Optimizing Expectation with Guarantees in POMDPs

Guillermo A. Pérez
joint with K. Chatterjee, P. Novotný, J.-F. Raskin, D. Zikelic

University of Antwerp

Workshop: Graph/Stochastic Games @ Mons, 2019

Why this model

Partially observable Markov decision processes

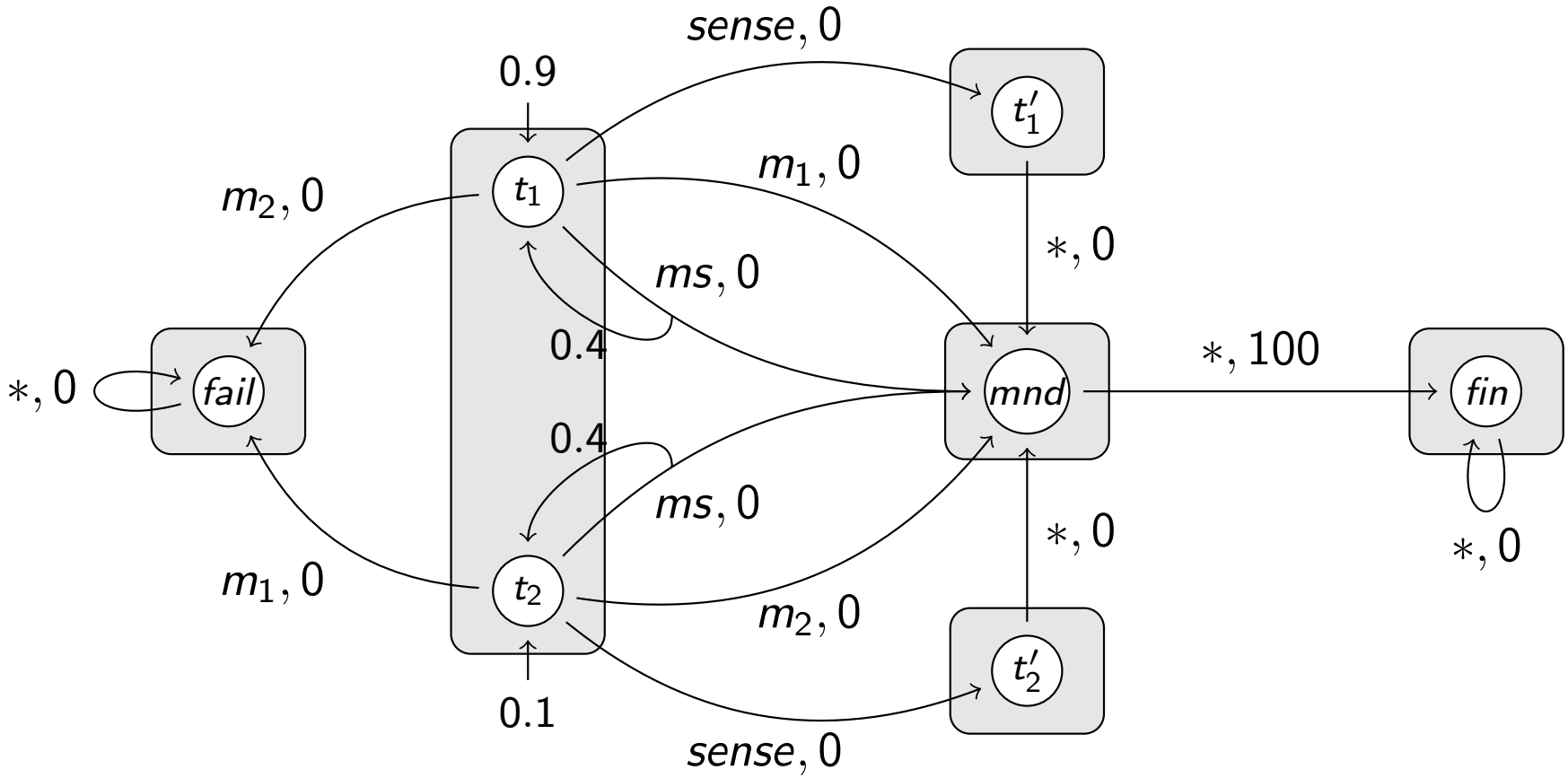
POMDPs are useful models for

- Decision-making under uncertainty
- Planning
- Reinforcement learning
- Robotics

Main features of POMDPs

1. Rewards: capture how good or bad an action is, per state
2. Probabilities: imprecise actuators, unknown environment, etc.
3. Partial observation: e.g. fixed amount of sensors with limited precision

POMDPs, more graphically



The initial distribution assigns $\frac{9}{10}$ to state t_1 and $\frac{1}{10}$ to t_2. With probability $\frac{2}{5}$ action *ms* from t_i loops back.

