Online Robust Planning under Model Uncertainty: A Sample-Based Approach

AAAI 2026

Presented by: Tamir Shazman

Advisor: Assoc. Prof. Vadim Indelman

TECHNION Israel Institute of Technology

ANPL - Autonomous Navigation and Perception Lab
Technion - Israel Institute of Technology

ANPL Autonomous Navigation and Perception Lab

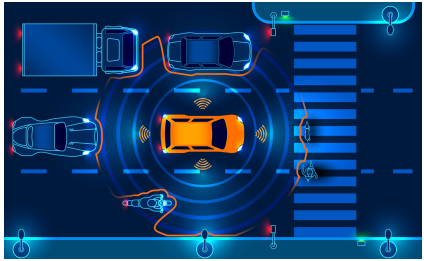
February 2026

Outline

Autonomous Planning

Context

- Autonomous agents operate in complex, dynamic environments.
- They must plan sequences of actions to maximize long-term rewards.
- **Example:** A robot navigating a warehouse or a drone delivering packages.



The Core Problem

- Most planners assume the world behaves exactly like their internal model.
- Reality is different: Friction changes, motors degrade, wind blows.

Background: Planning framework MDP Definition

A standard mathematical framework for planning is the Markov Decision Process (MDP), defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R})$:

- **State Space ($\mathcal{S}$):** All possible configurations of the environment.
  Example: The robot's position and velocity $(x, y, \dot{x}, \dot{y})$.
- **Action Space ($\mathcal{A}$):** All available actions to the agent.
  Example: Motor controls or steering angle.
- **Transition Model ($P(s'|s, a)$):** The probability of reaching state s' after taking action a in state s.
  Assumption: In a regular MDP, this probability distribution is considered absolute truth.
- **Reward Function ($\mathcal{R}(s, a)$):** The immediate feedback received.
  Example: $+100$ for reaching the goal, -10 for a collision.

Background: The MDP Objective

The Agent's Goal

- Find an optimal policy $\pi : \mathcal{S} \to \mathcal{A}$ that maps states to actions.
- Maximize the expected cumulative discounted reward over time.

The Value Function

The expected return from a starting state s, following policy π under a specific transition model P, is defined as:

$$V^{\pi,P}(s) = \mathbb{E}_P \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 = s, a_t = \pi(s_t) \right]$$

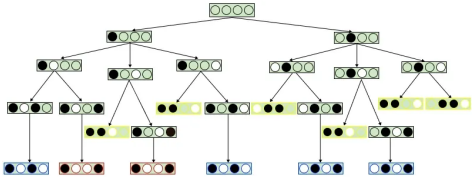
Key Components:

- $\gamma \in [0, 1)$**:** The discount factor, which prioritizes immediate rewards over distant, uncertain ones.
- P**:** The assumed transition dynamics (which MDPs assume is perfectly identical to the real world).

Background: Online Planning & Tree Search

Offline vs. Online Planning

- **Offline:** Computes a policy $\pi(s)$ for the entire state space. Intractable for large or continuous domains.
- **Online:** Computes the best action only for the *current* state, avoiding planning for unreachable states.



Common Online Solvers

- **Monte Carlo Tree Search (MCTS)**.
- **Sparse Sampling**.

  Key Property: Computation time is entirely independent of the size of the state space $|\mathcal{S}|$.

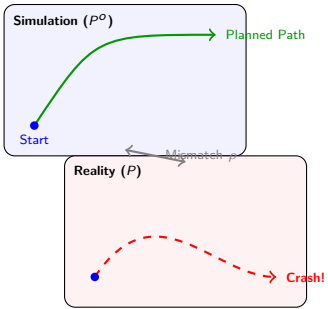
The Reality Gap

The Standard Assumption

- Online planners (e.g., MCTS, Sparse Sampling) rely entirely on their simulator of the transition model P^o.

The Reality

- Simulators are estimated from finite data or simplified physics.
- **Model Mismatch:** The simulator (P^o) $\neq$ Reality (P).
- Consequence: The agent optimizes for the *wrong* dynamics, leading to catastrophic failures when deployed.



