Simplified POMDP Algorithms with Performance Guarantees

Moran Barenboim

Supervisor: Assoc. Prof. Vadim Indelman

TECHNION Israel Institute of Technology

ANPL Autonomous Navigation and Perception Lab

Simplified POMDP Algorithms with Performance Guarantees

TECHNION Israel Institute of Technology

ANPL Autonomous Navigation and Perception Lab

Introduction

Sequential decision-making under uncertainty

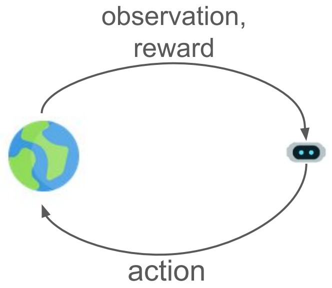
Examples include,

Urban IROS 2013 workshop | **Indoor** Upenn | **Autonomous Cars** Georgia Tech

Introduction - Formalism

Partially Observable Markov Decision Process (POMDP)

$$\mathcal{M} = \langle \mathcal{X}, \mathcal{A}, \mathcal{Z}, \mathcal{P}_0, \mathcal{P}_T, \mathcal{Z}, \mathcal{R}, \mathcal{T} \rangle$$



- State - $x_t \in \mathcal{X}$
- Action - $a_t \in \mathcal{A}$
- Observation - $z_t \in \mathcal{Z}$
- Reward - $r(x_t, a_t)$
- Transition function - $P_T(x_{t+1}, x_t, a_t) = \mathbb{P}(x_{t+1} | x_t, a_t)$
- Observation function - $P_z(z_t, x_t) = \mathbb{P}(z_t | x_t)$
- Prior distribution (also prior belief) - b_0
- Horizon - $\mathcal{T}$

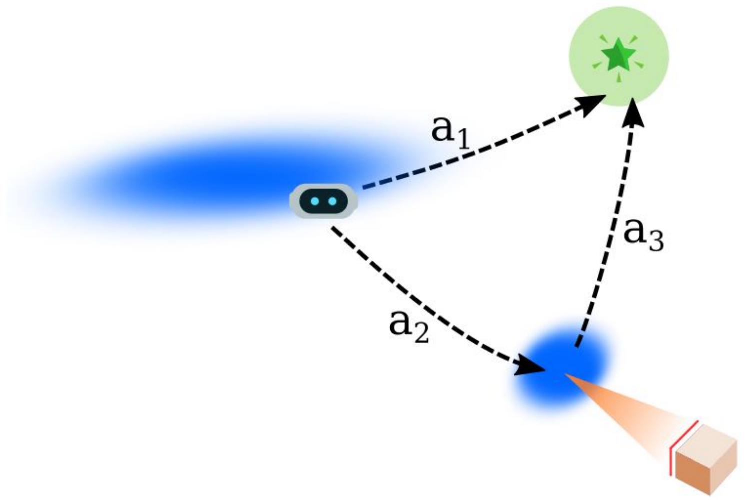
Introduction - Formalism

- History - $H_t = \{b_0, a_0, z_1, a_1, ...a_{t-1}, z_t\}$
- Belief - $b_t = \mathbb{P}(x_t \mid H_t)$
- Policy - $\pi_t(b_t)$
- Value function $V^\pi(b_t)$, where,

$$
\begin{aligned}
V^{\pi_t}(b_t) &= \mathbb{E}_{z_{t+1:\mathcal{T}}}\left[\sum_{i=t}^{\mathcal{T}} r(b_i, \pi_i)\right] \\
&= r(b_t, \pi_t) + \mathbb{E}_{z_{t+1}}\left[V^{\pi_t}(b_{t+1})\right]
\end{aligned}
$$

Introduction - Formalism

The (optimal) solution for a POMDP optimally trades off information-gathering actions versus other actions.

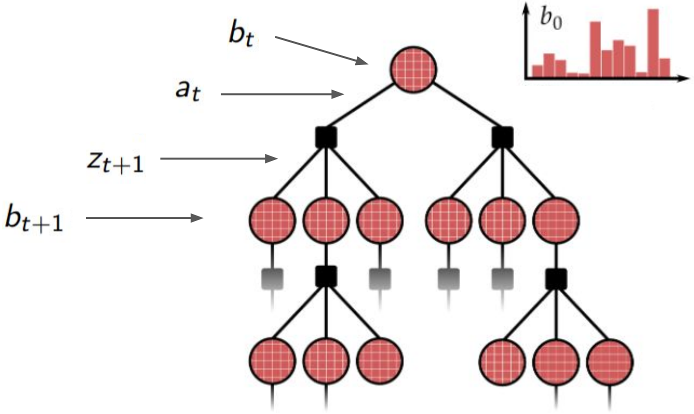


Introduction - Solutions

We will be focusing on online tree search methods

- Each node represents a belief
- Each edge represents an action or an observation
- Given a prior belief, the posterior belief is calculated via probabilistic inference

$$b_{t+1} = \psi(b_t, a_t, z_{t+1})$$

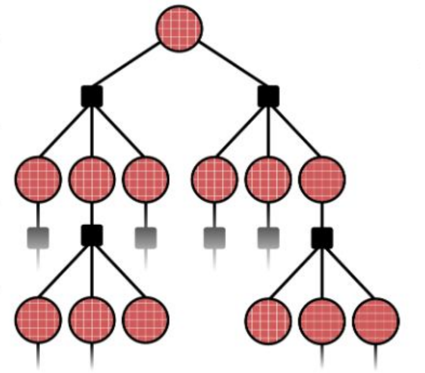


TECHNION Israel Institute of Technology
ANPL Autonomous Navigation and Perception Lab

Introduction - Solutions

How to get an exact solution?

Given a POMDP definition, construct a tree of all states, actions and observations



Problem?

Size of the tree - $O(T^{|O| \times |A|})$

Only relevant for very small POMDPs

Introduction - Solutions

Approximate planners:

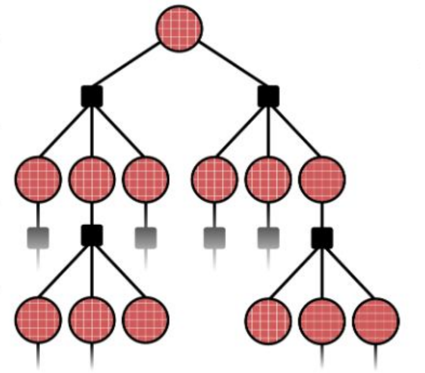
	Planning efficiency	Aware of state uncertainty	Optimal (in some sense)
Gradient-based, open-loop	Yes	No	No
Deterministic approximations	Yes	No	No
Monte-Carlo Sampling	Yes	Yes	Yes

POMCP, DESPOT, POMCPOW, PFT-DPW, AdaOPS…

Introduction - Our Approach

In this research, we derive a framework instead of solving the original POMDP, considers a simplified version of that POMDP.

$$\mathcal{M} \rightsquigarrow \bar{\mathcal{M}}$$



Then, we aim at deriving a mathematical relationship between the solution of the simplified, and the theoretical POMDP.

$$V(b) \quad ? \quad \bar{V}(b)$$

Simplified POMDP Algorithms with Performance Guarantees

Published:

- M. Barenboim and V. Indelman. Adaptive information belief space planning. In the 31st International Joint Conference on Artificial Intelligence and the 25th European Conference on Artificial Intelligence (IJCAI-ECAI), July 2022

TECHNION Israel Institute of Technology

ANPL Autonomous Navigation and Perception Lab

Belief Based Rewards - Motivation

A generalization of POMDPs (limited to state-based rewards).