Motivation

Formal guarantees

- In safety critical applications, the worst-case behaviour has priority, yet
- we would still like to maximize the expected payoff.
- Inspired by the beyond worst-case approach from formal verification.
- Solution implemented as an extension of the partially-observable Monte Carlo planning (POMCP) algorithm.

Our contribution

Optimizing the expected discounted-sum payoff in a given POMDP while guaranteeing a payoff of at least a given threshold.

POMDPs, more formally

Definition (POMDPs)

A POMDP is a tuple $P = (S, \mathcal{A}, \delta, r, \mathcal{O}, \lambda)$ where:

- S are states,
- $\mathcal{A}$ are actions,
- $\delta : S \times \mathcal{A} \to \mathcal{D}(S)$ is a probabilistic transition function,
- $r : S \times \mathcal{A} \to \mathbb{R}$ is a reward function,
- $\mathcal{Z}$ is a set of observations (a partitioning of S),
- $\mathcal{O} : S \to \mathcal{Z}$ maps every state s to its observation $\mathcal{O}(s)$,
- $\lambda : S \to \mathcal{D}(S)$ is the initial belief on the states of P.

What does an agent win? How do they play?

Discounted sum

Given a play $\varrho = s_0 a_0 s_1 a_1 s_2 a_2 \ldots$ and a discount factor[1] $0 \leq \gamma < 1$, the infinite-horizon discounted payoff Disc_γ of ϱ is:

$$\mathrm{Disc}_\gamma(\varrho) := \sum_{i=0}^{\infty} \gamma^i r(s_i, a_i).$$

Belief

Agent/system maintains a belief b about the current state:

- $b(s)$ is the probability of the current state being s.

Strategies

A strategy is a function assigning to each history, i.e. an observation-action sequence $a_0 o_1 a_1 o_2 \ldots o_n$, a distribution over actions.

[1] assumed to be $1/2$ for this talk

Maximizing the discounted-sum in POMDPs

Values of a strategy

Given a strategy σ, discount factor γ, and initial belief λ, we have the expected value of σ from an initial distribution λ:

