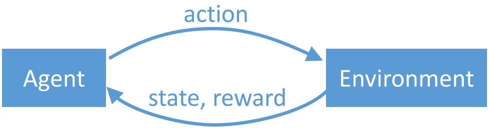
Practical Open-Loop Optimistic Planning

Edouard Leurent[1,2], **Odalric-Ambrym Maillard**[1]
[1] SequeL, Inria Lille – Nord Europe
[2] Renault Group

ECML PKDD 2019

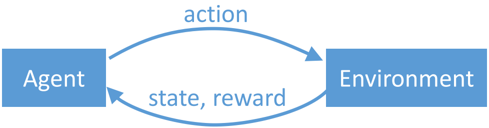
Würzburg, September 2019

Motivation — Sequential Decision Making



Markov Decision Processes

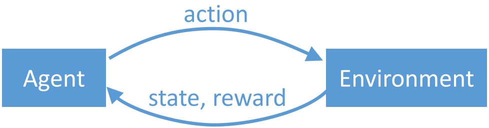
Motivation — Sequential Decision Making



Markov Decision Processes

1. Observe state $s \in S$;

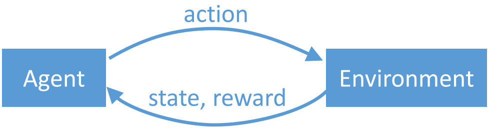
Motivation — Sequential Decision Making



Markov Decision Processes

1. Observe state $s \in S$;
2. Pick a discrete action $a \in A$;

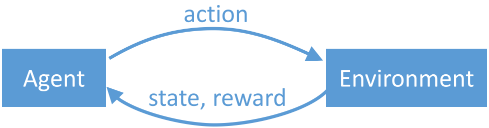
Motivation — Sequential Decision Making



Markov Decision Processes

1. Observe state $s \in S$;
2. Pick a discrete action $a \in A$;
3. Transition to a next state $s' \sim \mathbb{P}\left(s'|s, a\right)$;

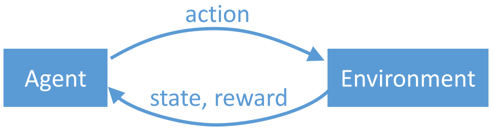
Motivation — Sequential Decision Making



Markov Decision Processes

1. Observe state $s \in S$;
2. Pick a discrete action $a \in A$;
3. Transition to a next state $s' \sim \mathbb{P}\left(s'|s,a\right)$;
4. Receive a bounded reward $r \in [0,1]$ drawn from $\mathbb{P}\left(r|s,a\right)$.

Motivation — Sequential Decision Making



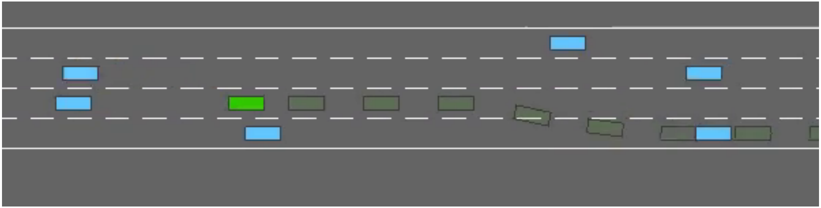
Markov Decision Processes

1. Observe state $s \in S$;
2. Pick a discrete action $a \in A$;
3. Transition to a next state $s' \sim \mathbb{P}\left(s'|s,a\right)$;
4. Receive a bounded reward $r \in [0,1]$ drawn from $\mathbb{P}\left(r|s,a\right)$.

Objective: maximise $V = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$

Motivation — Example

The `highway-env` environment

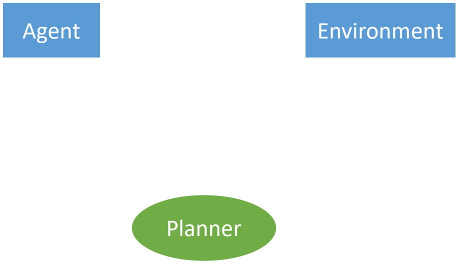


We want to handle stochasticity.

Motivation — How to solve MDPs?

Online *Planning*

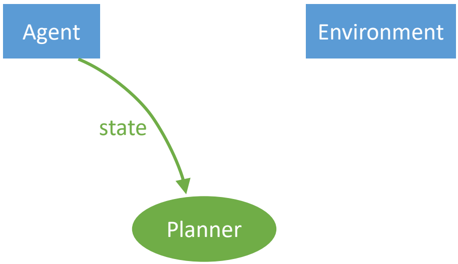
- we have access to a generative model:
  ↳ yields samples of $s', r \sim \mathbb{P}(s', r|s, a)$ when queried



Motivation — How to solve MDPs?

