Mission Planning in Unknown Environments as Bayesian Reinforcement Learning

Matthew Budd , **Paul Duckworth** , **Nick Hawes** and **Bruno Lacerda**
Dept. of Engineering Science, University of Oxford
{mbudd, pduckworth, nickh, bruno}@robots.ox.ac.uk

Abstract

To operate reliably in challenging real-world environments, a robot should consider that it has an incomplete model of its environment and plan to take measurements to improve this. However, unmodelled disturbances, sensor noise and the limitation of measurements to the robot's current location make this difficult. We propose a Bayesian reinforcement learning-based modelling and planning framework which uses Gaussian processes to model environmental uncertainty in a principled manner. We exploit the Bayesian RL formulation to plan more efficiently in these types of uncertain environments than previous methods are able to.

1 Introduction

When acting in uncertain environments, *a priori* unknown environmental dynamics may affect a robot by imposing unexpected costs or changing the robot's action outcome probabilities. Examples of such scenarios include unknown water currents acting on an autonomous underwater vehicle, or an unknown distribution of radiation that may harm the robot if cumulative exposure exceeds a given level. Enabling robots to operate in areas where they have incomplete information requires them to be able to take measurements (or *observations*) of unknown environment features and incorporate these into their plans.

A robot is often only able to take noisy, local measurements of unknown environment features at its current state – for example by measuring the radiation level at its current position. We follow previous works [Budd *et al.*, 2020; Turchetta *et al.*, 2016] and make use of a Gaussian process (GP) model [Rasmussen and Williams, 2006] to predict the unknown environment dynamics away from its local position. This modelling framework is commonly used to provide regression with confidence intervals to fit noisy spatio-temporal data distributions.

When it is not feasible to pre-plan for every possible environment, online planning and modelling (where observations, planning and execution are interleaved) is a suitable approach. We would like to enable the robot to plan action/observation sequences in a principled manner, which requires the robot to maintain a belief over the real underlying dynamics of the environment.

Previous literature [Morere *et al.*, 2017; Flaspohler *et al.*, 2019] often approaches the problem using a partially observable MDP (POMDP) formulation, where the entirety of the environment dynamics are incorporated into the partially observable state. However, this is not the most intuitive way to represent the problem – a POMDP model generally implies that the true underlying state (visible via probabilistic observations) can change, whereas in our case the environment is understood to be fixed but *a priori* unknown. Instead we discuss how exploration of a GP-modelled environment can conveniently be framed as a Bayesian reinforcement learning problem with a GP belief over the transition function. This allows us to plan more efficiently using techniques developed in that context, such as BAMCP [Guez *et al.*, 2013] – a Monte Carlo tree search (MCTS)-based algorithm. Such sampling-based methods are able to plan effectively in very large, continuous state spaces by only searching the reachable state space from the current state.

In summary, our contribution is to pose the online planning problem as a Bayes-adaptive MDP (BAMDP) and make use of the root sampling method originally proposed in POMCP [Silver and Veness, 2010] to improve search scalability in this model by allowing us to avoid expensive GP belief updates inside the MCTS search tree. For each MCTS trial, a possible environment distribution is drawn as a sample from the belief at the tree root node and used to generate the necessary samples during the MCTS trial. We argue that the BAMDP formulation better illustrates that the robot has local observability of the state of the environment at its current location and allows us to pose the robot transition function over unknown environment features in a clearer way, compared to the common POMDP problem formulation. We also show that both formulations lead to MCTS planning in similarly structured search trees.

2 Related Work

Online planning in environments modelled with GPs has been approached in previous literature using Markov decision process (MDP) models and partially observable MDP (POMDP) [Kaelbling *et al.*, 1998] models.

Turchetta *et al.* [2016] use GPs to model a safety function over an environment with a deterministic transition MDP, but

only use this model to restrict the robot to some parts of the state space where the GP confidence intervals do not cross a safety bound value. Wachi *et al*. [2018] also combine deterministic transition MDPs with GPs, using separate GPs to model both a safety function and reward function. They also define an information gain criterion to encourage the robot to visit and observe areas of high uncertainty.

In previous work [Budd *et al.*, 2020] we built on top of the approach of Turchetta *et al*. to pose the unknown environment as an MDP with unknown feature values (U-MDP). In this formalism, the dynamics of *known value* state features are known *a priori* and the dynamics of *unknown value* state features are determined by the predictions of the environment model. This MDP model is then solved online to plan cost-optimal paths to goal locations to observe, with the overall aim of improving the environment model accuracy. The approach is able to handle transition structures with probabilistic outcomes that may depend on environment feature values. However, all of these MDP-based approaches are myopic as they greedily select the next state to explore in a "next-best-view" fashion rather than planning informative sequences of observations.

Other recent literature [Morere *et al.*, 2017; Flaspohler *et al.*, 2019] carries out "informative path planning" across unknown environments, also using a GP model. The goal for both is to carry out Bayesian optimisation to find the maximum of an unknown function, modelled as a GP, which is combined with the robot pose to form the state of a partially observable MDP (POMDP). In general, solving POMDPs exactly is infeasible for all but the smallest problems, due to planning taking place in a continuous belief space and the exponential growth of the number of possible action/observation histories as the planning horizon increases. Both of the solution methods used in these papers are therefore based on MCTS. However, these approaches assume robot actions have deterministic outcomes, and do not consider the case where the unknown environment features can affect robot transition dynamics. The MCTS algorithms proposed in these papers also carry out expensive belief updates within the MCTS search tree.