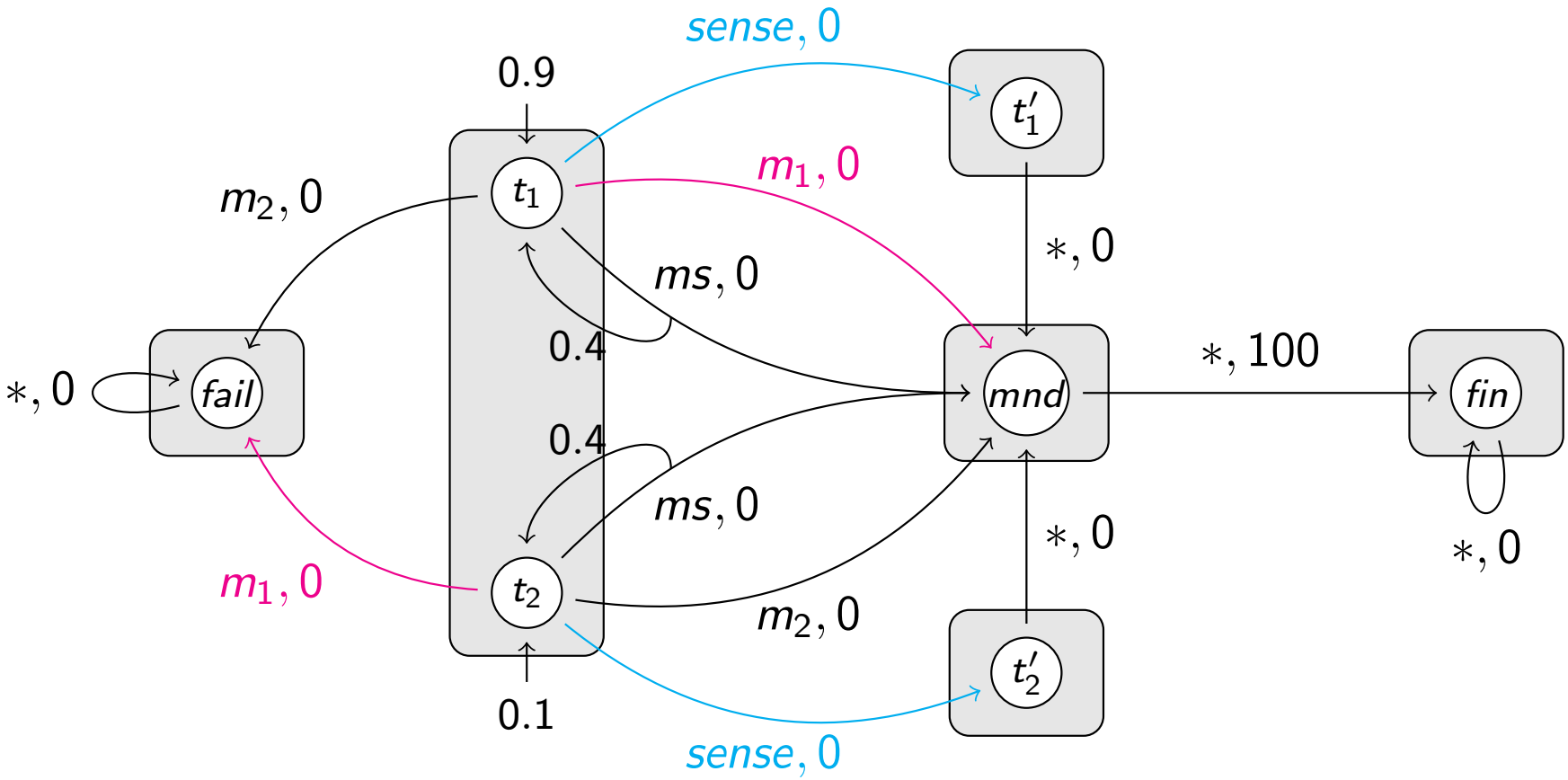
$$\mathit{eVal}(\sigma) := \mathbb{E}^{\sigma}_{\lambda}[\mathsf{Disc}_{\gamma}].$$

The worst-case value of σ from belief λ is

$$\mathit{wVal}(\sigma) := \inf_{\varrho_\sigma} \mathsf{Disc}_{\gamma}(\varrho_\sigma)$$

where ϱ_σ is any play consistent with σ and which starts in a state sampled from λ.

Can't we maximize both values at the same time?



- To get $eVal = 45$, m_1 is played as first action. With non-zero probability, this *fails* and the payoff is 0.
- To get $eVal = wVal = 25$, *sense* is played as first action.

The Guaranteed Payoff Optimization Problem

Expected value with guarantees

The best expected value obtainable while ensuring a worst-case payoff of at least t is