Online *Planning*

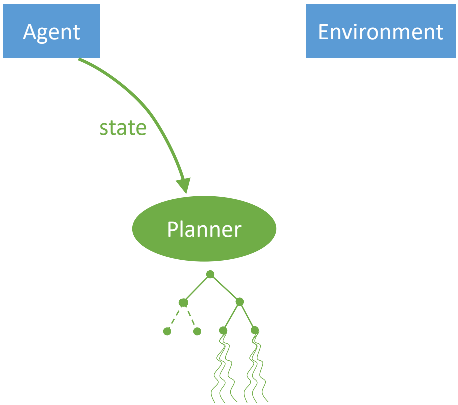
- we have access to a generative model:
  ↳ yields samples of $s', r \sim \mathbb{P}(s', r|s, a)$ when queried



Motivation — How to solve MDPs?

Online *Planning*

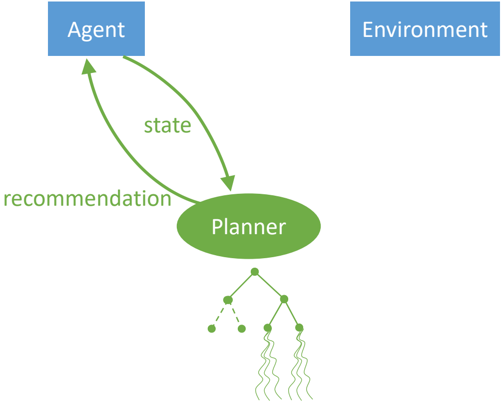
- we have access to a generative model:
  ↳ yields samples of $s', r \sim \mathbb{P}(s', r | s, a)$ when queried



Motivation — How to solve MDPs?

Online *Planning*

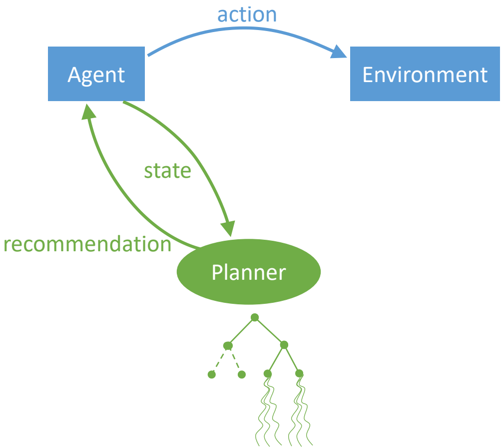
- we have access to a generative model:
  ↳ yields samples of $s', r \sim \mathbb{P}(s', r|s, a)$ when queried



Motivation — How to solve MDPs?

Online *Planning*

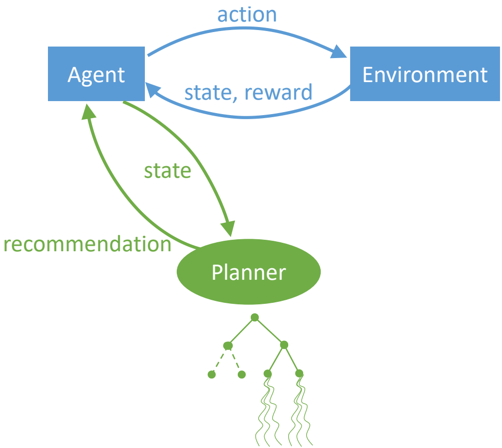
- we have access to a generative model:
  ↳ yields samples of $s', r \sim \mathbb{P}(s', r|s, a)$ when queried



Motivation — How to solve MDPs?

Online *Planning*

- we have access to a generative model:
  ↳ yields samples of $s', r \sim \mathbb{P}(s', r|s, a)$ when queried



Motivation — How to solve MDPs?

Online *Planning*

- fixed budget: the model can only be queried n times

$$\text{Objective: minimize } \mathbb{E} \underbrace{V^* - V(n)}_{\text{Simple Regret } r_n}$$

An exploration-exploitation problem.

Optimistic Planning

Optimism in the Face of Uncertainty

Given a set of options $a \in A$ with uncertain outcomes, try the one with the highest possible outcome.

Optimistic Planning

Optimism in the Face of Uncertainty

Given a set of options $a \in A$ with uncertain outcomes, try the one with the highest possible outcome.

- Either you performed well;

Optimistic Planning

Optimism in the Face of Uncertainty

Given a set of options $a \in A$ with uncertain outcomes, try the one with the highest possible outcome.

- Either you performed well;
- or you learned something.

Optimistic Planning

Optimism in the Face of Uncertainty

Given a set of options $a \in A$ with uncertain outcomes, try the one with the highest possible outcome.