Our BAMDP model could also be considered to be equivalent to a POMDP-lite [Chen *et al.*, 2016], with the true environment represented as a constant state variable. However, the solution method of Chen *et al*. encourages exploration by including an exploration bonus in the reward function, which can be difficult to tune [Guez *et al.*, 2013].

MCTS planning in time-varying unknown environments using GPs has also been carried out by Duckworth *et al*. [2021], for the case where the environment features affect only transition durations in a semi-MDP. This planning method makes use of the GP mean and variance predictions directly, rather than sampling from the GP posterior as we do here. Although the approach avoids some of the computational expense of in-tree GP belief updates, they do not carry out MCTS root sampling to improve search efficiency as we do.

3 Preliminaries

3.1 Markov Decision Process

A Markov decision process (MDP) is defined as a tuple $\mathcal{M} = \langle S, \overline{s}, A, T, C \rangle$, where S is a finite set of states; $\overline{s} \in S$ is the initial state; A is a finite set of actions; $T : S \times A \times S \to [0, 1]$ is a probabilistic transition function; and $C : S \times A \to \mathbb{R}_{\geq 0}$ is a cost function. A (stationary) policy is a mapping $\pi : S \times A \to [0, 1]$ that defines the probability of choosing a given action in a given state.

A stochastic shortest path (SSP) MDP also includes a set of absorbing, zero-cost goal states $G \subset S$. In an SSP MDP there must exist a policy that is *proper* in all states. A policy is proper in a state s if it reaches a state $s_g \in G$ when starting from s with probability 1, and an improper policy at a state will always incur infinite cost. Although our definition of a policy is stochastic, it is known that there exists a cost-optimal deterministic policy for an SSP.

3.2 Bayesian Reinforcement Learning: Bayes-Adaptive MDP

In a Bayes-adaptive MDP (BAMDP) [Duff, 2003] the transition function T and (optionally) the cost function C are unknown – the agent only has access to a prior probability distribution over their dynamics. To minimise its expected cumulative cost, the agent must maintain a *belief* about the actual dynamics, in the form of probability distributions over T and C.

A *history* in a BAMDP is a sequence of actions and states $h_t = h_0 a_1 s_2 a_2 ... a_{t-1} s_t$ observed during execution. A BAMDP manages uncertainty in T and C by planning in an augmented state space $S^+ = S \times H$ where H is the set of possible histories. The history is a sufficient statistic for the belief, as it is possible to transform any history starting from h_0 and its equivalent belief b_0 using successive applications of Bayes' rule: $p(\{T, C\} \mid h_t) \propto p(h_t \mid \{T, C\}) p(\{T, C\})$. We only consider the case where the transition dynamics of the BAMDP are unknown.

A BAMDP is a tuple $\mathcal{M}^+ = \langle S^+, \overline{s}^+, A, T^+, C^+ \rangle$, where $C^+((s, h), a) = C(s, a)$, $\overline{s}^+ = (\overline{s}, h_0)$ and

$$T^+((s, h), a, (s', has')) = \int_T T(s, a, s') p(T \mid h) dT. \quad (1)$$

A policy in a BAMDP is a history dependent probabilistic mapping from histories to actions: $\pi : H \times A \to [0, 1]$. The optimal policy π^* is that which minimises expected cumulative cost (or equivalently maximises cumulative expected reward) given the prior over T, up to some finite or indefinite horizon.

3.3 Gaussian Process

A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution [Rasmussen and Williams, 2006]. A GP regression is of the form $f(s) \sim \mathcal{GP}(m(s), k(s, s'))$. This represents a probability distribution over functions, fully specified by the mean function $m(s)$ and kernel function $k(s, s')$. We can let $m(s) = 0$ without loss of generality.

Given a dataset of n noisy observations $\mathbf{o} = \{(o(s_i) + \epsilon_i), i = 1, \ldots, n\}$ at states $\mathbf{s}_n$, GP regression can be used to predict values of the unknown environment features at all states $\mathbf{s}_*$, where $\epsilon_i \sim \mathcal{N}(0, \sigma_n^2)$ is Gaussian observation noise. The resulting Gaussian posterior, conditioning on the observations, is a multivariate normal $\mathcal{N}(\boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*)$:

$$\boldsymbol{\mu}_* = \mathbf{K}_*^\mathsf{T}(\mathbf{K}_n + \sigma_n^2\mathbf{I})^{-1}\mathbf{o}, \tag{2}$$

$$\boldsymbol{\Sigma}_* = \mathbf{K}_{**} - \mathbf{K}_*^\mathsf{T}(\mathbf{K}_n + \sigma_n^2\mathbf{I})^{-1}\mathbf{K}_*, \tag{3}$$

where the positive semi-definite kernel matrix $\mathbf{K}_n = [k(s, s')]_{s,s' \in \mathbf{s}_n}$, $\mathbf{K}_* = [k(s, s')]_{s \in \mathbf{s}_n, s' \in \mathbf{s}_*}$, $\mathbf{K}_{**} = [k(s, s')]_{s,s' \in \mathbf{s}_*}$, and $\mathbf{I} \in \mathbb{R}^{n \times n}$ is the identity matrix.

