Efficient Bayes-Adaptive Reinforcement Learning using Sample-Based Search

Arthur Guez*, **David Silver**†, **and Peter Dayan***

* Gatsby Computational Neuroscience Unit, UCL † Dept. of Computer Science, UCL

Introduction

We introduce a tractable, sample-based method for approximate Bayes-optimal planning which exploits Monte-Carlo tree search. Our approach avoids expensive applications of Bayes rule within the search tree by lazily sampling models from the current beliefs at the root. It outperfoms existing approaches on standard benchmark problems and it can deal with large state spaces with structured priors.

Reminder: Model-based Bayesian Exploration

- Typical MDP description $M = \langle S, A, \mathcal{P}, \mathcal{R}, \gamma \rangle$, but here $\mathcal{P}$ is a latent variable distributed according to a prior $P(\mathcal{P})$.
- **Goal:** Find exploration policy $\pi : S \times \mathcal{H} \to A$ that maximizes $\int_{\mathcal{P}} P(\mathcal{P}) \mathbb{E}^{\pi}_{M(\mathcal{P})}[\sum_{t=0}^{\infty} \gamma^t r_t | s_0 = s, h_0 = s]$; the resulting policy trades off exploration and exploitation. ($\mathcal{H} \equiv$ Set of all possible histories)
- Equivalent to solving augmented MDP M^+ in belief space: **Bayes-Adaptive MDP (BAMDP)** where $\mathcal{P}^+(\langle s,h \rangle, a, \langle s', h' \rangle) = \mathbf{1}_{h'=has'} \int_{\mathcal{P}} \mathcal{P}(s,a,s') P(\mathcal{P}|h)$.

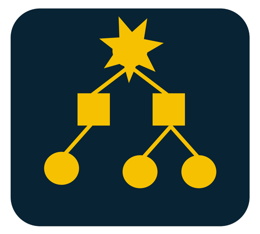
Major obstacle: Computationally intractable to solve exactly even for tiny state spaces.

Our approach

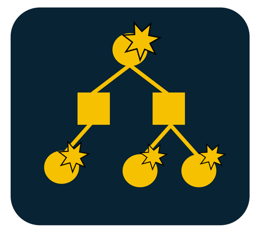
Problem Formulation: We want to find a tractable approximation to BAMDP's optimal policy compatible with a *large class of priors.*

Proposed solutions:

- **BA-UCT**
  - Tackle the BAMDP, a particular MDP, with Monte-Carlo Tree Search/UCT.
  - Solves BAMDP online approximately for the current state; UCT focuses search effort where it matters; converges to Bayes-optimal policy.
  - **Issue:** Expensive belief updates at every tree node, not practical for most priors.



- **BA-UCT + Root sampling:**
  - Restrict posterior sampling to the root node (as in Silver's & Veness' POMCP alg.).
  - Only need to perform 1 belief update and generate posterior samples at tree root.
  - **Issue:** Generating full samples $\mathcal{P}$ not feasible in large MDPs.



- **BA-UCT + Root sampling + Lazy sampling:**
  - Use factorization of the posterior to minimize sampling for each simulation.

BA-UCT + Root Sampling + Lazy Sampling + Rollout Learning ≡ **BAMCP** algorithm (**B**ayes-**A**daptive **M**onte-**C**arlo **P**lanning).

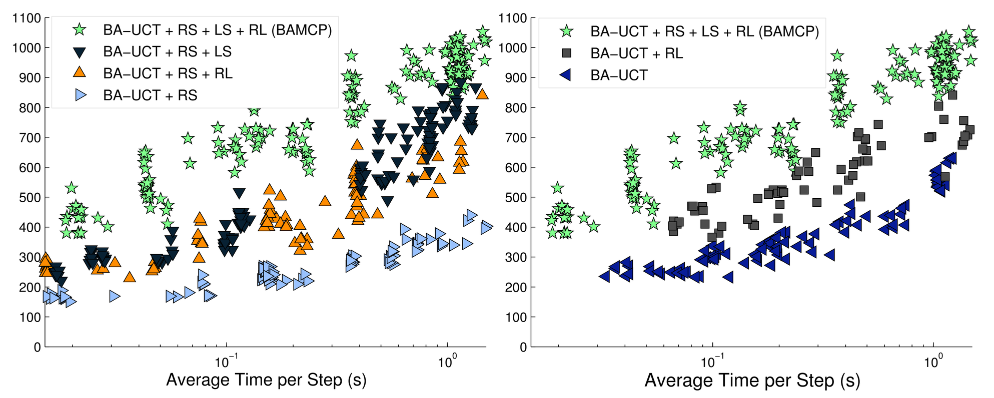
Theoretical Properties

- BAMCP converges to the Bayes-optimal policy.
  $V(\langle s_t, h_t \rangle) \xrightarrow{p} V^*_\epsilon(\langle s_t, h_t \rangle)$
- Rate of convergence at the nodes as in UCT.
  Bias decreases as $\log(N(\langle s,h \rangle))/N(\langle s,h \rangle)$.