Supports explicit reasoning of uncertainty, e.g.,

- Pose uncertainty (of the robot, other agents, etc.)
- Map representation
- Semantic uncertainty

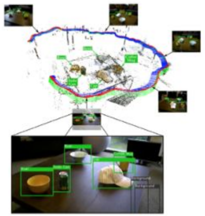


Figure: Pillai et al.

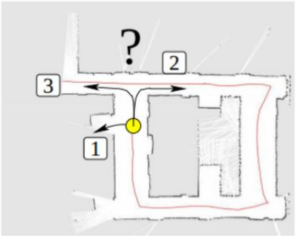


Figure: Stachniss et al.

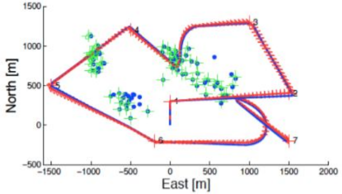


Figure: Indelman et al.

Belief Based Rewards - Motivation

Information-Theoretic functions may be used as uncertainty measures, commonly used as reward functions. E.g.,

- Differential entropy

$$\mathcal{H}(x_t) = -\int_x b_t \cdot log(b_t)dx$$

- Information Gain
- Mutual information
- Kullback-Leibler divergence
- and more…

Belief Based Rewards - Introduction

We focused on entropy as an information-theoretic reward function,

$$r(b_t, a_t, b_{t+1}) = \omega_1 \mathbb{E}_{s \sim b_{t+1}} [r(s, a_t)] + \omega_2 \mathcal{H}(b_{t+1}),$$

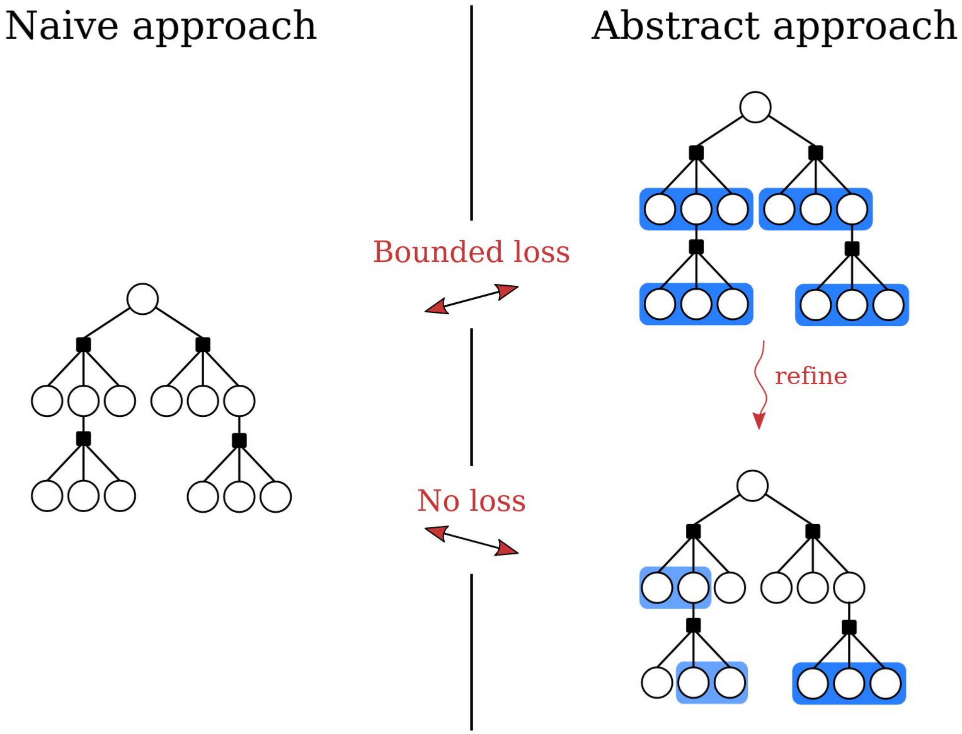
a weighted sum of state-dependent reward and entropy (discrete or continuous)

Belief Based Rewards - The Challenge

The difficulty - Information theoretic functions are generally intractable

And even approximations are computationally difficult - $O(|\mathcal{X}|^2)$ - for every reward calculation

Belief Based Rewards - Our Contribution

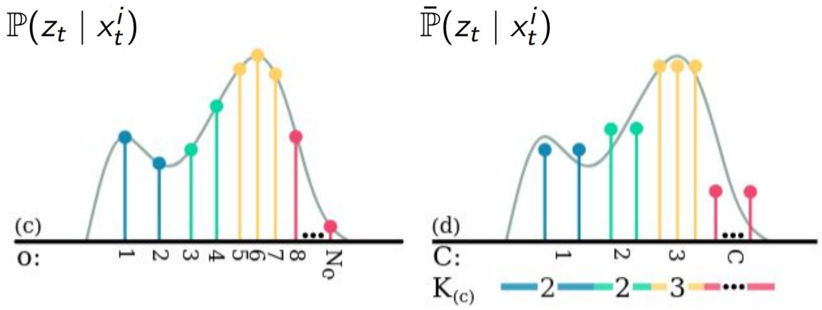


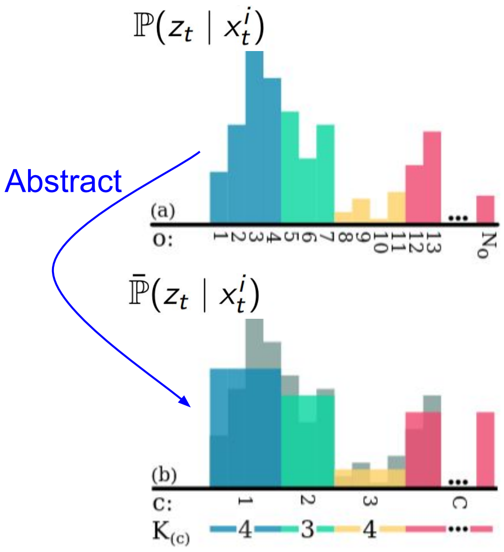
Belief Based Rewards - Our Contribution

We introduced an abstract observation model,

$$\bar{\mathbb{P}}(z_t^j \mid x_t^i) = \frac{\sum_k^K \mathbb{P}(z_t^k | x_t^k)}{K}$$

