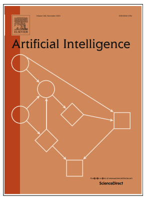
Contents lists available at ScienceDirect

Artificial Intelligence 350 (2026) 104442

Artificial Intelligence

journal homepage: www.elsevier.com/locate/artint



Online POMDP planning with anytime deterministic optimality guarantees

Moran Barenboim [a,*], Vadim Indelman [a,b,c]

[a] Technion Autonomous Systems Program (TASP), Technion – Israel Institute of Technology, Israel
[b] Faculty of Data and Decision Sciences, Technion – Israel Institute of Technology, Israel
[c] Faculty of Aerospace Engineering, Stephen B. Klein, Technion – Israel Institute of Technology, Israel

ARTICLE INFO

Keywords:
Decision-making under uncertainty
POMDP
Planning under uncertainty

ABSTRACT

Decision-making under uncertainty is a critical aspect of many practical autonomous systems due to incomplete information. Partially Observable Markov Decision Processes (POMDPs) offer a mathematically principled framework for formulating decision-making problems under such conditions. However, finding an optimal solution for a POMDP is generally intractable. In recent years, there has been a significant progress of scaling approximate solvers from small to moderately sized problems, using online tree search solvers. Often, such approximate solvers are limited to probabilistic or asymptotic guarantees towards the optimal solution. In this paper, we derive a deterministic relationship for discrete POMDPs between an approximated and the optimal solution. We show that at any time, we can derive bounds that relate between the existing solution and the optimal one. We show that our derivations provide an avenue for a new set of algorithms and can be attached to existing algorithms that have a certain structure to provide them with deterministic guarantees with marginal computational overhead. In return, not only do we certify the solution quality, but we demonstrate that making a decision based on the deterministic guarantee may result in superior performance compared to the original algorithm without the deterministic certification.

1. Introduction

Decision-making under uncertainty is a common challenge in many practical autonomous systems. In such systems, agents often operate with incomplete information about their environment. This uncertainty can arise from various sources, including sensor noise, hardware limitations, modeling approximations, and the inherent unpredictability of the environment. Mathematically, Decision-making under uncertainty can be formalized as Partially Observable Markov Decision Process (POMDP).

Unfortunately, finding an optimal solution to most POMDP problems is computationally intractable, mostly due to a large number of possibilities for the ground truth of the current state, and exponentially increasing possibilities of the future outcomes, commonly referred to as the curse of dimensionality, and the curse of history. As such, most state-of-the-art (SOTA) algorithms aim to find an approximate solution.

One prominent approach to deriving approximate solutions employs an online tree-search paradigm. In this framework, following each real-world decision, an online solver evaluates the current state and projects potential future scenarios. These scenarios are organized within a tree graph structure. As the tree is constructed, the agent assesses the implications of selecting a particular action, subsequently receiving feedback from the environment. This feedback informs the estimation of probabilities for new states, guiding the selection of subsequent actions based on accumulated knowledge. This iterative process continues, building on past outcomes to navigate the decision space.

Given the inherent approximation in these solutions, a natural inquiry regarding the connection between the approximate solution and the actual problem at hand. Some state of the art online algorithms, e.g. [1], offer asymptotic guarantees thus having no finite time guarantees on the solution quality. A different class of algorithms suggests finite time, but probabilistic guarantees such as [2]. Many algorithms have shown good empirical performance, at the advent of the practical use case of POMDP problems, e.g. [3], but fall short of providing a framework that bridges between the policy found and the underlying POMDP.

In this paper, we focus on deriving deterministic guarantees for POMDPs with discrete state, action and observation spaces. Unlike existing black-box sampling mechanisms employed in algorithms such as [3–5], our approach assumes access not only to the observation model but also to the transition and the prior models. By leveraging this additional information, we develop novel bounds that necessitate only a subset of the state and observation spaces, enabling the computation of deterministic bounds with respect to the optimal policy at any belief node within the constructed tree. From a practical standpoint, we

[*] Corresponding author.
E-mail addresses: moranboim@gmail.com, moranbar@campus.technion.ac.il (M. Barenboim), vadim.indelman@technion.ac.il (V. Indelman).

https://doi.org/10.1016/j.artint.2025.104442
Received 13 July 2024; Received in revised form 7 October 2025; Accepted 17 October 2025
Available online 24 October 2025