Why can we get away with root sampling (Silver & Veness 2010)? Compare distribution of $\mathcal{P}$ at the tree nodes using BA-UCT (posterior) versus using BAMCP ($\tilde{P}$), assume equivalent up to node h, then:

$$P(\mathcal{P}|has') \propto P(\mathcal{P}|h)\mathcal{P}(s,a,s') = \tilde{P}_h(\mathcal{P})\mathcal{P}(s,a,s') = \tilde{P}_{ha}(\mathcal{P})\mathcal{P}(s,a,s') \propto \tilde{P}_{has'}(\mathcal{P})$$

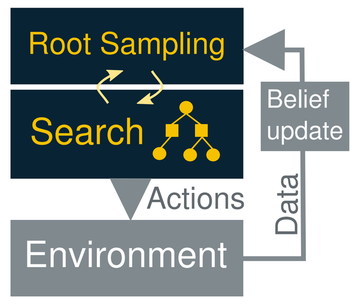
Example on Dearden's Maze



Sum of rewards after 20K steps. RS=Root Sampling, LS=Lazy Sampling, RL=Rollout Learning.

Even in small state spaces (264 states) and relatively simple prior (Sparse Dir-Mult), BAMCP benefits from root sampling, lazy sampling, and rollout learning.

BAMCP



- *Sampler* provides (lazy) posterior MDP samples $P(\mathcal{P}|h)$ for current history h.
- *Search* treats sampler as black box, uses MDP samples to run UCT.

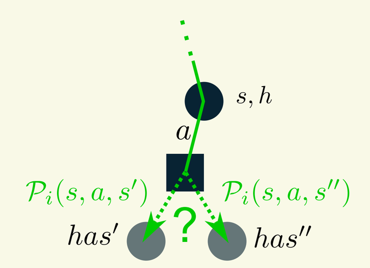
BAMCP Algorithm

Initialize empty search tree
Repeat:
- Root sampling of posterior dynamics
- Sample tree trajectory until leaf is reached

see below

- Run rollout policy and extend tree at leaf node
- Update value at each traversed tree node using MC Backup
- Update visit count at each traversed node

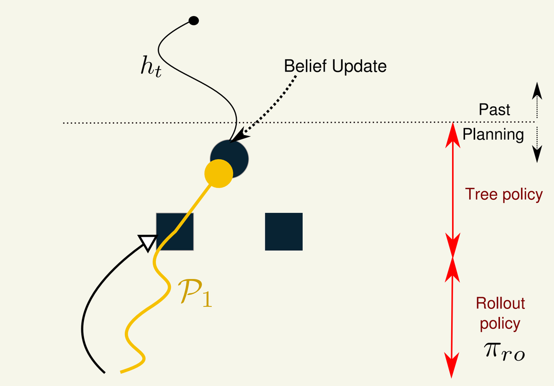
How are tree trajectories generated?



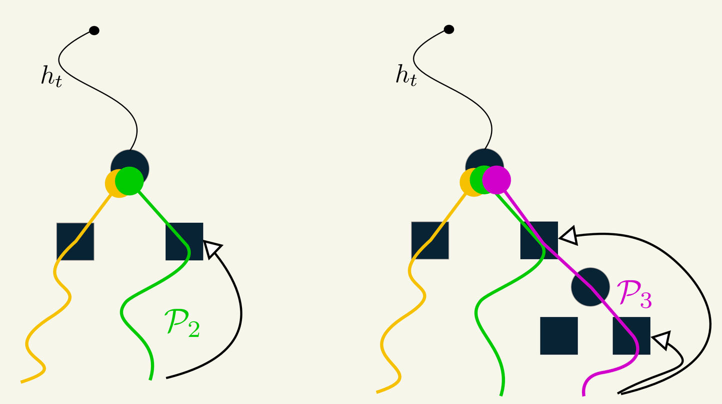
- Time-step t, i-th simulation
- $\mathcal{P}_i \sim P(\mathcal{P}|h_t)$ (lazily)
- Generate succ. states according to $\mathcal{P}_i(s,a,\cdot)$
- UCT selects actions (independently of $\mathcal{P}_i$!)