$$gVal(t) := \sup_{\sigma} \{eVal(\sigma) \mid wVal(\sigma) \geq t\}.$$

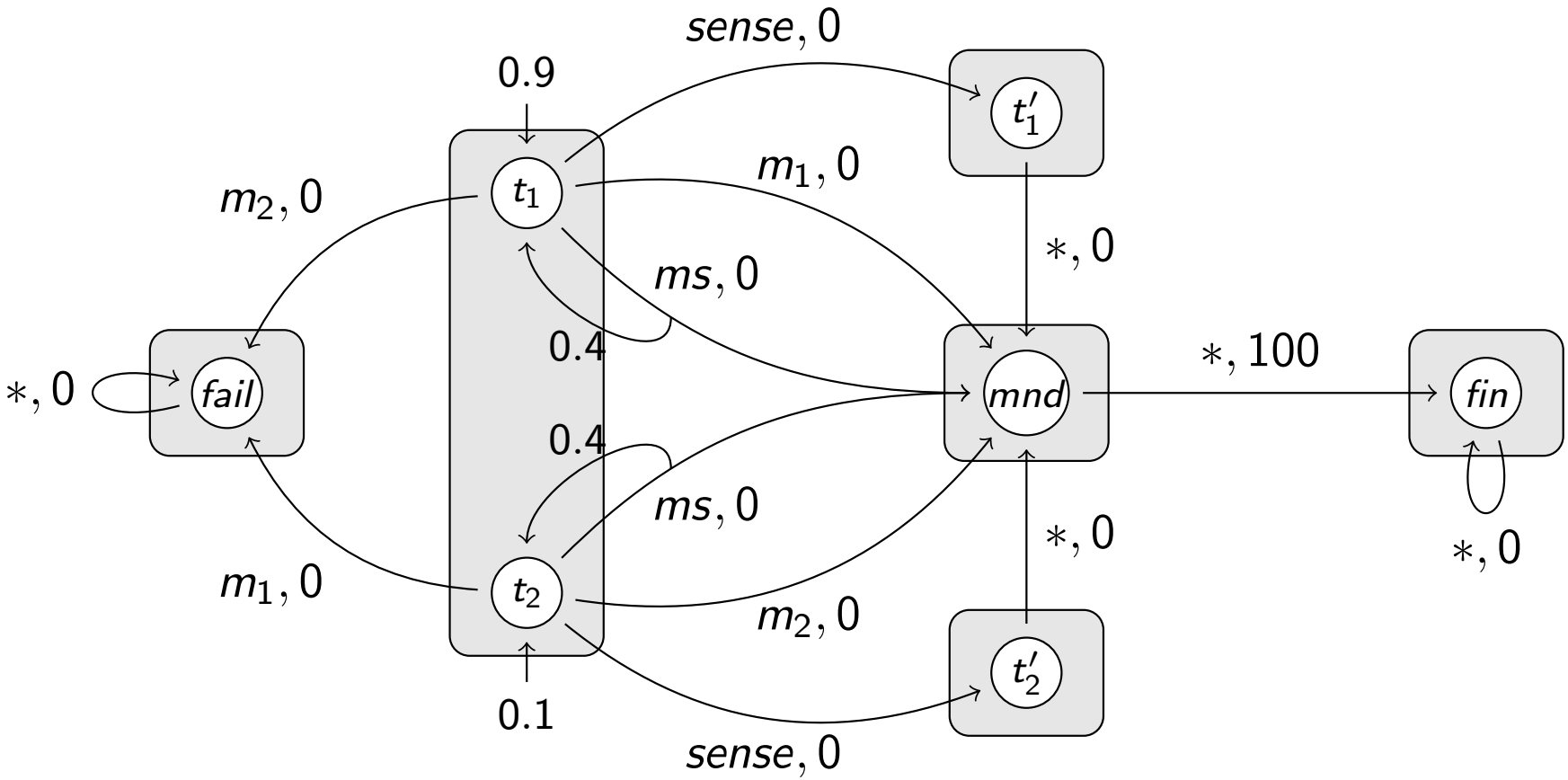
Guaranteed payoff (expectation) optimization (GPO)

Given a POMDP and a threshold $t \in \mathbb{R}$, we want to produce a strategy σ such that

1. σ satisfies a hard threshold constraint: $wVal(\sigma)$ is at least t.
2. Among all policies that satisfy item 1, σ has ε-optimal expected value, i.e.

$$eVal(\sigma) \geq gVal(t) - \varepsilon.$$

Belief-based strategies are not sufficient



- For threshold $t = 5$, we should play *ms* twice and then *sense* if you have not yet succeeded.
- The above strategy is not belief-based!

Our solution: keep track of your debt

Remaining "payoff debt"

For every history h we complement the belief with the value $rem^t_\gamma(h)$.
That is, the remaining payoff that must be obtained to meet threshold t:

$$rem^t_\gamma(h) := \frac{\{t - \min\left(\mathrm{Disc}_\gamma(\varrho) \mid \mathrm{Hist}(\varrho) = h\right)\}}{\left(\gamma^{\mathrm{len}(h)}\right)}.$$

Future (worst-case) values

The future value of b_h is the maximal worst-case payoff obtainable from a state sampled from b_h:

$$fVal(b_h) := fVal(\mathrm{Supp}(b_h)) := \sup_\sigma wVal^{b_h}(\sigma).$$

Combining both new values

Allowed actions

If the current belief after witnessing a history h is b_h, we allow an action a only if from all a-reachable beliefs we can meet our payoff debt:

$$\min_{s \in \mathrm{Supp}(b_h)} \min_{o \in \mathcal{Z}} \left(r(s,a) + \gamma \cdot fVal(b_{hao}) \right) \geq rem^t_\gamma(h).$$

Aren't those values hard to compute?