The kernel function k may include hyperparameters including variance and lengthscale, which we optimise using maximum likelihood estimation (MLE) to fit the dataset. Regularity and Lipschitz continuity modelling assumptions must be made to make predictions of unknown environment features with the GP [Rasmussen and Williams, 2006].

We can sample functions from the posterior of a GP, at a finite set of points $\mathbf{s}_*$, by transformation of a standard normal distribution $\boldsymbol{\phi} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$:

$$\mathbf{o}_* = \boldsymbol{\mu}_* + \boldsymbol{\Sigma}_*^{1/2}\boldsymbol{\phi}. \tag{4}$$

The matrix inverse and square root operations above, required for carrying out the GP regression and generating samples from the GP posterior, can be carried out with Cholesky decomposition giving $\sim O(n^3)$ complexity.

4 Approach

4.1 Problem Formulation

In order to clearly separate robot transition dynamics from the unknown environment dynamics, we represent the unknown environment and its effect on the robot as an *MDP with Unknown Feature Values (U-MDP)*. As originally described in [Budd *et al.*, 2020], a U-MDP is a tuple $\mathcal{M}^o = \langle S^o, \overline{s}, A, C, T^o \rangle$, where:

- The state space is factored, $S^o = S_k \times S_e$ where $S_k = S_k^1 \times \ldots \times S_k^{n_k}$ is a set of state features with discrete, known values (e.g. the pose of the robot), and $S_e = S_e^1 \times \ldots \times S_e^{n_e}$ is a set of state features with unknown values (e.g. the environment state at a pose);
- An *a priori* unknown mapping $o : S_k \to S_e$ defines which values $o(s_k) \in S_e$ are observed at $s_k \in S_k$;
- $\overline{s}$ is the initial state $\overline{s} = \big(\overline{s}_k, o(\overline{s}_k)\big)$;
- A is a finite set of actions;
- $C : S \times A \to \mathbb{R}$ is the cost function; and
- T^o is the U-MDP transition structure $T^o : (S_k \times S_e) \times A \times S_k \to [0, 1]$. As the state of the U-MDP is uniquely defined by the value of the known state feature $s_k \in S_k$, the transition structure of the U-MDP only represents the change in the known state feature.

We define a set of goal states G which are absorbing and incur zero cost, making this a stochastic shortest path problem and making the U-MDP a variant of an SSP MDP. As the value of the mapping o is not known at a goal state, and the goal state should be reachable irrespective of the mapping o, goal states are defined only across the known value state features: $G \subset S_k$.

Our problem statement is therefore to generate a policy π^* that minimises the expected cumulative cost incurred to reach a goal state, given that the robot takes an observation of the mapping function o at each new combination of known value state features that it transitions to.

The SSP MDP formulation can be generalised to objectives other than the undiscounted cost, indefinite horizon problem we describe here [Mausam and Kolobov, 2012].

4.2 U-MDP Environment as a BAMDP

Our first contribution is to pose the task of planning in a U-MDP environment in a Bayesian RL manner by defining it as a BAMDP. The BAMDP root node belief representation is the GP $\mathcal{GP}_{\mathbf{o}_t}$ which maintains the robot's current belief about the unknown mapping o in the U-MDP, given a dataset of observations $\mathbf{o}_t$ up to timestep t. This BAMDP is defined as:

- The state space S^+ is $S^o \times H$, where $S^o = S_k \times S_e$ is the full U-MDP state space and H is the set of possible histories;
- The start state $\overline{s}^+$ is $(\overline{s}^o, h_0)$ where $\overline{s}^o$ is the initial state of the U-MDP and h_0 is the empty initial history;
- The set of actions A and cost structure C are the same as in the U-MDP; and
- T^+ is defined as in (1), where the term inside the integral considers the transition function of the U-MDP and the current GP estimate:

$$\begin{aligned} &T\big((s_k, s_e), a, (s'_k, s'_e)\big)p\big(T \mid h\big) \\ &= T^o\big((s_k, s_e), a, s'_k\big) \cdot p\big(o(s'_k) = s'_e \mid \mathcal{GP}_{\mathbf{o}_t}\big). \end{aligned} \tag{5}$$

As the integral in (1) then only has value for a single case of the transition dynamics:

$$\begin{aligned} &T^+((s, h), a, (s', has')) \\ &= T^o\big((s_k, s_e), a, s'_k\big) \cdot f\big(s'_e \mid s'_k, \mathcal{GP}_{\mathbf{o}_t}\big). \end{aligned} \tag{6}$$

where f is the CDF of the Gaussian distribution.

This transition function represents the combination of i) the known state dynamics from the U-MDP, providing the transition mapping in the discrete known value state space, and ii) the current GP estimate of the dynamics of the unknown value state features in the U-MDP.

When the state includes multiple unknown value state features (i.e. $n_e > 1$), $\mathcal{GP}_{\mathbf{o}_t}$ is a multi-output GP. The multi-output GP could also be factorised into multiple single-output GPs if the unknown value state features are assumed to have independent dynamics.