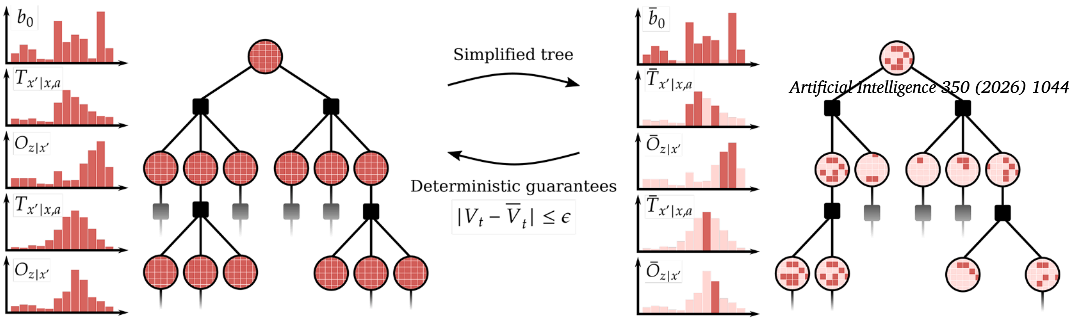


Fig. 1. The figure depicts two search trees: a complete tree (left) that considers all states and observations at each planning step, and a simplified tree (right) that incorporates only a subset of states and observations, linked to simplified models. Our methodology establishes a deterministic link between these two trees.

demonstrate how to harness the theoretical derivations to recent advancements in POMDP approximate solvers, by attaching the bounds to existing state-of-the-art algorithms. We show that despite their stochastic nature, we can guarantee deterministic linkage to the optimal solution with marginal computational overhead. We extend the approach even further by demonstrating how to utilize the bounds to explore the tree and finally select an action based on the deterministic guarantees.

In this paper, our main contributions are as follows. First, we introduce a simplified POMDP that uses a subset of the state and observation spaces to increase the computational efficiency. Then, we derive deterministic bounds that relate between the former and the non-simplified POMDP. Notably, the bounds are only a function of the states and observations known to the simplified POMDP and hence can be calculated while planning to guide the decision-making and even exploration. We also show a tighter version of the bounds considered in the conference version of this paper, [6]. We further extend the approach and show that utilizing these bounds for exploration results in convergence to the optimal solution of the POMDP in finite time; While the optimality guarantees applied only to observation-space simplification in the conference version, we extend the results in this paper by deriving optimality guarantees for both state- and observation-space simplification. Based on the derived bounds, we illustrate how to incorporate the bounds into a general structure of common state-of-the-art algorithms. We utilize the bounds for exploration, decision-making and pruning of suboptimal actions while planning. Last, we demonstrate the practicality of the bounds by experimenting with our novel algorithms, suggested in this paper, namely DB-POMCP, RB-POMCP and DB-DESPOT, which are variants of the POMCP and DESPOT algorithms, to improve the empirical results in finite-horizon problems.

2. Related work

Over the last two decades there has been significant progress in online POMDP planning, aiming to balance the trade-off between computational efficiency and the quality of the solution.

The Heuristic Search Value Iteration (HSVI) [7] algorithm marked a significant milestone in POMDP planning by introducing an efficient point-based value iteration method that provides convergence guarantees. HSVI leverages a heuristic to focus the search on the most promising regions of the belief space, thus improving computational efficiency while maintaining solution quality. Another pivotal algorithm, Successive Approximations of the Reachable Space under Optimal Policies (SARSOP) [8], builds on this idea by further refining the focus on reachable belief states under optimal policies. SARSOP's ability to prune irrelevant parts of the belief space enables it to handle larger POMDPs more effectively. However, these approaches were limited in their scalability to large state spaces due to the necessity of computing a complete belief state at each posterior node in the planning tree.

SARSOP, like other point-based offline solvers, provides suboptimality guarantees *after* completion of an offline solve, whereas our setting is online decision making under tight per-decision budgets, yielding a certificate at each iteration. In large spaces with concentrated posteriors, exhaustive belief updates can allocate computation to low-probability regions; sampling-based online planners (including ours) naturally bias effort toward high-probability trajectories. Consequently, SARSOP is not our primary baseline for time-constrained online planning. That said, its focus on "optimally reachable" beliefs is complementary and can inform our selection of simplified state/observation models without changing guarantees.