- Aggregates a set of K observations
- The new probability value is the aggregate average





Belief Based Rewards - Our Contribution

Using the abstract observation model, we derived analytical bounds compared to the non-abstract model,

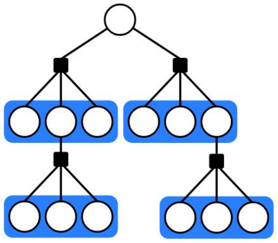
Corollary

The difference between the theoretical value function and the abstract value function is bounded by,

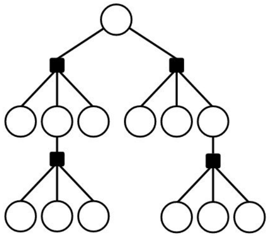
$$0 \leq \bar{V}^{\pi}(b_t) - V^{\pi}(b_t) \leq \mathcal{T} \cdot \omega_2 log(K).$$

$\mathcal{T}$	-	Horizon
ω_2	-	Entropy weight
K	-	Num clustered observations

Abstract approach



Naive approach



Belief Based Rewards - Our Contribution

Using the abstract observation model, we derived analytical bounds compared to the non-abstract model,

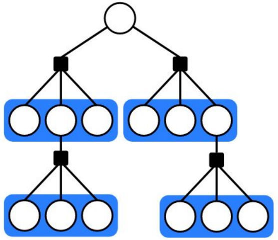
Corollary

The difference between the theoretical value function and the abstract value function is bounded by,

$$0 \leq \bar{V}^{\pi}(b_t) - V^{\pi}(b_t) \leq \mathcal{T} \cdot \omega_2 log(K).$$

$\mathcal{T}$	-	Horizon
ω_2	-	Entropy weight
K	-	Num clustered observations

Abstract approach



Here, K=3

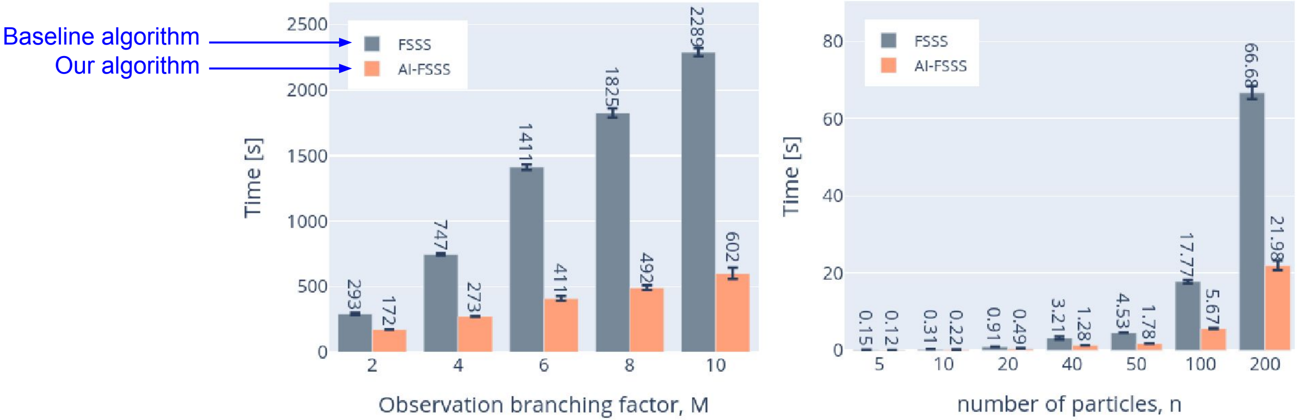
Some interesting observations:

- There is no loss for abstracting the state-dependent reward
- The bound can be made adaptive by reducing the size of K
- K does not need to remain constant throughout the tree

Belief Based Rewards - Results

Speed-up for free

- Exact same solution
- A fraction of the planning time



Technion Israel Institute of Technology
ANPL Autonomous Navigation and Perception Lab

Simplified POMDP Algorithms with Performance Guarantees

Published:

- Barenboim, M.; Shienman, M.; and Indelman, V. 2023., "Monte Carlo planning in hybrid belief POMDPs," IEEE Robotics and Automation Letters (RA-L).
- Barenboim, M.; Lev-Yehudi, I.; and Indelman, V. 2023. Data Association Aware POMDP Planning with Hypothesis Pruning Performance Guarantees. IEEE Robotics and Automation Letters (RA-L).

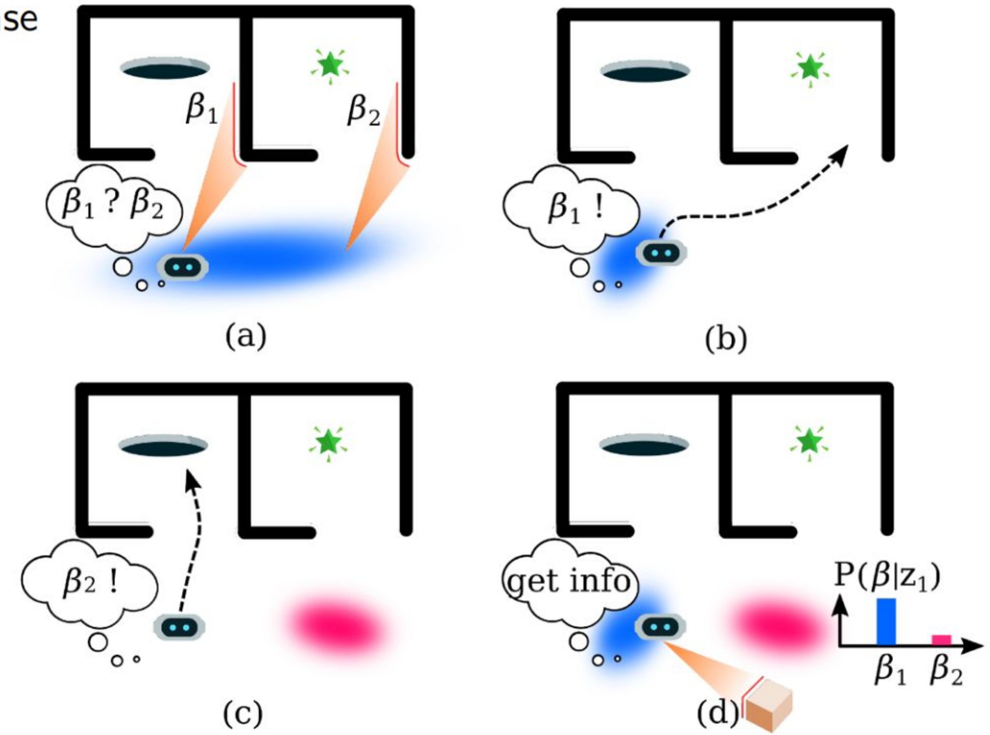
TECHNION Israel Institute of Technology

ANPL Autonomous Navigation and Perception Lab

Continuous-Discrete State Spaces - Motivation

As an accompanying example, consider the case of ambiguous data associations:

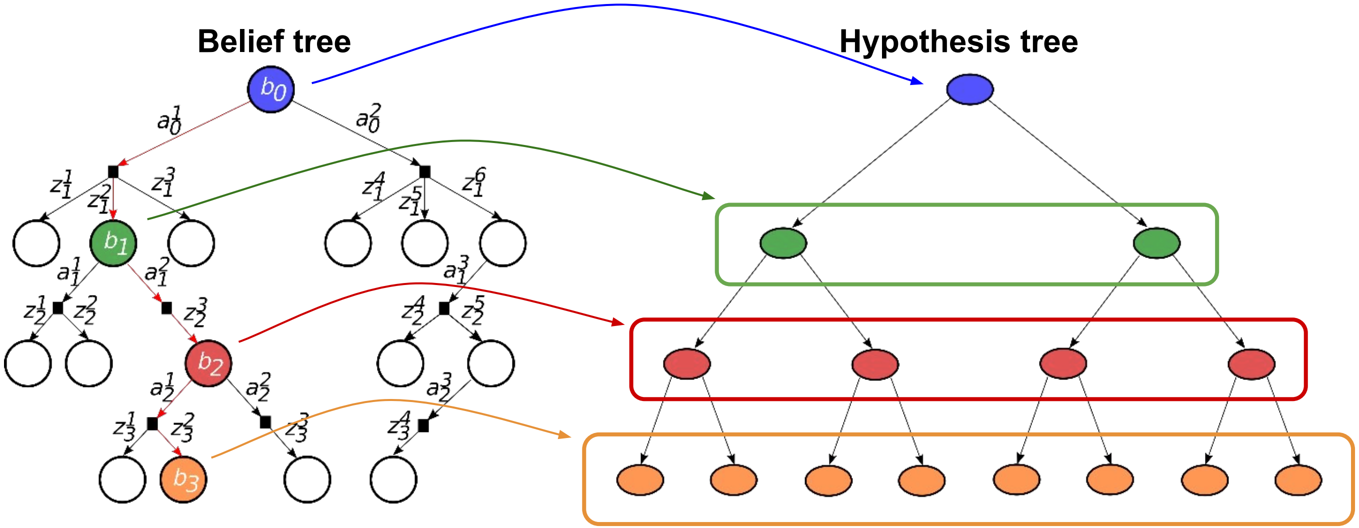
- Uncertain observation source
- Optimality requires reasoning about different hypotheses



Continuous-Discrete State Spaces - The Challenge

Computing the reward function requires explicit knowledge of the hypotheses

However, the number of hypotheses may grow **exponentially** with the horizon!

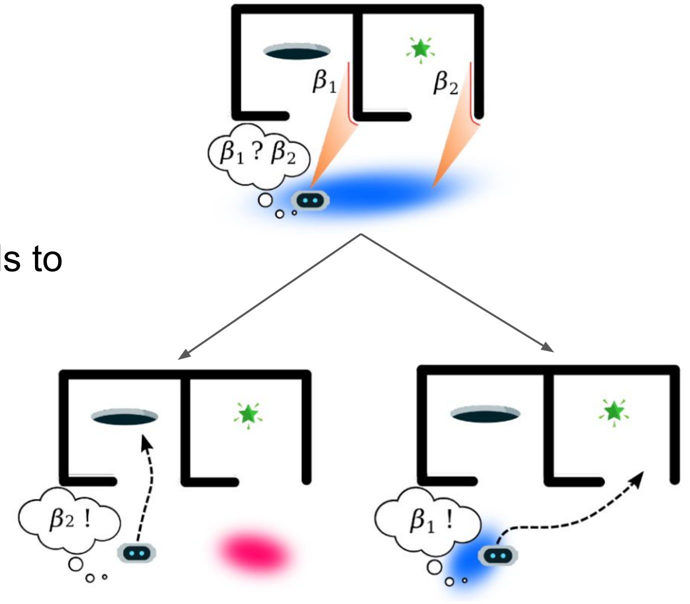


Continuous-Discrete State Spaces - Motivation

What others have done?

- Either implicitly assume known observation source
- Or prune hypotheses based on heuristics

It is not hard to show that a pruned set of hypotheses leads to a biased estimation



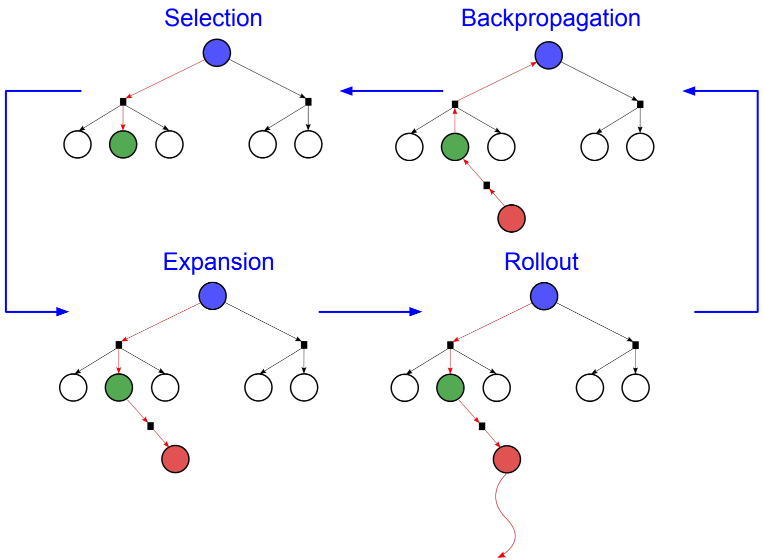
Continuous-Discrete State Spaces - Our Contributions (1)

Instead of computing all possible hypotheses, we utilize MCTS sampling and exploration approach.
MCTS:

- An MDP solver
- Uses UCT to tradeoff exploration-exploitation for actions

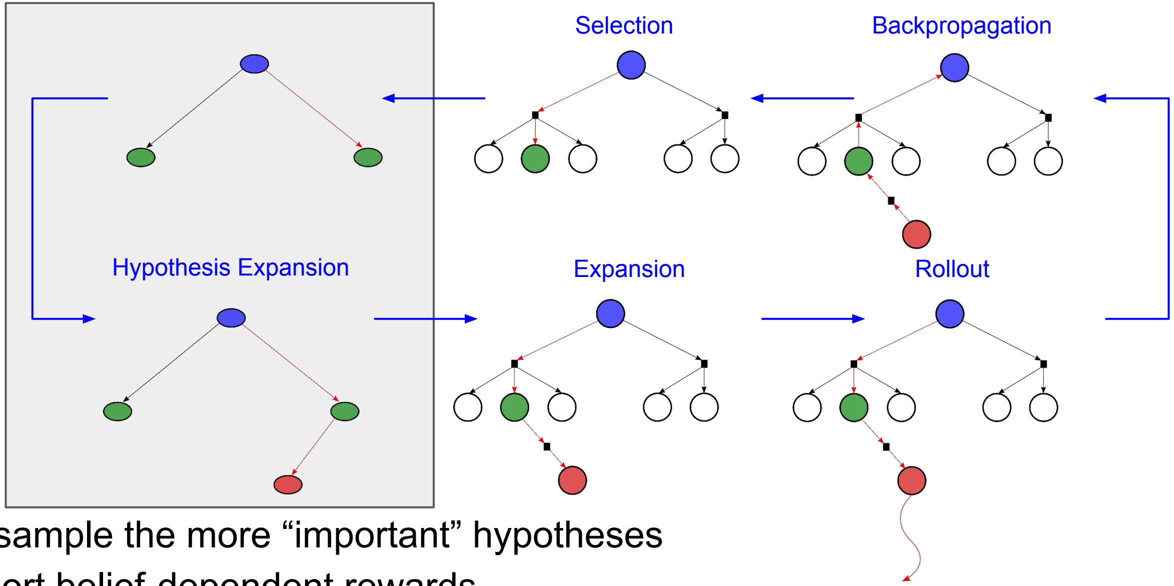
$$UCT(x_t, a_t) = \hat{Q}(x_t, a_t) + c \cdot \sqrt{\frac{\log N(x_t)}{n(x_t, a_t)}}$$

- Given an action, samples the next state



Continuous-Discrete State Spaces - Our Contributions (1)

We add a layer that samples hypotheses via Monte-Carlo sampling



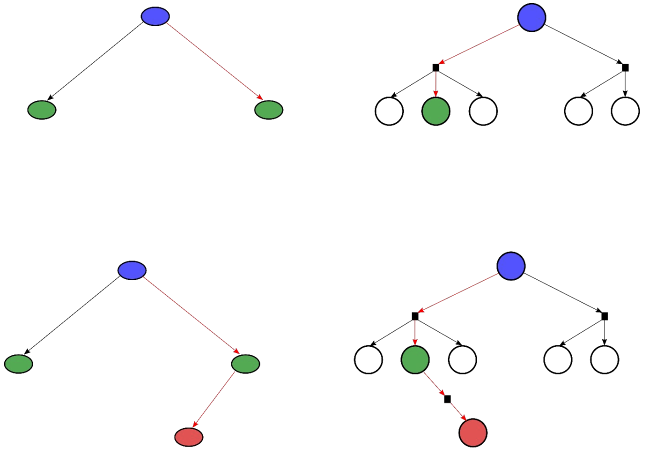
- Tends to sample the more “important” hypotheses
- Can support belief-dependent rewards

Continuous-Discrete State Spaces - Our Contributions (1)

We have derived a corresponding reward estimator, $\hat{\mathcal{R}}_X$, and have shown that it leads to an unbiased estimator,

Lemma

The sampled-based, state-dependent reward estimator, $\hat{\mathcal{R}}_X \triangleq \frac{1}{N}\sum_{i,j=1}^{N} \lambda_t^{i,j} \frac{1}{n_X}\sum_{k=1}^{n_X} r(X_t^{i,j,k}, a_t)$, *is unbiased.*

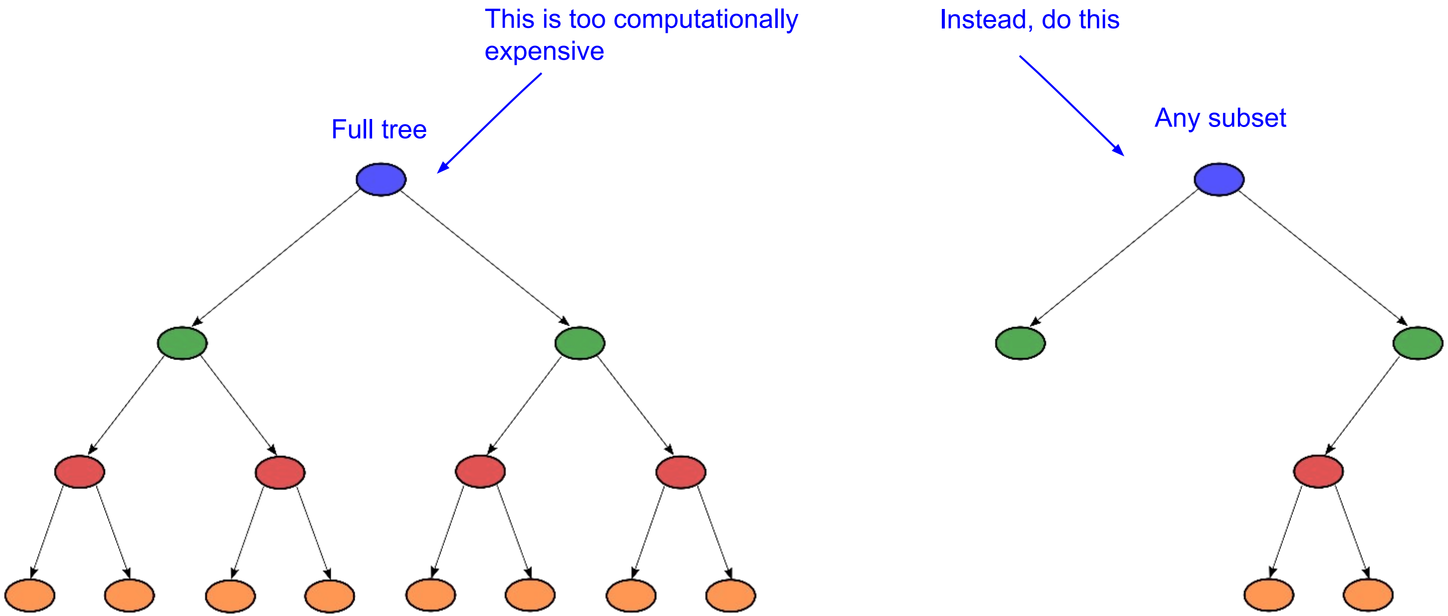


Technion Israel Institute of Technology

ANPL Autonomous Navigation and Perception Lab

Continuous-Discrete State Spaces - Our Contributions (2)