4.3 BAMCP on the U-MDP

As the GP belief over the environment dynamics is continuous and the space of possible BAMDP histories grows

exponentially with the horizon, attempting to exactly solve this BAMDP does not scale. Having posed the problem as a BAMDP we can make use of MCTS planning frameworks that were developed in this context, specifically BAMCP [Guez *et al.*, 2013]. We also carry out online planning, which is common with MCTS algorithms.

MCTS planners of this type build a search tree consisting of alternating state and action nodes. The search tree is constructed over the course of many Monte Carlo trials starting from the root node and sampling action outcomes. While inside the search tree, actions are chosen using a *tree policy*, most commonly UCT to provide an exploration-exploitation trade-off based on the average cost or reward returns and action counts of child action nodes. Heuristic estimates of leaf nodes' values are provided by continuing the trial trajectory from the leaf node using a *rollout policy*, up to a definite or indefinite horizon. MCTS's performance results from quickly focusing search on promising sections of the state space, without expanding states that are unreachable from the tree root state.

To avoid the notational and algorithmic complexity of continuous variables such as the mapping o, we focus here on a solution approach that discretises continuous environment feature values into a set of ranges to give a BAMDP with discrete state and action spaces. Methods suggested by Sunberg and Kochenderfer [2018] could be used to extend the algorithm to handle continuous state and action spaces – we discuss this further in Section 6.

To plan in BAMDPs, the BAMCP search tree belief nodes contain BAMDP histories. As discussed in Section 3.2, histories from a root belief node in a BAMDP are equivalent to beliefs over the environment dynamics. BAMCP adapts from POMCP [Silver and Veness, 2010] the concept of *root sampling*, where for each MCTS trial an MDP transition function is sampled from the root belief node and used throughout the trial. The equivalent root sampling in POMCP samples a POMDP state from the root belief node.

The validity of root sampling for BAMDPs is demonstrated in Lemma 1 of [Guez *et al.*, 2014], which shows that the rollout distribution function resulting from root sampling is the same as the rollout distribution when maintaining and updating full belief representations at each step. We also provide a proof for the rollout equivalence between our root sampling-based method and previous full belief representation methods in Section 4.4.

In our case, sampling an MDP from the root belief node corresponds to sampling a possible environment that is consistent with our current GP environment model. An environment is sampled from the GP posterior using (4). Specifically, we sample $\hat{o} \sim \mathcal{GP}_{\mathbf{o}_t}$ where $\hat{o}$ is a function $\hat{o} : S_k \rightarrow S_e$ that maps each known state to a possible value of the unknown state features. Figure 1a shows an example of a BAMCP search tree constructed from histories, where each history has been generated from interacting with a root-sampled environment model. A new leaf node can be added by appending the parent node history with a new state sampled using the MDP transition function sampled from the root for this MCTS trial. By comparison, in the POMDP belief search tree (Figure 1b) the belief in a new leaf node is generated from a belief update of its parent node using the observation that created the new leaf node.

Due to the high cost of belief updates, discussed below, the BAMDP search tree can therefore add a new leaf node with much less computational effort than a POMDP belief search tree. The usage of root sampling therefore makes our MCTS planning approach significantly more efficient than the previous methods discussed in the following section.

4.4 Connections to POMDP Formulation

We will now discuss how our approach builds the same MCTS search tree structure as similar literature, but in a more efficient way.

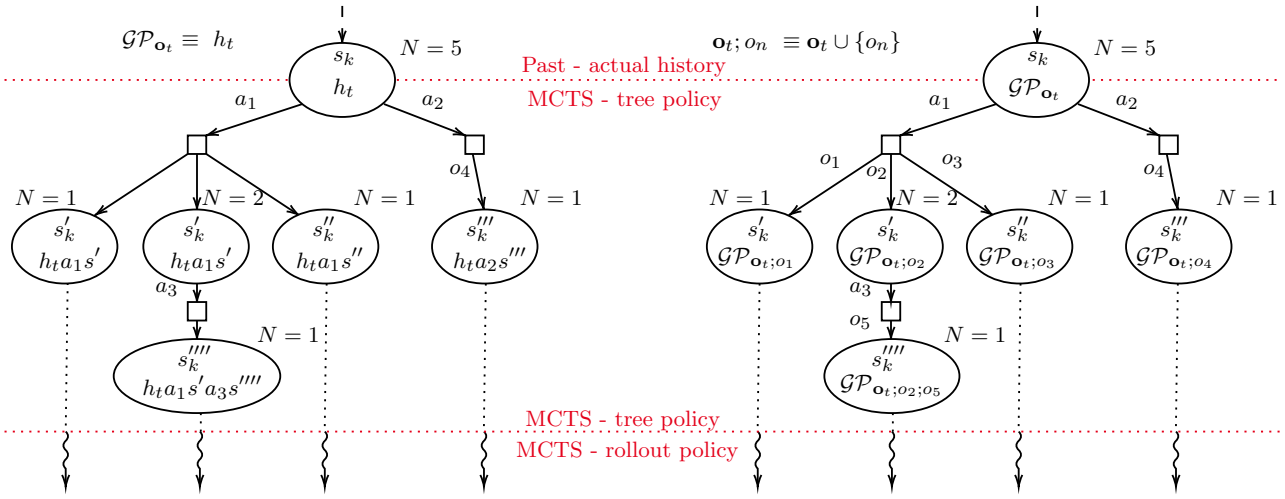
Previous approaches [Morere *et al.*, 2017; Flaspohler *et al.*, 2019] pose the unknown environment as a POMDP, with a GP belief over the environment dynamics. The POMDP formulation assumes a known transition and cost structure, but has the current state only partially observable via some observation function. POMCP solves POMDPs with MCTS in a similar way as described above for BAMCP, by sampling from the state transition function and observation function to build a *belief MDP* on the fly. Figure 1b shows an example POMDP MCTS search tree being constructed.