Example BAMCP run

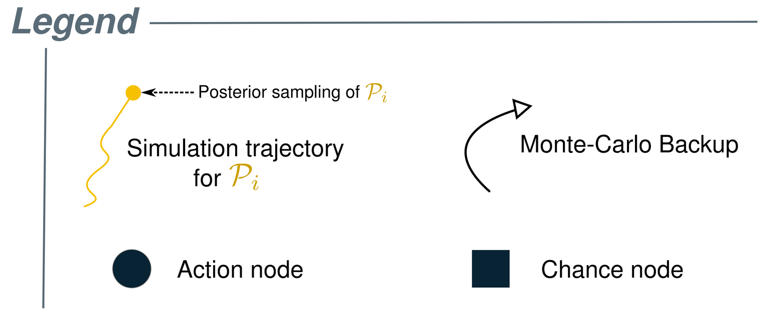
1st simulation (UCT sim. with $\mathcal{P}_1$)



2nd simulation 3rd simulation



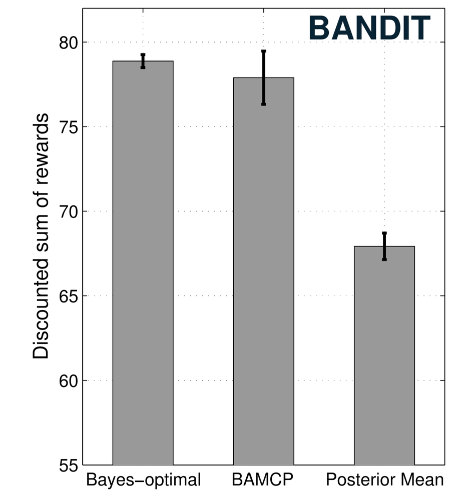
Legend

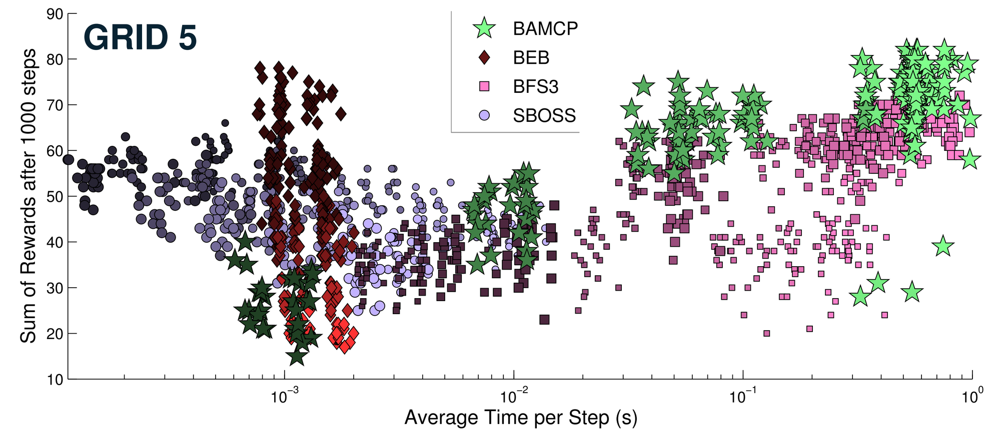


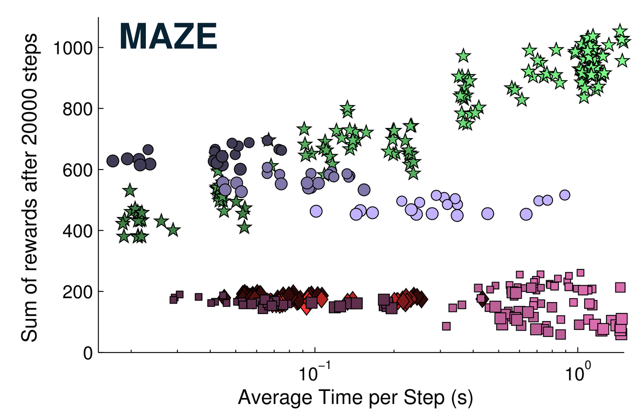
Results: Standard Domains

	Double-loop	Grid5	Grid10	Dearden Maze
BAMCP	**387.6 ± 1.5**	**72.9 ± 3**	**32.7 ± 3**	**965.2 ± 73**
BFS3	382.2 ± 1.5	66 ± 5	10.4 ± 2	240.9 ± 46
SBOSS	371.5 ± 3	59.3 ± 4	21.8 ± 2	671.3 ± 126
BEB	386 ± 0	67.5 ± 3	10 ± 1	184.6 ± 35
Bayesian DP*	377 ± 1	-	-	-
Bayes VPI+MIX*	326 ± 31	-	-	817.6 ± 29
IEQL+*	264 ± 1	-	-	269.4 ± 1
QL Boltzmann*	186 ± 1	-	-	195.2 ± 20

Table : Cumulative sum of reward (Double-loop,Grid5: after 1K steps, Grid10: after 2K steps, Maze 264: after 20K steps), $\gamma = 0.95$. (*) Reported results from Strens 2000.







