ROSARL: Reward-Only Safe Reinforcement Learning

Geraud Nangue Tasse, Tamlin love, Mark Nemecek, Steven James and Benjamin Rosman
University of the Witwatersrand, Johannesburg, South Africa

RLC WITS RAIL LAB

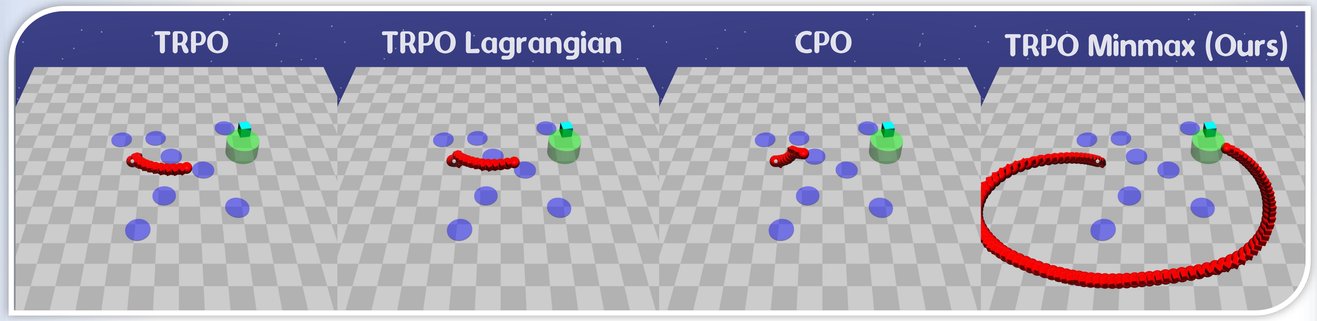
Is a scalar reward enough to solve tasks safely? Yes!
How? Replace the rewards for unsafe transitions with the Minmax penalty, defined by the diameter and controllability of the environment

What we want

We want agents (e.g robots) that can **optimally reach safe** goal states in an environment (e.g real world) **while minimising the reach probability of unsafe** goal states

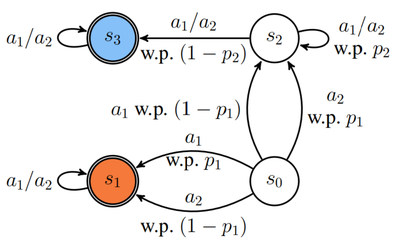
Too small penalties may result in unsafe behaviour

Too large penalties may result in longer learning times

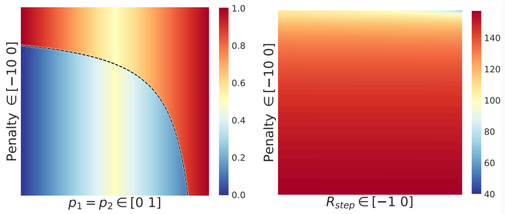


Motivating example

- Consider the simple Chain-walk MDP



- Learning the optimal policy for varying choices of penalty gives the following observation



Failure rates *Total timesteps to converge*

- **Prior works** generally use constrained optimization based on a cost function, but **usually have a large cost threshold (e.g. 25).** Hence they generally **fail at minimising the reach probability of unsafe goal states**, i.e. when the cost threshold is 0.

Solution: Minmax penalty

- The Minmax penalty is the smallest penalty for unsafe states that leads to safe optimal policies, **regardless of task rewards**.
- We identify that it can be bounded by considering a notion of diameter (D) and controllability (C) of the environment:

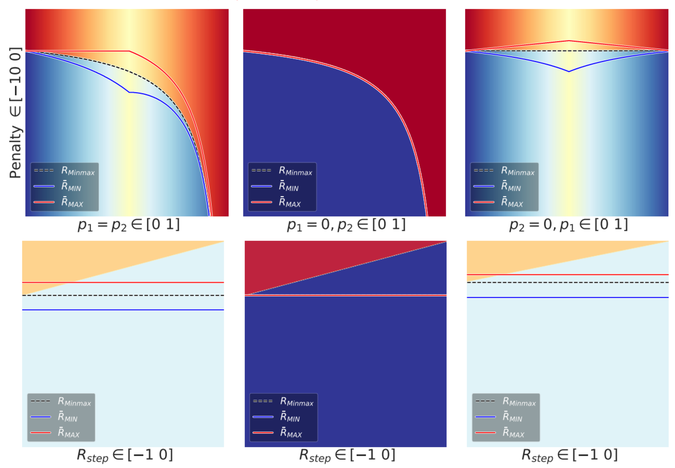
$$D' := \max_{s \in \mathcal{S} \setminus \mathcal{G}} \max_{\pi \in \Pi} \mathbb{E}\left[T(s_T \in \mathcal{G} \setminus \mathcal{G}' | \pi\right]$$

$$C := \min_{s \in S \setminus \mathcal{G}} \min_{\substack{\pi \in \Pi \\ P_s^\pi(s_T \notin \mathcal{G}') \neq 0}} P_s^\pi(s_T \notin \mathcal{G}')$$

$$\bar{R}_{\text{MAX}} := R_{\text{MIN}} D'$$

$$\bar{R}_{\text{MIN}} := (R_{\text{MIN}} - R_{\text{MAX}}) \frac{D}{C}$$

Failure rates of optimal policies in the chain-walk MDP



Theoretical results

Theorem 1 (Estimation)