For these previous approaches, the POMDP observation function is the GP prediction at a robot pose, and in our notation their root belief state is $\{s_k, \mathcal{GP}_{\mathbf{o}_t}\}$ as they also assume that the robot pose (s_k in our notation) is fully observable.

A BAMDP can be interpreted as a special case of a POMDP belief MDP, where the dynamics T is included in the partially observable state. The BAMDP search tree is therefore equivalent to a POMDP belief search tree where POMDP observations are BAMDP state transition outcomes. In other words, a POMDP action-observation history $h_0 a_1 o_1 a_2 o_2$ is equivalent to a BAMCP history $h_0 a_1 s_1 a_2 s_2$. Our approach therefore produces the same structure of search tree as the POMDP belief MCTS tree approaches. The two search trees in Figure 1 are the result of identical sequences of actions and observations and show the equivalence between the POMDP belief and BAMDP search trees.

The previous POMDP belief MCTS search methods [Morere *et al.*, 2017; Flaspohler *et al.*, 2019] maintain a full belief, in the form of a GP, at each search tree node. Given a sampled observation resulting from an action at a parent node, a belief update must be performed to produce the new belief at the child node. Carrying out GP belief updates in the search tree requires drawing a hypothesised data point $\hat{y} \sim \mathcal{GP}_{\mathbf{o}_k}$ from the parent node GP belief posterior and adding this to the GP dataset at the new child belief node $\mathcal{GP}_{\mathbf{o}_k \cup \{\hat{y}\}}$. This is shown in Figure 1b and can incur a large computational cost depending on the dataset size of the GP belief. More GP predictions may also need to be made in the rollout phase – for example, Flaspohler et al. use a random rollout policy with reward based on the GP belief at the leaf node.

Limiting the dataset update to adding a single new data point can be shown to reduce the Cholesky decomposition complexity from $O(N^3)$ to $O(N^2)$ [Osborne *et al.*, 2008] where N is the number of data and sample points. Even so, this belief update is still highly computationally expensive as



(a) MCTS search tree for a BAMDP, with root sampling from the current"real-observations" GP at the root node, and action-observation histories at search nodes. Initialising the 5 search nodes has required 5 samples from the root GP belief $\mathcal{GP}_{\mathbf{o}_t}$ at the root node.

(b) MCTS search tree in POMDP belief space using a separate GP to maintain belief at each node. Initialising the 5 search nodes has required 5 instances of single-datapoint GP updates. $o_{\{1,\dots,5\}}$ are observations sampled from their parent node GPs.

Figure 1: Example equivalent MCTS search trees. The robot's current belief state is at the root of the tree, and 5 MCTS trajectories have been simulated. The process followed to add new leaf nodes to either tree is described in Section 4.3

the dataset of real and hypothesised observations grows and must take place once during each MCTS trial when a new leaf node is added to the tree. For our root-sampled environment models, once the original Cholesky decomposition in (4) has been carried out to incorporate any new real observations, this can be used to draw an arbitrary number of samples at little extra computational cost. The lower computational cost of sampling transition functions from the root belief node means that we can build a larger MCTS search tree within the computational budget than previous approaches.

4.5 Theoretical Analysis

We wish to show the equivalence of rollout distributions between our root sampling BAMDP approach and the previous POMDP belief MCTS search methods, by showing that the probability of generating a history from the BAMDP with individual belief updates at each node is the same as the probability of generating the history when performing root sampling. Let $\mathcal{P}_\pi^{h_t}(h_{t+\tau})$ be the probability of a history $h_{t+\tau}$ in the BAMDP, starting at history h_t under policy π, when carrying out individual belief updates at every stage. Similarly, let $\tilde{\mathcal{P}}_\pi^{h_t}(h_{t+\tau})$ be the history probability when carrying out root sampling.

Proposition 1. *$\mathcal{P}_\pi^{h_t}(h_{t+\tau}) = \tilde{\mathcal{P}}_\pi^{h_t}(h_{t+\tau})$ for all policies π and all histories $h_{t+\tau}$ of length τ.*