Combining both new values

Allowed actions

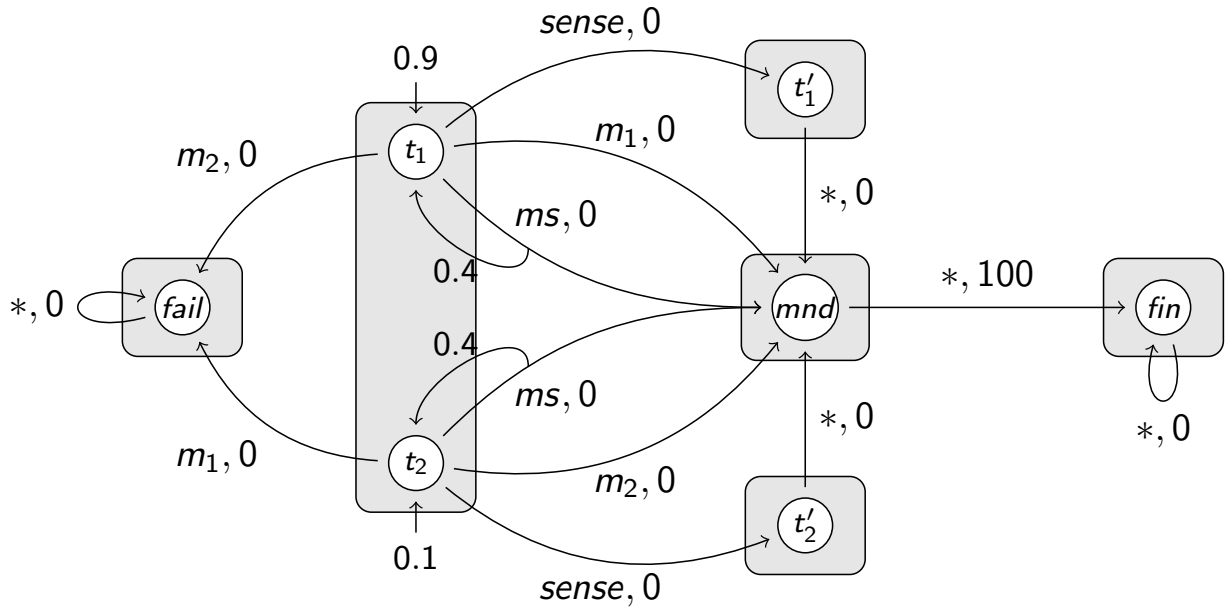
If the current belief after witnessing a history h is b_h, we allow an action a only if from all a-reachable beliefs we can meet our payoff debt:

$$\min_{s \in \mathrm{Supp}(b_h)} \min_{o \in \mathcal{Z}} \left(r(s,a) + \gamma \cdot \mathit{fVal}(b_{hao}) \right) \geq \mathit{rem}^t_\gamma(h).$$

Aren't those values hard to compute?

- Future-value lower bounds can be used instead of the exact ones;
- for observable-reward POMDPs, we give algorithms to compute the exact values in the paper.
- (For general POMDPs, *fVal* is not known to be computable!)

Example: disallowing actions for t = 12



Initially $rem^{12}_{0.5}(\cdot) = 12$, so only *ms* and *sense* are allowed since

$$r(t_i, m_{3-i}) + \gamma \cdot fVal(\{fail\}) = 0 < rem^{12}_{0.5}(\cdot) = 12, \text{ for } i \in \{1, 2\}.$$

- If *sense* is played, we get a payoff of 25.
- If *ms* is played and the next observation witnessed is $\mathcal{O}(t_1) = \mathcal{O}(t_2)$, the only allowed action is *sense* because for all $i \in \{1, 2\}$

$$r(t_i, ms) + \gamma \cdot fVal(\{t_1, t_2\}) = 12.5 < rem^{12}_{0.5}(ms\ \mathcal{O}(t_1)) = 24$$

So *sense* is played and we get $100/2^3 = 12.5$.

Implementation

Partially-observable Monte Carlo planning

POMCP is an online planning method which in each decision epoch aims to select the best action to optimize the expected payoff, given the current history h.