Our second contribution bridges the gap between the full hypothesis tree and a simplified tree



Continuous-Discrete State Spaces - Our Contributions (2)

Derived a deterministic bound to relate the full set of hypotheses to a subset thereof,

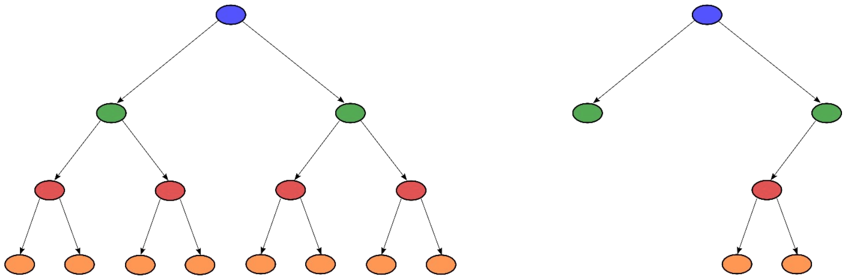
Corollary

For any policy π, and selection of hypotheses set $\{\beta^i_{0:\mathcal{T}}\}_{i=0}^{|\mathcal{B}|}$ the following holds,

$$\left|V^{\pi}(b_0) - \bar{V}^{\pi}(\bar{b}_0)\right| \leq \mathcal{R}_{max}\left[\mathcal{T}\delta_0^{\beta} + \sum_{k=1}^{\mathcal{T}}\sum_{\tau=1}^{k}\mathbb{E}_{z_{1:\tau}}\left[\delta_{\tau}^{\beta}\right]\right].$$

Full tree

Any subset



Importantly, the bound relies on the available hypotheses

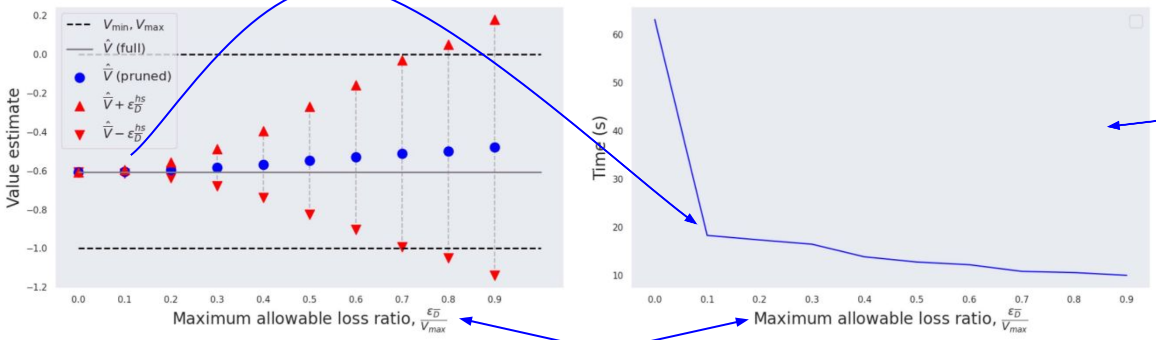
Can bound the theoretical value with access only to the simplified tree

Continuous-Discrete State Spaces - Results

Small loss in the value function, may lead to significant improvement in planning time

Value estimation

Planning time



(a) (b)

Tunable loss limit (hyperparameter)

Simplified POMDP Algorithms with Performance Guarantees

Published:

- M. Barenboim and V. Indelman. Online pomdp planning with anytime deterministic guarantees. In Advances in Neural Information Processing Systems, 2023

To be submitted:

- M. Barenboim and V. Indelman. Online pomdp planning with anytime deterministic guarantees - Extended version. To be submitted

TECHNION Israel Institute of Technology

ANPL Autonomous Navigation and Perception Lab

POMDPs with Deterministic Guarantees - Motivation

Short reminder:

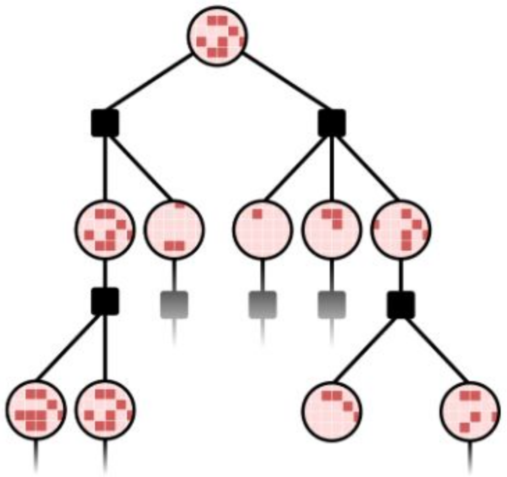
- POMDP is a formal framework for decision-making under uncertainty
- Finding an optimal policy is generally intractable
- Must resort to approximate solvers

Note - in this section we focus on discrete spaces

POMDPs with Deterministic Guarantees - Motivation

SOTA approximate solvers rely on sampling

They choose a subset of the state and observation spaces



Autonomous Navigation and Perception Lab

POMDPs with Deterministic Guarantees - The Challenge

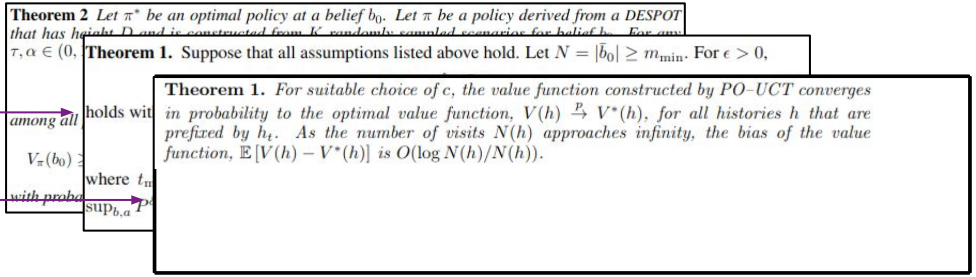
Naturally, sampling comes with probabilistic theoretical guarantees

DESPOT →

AdaOPS →

POMCP →

…


Theorem 2 *Let π^* be an optimal policy at a belief b_0. Let π be a policy derived from a DESPOT*

Theorem 1. Suppose that all assumptions listed above hold. Let $N = |\bar{b}_0| \geq m_{\min}$. For $\epsilon > 0$,

Theorem 1. *For suitable choice of c, the value function constructed by PO–UCT converges in probability to the optimal value function, $V(h) \xrightarrow{p} V^*(h)$, for all histories h that are prefixed by h_t. As the number of visits $N(h)$ approaches infinity, the bias of the value function, $\mathbb{E}[V(h) - V^*(h)]$ is $O(\log N(h)/N(h))$.*

Can we get deterministic guarantees?

POMDPs with Deterministic Guarantees - Approach

In our work, we show that deterministic guarantees are indeed possible!