Robust MDPs: Formal Definition

To mathematically address model uncertainty, a **Robust MDP (RMDP)** replaces the single transition model with an uncertainty set. It is defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R})$:

- **State Space $\mathcal{S}$, Action Space $\mathcal{A}$, Reward $\mathcal{R}$:**
  Defined exactly the same as in a regular MDP.
- **Uncertainty Set $\mathcal{P}$:** The Core Difference.
  Instead of a single known transition model P, we define a set of plausible transition models $\mathcal{P}$.
  Assumption: The true environment dynamics are unknown, but we assume they lie somewhere within this bounded set $\mathcal{P}$.
- **The Objective (Max-Min Planning):**

$$V^*(s) = \max_{\pi} \min_{P' \in \mathcal{P}} V^{\pi, P'}(s)$$

Robust MDPs: Our Uncertainty Model

How do we define the uncertainty set $\mathcal{P}$ in practice?

Total Variation Uncertainty Set

We define the set of plausible models $\mathcal{P}$ centered around our simulator's estimate P^o, bounded by a radius ρ:

$$\mathcal{P}_{s,a} = \{P'_{s,a} \in \Delta(\mathcal{S}) : D_{TV}(P'_{s,a}, P^o_{s,a}) \leq \rho\}$$

The Max-Min Bellman Equation

This leads to our robust objective function:

$$V^*(s) = \max_{a \in \mathcal{A}} \left[r(s,a) + \gamma \min_{P' \in \mathcal{P}_{s,a}} \mathbb{E}_{s' \sim P'}[V^*(s')] \right]$$

The Challenge: Solving this inner minimization over a continuous space of probability distributions is computationally expensive and intractable for large state spaces.

Contribution: Robust Sparse Sampling (RSS)

The Limitation of Prior Work

- Existing algorithms for the robust Bellman equation are designed for *offline* planning.
- They require iterating over the entire state space ($|\mathcal{S}|$), making them intractable for large or continuous domains.

Our Solution: Robust Sparse Sampling (RSS)

In our AAAI 2026 paper, *Online Robust Planning under Model Uncertainty: A Sample-Based Approach*, we introduce **RSS**—the **first** online algorithm to efficiently solve RMDPs with Theoretical Guarantees.

- Computes the robust update at every node of an online lookahead tree.
- Computation time is **strictly independent** of the state space size.

Challenge 1: Computation Complexity

The Problem:

- We need to solve the inner minimization: $\min_{P' \in \mathcal{P}_{s,a}} \mathbb{E}_{s' \sim P'}[V^*(s')]$.
- This requires searching over a continuous space of probability distributions.
- For large state spaces, doing this online is computationally intractable.

How can we make this tractable for online planning?

Solution: Dual Formulation

Key Insight: We can transform the complex optimization over distributions into a tractable scalar optimization using strong duality.

Dual Objective

$$\min_{P' \in \mathcal{P}_{s,a}} \mathbb{E}_{s' \sim P'}[V^*(s')] = \min_{\eta \in [0, \frac{2}{\rho(1-\gamma)}]} \underbrace{(\mathbb{E}_{s' \sim P^o}[(\eta - V^*(s'))_+] - \eta(1-\rho))}_{F^{\rho}_{s,a}(\eta)}$$

- **Benefit:** We now optimize over a single scalar variable η instead of a high-dimensional distribution P'.
- **Issue:** The expectation $\mathbb{E}_{s' \sim P^o}$ is still an integral over the entire next-state space.

Challenge 2: Intractable Expectation

The Problem:

- The dual formulation requires evaluating the expectation: $\mathbb{E}_{s' \sim P^o}[(\eta - V^*(s'))_+]$.
- In continuous spaces or with complex physics engines, we cannot compute this integral in closed form.

How do we approximate this expectation efficiently?

Solution: Sample Average Approximation (SAA)

Approach: We approximate the expectation using C discrete samples drawn from our generative model P^o.

$$\hat{F}^{\rho}_{s,a}(\eta) = \frac{1}{C}\sum_{i=1}^{C}(\eta - V^*(s'_i))_+ - \eta(1-\rho)$$

Properties of this formulation:

- This function is **piecewise-linear and convex**.
- It can be solved exactly in $O(C \log C)$ time simply by sorting the sampled values $V^*(s'_i)$.
- It successfully bridges the gap between sample-based algorithms (online) and robust uncertainty sets (theory).

