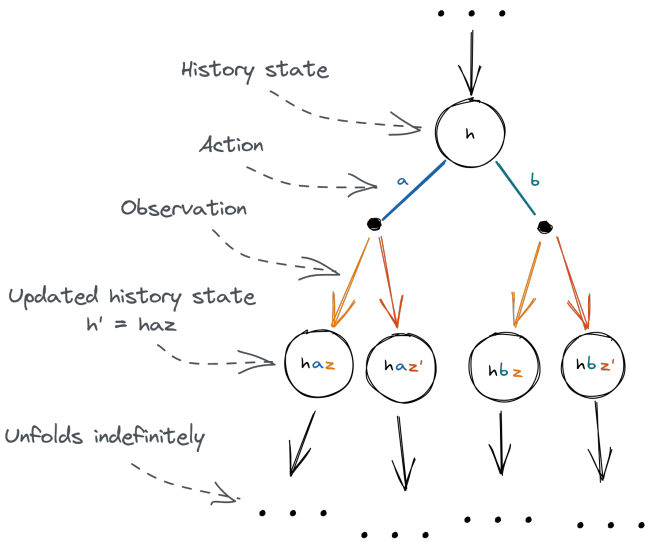
- S, A, P, R, γ as in MDPs
- Z observations
- $O\colon S \times A \to \mathcal{D}(Z)$ observation function

w.l.o.g. we can assume deterministic state-based observations: $O\colon S \to Z$.



POMDP Semantics

- Equivalent to a fully observable history MDP
- Histories can be compressed into distributions over states (beliefs)

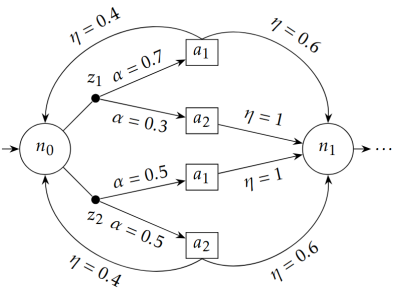


Solving POMDPs: policies

- Most objectives in POMDPs are undecidable; policies are history-based and need memory: $\pi\colon (Z \times A)^* \times Z \to \mathcal{D}(A)$
- Finite-memory policies $\pi\colon (Z \times A)^k \times Z \to \mathcal{D}(A)$ often yield good approximations

Solving POMDPs: policies

- Most objectives in POMDPs are undecidable; policies are history-based and need memory: $\pi\colon (Z \times A)^* \times Z \to \mathcal{D}(A)$
- Finite-memory policies $\pi\colon (Z \times A)^k \times Z \to \mathcal{D}(A)$ often yield good approximations
- Can be succinctly represented by finite-state controllers (FSCs)
- Product of a POMDP and FSC yields a standard Markov chain, which uniquely defines the value.



Solving POMDPs: algorithms

- Extensions of value iteration (point-based VI, heuristic search VI)
- Monte-Carlo tree search (POMCP)
- Policy iteration over FSCs
- Convex optimization directly optimizing an FSC

Robust (partially observable) MDPs

Motivation

MDPs and POMDPs assume probabilities are precisely known

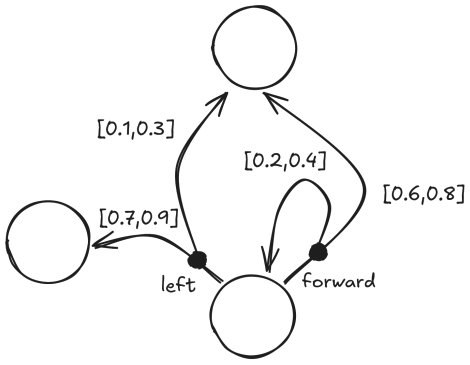
Unrealistic in practice as

- (Optimal) policies and values are highly sensitive to small perturbations
- Probabilities are often estimated from (historical) data
- Models are almost always an abstraction

Robust MDPs

Robust MDP:

- S, A, R, γ as in MDPs
- $\mathcal{U} \subset [0,1]^{S \times A \times S}$ uncertainty set
- each $\boldsymbol{u} \in \mathcal{U}$ is required to satisfy $\forall s, a \sum_{s'} \boldsymbol{u}(s,a,s') = 1$
- $(S, A, P_{\boldsymbol{u}}, R, \gamma)$ defines a standard MDP for each $\boldsymbol{u} \in \mathcal{U}$