Proof. This proof is based on Lemma 1 of [Guez *et al.*, 2014]. With individual belief updates at every stage, the history density is (shortening $\mathcal{P}_\pi^{h_t}(h_{t+\tau})$ to $\mathcal{P}_\pi^{h_t}$ and $\tilde{\mathcal{P}}_\pi^{h_t}(h_{t+\tau})$ to $\tilde{\mathcal{P}}_\pi^{h_t}$):

$$\begin{aligned}\mathcal{P}_\pi^{h_t} &= p(a_t s_{t+1} a_{t+1} \dots s_{t+\tau} \mid h_t, \pi) \\ &= p(a_t \mid h_t, \pi) p(s_{t+1} \mid h_t, \pi, a_t) p(a_{t+1} \mid h_{t+1}, \pi) \dots \\ &\quad p(s_{t+\tau} \mid h_{t+\tau-1}, a_{t+\tau}, \pi) \qquad (7) \\ &= \prod_{t \le t' < t+\tau} \pi(h_{t'}, a_{t'}) \prod_{t < t' \le t+\tau} p(s_{t'} \mid h_{t'-1}, a_{t'-1}) \qquad (8)\end{aligned}$$

$$= \prod_{t \le t' < t+\tau} \pi(h_{t'}, a_{t'}) \cdot \prod_{t < t' \le t+\tau} \int_T T(s_{t'-1}, a_{t'-1}, s_{t'}) p(T \mid h_{t'-1}) dT. \qquad (9)$$

where $s_t^+ = (s_t, h_t)$ and $s_t = (s_k, s_e)$.

Given the definition of T^+ in (6), and the fact that a history h_t uniquely specifies a GP $\mathcal{GP}_{\mathbf{o}_t}$ of observations up to time t:

$$\mathcal{P}_\pi^{h_t} = \prod_{t \le t' < t+\tau} \pi(h_{t'}, a_{t'}) \cdot \prod_{t < t' \le t+\tau} \left[T^o(s_{t'-1}, a, s_{k,t'}) f(s_{e,t'} \mid s_{k,t'}, \mathcal{GP}_{\mathbf{o}_{t'-1}}) \right]. \qquad (10)$$

The GP posterior of $\mathcal{GP}_{\mathbf{o}_{t-1}}$ is a multivariate normal distribution (MVN). A GP belief update with a noise-free sampled observation (e.g. without adding the noise value to the observation variance in the new $\mathbf{K}_{n+1}$ matrix) is performed by

conditioning the posterior MVN on the sampled value (for ease of notation we remove s_k from the MVN probability density function f):

$$\mathcal{P}_{\pi}^{h_t} = \prod_{t \leq t' < t+\tau} \pi(h_{t'}, a_{t'}) \cdot \prod_{t < t' \leq t+\tau} T^o\big(s_{t'-1}, a, s_{k,t'}\big) \cdot \Big[f\big(s_{e,t+1} \mid \mathcal{GP}_{\mathbf{o}_t}\big) \cdot \prod_{t+1 < t' \leq t+\tau} f\big(s_{e,t'} \mid s_{e,t'-1}, ..., s_{e,t+1}\big)\Big]. \tag{11}$$

The repeated belief update product in the square brackets in (11) can be recognised as being equivalent (via the chain rule for probability) as being equivalent to the joint distribution across all values of $s_{e,t'}$:

$$\big[\cdots\big] = f\big(s_{e,t+1}, ..., s_{e,t+\tau} \mid \mathcal{GP}_{\mathbf{o}_t}\big) \tag{12}$$

Therefore the rollout distribution is identical between individual belief updates and root sampling:

$$\mathcal{P}_{\pi}^{h_t} = \prod_{t \leq t' < t+\tau} \pi(h_{t'}, a_{t'}) \cdot \prod_{t < t' \leq t+\tau} T^o\big(s_{t'-1}, a, s_{k,t'}\big) \cdot f\big(s_{e,t+1}, ..., s_{e,t+\tau} \mid \mathcal{GP}_{\mathbf{o}_t}\big) = \tilde{\mathcal{P}}_{\pi}^{h_t}. \tag{13}$$

□

4.6 Online Planning and Execution Loop

The robot starts in the U-MDP with a set of initial observations $\mathbf{o}_0$. At each discrete timestep t, it makes an observation of the mapping o at its current known state s_k, and updates its GP model $\mathcal{GP}_{\mathbf{o}_{t-1}}$ to incorporate the new dataset $\mathbf{o}_t$.

Throughout execution the robot maintains its BAMDP planning model and associated MCTS search tree. When the GP is updated, this forms the new root node h_t of the BAMCP search tree, and any search tree nodes which are incompatible with the new, actual history are removed from the tree.

The robot then selects a single new action by running MCTS on its current model, executing the action in the real world once sufficient MCTS trials have been run (up to a computational budget, i.e. an anytime algorithm). The action with the lowest expected cumulative cost to goal is selected.

5 Experiments

We focus on a stochastic shortest path problem in an environment with an unknown distribution of radiation in order to demonstrate the performance benefits of our approach.

In a 5m × 5m simulated world, a robot equipped with a radiation sensor must navigate to a physical goal location (which is at least 4m away from the initial location) while minimising its cumulative radiation exposure. In this setting the robot pose comprises the known value U-MDP state features: S_k is a finite set of (x, y) locations $\{x, y\} \subseteq S_k$, and the radiation exposure level is the only unknown value state feature $S_e = \mathbb{R}$ where $rad\_exp \in S_e$ is the radiation exposure level at a location. The solution approach discretises rad_exp into 8 discrete ranges $[0, 5), [5, 10), ...$ etc. The physical map is discretised into a grid of states with side length 0.3m, with