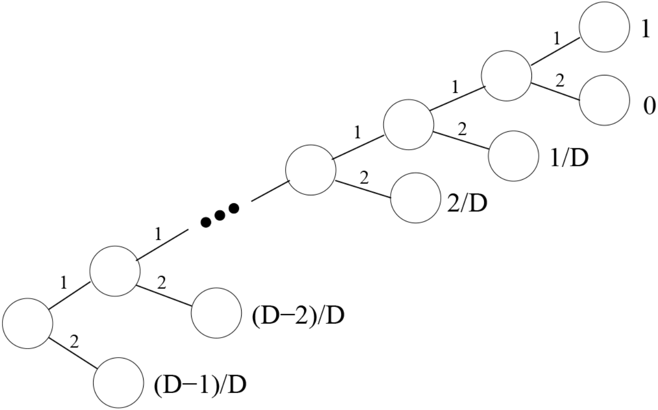
- Either you performed well;
- or you learned something.

Instances

- Monte-carlo tree search (`MCTS`) [Coulom 2006]: `CrazyStone`
- Reframed in the bandit setting as `UCT` [Kocsis and Szepesvári 2006], still very popular (e.g. `Alpha Go`).
- Proved asymptotic consistency, but no regret bound.

Analysis of UCT

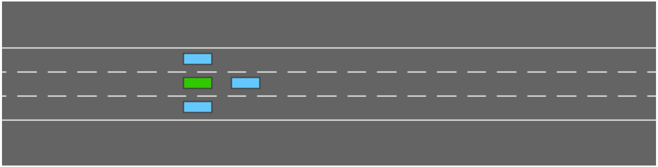
It was analysed in [Coquelin and Munos 2007]

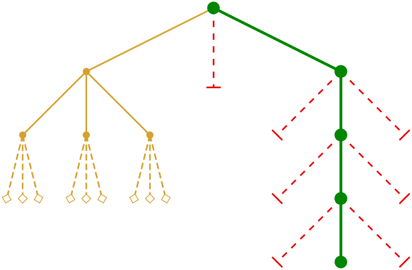


The sample complexity of is lower-bounded by $O(\exp(\exp(D)))$.

Failing cases of UCT

Not just a theoretical counter-example.





Can we get better guarantees?

OPD: Optimistic Planning for Deterministic systems

- Introduced by [Hren and Munos 2008]
- Another optimistic algorithm
- Only for deterministic MDPs

Theorem (OPD sample complexity)

$$\mathbb{E}\, r_n = \mathcal{O}\left(n^{-\frac{\log 1/\gamma}{\log \kappa}}\right), \textit{if } \kappa > 1$$

Can we get better guarantees?

OPD: Optimistic Planning for Deterministic systems

- Introduced by [Hren and Munos 2008]
- Another optimistic algorithm
- Only for deterministic MDPs

Theorem (OPD sample complexity)

$$\mathbb{E}\, r_n = \mathcal{O}\left(n^{-\frac{\log 1/\gamma}{\log \kappa}}\right), \textit{if } \kappa > 1$$

OLOP: Open-Loop Optimistic Planning

- Introduced by [Bubeck and Munos 2010]
- Extends OPD to the stochastic setting
- Only considers open-loop policies, i.e. sequences of actions

The idea behind OLOP

A direct application of Optimism in the Face of Uncertainty

1. We want

$$\max_a V(a)$$

The idea behind OLOP

A direct application of Optimism in the Face of Uncertainty

1. We want

$$\max_a V(a)$$

2. Form upper confidence-bounds of sequence values:

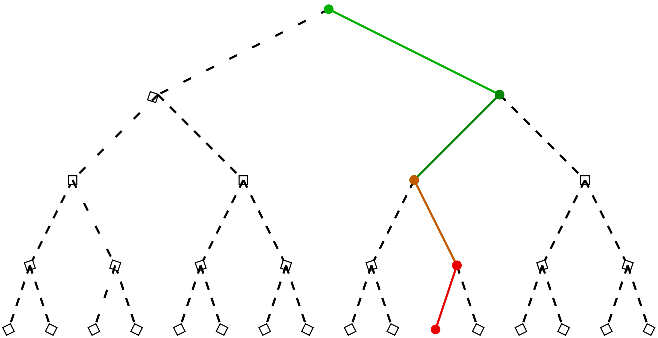
$$V(a) \leq U_a \quad \text{w.h.p}$$

The idea behind OLOP

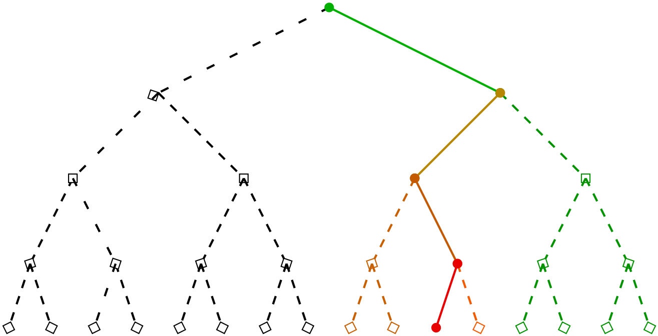
A direct application of Optimism in the Face of Uncertainty