Interval MDPs are a classical example of robust MDPs

Robust MDP Semantics

Robust MDPs form a stochastic game

- Agent selects actions
- Adversary selects transition functions from the uncertainty set $\mathcal{U}$

Agent maximizes its reward against a minimizing adversary

$$\pi^* = \arg\max_{\pi} \inf_{\boldsymbol{u} \in \mathcal{U}} \mathbb{E}_{\pi, P_{\boldsymbol{u}}} \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \mid s_0 = \iota \right]$$

Rectangularity

The uncertainty set $\mathcal{U}$ can satisfy certain structural assumptions

- $\mathcal{U}$ is (s, a)-rectangular if it is composed of independent uncertainty sets at each state-action pair: $\mathcal{U} = \times_{s,a} \mathcal{U}_{s,a}$ with $\mathcal{U}_{s,a} \subset [0, 1]^S$.
- Similarly, s-rectangularity allows for dependencies across different actions at the same state
- Uncertainty sets are also typically assumed to be convex, e.g., polytopes

Robust MDP Semantics

Static

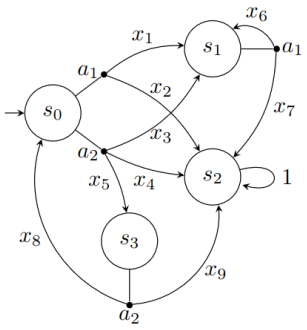
- Adversary chooses one unknown transition function $P_{\boldsymbol{u}}$ at the start

Dynamic

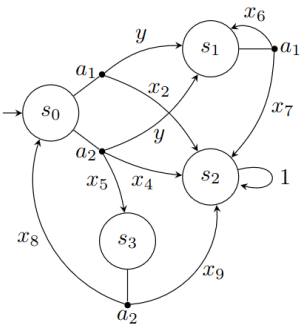
- Adversary chooses new transition $P_{\boldsymbol{u}_t}$ function at every step t

In robust MDPs memoryless deterministic policies are sufficient and static and dynamic uncertainty coincide (Iyengar, Mathematics of Operations Research 2005)

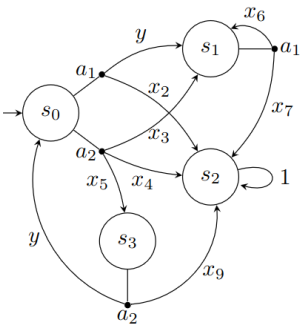
Robust MDP example



(a) An (s, a)-rectangular RMDP.



(b) An s-rectangular RMDP.



(c) A non-rectangular RMDP.

Solving (s, a)-rectangular Robust MDPs

Under (s, a)-rectangularity, the standard Bellman equation

$$V^{(t+1)}(s) = \max_{a \in A} R(s, a) + \gamma \sum_{s' \in S} P(s, a)(s') V^{(t)}(s')$$

can be extended to

$$V^{(t+1)}(s) = \max_{a \in A} R(s, a) + \gamma \inf_{\boldsymbol{u} \in \mathcal{U}_{s,a}} \sum_{s' \in S} P_{\boldsymbol{u}}(s, a)(s') V^{(t)}(s')$$

which still has a unique least fixed point that is the worst-case value

Robust POMDPs

An interesting timeline:

- Initial definition and value iteration based algorithms (Osogami, ICML 2015)
- Convex optimization based policy iteration schemes (Suilen et al., IJCAI 2020; Cubuktepe et al., AAAI 2021)
- Robust POMDP semantics for finite-horizon rewards (Bovy et al., IJCAI 2024)

Robust POMDP Semantics

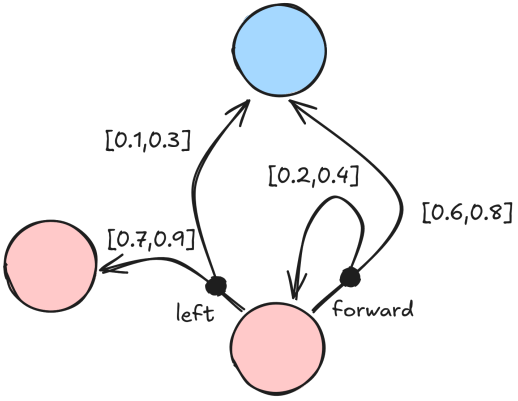
Based on the paper:

Imprecise Probabilities Meet Partial Observability:
Game Semantics for Robust POMDPs
Eline Bovy, Marnix Suilen, Sebastian Junges & Nils Jansen. IJCAI 2024