Given a POMDP: $$\mathcal{M} = \langle \mathcal{X}, \mathcal{Z}, \mathcal{A}, \mathcal{P}_0, \mathcal{P}_T, \bar{\mathcal{P}}_{\mathcal{Z}}, \mathcal{R}, \mathcal{T} \rangle$$

We define a simplified POMDP,

$$\bar{\mathcal{M}} = \langle \bar{\mathcal{X}}, \bar{\mathcal{Z}}, \mathcal{A}, \bar{\mathcal{P}}_0, \bar{\mathcal{P}}_T, \bar{\mathcal{P}}_{\mathcal{Z}}, \mathcal{R}, \mathcal{T} \rangle$$

$$\bar{\mathcal{X}}(H_t) \subset \mathcal{X}$$
$$\bar{\mathcal{Z}}(H_t) \subset \mathcal{Z}$$

$$\bar{b}_0(x) \triangleq \begin{cases} b_0(x) & , x \in \bar{\mathcal{X}}_0 \\ 0 & , otherwise \end{cases}$$

$$\bar{\mathbb{P}}(x_{t+1} \mid x_t, a_t) \triangleq \begin{cases} \mathbb{P}(x_{t+1} \mid x_t, a_t) & , x_{t+1} \in \bar{\mathcal{X}}(H^-_{t+1}) \\ 0 & , otherwise \end{cases}$$

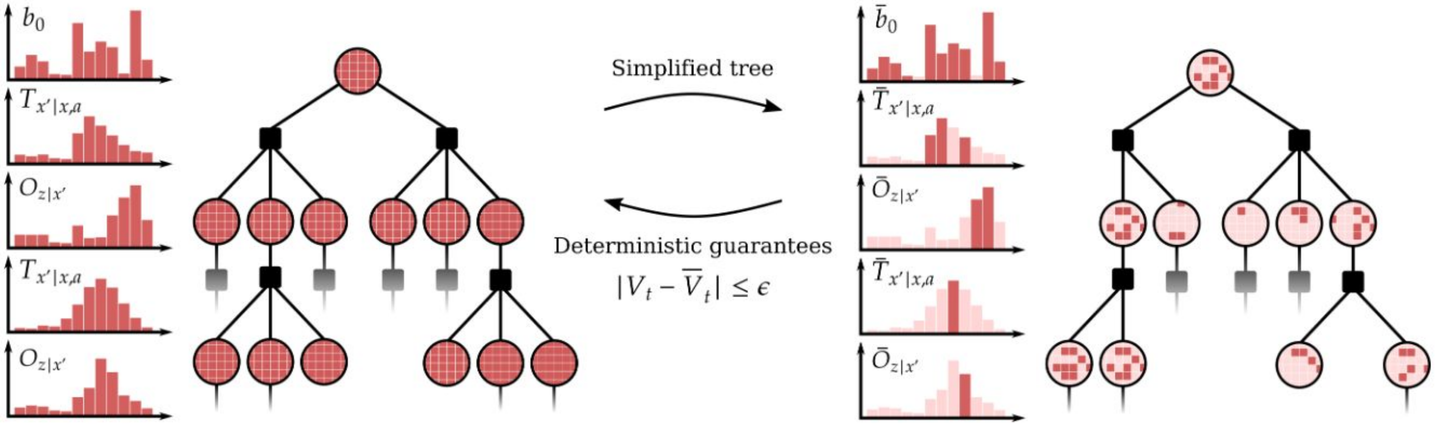
$$\bar{\mathbb{P}}(z_t \mid x_t) \triangleq \begin{cases} \mathbb{P}(z_t \mid x_t) & , z_t \in \bar{\mathcal{Z}}(H_t) \\ 0 & , otherwise \end{cases}$$

POMDPs with Deterministic Guarantees - Approach

With the simplified POMDP, we define a simplified value function,

$$\bar{V}^{\pi}(\bar{b}_t) \triangleq r(\bar{b}_t, \pi_t) + \bar{\mathbb{E}}_{z_{t+1:\mathcal{T}}}\left[\bar{V}^{\pi}(\bar{b}_{t+1})\right]$$

The formulation is flexible enough to allow any selection of the simplified state and observation spaces,



POMDPs with Deterministic Guarantees - Our Contribution

Derived upper and lower bounds for the optimal value function,

Lemma

Let $\mathcal{A}$ be the set of actions and $\mathcal{U}_0^\star(H_t)$, $\mathcal{L}_0^\star(H_t)$ be the upper and lower bounds of node H_t. Then, the optimal value at the root is bounded by,

$$\mathcal{L}_0^\star(H_0) \leq V^{\pi^*}(H_0) \leq \mathcal{U}_0^\star(H_0).$$

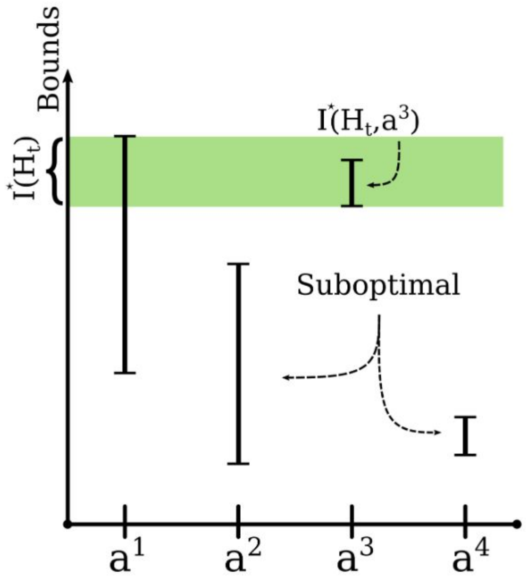
- The bounds are easier to compute than the optimal value function
- The bounds shrink monotonically as the algorithm explores the tree
- Converge to the optimal value function
- This is the first work to our knowledge to provide deterministic guarantees for anytime online POMDPs

POMDPs with Deterministic Guarantees - Our Contribution

Importantly, the bounds can be calculated during planning.

How can we use them?