1. We want
$$\max_a V(a)$$
2. Form upper confidence-bounds of sequence values:
$$V(a) \leq U_a \quad \text{w.h.p}$$
3. Sample the sequence with highest UCB:
$$\arg\max_a U_a$$

The idea behind OLOP



The idea behind OLOP



Under the hood

Upper-bounding the value of sequences

$$V(a) = \overbrace{\sum_{t=1}^{h} \gamma^t \mu_{a_{1:t}}}^{\text{follow the sequence}} + \overbrace{\sum_{t \geq h+1} \gamma^t \mu_{a^*_{1:t}}}^{\text{act optimally}}$$

Under the hood

Upper-bounding the value of sequences

$$V(a) = \overbrace{\sum_{t=1}^{h} \gamma^t \underbrace{\mu_{a_{1:t}}}_{\leq U^\mu}}^{\text{follow the sequence}} + \overbrace{\sum_{t \geq h+1} \gamma^t \underbrace{\mu_{a^*_{1:t}}}_{\leq 1}}^{\text{act optimally}}$$

Under the hood

OLOP main tool: the Chernoff-Hoeffding deviation inequality

$$\underbrace{U_a^{\mu}(m)}_{\text{Upper bound}} \stackrel{\text{def}}{=} \underbrace{\hat{\mu}_a(m)}_{\text{Empirical mean}} + \underbrace{\sqrt{\frac{2\log M}{T_a(m)}}}_{\text{Confidence interval}}$$

Under the hood

OLOP main tool: the Chernoff-Hoeffding deviation inequality

$$\underbrace{U_a^\mu(m)}_{\text{Upper bound}} \stackrel{\text{def}}{=} \underbrace{\hat{\mu}_a(m)}_{\text{Empirical mean}} + \underbrace{\sqrt{\frac{2\log M}{T_a(m)}}}_{\text{Confidence interval}}$$

OPD: upper-bound all the future rewards by 1

$$U_a(m) \stackrel{\text{def}}{=} \sum_{t=1}^{h} \underbrace{\gamma^t U_{a_{1:t}}^\mu(m)}_{\text{Past rewards}} + \underbrace{\frac{\gamma^{h+1}}{1-\gamma}}_{\text{Future rewards}}$$

Under the hood

`OLOP` main tool: the Chernoff-Hoeffding deviation inequality

$$\underbrace{U_a^\mu(m)}_{\text{Upper bound}} \stackrel{\text{def}}{=} \underbrace{\hat{\mu}_a(m)}_{\text{Empirical mean}} + \underbrace{\sqrt{\frac{2\log M}{T_a(m)}}}_{\text{Confidence interval}}$$

`OPD`: upper-bound all the future rewards by 1

$$U_a(m) \stackrel{\text{def}}{=} \sum_{t=1}^{h} \underbrace{\gamma^t U^\mu_{a_{1:t}}(m)}_{\text{Past rewards}} + \underbrace{\frac{\gamma^{h+1}}{1-\gamma}}_{\text{Future rewards}}$$

Bounds sharpening

$$B_a(m) \stackrel{\text{def}}{=} \inf_{1\leq t\leq L} U_{a_{1:t}}(m)$$

OLOP guarantees

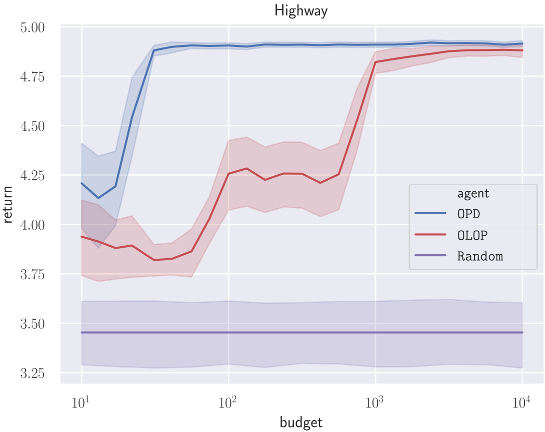
Theorem (OLOP Sample complexity)

OLOP satisfies:

$$\mathbb{E}\, r_n = \begin{cases} \widetilde{\mathcal{O}}\left(n^{-\frac{\log 1/\gamma}{\log \kappa'}}\right), & \textit{if } \gamma\sqrt{\kappa'} > 1 \\ \widetilde{\mathcal{O}}\left(n^{-\frac{1}{2}}\right), & \textit{if } \gamma\sqrt{\kappa'} \leq 1 \end{cases}$$

"Remarkably, in the case $\kappa\gamma^2 > 1$, we obtain the same rate for the simple regret as Hren and Munos (2008). Thus, in this case, we can say that planning in stochastic environments is not harder than planning in deterministic environments".