Robust POMDPs

- Intuition: Robust POMDP = robust MDP + partial observability
- S, A, $\mathcal{U}$, R as in robust MDPs
- Z, O_a, O_n observations and observation functions for both players
- We define semantics for finite-horizon rewards

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\pi,P}\left[\sum_{t=0}^{H} R(s_t, a_t)\right]$$



Stickiness

Notions of static/dynamic uncertainty from robust MDPs no longer coincide

Stickiness

Notions of static/dynamic uncertainty from robust MDPs no longer coincide

Stickiness determines when variable assignments 'stick'

- Zero stickiness: all variable assignments are only valid for one step
  i.e., dynamic uncertainty
- Full stickiness: all variable assignments remain fixed once and for all
  i.e., static uncertainty

Stickiness

Notions of static/dynamic uncertainty from robust MDPs no longer coincide

Stickiness determines when variable assignments 'stick'

- Zero stickiness: all variable assignments are only valid for one step
  i.e., dynamic uncertainty
- Full stickiness: all variable assignments remain fixed once and for all
  i.e., static uncertainty
- Can also define partial stickiness

Zero stickiness provides a conservative lower bound

Stickiness influences value

Theorem
There exist finite-horizon robust POMDPs M_1, M_2 that only differ in stickiness, which have different optimal finite-horizon values: $V^*_{M_1} \neq V^*_{M_2}$.

Proof by explicitly constructing M_1 and M_2 and comparing their values

A similar result can be obtained for changing the order of play

Robust POMDP semantics

Theorem
Robust POMDPs are equivalent to a two-sided partially observable stochastic game

Proof by showing:

- Bijections between paths, histories, policies
- Values coincide

Theorem
The associated game satisfies a saddle point condition, and a Nash equilibrium exists

Classification of existing RPOMDP work

Reference	Stickiness	Order of play
[Osogami, 2015]	Zero	Agent first
[Chamie and Mostafa, 2018]	Zero	Agent first
[Saghafian, 2018]	Zero	Agent first
[Nakao *et al.*, 2021]	Zero	Agent first
[Suilen *et al.*, 2020]	Full	Nature first
[Cubuktepe *et al.*, 2021]	Full	Nature first

Table 1: Classification of existing RPOMDP literature.

Algorithms for Robust POMDPs

Algorithms for POMDPs

We focus on two policy iteration-like methods based on:

1. Convex optimization
   Robust Finite-State Controllers for Uncertain POMDPs
   Murat Cubuktepe, Nils Jansen, Sebastian Junges, Ahmadreza Marandi, Marnix Suilen & Ufuk Topcu. AAAI 2021
2. Recurrent neural networks
   Pessimistic iterative planning with RNNs for robust POMDPs
   Maris Galesloot, Marnix Suilen, Thiago D. Simão, Steven Carr, Matthijs T.J. Spaan, Ufuk Topcu & Nils Jansen. ECAI 2025

Robust Finite-State Controllers

Convex Optimization Method

Pre-processing:

- Fix an FSC memory structure
- Encode FSC memory into RPOMDP state space

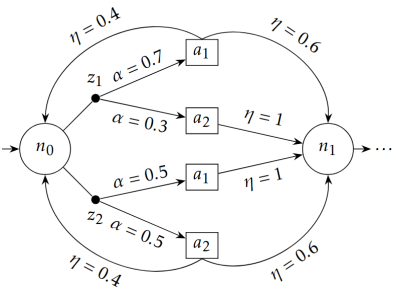
Convex Optimization Method

Pre-processing:

- Fix an FSC memory structure
- Encode FSC memory into RPOMDP state space
- Computing an optimal stochastic memoryless policy is a non-convex semi-infinite optimization problem

constraints of the form

$$\forall \boldsymbol{u} \in \mathcal{U}_{s,a} \colon v_s \leq \sum_{a \in A} \pi_{s,a} \left(R(s,a) + \gamma P_{\boldsymbol{u}}(s,a)(s') v_{s'} \right)$$



Iteratively solve

Iterative procedure:

1. Take the non-convex semi-infinite optimization problem and split concerns:
   1.1 dualize semi-infinite constraints
   1.2 linearize non-convex constraints around a previous solution
2. Solve resulting linear program and repeat 1.2 around new solution
   until convergence/time-out/max-iters/...

Iteratively solve

Iterative procedure:

1. Take the non-convex semi-infinite optimization problem and split concerns:
   1.1 dualize semi-infinite constraints
   1.2 linearize non-convex constraints around a previous solution
2. Solve resulting linear program and repeat 1.2 around new solution until convergence/time-out/max-iters/...