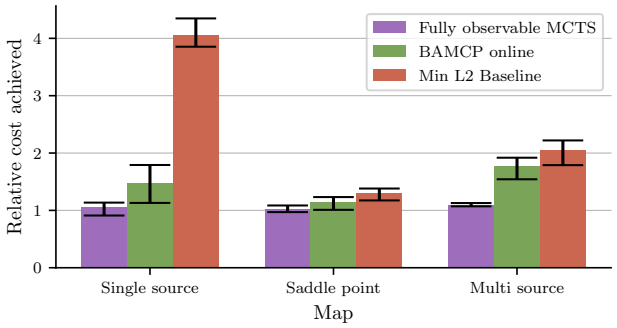


Figure 2: Achieved incurred cost to goal over 3 maps, with 5 runs per algorithm on each map.

an action defined for each of the 4 cardinal directions that has a success probability of 0.9, where success implies transitioning to the state in that direction and failure implies remaining in the same state. To demonstrate a scenario where unknown environment features affect transition probabilities, we state that when the robot's local radiation exposure level exceeds a set limit of 30 its localisation sensor performance worsens and its navigation success probability decreases to 0.5. Simulated radiation level distribution data is generated as described in [Budd *et al.*, 2020].

Figure 2 summarises the the achieved cumulative cost to reach the goal for three "agents", each demonstrating an algorithm. One agent uses the BAMDP approach described above, one carries out the same MCTS planning but with full knowledge of the actual underlying radiation distribution, and one is a shortest path baseline. For the MCTS-based algorithms, the chosen rollout policy is to choose the action that minimises the L2 distance between the next state (if the action is successful) and the goal state. The baseline agent always uses this rollout policy and so attempts to take the shortest physical path to the goal, ignoring radiation cost.

For consistency across the test environments, total cost in Figure 2 is normalised relative to the ground-truth stochastic shortest path cost calculated using an exact method and the actual underlying radiation distribution. MCTS agents were given a budget of 1000 trials, and the UCT exploration constant is dynamically set to equal the value of the decision node.

In general the fully observable MCTS algorithm (which has access to the true radiation distribution) achieves close to the optimal expected score – it can achieve less than the optimal expected score in some runs due to the probabilistic transition function. The BAMDP agent's performance largely depends on how well it can predict with its GP – in complex environments such as the multi radiation source map the BAMCP agent could not predict values accurately many states away from its current location. Conversely, the saddle point map (formed by two large radiation sources at corners of the map) has relatively smooth dynamics and is well modelled online by the BAMCP agent.

For the same experimental setup, Figure 3 compares our root sampling approach with one that maintains a GP belief in

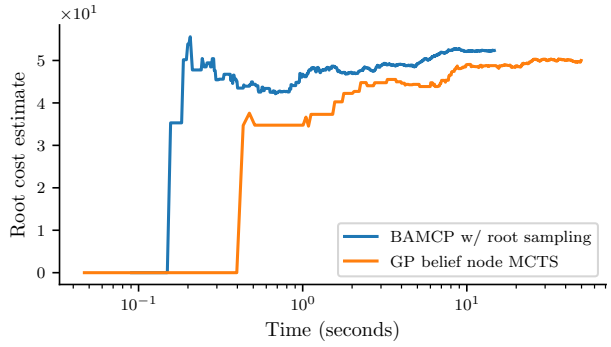


Figure 3: Root state cost estimate vs time for both MCTS belief tree variants.

each search node as in Figure 1b. When both algorithms are given the same root belief state at a single planning epoch, the figure illustrates that the root sampling approach converges to the same value at the root node as the full belief node approach, as expected. The root sampling approach converges significantly faster than the GP belief node approach.

In the experiment scenario, BAMCP with root sampling approach carried out 1000 MCTS trials in 3.66 seconds, compared to 1000 MCTS trials in 49.78 seconds for the GP belief node approach – a relative speed factor of 13.6. The root GP belief node contained only 6 observations – in the case where there were more observations this would further decrease the speed of carrying out GP belief node MCTS relative to carrying out BAMCP with root sampling.

6 Conclusion

We have proposed a framework for planning in uncertain worlds, and demonstrated its ability to plan in environments with unknown features that may affect both robot costs and transitions. We have also demonstrated that we are able to plan more efficiently in these models than previous approaches which plan in unknown environments modelled by GPs.

Our current solution method relies on discretising the values of continuous state features. An obvious extension to this work would be to plan directly with continuous unknown value state features S_e, as several previous methods are able to. Combining root sampling concepts with techniques to manage search tree branching factors in continuous observation MCTS (e.g. progressive widening as in [Flaspohler *et al.*, 2019]) requires some thought as the probability of two sampled environment feature values s_e at location s_k being equal is 0 when the state space S_e is continuous in the search tree. Techniques presented in POMCPOW [Sunberg and Kochenderfer, 2018] may make this feasible while still maintaining some of the benefits of root sampling.

Extending to continuous state spaces in the known value state space S_k is complicated by the need to sample from the GP at a finite set of sample points – which would not be known before the trials if the outcomes of actions could lead to any continuous x and y robot pose, and would not be consistent between trials to enable the same sampling points to be used. Other than simply linearly interpolating between sampled points, one promising avenue is to make use of more complex GP sampling techniques such as described in [Wilson *et al.*, 2020] to freely evaluate GP predictions at continuous values of s_k.