Does it work?



Our objective: understand and bridge this gap.

Make OLOP *practical*.

What's wrong with OLOP?

Explanation: inconsistency

- Unintended behaviour happens when $U_a^\mu(m) > 1, \forall a$.

$$U_a^\mu(m) = \underbrace{\hat{\mu}_a(m)}_{\in[0,1]} + \underbrace{\sqrt{\frac{2 \log M}{T_a(m)}}}_{>0}$$

What's wrong with OLOP?

Explanation: inconsistency

- Unintended behaviour happens when $U^\mu_a(m) > 1, \forall a$.

$$U^\mu_a(m) = \underbrace{\hat{\mu}_a(m)}_{\in[0,1]} + \underbrace{\sqrt{\frac{2\log M}{T_a(m)}}}_{>0}$$

- Then the sequence $(U_{a_{1:t}}(m))_t$ is increasing

$$U_{a_{1:1}}(m) = \gamma U^\mu_{a_1}(m) + \gamma^2 1 \qquad + \gamma^3 1 + \dots$$
$$U_{a_{1:2}}(m) = \gamma U^\mu_{a_1}(m) + \gamma^2 \underbrace{U^\mu_{a_2}}_{>1} \qquad + \gamma^3 1 + \dots$$

What's wrong with OLOP?

Explanation: inconsistency

- Unintended behaviour happens when $U_a^\mu(m) > 1, \forall a$.

$$U_a^\mu(m) = \underbrace{\hat{\mu}_a(m)}_{\in[0,1]} + \underbrace{\sqrt{\frac{2\log M}{T_a(m)}}}_{>0}$$

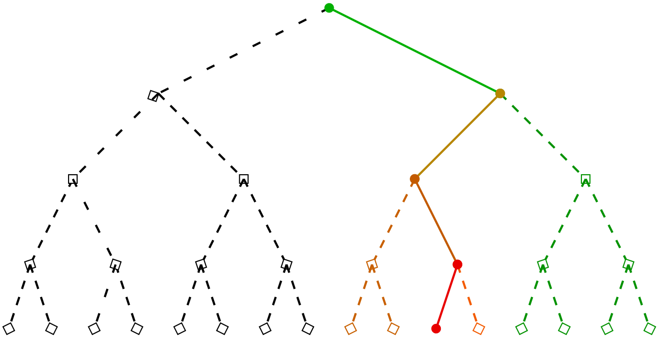
- Then the sequence $(U_{a_{1:t}}(m))_t$ is increasing

$$U_{a_{1:1}}(m) = \gamma U_{a_1}^\mu(m) + \gamma^2 1 \qquad +\gamma^3 1 + \dots$$
$$U_{a_{1:2}}(m) = \gamma U_{a_1}^\mu(m) + \gamma^2 \underbrace{U_{a_2}^\mu}_{>1} \qquad +\gamma^3 1 + \dots$$

- Then $B_a(m) = U_{a_{1:1}}(m)$

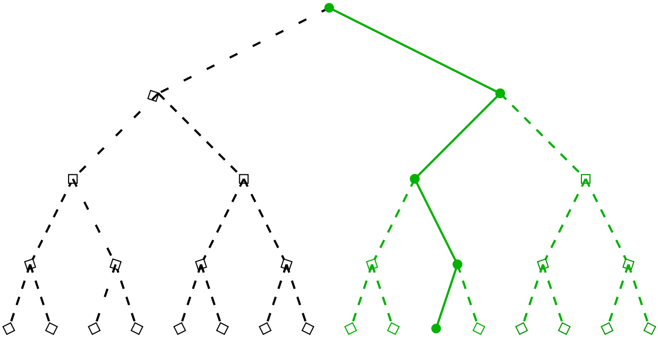
What's wrong with OLOP?

What we were promised



What's wrong with OLOP?

What we actually get



OLOP behaves as uniform planning!