- Guaranteed to converge to a local optimum
- Found policy/value is a sound lower bound to the optimal policy/value

Pessimistic Iterative Planning for Robust POMDPs

FSCs vs. RNNs

Finite-state controllers (FSCs):

- Finite-memory policies
- Efficient exact policy evaluation
- Pre-defined memory structure
- How do we find a good memory structure?

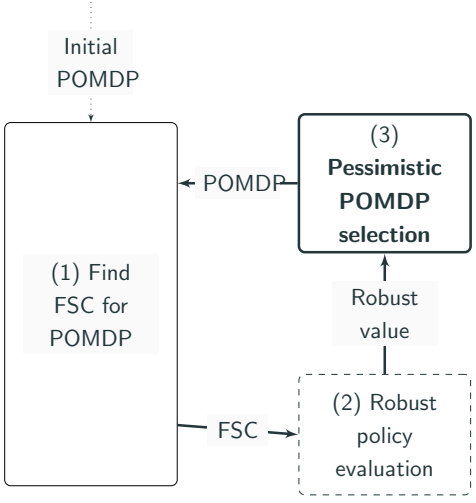
Recurrent neural networks (RNNs):

- Infinite-state machines.
- Learn memory structure from data
- Policy evaluation via simulation
- How do we evaluate performance?

The Pessimistic Iterative Planning Framework

General idea: Iterate over POMDPs

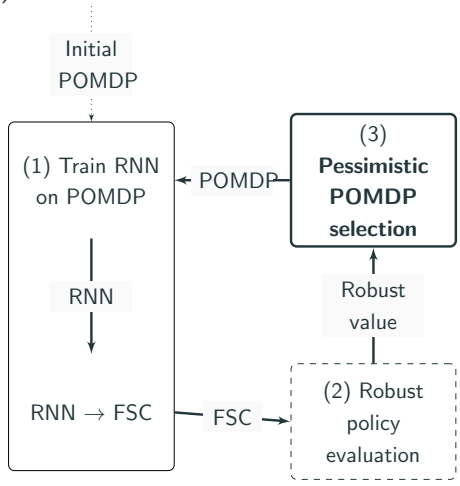
1. within the uncertainty set of the RPOMDP
2. ensure robustness by robust policy evaluation
3. pessimistic POMDP selection



Key steps of our approach

rFSCNet uses recurrent neural networks (RNNs):

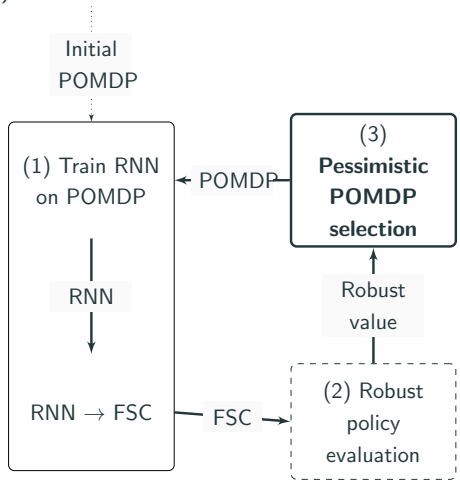
1. Train an RNN *pessimistic* POMDPs and extract an FSC.
2. Evaluate the FSC on the RPOMDP, giving an exact *robust value*.
3. Find a new *pessimistic* POMDP with respect to the FSC for the next iteration.



Key steps of our approach

rFSCNet uses recurrent neural networks (RNNs):

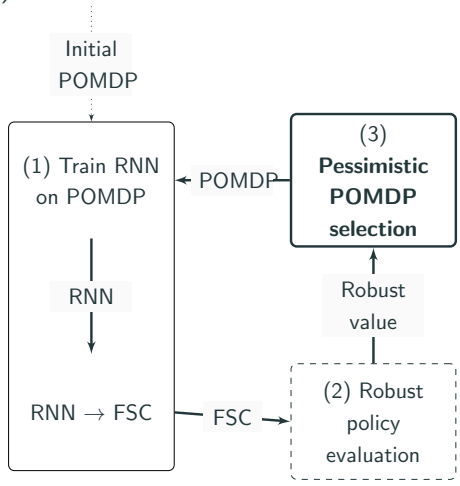
1. Train an RNN *pessimistic* POMDPs and extract an FSC.
2. Evaluate the FSC on the RPOMDP, giving an exact *robust value*.
3. Find a new *pessimistic* POMDP with respect to the FSC for the next iteration.