The advent of Monte Carlo methods brought a significant shift in online POMDP planning. The Partially Observable Monte Carlo Planning (POMCP) [1] algorithm introduced a particle filter-based approach combined with Monte Carlo tree search (MCTS). POMCP uses a set of particles to represent the belief state and UCT (Upper Confidence bounds applied to Trees, [9]) to guide the search, making it much more scalable and for large state and observation spaces. POMCP is a forward search algorithm which handles the large state and observation spaces by aggregating Monte-Carlo rollouts of future scenarios in a tree structure. During each rollout, a single state particle is recursively propagated from the root node to the leaves of the tree. It adaptively trades off between actions that lead to unexplored areas of the tree and actions that lead to rewarding areas of the tree search by utilizing UCT [10]. The guarantees on the provided solution by POMCP are asymptotic, implying that the quality of the solution remains unknown within any finite time frame.

Another notable approximate solver, Anytime Regularized DESPOT (AR-DESPOT) [2,11] is derived from Regularized DESPOT, which holds theoretical guarantees for the solution quality with respect to its optimal value. Similar to POMCP, AR-DESPOT performs forward search and propagates a single particle from the root node down to its leaves. It relies on branch-and-bound approach in the forward search, and utilizes dynamic programming techniques to update the value function estimate at each node. In contrast to POMCP, Regularized DESPOT offers a probabilistic lower bound on the value function obtained at the root node, providing a theoretical appeal by measuring its proximity to the optimal policy.

While the primary focus of this paper is on discrete POMDP planning, it is essential to acknowledge recent advancements in POMDP planning that encompass both discrete and continuous observation spaces. Few notable approaches include POMCPOW [3], LABECOP [4] and AdaOPS [5], which leverage explicit use of observation models. These algorithms employ importance sampling mechanisms to weigh each state sample based on its likelihood value, which is assumed to be known. Although these methods have exhibited promising performance in practical scenarios, POMCPOW and LABECOP currently lack formal guarantees, while [5] derives probabilistic guarantees which do not hold in practice for AdaOPS algorithm, due to assumption relaxations. To address this gap, [12,13] introduced a simplified solver aimed at bridging the theoretical gap between the empirical success of these algorithms and the absence of theoretical guarantees for continuous observation spaces. Lim et al. [13], derive (i) high-probability bounds on the value loss of their particle-belief solver and (ii) a deterministic guarantee on its *expected* return-obtained after averaging over both the solvers internal randomness and the POMDPs stochastic dynamics. Because this guarantee is only in expectation, a single execution may still violate the bound, whereas the results presented in this paper hold deterministically for every run.

3. Preliminaries

A finite horizon POMDP M is defined as a tuple $\langle \mathcal{X}, \mathcal{A}, \mathcal{Z}, T, O, \mathcal{R}, b_0 \rangle$, where $\mathcal{X}$, $\mathcal{A}$, and $\mathcal{Z}$ represent a discrete state, action, and observation spaces, respectively. The transition density function $T(x_t, a_t, x_{t+1}) \triangleq \mathbb{P}(x_{t+1}|x_t, a_t)$ defines the probability of transitioning from state $x_t \in \mathcal{X}$ to state $x_{t+1} \in \mathcal{X}$ by taking action $a_t \in \mathcal{A}$. The observation density function $O(x_t, z_t) \triangleq \mathbb{P}(z_t|x_t)$ expresses the probability of receiving observation $z_t \in \mathcal{Z}$ from state $x_t \in \mathcal{X}$. $b_0 \equiv \mathbb{P}(x_0 \mid H_0)$ represents the prior probability function, which is the distribution function over the state space at time $t = 0$.

Given the limited information provided by observations, the true state of the agent is uncertain and a probability distribution function over the state space, also known as a belief, is maintained. The belief depends on the entire history of actions and observations, denoted as $H_t \triangleq \{z_{1:t}, a_{0:t-1}\}$. We also define the propagated history as $H_t^- \triangleq \{z_{1:t-1}, a_{0:t-1}\}$. At each time step t, the belief is updated by applying Bayes' rule using the transition and observation models, given the previous action a_{t-1} and the current observation z_t, $b(x_t) = \eta_t \mathbb{P}(z_t|x_t) \sum_{x_{t-1}\in\mathcal{X}} \mathbb{P}(x_t|x_{t-1}, a_{t-1}) b(x_{t-1})$, where η_t denotes a normalization constant and $b_t \triangleq \mathbb{P}(x_t \mid H_t)$ denotes the belief at time t. The updated belief, b_t, is sometimes referred to as the posterior belief, or simply the posterior. We will use these interchangeably throughout the paper.