Our contribution: Kullback-Leibler OLOP

We summon the upper-confidence bound from kl-UCB [Cappé et al. 2013]:

$$U_a^{\mu}(m) \stackrel{\text{def}}{=} \max\{q \in I : T_a(m) d(\hat{\mu}_a(m), q) \leq f(m)\}$$

Our contribution: Kullback-Leibler `OLOP`

We summon the upper-confidence bound from `kl-UCB` [Cappé et al. 2013]:

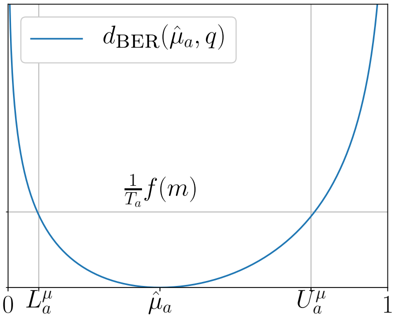
$$U_a^{\mu}(m) \overset{\text{def}}{=} \max\left\{q \in I : T_a(m) d(\hat{\mu}_a(m), q) \leq f(m)\right\}$$

Algorithm	`OLOP`	`KL-OLOP`
Interval I	$\mathbb{R}$	$[0, 1]$
Divergence d	$d_{\texttt{QUAD}}$	$d_{\texttt{BER}}$
$f(m)$	$4 \log M$	$2 \log M + 2 \log \log M$

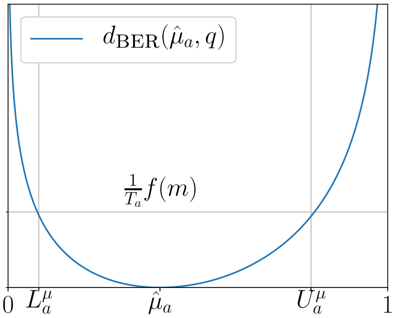
$$d_{\texttt{QUAD}}(p, q) \overset{\text{def}}{=} 2(p-q)^2$$

$$d_{\texttt{BER}}(p, q) \overset{\text{def}}{=} p \log \frac{p}{q} + (1-p) \log \frac{1-p}{1-q}$$

Our contribution: Kullback-Leibler OLOP



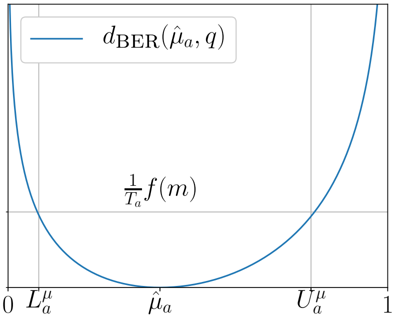
Our contribution: Kullback-Leibler OLOP



And now,

- $U_a^\mu(m) \in I = [0, 1], \forall a$.

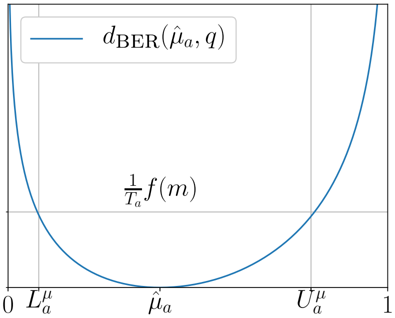
Our contribution: Kullback-Leibler OLOP



And now,

- $U_a^\mu(m) \in I = [0, 1], \forall a$.
- The sequence $(U_{a_{1:t}}(m))_t$ is non-increasing

Our contribution: Kullback-Leibler OLOP



And now,

- $U_a^\mu(m) \in I = [0, 1], \forall a$.
- The sequence $(U_{a_{1:t}}(m))_t$ is non-increasing
- $B_a(m) = U_a(m)$, the *bound sharpening* step is superfluous.

Sample complexity

Theorem (Sample complexity)

KL-OLOP enjoys the same regret bounds as OLOP. More precisely, KL-OLOP satisfies:

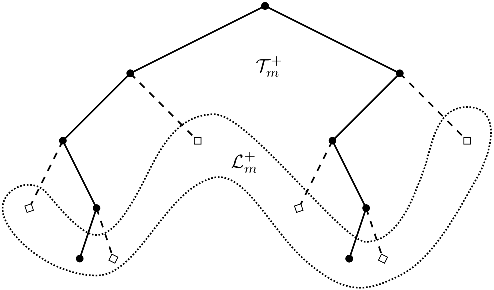
$$\mathbb{E}\, r_n = \begin{cases} \widetilde{\mathcal{O}}\left(n^{-\frac{\log 1/\gamma}{\log \kappa'}}\right), & \textit{if } \gamma\sqrt{\kappa'} > 1 \\ \widetilde{\mathcal{O}}\left(n^{-\frac{1}{2}}\right), & \textit{if } \gamma\sqrt{\kappa'} \leq 1 \end{cases}$$