- Pruning of sub-optimal branches
  - → Made possible by the deterministic guarantees
- Stopping criteria for the planning phase
  - → Made possible by the deterministic guarantees
- Finding the optimal solution in finite time
  - → Without recovering the theoretical tree



TECHNION Israel Institute of Technology

ANPL Autonomous Navigation and Perception Lab

POMDPs with Deterministic Guarantees - Our Contribution

Algorithm blueprint

represents most SOTA algorithms

(similar structure)

Can attach our bounds to any such algorithm

Algorithm 1 ALGORITHM-$\mathcal{A}$:

function SEARCH
while time permits **do**
  Generate states x from b_0.
  $\tau_0 \leftarrow x$
  $\bar{\mathbb{P}}_0 \leftarrow b(x = \tau_0 \mid h_0)$
  if $\tau_0 \notin \tau(h_0)$ **then**
    $\bar{\mathbb{P}}(h_0) \leftarrow \bar{\mathbb{P}}(h_0) + \bar{\mathbb{P}}_0$
  end if
  SIMULATE$(h_0, D, \tau_0, \bar{\mathbb{P}}_0)$.
end while
return

function SIMULATE$(h, d, \tau_d, \bar{\mathbb{P}}_d)$
if $d = 0$ **then**
  return
end if
Select action a.
Generate next states and observations, x', z.
$\tau_d,\ \bar{\mathbb{P}}_\tau \leftarrow$ FWDUPDATE$(ha, haz, \tau_d, \bar{\mathbb{P}}_\tau, x')$
Select next observation z.
SIMULATE$(haz, d-1, \tau_d, \bar{\mathbb{P}}_\tau)$
BWDUPDATE(h, ha, d)
return

function FWDUPDATE$(ha, haz, \tau_d, \bar{\mathbb{P}}_\tau, x')$
if $\tau_d \notin \tau(ha)$ **then**
  $\tau(ha) \leftarrow \tau(ha) \cup \{\tau_d\}$
  $\bar{R}(ha) \leftarrow \bar{R}(ha) + \bar{\mathbb{P}}_\tau \cdot r(x, a)$
end if
$\tau_d \leftarrow \tau_d \cup \{x'\}$
$\bar{\mathbb{P}}_\tau \leftarrow \bar{\mathbb{P}}_\tau \cdot Z_{z|x'} \cdot T_{x'|x,a}$
if $\tau_d \notin \tau(haz)$ **then**
  $\bar{\mathbb{P}}(haz) \leftarrow \bar{\mathbb{P}}(haz) + \bar{\mathbb{P}}_\tau$
  $\tau(haz) \leftarrow \tau(haz) \cup \{\tau_d\}$
end if
return

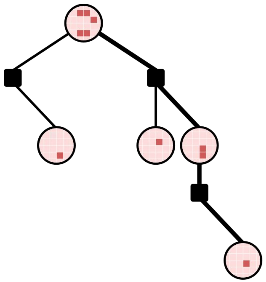
function BWDUPDATE(h, ha, d)
$\epsilon(ha) = \gamma^{D-d} V_{\max,d}(\bar{\mathbb{P}}(h) - \bar{\mathbb{P}}(ha)) + \gamma^{D-d-1} V_{\max,d+1}(\bar{\mathbb{P}}(ha) - \sum_{z|ha} \bar{\mathbb{P}}(haz))$
$U(ha) = \bar{R}(ha) + \gamma \sum_{z|ha} U(haz) + \epsilon(ha)$
$L(ha) = \bar{R}(ha) + \gamma \sum_{z|ha} L(haz) - \epsilon(ha)$
$U(h) \leftarrow \max_{a'} \{U(ha')\}$
$L(h) \leftarrow \max_{a'} \{L(ha')\}$
return

TECHNION Israel Institute of Technology

ANPL Autonomous Navigation and Perception Lab

POMDPs with Deterministic Guarantees - Our Contribution

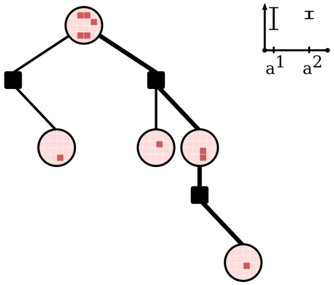
POMCP



$$\mathrm{UCT}(b_t, a_t) = \hat{Q}(b_t, a_t) + c \cdot \sqrt{\frac{\log N(b_t)}{n(b_t, a_t)}}$$

$$a_t = \arg\max_a \{UCT(b_t, a)\}$$

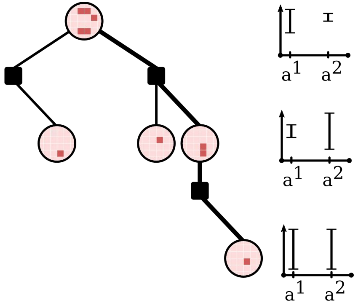
DB-POMCP



$$a_0 = \arg\max_a \{\mathcal{L}_0^{\star}(b_0, a)\}$$

$$a_t = \arg\max_a \{UCT(b_t, a)\}$$

RB-POMCP



$$a_0 = \arg\max_a \{\mathcal{L}_0^{\star}(b_0, a)\}$$

$$a_t = \arg\max_a \{\mathcal{U}_0^{\star}(b_0, a)\}$$

	POMCP	DB-POMCP	RB-POMCP
Deterministic guarantees	No	Yes	Yes
Pruning	No	Yes	Yes
Finite-time optimality	No	No	Yes

Autonomous Navigation and Perception Lab

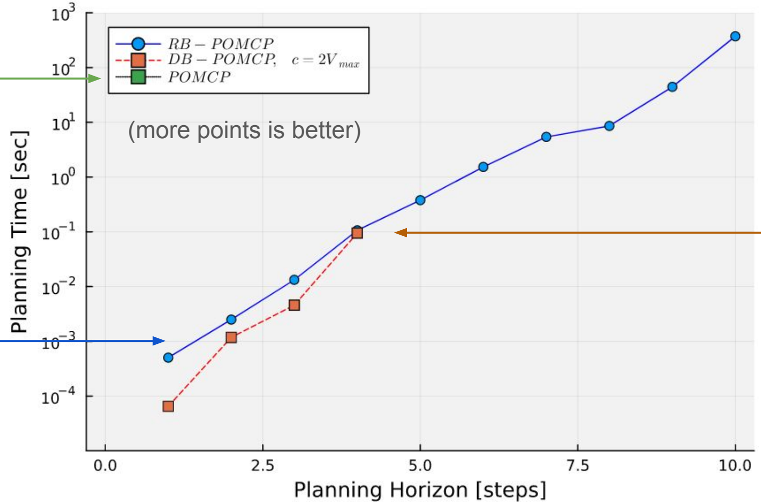
POMDPs with Deterministic Guarantees - Results

We compared the time it takes for the algorithms to find the **optimal** action

- SOTA algorithms excluded as they don't ensure optimal solutions
- Each point in the graph corresponds to the time it took to find the optimal action

Can't guarantee optimality

RB-POMCP empirically scales linearly with problem size



UCT effectively stops exploring before guaranteeing the optimal solution

TECHNION Israel Institute of Technology

ANPL Autonomous Navigation and Perception Lab

Simplified POMDP Algorithms with Performance Guarantees

Summary

TECHNION Israel Institute of Technology

ANPL Autonomous Navigation and Perception Lab

Thank you for listening!

TECHNION
Israel Institute
of Technology

ANPL | Autonomous Navigation and Perception Lab