A policy function $a_t = \pi_t(H_t)$ determines the action to be taken at time step t, based on the history H_t and time t. In the rest of the paper we write $\pi_t \equiv \pi_t(H_t)$ for conciseness. The reward is defined as an expectation over a state-dependent function, $r(b_t, a_t) = \mathbb{E}_{x\sim b_t}[r_x(x, a_t)]$, and is assumed to be bounded by $-\mathcal{R}_{\max} \le r_x(x, a_t) \le \mathcal{R}_{\max}$. The value function for a policy π over a finite horizon T is defined as the expected cumulative reward received by executing π and can be computed using the Bellman update equation,

$$V_t^{\pi}(b_t) = r(b_t, \pi_t) + \mathop{\mathbb{E}}_{z_{t+1:T}} \left[\sum_{\tau=t+1}^{T} r(b_\tau, \pi_\tau) \right]. \tag{1}$$

We use $V_t^{\pi}(b_t)$ and $V_t^{\pi}(H_t)$ interchangeably throughout the paper. The action-value function is defined by executing action a_t and then following policy π,

$$Q_t^{\pi}(b_t, a_t) = r(b_t, a_t) + \mathop{\mathbb{E}}_{z_{t+1:T}} \left[\sum_{\tau=t+1}^{T} r(b_\tau, \pi_\tau) \right]. \tag{2}$$

The optimal value function may be computed using Bellman's principle of optimality,

$$V_t^{\pi^*}(b_t) = \max_{a_t} \{ r(b_t, a_t) + \mathop{\mathbb{E}}_{z_{t+1}|a_t, b_t} \left[V_{t+1}^{\pi^*}(b_{t+1}) \right] \}. \tag{3}$$

The goal of the agent is to find the optimal policy π^* that maximizes the value function.

For notational convenience, we introduce a few more simplifying notations; We use $\mathcal{V}_{max,t}, \mathcal{V}_{min,t}$ to denote upper an lower bounds on the value function at time step t. In the simplest case, these may be $\mathcal{V}_{max,t} = (T - t) \cdot \mathcal{R}_{\max}$, $\mathcal{V}_{min,t} = (t - T) \cdot \mathcal{R}_{\max}$. Additionally, we denote a trajectory as, $\tau_t = \{x_0, a_0, z_1, x_1, a_1, \dots, a_{t-1}, x_t, z_t\}$, and a corresponding probability distribution over the possible trajectories, $\mathbb{P}(\tau_t)$. We denote a policy-dependent trajectory distribution as $\mathbb{P}^{\pi}(\tau_t) \equiv \mathbb{P}(\tau_t \mid b_0, \pi_0, \dots, \pi_t)$.

4. Simplified POMDP

Typically, it is infeasible to fully expand a Partially Observable Markov Decision Process (POMDP) tree due to the extensive computational resources and time required. To address this challenge, we propose two approaches. In the first approach, presented in Section 5.1, we propose a solver that selectively chooses a subset of the observations to branch from, while maintaining a full posterior belief at each node. This allows us to derive an hypothetical algorithm that directly uses our suggested deterministic bounds to choose which actions to take while exploring the tree. As in most scenarios computing a complete posterior belief may be too expensive, in Section 5.2 we suggest an improved method that in addition to branching only a subset of the observations, selectively chooses a subset of the states at each encountered belief.

The presented approaches diverge from many existing algorithms that rely on black-box prior, transition, and observation models. Instead, our method directly utilizes state and observation probability values to evaluate both the value function and the associated bounds. In return, an anytime deterministic guarantee on the value function for the derived policy concerning its deviation from the optimal value function is derived.