- *For any given controllable environment,*
- *We can learn D and C to convergence by using policy evaluation.*

Theorem 2 (Safety Bounds)

- *Let the task rewards be bounded by* $[R_{Min}, R_{Max}]$
- *Then* $\bar{R}_{Min} \leq R_{Minmax} \leq \bar{R}_{Max}$

Theorem 3 (Complexity)

- Estimating the Minmax penalty R_{Minmax} accurately is NP-hard.

Practical algorithm

In practice, we can learn safe policies by **estimating the lower-bound penalty** using the learned value function

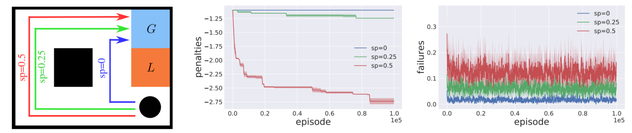
Algorithm 2: RL while learning Minmax penalty

Input : RL algorithm $\mathbf{A}$, max timesteps T
Initialise : $R_{\text{MIN}} = 0, R_{\text{MAX}} = 0, V_{\text{MIN}} = R_{\text{MIN}}, V_{\text{MAX}} = R_{\text{MAX}}$, π and V as per $\mathbf{A}$
for t in T **do**
  observe a state s_t, **take** an action a_t using π as per $\mathbf{A}$, and **observe** s_{t+1}, r_t
  $R_{\text{MIN}}, R_{\text{MAX}} \leftarrow \min(R_{\text{MIN}}, r_t), \max(R_{\text{MAX}}, r_t)$
  $V_{\text{MIN}}, V_{\text{MAX}} \leftarrow \min(V_{\text{MIN}}, R_{\text{MIN}}, V(s_t)), \max(V_{\text{MAX}}, R_{\text{MAX}}, V(s_t))$
  $\bar{R}_{\text{MIN}} \leftarrow V_{\text{MIN}} - V_{\text{MAX}}$
  $r_t \leftarrow \bar{R}_{\text{MIN}}$ **if** $s_{t+1} \in \mathcal{G}'$ **else** r_t
  update π and V with (s_t, a_t, s_{t+1}, r_t) as per $\mathbf{A}$
end for

Experiments

When theoretical assumptions hold

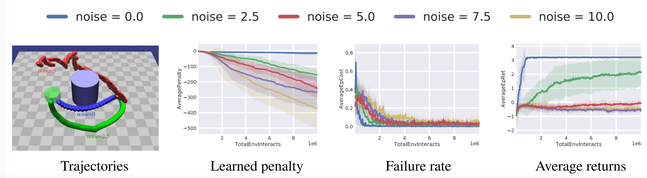
Consider the lava grid-world **with different noise levels (sp)**. Using Algorithm 2 with Q-learning, the agent **learns the short or long safe policies depending on noise (sp),** to reach G while avoiding the lava L



(a) Trajectories (b) Learned penalty (c) Failure rate

When theoretical assumptions do not hold

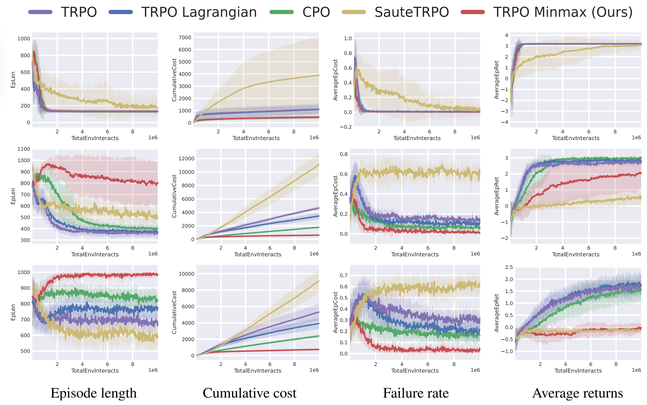
Consider the SafetyGym domain **with different noise levels**. Using Algorithm 2 with function approximation (TRPO), the agent again **learns the short or long safe policies depending on noise**



Trajectories Learned penalty Failure rate Average returns

Comparison to prior Safe RL approaches

*We learn **safer policies than baselines,** while still solving task goals when doing so does not compromise safety*



Episode length Cumulative cost Failure rate Average returns

(top) noise $= 0$, **(middle)** noise $= 2.5$, and **(bottom)** noise $= 5$