Key steps of our approach

rFSCNet uses recurrent neural networks (RNNs):

1. Train an RNN *pessimistic* POMDPs and extract an FSC.
2. Evaluate the FSC on the RPOMDP, giving an exact *robust value*.
3. Find a new *pessimistic* POMDP with respect to the FSC for the next iteration.



Empirical evaluation

The experimental evaluation compares to the following:

- SCP: Sequential convex programming
- Baselines: using RNNs to compute FSCs (Carr et al., JAIR 2021) on fixed POMDPs

Empirical evaluation

The experimental evaluation compares to the following:

- SCP: Sequential convex programming
- Baselines: using RNNs to compute FSCs (Carr et al., JAIR 2021) on fixed POMDPs
  - Initialized with upper/lower bound of intervals.
  - Initialized with a random selection of probabilities.
  - Randomly selected at each iteration, somewhat like *domain randomization*

Environments: aircraft collision avoidance and grid-worlds with adversaries.

Objective: stochastic shortest path (minimize expected costs).

Empirical evaluation

The experimental evaluation compares to the following:

- SCP: Sequential convex programming
- Baselines: using RNNs to compute FSCs (Carr et al., JAIR 2021) on fixed POMDPs
  - Initialized with upper/lower bound of intervals.
  - Initialized with a random selection of probabilities.
  - Randomly selected at each iteration, somewhat like *domain randomization*

Environments: aircraft collision avoidance and grid-worlds with adversaries.

Objective: stochastic shortest path (minimize expected costs).

Metric: the robust performance of the FSCs (worst-case expected costs).

Comparison to the state-of-the-art

Lower is better. **Bold** indicates the best robust performance.

$\underline{V}^*$ is a lower bound, given by the optimal robust value of the underlying RMDP.

		Aircraft	*Avoid*	*Evade*	*Intercept*
RMDP	$\underline{V}^*$	94.24	18.05	31.19	16.99
SCP	$k = 3$	116.03	20.07	37.97	**31.57**
	$k = 9$	116.58	29.51	39.78	101.12
rFSCNet	med.	**103.30**	**18.51**	**37.61**	47.82
	min.	102.03	18.16	36.65	31.61

For rFSCNet, we report the *median* (med.) and *minimum* (min.) of the robust performance across 20 seeds.

Comparison to RNN baseline

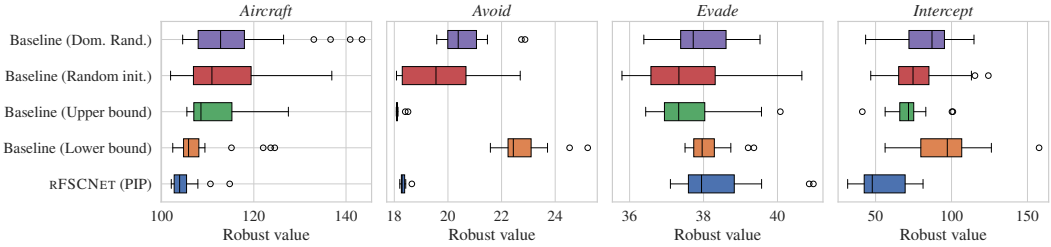


Figure 1: The boxplots depict the minimum (i.e., best) *robust performance* of the extracted FSC policies for both the baselines and rFSCNet reported across 20 seeds.

rFSCNet shows an improvement in robust performance by using the PIP framework, compared to the baselines.

Conclusion

Robust POMDPs are not just robust MDPs with partial observability

- Assumptions matter
- Approximation techniques used in algorithms may muddy the semantics

Open problems

- Game semantics for infinite horizon objectives (discounted reward)
- Computational complexity of decidable POMDP problems
- Efficient algorithms for less conservative settings

References

POMDPs / Robust MDPs:

- Kaelbling et al. Planning and acting in partially observable stochastic domains. Artificial intelligence. 1998
- Iyengar. Robust dynamic programming. Mathematics of Operations Research. 2005
- Suilen et al. Robust Markov decision processes: A place where AI and formal methods meet. LNCS. 2024

Robust POMDPs:

- Bovy et al. Imprecise Probabilities Meet Partial Observability: Game Semantics for Robust POMDPs. IJCAI. 2024
- Cubuktepe et al. Robust finite-state controllers for uncertain POMDPs. AAAI. 2021
- Galesloot et al. Pessimistic iterative planning with RNNs for robust POMDPs. ECAI. 2025