Time complexity

Original KL-OLOP

Compute $B_a(m-1)$ *from* (14) *for all* $a \in A^L$

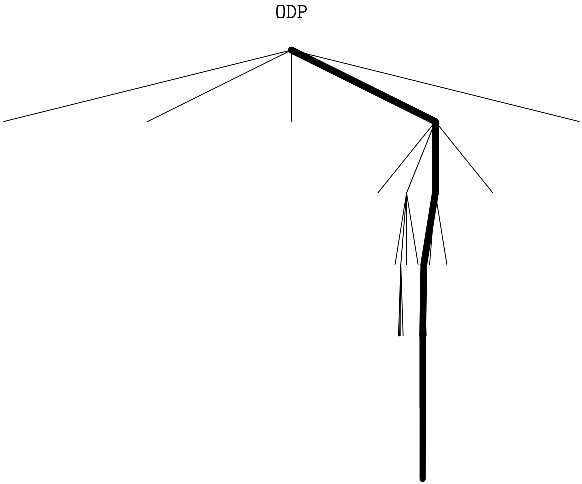
Lazy KL-OLOP



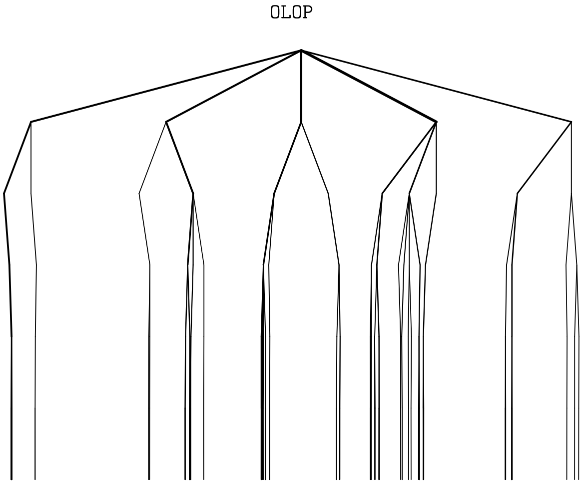
Property (Time and memory complexity)