Infinite 2D Grid Task

- Infinite combinatorial state space,
- Correlated reward locations:
  - Latent ith column parameters: $p_i \sim \text{Beta}(\alpha_1, \beta_1)$
  - Latent jth row parameters: $q_j \sim \text{Beta}(\alpha_2, \beta_2)$
  - Pr(Reward(grid cell ij) = 1) = $p_i q_j$

  Rewards can only be consumed once.
- Posterior inference needs approx. (Metropolis-Hastings).

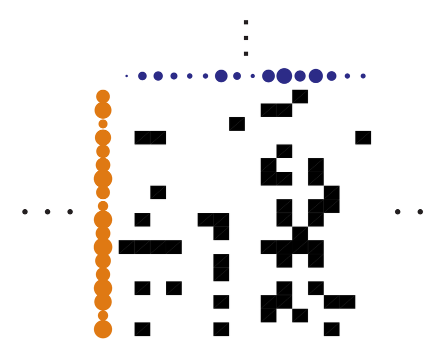


Figure : Example section of a grid with $\alpha_1 = 1, \beta_1 = 2, \alpha_2 = 2, \beta_2 = 1$. Circles represent the p_i and q_j parameters.

Intractable task for existing methods (huge state space + expensive belief updates). With BAMCP:
- *Root sampling* avoids expensive MCMC at every tree node,
- *Lazy sampling* only samples small finite set of parameters for each sim,
- *Forward-search/UCT* can deal with large state space.

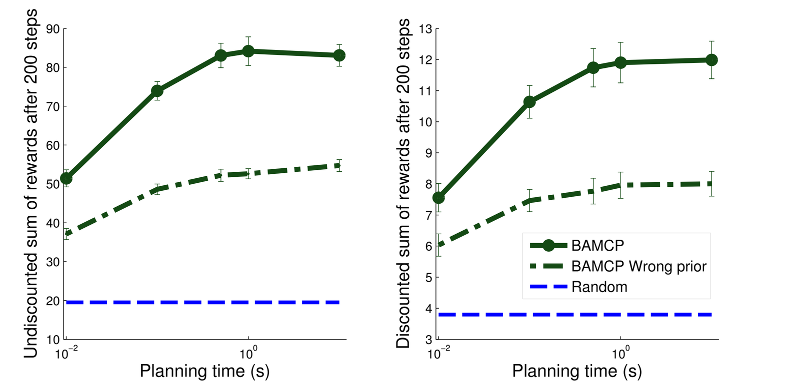


Figure : Performance for the first 200 steps in the environment, averaged over 50 sampled environments ($\gamma = 0.97$). In this example, grids are generated with Beta parameters $\alpha_1 = 1, \beta_1 = 2, \alpha_2 = 2, \beta_2 = 1$.

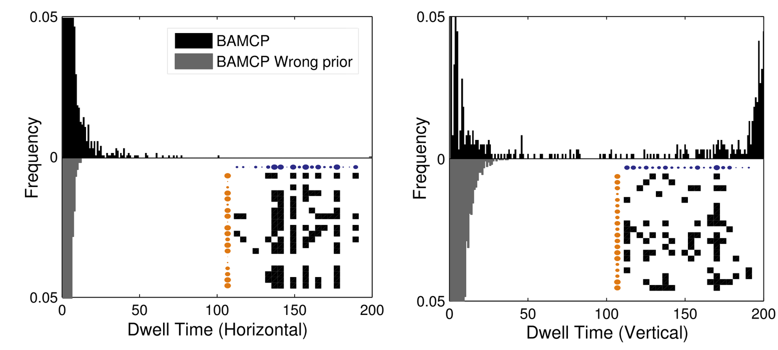


Figure : For two different prior scenarios, the distribution of behavior in terms of row or column dwell times across many trials.

Summary

- Introduce tractable sample-based algorithm for Bayesian RL,
- State-of-the-art performance results on standard domains,
- Scales to large tasks,
- Can exploit structured priors.

This work was funded by the Gatsby Charitable Foundation and the Natural Sciences and Engineering Research Council of Canada.

Contact: aguez@gatsby.ucl.ac.uk