Challenge 3: Unknown Future Values

The Problem:

- The SAA equation assumes we already know the true optimal robust values $V^*(s'_i)$ of all the next states.
- In an online setting, we are exploring these states for the first time; their values are unknown.

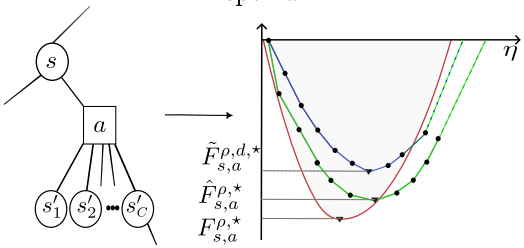
How do we estimate these robust values on the fly?

Solution: Recursive Tree Search (RSS)

Recursive Estimator: We build a lookahead tree. At depth d, we approximate the unknown true value using the value returned by our algorithm at depth $d-1$.

$$\tilde{F}_{s,a}^{\rho,d}(\eta) = \frac{1}{C}\sum_{i=1}^{C}(\eta - \hat{V}_{d-1}(s_i'))_+ - \eta(1-\rho)$$

1. **Sample** C next states from the simulator.
2. **Recurse** down the tree to get children's values.
3. **Sort** the returned children's values.
4. **Solve** for η to execute a robust backup.



Theoretical Guarantees

RSS is the **first online robust planner** with finite-sample guarantees.

Theorem 1 (Informal)

For any state s and desired accuracy ϵ, RSS returns a policy π such that:

$$|V^{\pi}(s) - V^{*}(s)| \leq \epsilon$$

provided the planning horizon H and sample width C are sufficiently large.

Key Properties:

- **State-Space Independent:** Complexity depends on H and C, not $|\mathcal{S}|$. Suitable for continuous domains.
- **Finite-Sample:** We bound the error introduced by SAA and recursion.

Results: FrozenLake (Grid World)

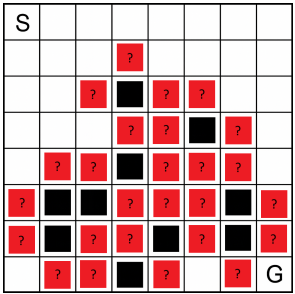
Setup

- 8×8 Grid with hazardous holes.
- The simulator assumes a safer floor, underestimating the true slip probability by ρ.

Findings

- Standard Sparse Sampling (SS) is dangerously overconfident.
- **RSS (Ours)** anticipates the model error and plans robust, safer paths.

ρ	RSS (Ours)	SS
0.2	**0.171**	0.123
0.3	**0.145**	0.109
0.4	**0.126**	0.098
0.5	**0.127**	0.080

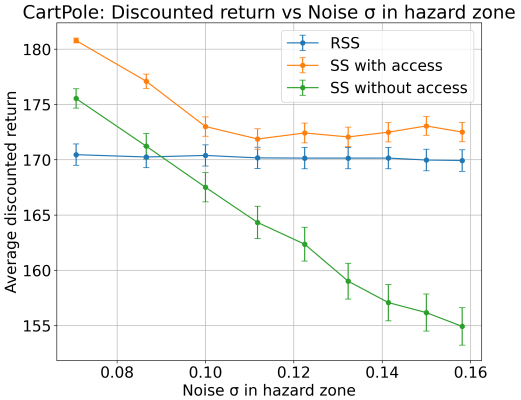


Results: Robust CartPole

Setup: Continuous control domain featuring a "Hazard Zone" where environmental noise is significantly higher than modeled by the simulator.

Comparison

- Standard SS: Performs well in the safe zone, but collapses entirely as unmodeled noise increases in the hazard zone.
- RSS: Is slightly conservative in perfectly safe scenarios, but provides **significant safety** improvements by hedging against high-risk outcomes.



Conclusions

- We introduced **Robust Sparse Sampling (RSS)**.
- **Bridge:** Successfully connects offline Robust MDP theory with scalable online, sample-based planning.
- **Efficiency:** Solves the robust Bellman update efficiently ($O(C \log C)$) at every node using Dual SAA.
- **Guarantees:** Provides the first online robust planner with finite-sample performance bounds that remain independent of the state space size.
- **Impact:** Enables safe, resilient planning in continuous domains despite severe model mismatch.

Thank You!

Questions?

Paper presented at AAAI-26, Singapore