$$\frac{C(\texttt{Lazy KL-OLOP})}{C(\texttt{KL-OLOP})} = \frac{nK}{K^L}$$

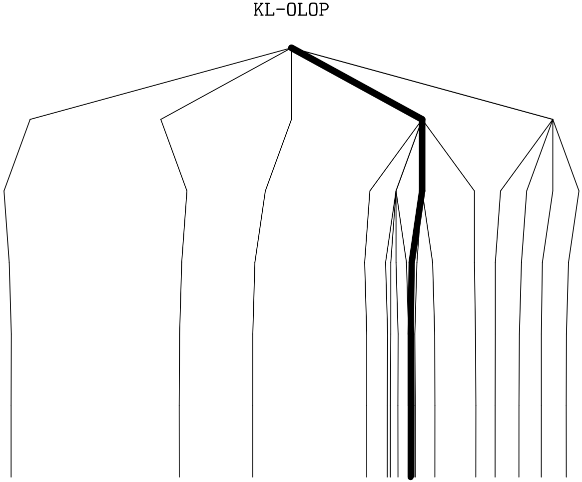
Experiments — Expanded Trees



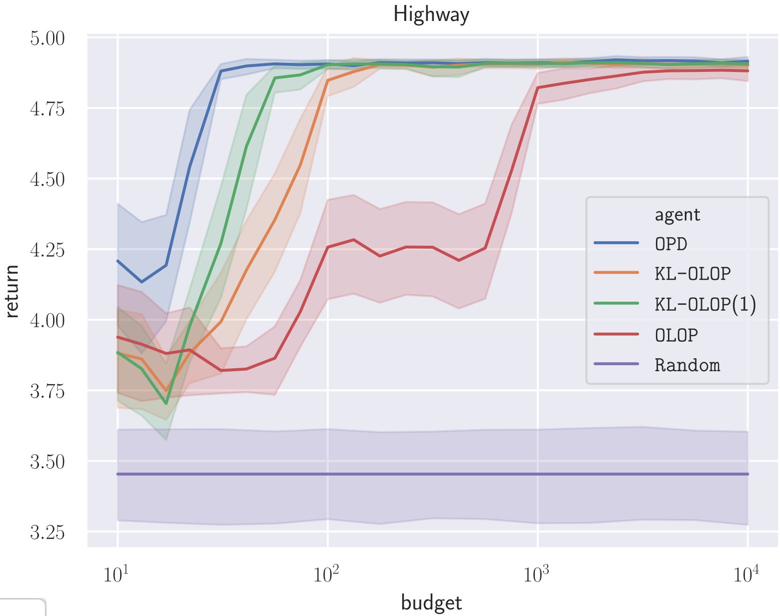
Experiments — Expanded Trees



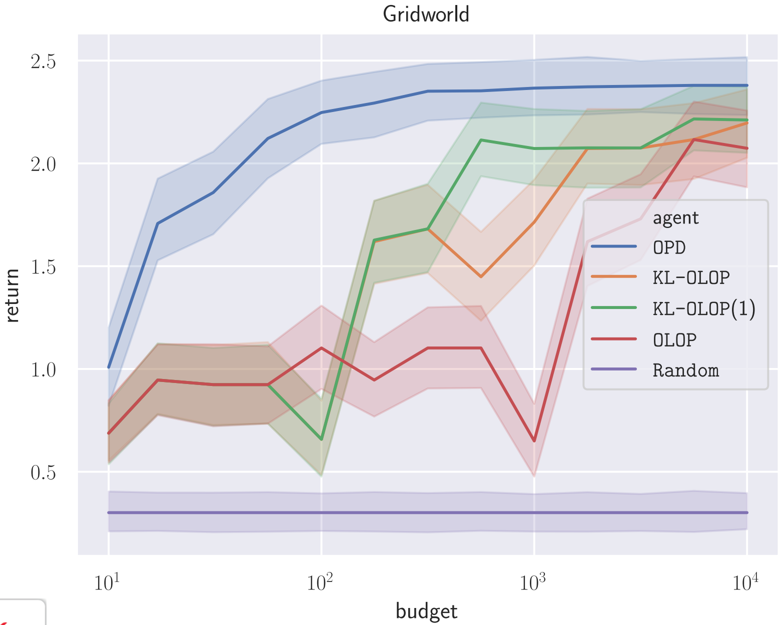
Experiments — Expanded Trees



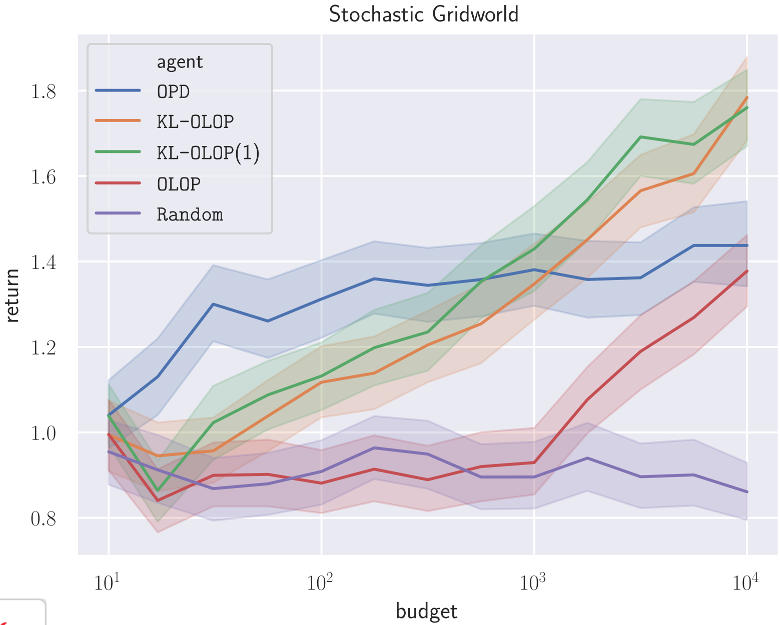
Experiments — Highway



Experiments — Gridworld



Experiments — Stochastic Gridworld



References

Sébastien Bubeck and Rémi Munos. "Open Loop Optimistic Planning". In: *Proc. of COLT*. 2010.

Olivier Cappé, Aurélien Garivier, Odalric-Ambrym Maillard, Rémi Munos, and Gilles Stoltz. "Kullback-Leibler Upper Confidence Bounds for Optimal Sequential Allocation". In: *The Annals of Statistics* 41.3 (2013), pp. 1516–1541.

Pierre-Arnaud Coquelin and Rémi Munos. "Bandit Algorithms for Tree Search". In: *Proc. of UAI* (2007).

Rémi Coulom. "Efficient Selectivity and Backup Operators in Monte-Carlo Tree Search". In: *Proc. of International Conference on Computer and Games*. 2006.

Jean François Hren and Rémi Munos. "Optimistic planning of deterministic systems". In: *Lecture Notes in Computer Science* (2008).

Levente Kocsis and Csaba Szepesvári. "Bandit Based Monte-carlo Planning". In: *Proc. of ECML PKDD*. 2006.

Thank You.