In the future we would also like to use the framework described to tackle safety-constrained problems where the robot has hard limits on environmental feature values that it can safely withstand. Our Bayesian RL formalisation gives us the opportunity to tackle problems that have previously been investigated in other contexts. An example would be the extension of safe exploration for MDPs [Budd *et al.*, 2020; Turchetta *et al.*, 2016] to non-myopically plan where a robot should safely observe.

Extending the BAMDP representation to consider partial observability separately from the unknown environment could be handled in future work by planning with a BAPOMDP rather than a BAMDP, and by adding sampling noise from the GP's observation function to BAMDP root samples. This would allow a robot to plan multiple observations at the same physical location in order to improve the accuracy of its predictions, which is not possible with the BAMDP model where the mapping o is fixed during a single trial. The safety-constrained aspect of the problem would then likely be tackled using methods from cost constrained POMDP literature [Lee *et al.*, 2018].

Acknowledgments

This work has been funded by UK Research and Innovation and EPSRC, through the Robotics and Artificial Intelligence for Nuclear (RAIN) and Offshore Robotics for Certification of Assets (ORCA) hubs [EP/R026084/1, EP/R026173/1]. Budd was sponsored by a Lighthouse Scholarship from Amazon Web Services.

References

[Budd *et al.*, 2020] Matthew Budd, Bruno Lacerda, Paul Duckworth, Andrew West, Barry Lennox, and Nick Hawes. Markov decision processes with unknown state feature values for safe exploration using Gaussian processes. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2020.

[Chen *et al.*, 2016] Min Chen, Emilio Frazzoli, David Hsu, and Wee Sun Lee. POMDP-lite for robust robot planning under uncertainty. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5427–5433. IEEE, 2016.

[Duckworth *et al.*, 2021] Paul Duckworth, Bruno Lacerda, and Nick Hawes. Time-bounded mission planning in time-varying domains with semi-MDPs and Gaussian processes. In *Conference on Robot Learning (CoRL)*. Journal of Machine Learning Research, 2021.

[Duff, 2003] Michael O Duff. Optimal learning: Computational procedures for Bayes-adaptive Markov decision processes. 2003.

[Flaspohler *et al.*, 2019] G. Flaspohler, V. Preston, A. P. M. Michel, Y. Girdhar, and N. Roy. Information-guided robotic maximum seek-and-sample in partially observable continuous environments. *IEEE Robotics and Automation Letters (IEEE RA-L)*, 4(4), 2019.

[Guez *et al.*, 2013] Arthur Guez, David Silver, and Peter Dayan. Scalable and efficient Bayes-adaptive reinforcement learning based on Monte-Carlo tree search. *Journal of Artificial Intelligence Research (JAIR)*, 48:841–883, 2013.

[Guez *et al.*, 2014] Arthur Guez, Nicolas Heess, David Silver, and Peter Dayan. Bayes-adaptive simulation-based search with value function approximation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 27, pages 451–459, 2014.

[Kaelbling *et al.*, 1998] Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *AIJ*, 101(1-2), 1998.

[Lee *et al.*, 2018] Jongmin Lee, Geon-Hyeong Kim, Pascal Poupart, and Kee-Eung Kim. Monte-carlo tree search for constrained POMDPs. In *NeurIPS*, pages 7934–7943, 2018.

[Mausam and Kolobov, 2012] Mausam and Andrey Kolobov. *Planning with Markov decision processes: An AI perspective*. Morgan & Claypool Publishers, 2012.

[Morere *et al.*, 2017] Philippe Morere, Roman Marchant, and Fabio Ramos. Sequential bayesian optimization as a POMDP for environment monitoring with uavs. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 6381–6388. IEEE, 2017.

[Osborne *et al.*, 2008] Michael A Osborne, Stephen J Roberts, Alex Rogers, Sarvapali D Ramchurn, and Nicholas R Jennings. Towards real-time information processing of sensor network data using computationally efficient multi-output Gaussian processes. In *International Conference on Information Processing in Sensor Networks (ISPN)*, pages 109–120. IEEE, 2008.

[Rasmussen and Williams, 2006] CE. Rasmussen and CKI. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[Silver and Veness, 2010] David Silver and Joel Veness. Monte-Carlo Planning in Large POMDPs. *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1–9, 2010.

[Sunberg and Kochenderfer, 2018] Zachary Sunberg and Mykel Kochenderfer. Online algorithms for POMDPs with continuous state, action, and observation spaces. In *International Conference on Automated Planning and Scheduling (ICAPS)*, 2018.

[Turchetta *et al.*, 2016] Matteo Turchetta, Felix Berkenkamp, and Andreas Krause. Safe exploration in finite Markov decision processes with Gaussian processes. In *Advances in Neural Information Processing Systems (NeurIPS)*, NIPS'16, pages 4312–4320, 2016.

[Wachi *et al.*, 2018] Akifumi Wachi, Yanan Sui, Yisong Yue, and Masahiro Ono. Safe exploration and optimization of constrained MDPs using Gaussian processes. In *AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[Wilson *et al.*, 2020] James Wilson, Viacheslav Borovitskiy, Alexander Terenin, Peter Mostowsky, and Marc Deisenroth. Efficiently sampling functions from gaussian process posteriors. In *International Conference on Machine Learning (ICML)*, pages 10292–10302. PMLR, 2020.