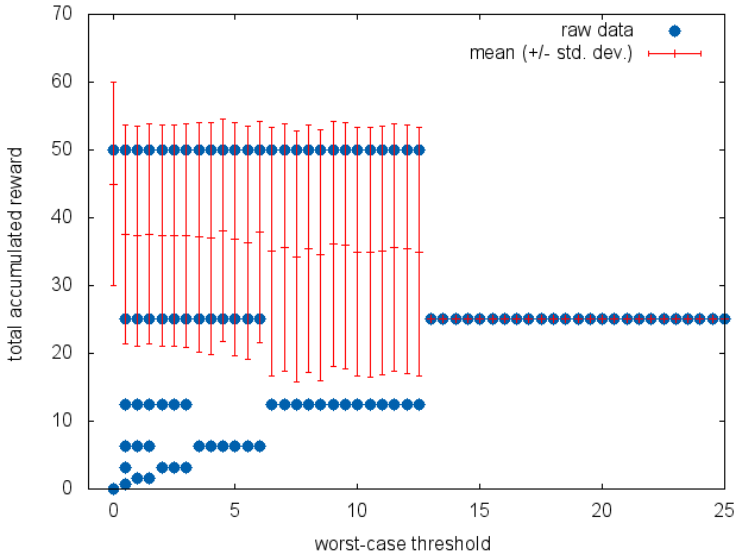
- POMCP has been used for learning to play video games
- It performs a number of finite-horizon simulations starting from the current to compute a local approximation of the optimal expected value function.

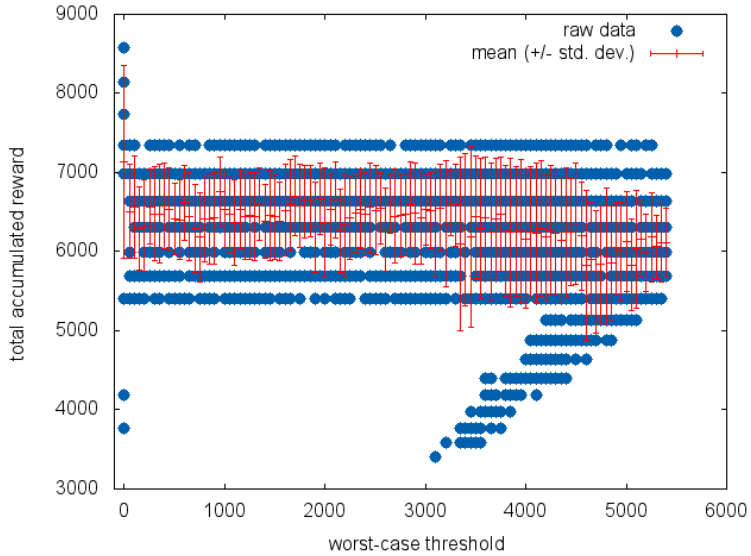
Instrumenting POMCP

We hacked the POMCP algorithm to use allowed actions only.

- Nowadays, this has been termed a shield.
- (So we were doing shielded RL before it was cool!)

Experimental results





- A circle with coordinates (x, y) corresponds to a run of our POMCP with worst-case threshold x, that obtained y as accumulated payoff

1. The POMDP from the example; and
2. a toy robot-motion planning example, all with increasing worst-case thresholds until $\mathit{fVal}(\mathrm{Supp}(\lambda))$.

Latency with planning horizon of 1K

No.	States	Actions	Obs.	P.-proc.	Avg. Lat.
Tiger	7	4	6	< 0.001s	< 0.009s
RSample	10207	7	168	184s	0.816s
Hallway	2039	3	18	2.02s	1.308s

Conclusions

Want probabilistic (not worst-case) guarantees when optimizing the expected value?

- Krishnendu Chatterjee, Adrián Elgyütt, Petr Novotný, Owen Rouillé: **Expectation Optimization with Probabilistic Guarantees in POMDPs with Discounted-Sum Objectives.** IJCAI 2018: 4692–4699.

SYNT camp @ ETAPS'19

- Do you want to learn more about efficiently solving games for planning/synthesis?
- Are you interested in the synthesis competition?
- Sunday workshop/tutorial before main conference.

https://conf.researchr.org/track/etaps-2019/syntcompcamp-2019-tutorial