To that end, we define a simplified POMDP, which is a reduced version of the original POMDP that abstracts or ignores certain states and/or observations. A simplified POMDP, $\bar{M}$, is a tuple $\langle \bar{\mathcal{X}}, \mathcal{A}, \bar{\mathcal{Z}}, \bar{T}, \bar{O}, \mathcal{R}, \bar{b}_0 \rangle$, where $\bar{\mathcal{X}}, \bar{\mathcal{Z}}, \bar{T}$ and $\bar{O}$ are the simplified versions of the state and observation spaces, and their corresponding transition and observation models,

$$\bar{b}_0(x) \triangleq \begin{cases} b_0(x) & , x \in \bar{\mathcal{X}}_0 \\ 0 & , otherwise \end{cases} \tag{4}$$

$$\bar{\mathbb{P}}(x_{t+1} \mid x_t, a_t) \triangleq \begin{cases} \mathbb{P}(x_{t+1} \mid x_t, a_t) & , x_{t+1} \in \bar{\mathcal{X}}(H_{t+1}^-) \\ 0 & , otherwise \end{cases} \tag{5}$$

$$\bar{\mathbb{P}}(z_t \mid x_t) \triangleq \begin{cases} \mathbb{P}(z_t \mid x_t) & , z_t \in \bar{\mathcal{Z}}(H_t) \\ 0 & , otherwise \end{cases} \tag{6}$$

where $\bar{\mathcal{X}}(H_{t+1}^-) \subseteq \mathcal{X}$ and $\bar{\mathcal{Z}}(H_t) \subseteq \mathcal{Z}$ may be chosen arbitrarily, e.g. by sampling or choosing a fixed subset a-priori, as the derivations of the bounds are independent of the subset choice. Note that the simplified prior, transition and observation models are unnormalized and do not aim to represent valid distribution functions. For the rest of the sequel we drop the explicit dependence on the history, and denote $\bar{\mathcal{X}}(H_{t+1}^-) \equiv \bar{\mathcal{X}}$, $\bar{\mathcal{Z}}(H_t) \equiv \bar{\mathcal{Z}}$. The action space, $\mathcal{A}$ and prior probability, b_0 are as defined in the original POMDP, M.

With the definition of the simplified POMDP, we define a corresponding simplified value function,

$$\bar{V}^{\pi}(\bar{b}_0) \triangleq \bar{\mathbb{E}} \left[\sum_{t=0}^{T} r(x_t, a_t) \right] \tag{7}$$

$$= \sum_{t=0}^{T} \sum_{z_{1:t}} \sum_{x_{0:t}} \prod_{k=1}^{t} \bar{\mathbb{P}}(z_k \mid x_k) \bar{\mathbb{P}}(x_k \mid x_{k-1}, \pi_{k-1}) \bar{b}(x_0) r(x_t, a_t) \tag{8}$$

$$= \sum_{t=0}^{T} \sum_{\tau_t} \bar{\mathbb{P}}^{\pi}(\tau_t) r(x_t, a_t), \tag{9}$$

where the simplified expectation-like operator, $\bar{\mathbb{E}}[\cdot]$, is taken with respect to the simplified prior, transition and observation models, which do not include the entire distribution, and thus is not a complete expectation.

We use the simplified value function as a computationally-efficient replacement for the theoretical value function; For clarity, the simplified POMDP and consequently all derivations consider a finite-horizon POMDP, but its extension to the discounted infinite horizon case is straightforward, by introducing the discount factor whenever the reward is being used, and an additive term for truncating the tree, $\gamma^t V_{max,t}$, as suggested in, e.g., [14].

In the following sections, we will derive upper and lower bounds between the simplified and the theoretical values of a given policy. Then, we will show how to use the simplification to achieve guarantees with respect to the optimal value function of the original POMDP, and how to utilize these bounds for planning.

5. Anytime dnistic guarantees for implified POMDPs

5.1. Simplified observation space

We first analyze the performance guarantees of a simplified observation space, while assuming a complete belief update at each considered history node, i.e., $\bar{\mathcal{X}} \equiv \mathcal{X}$. Such an approach is viable when the posterior belief can be calculated efficiently, e.g. when the state spae is sufficiently small. We start by presenting a bound between the simplified value function and the theoretical one of a given policy; then, we provide optimality guarantees for any policy, obtained by solving the simplified POMDP, both in terms of convergence and a deterministic bound, in which the optimal value, for an unknown policy must reside in.

5.1.1. Fixed olicy uarantees for sied bservation paces

The following theorem describes the guarantees of the observation-simplified value function with respect to its theoretical value,

Theorem 1. *Let b_t belief state at time t, and T be the last time step of the POMDP. Let $V^\pi(b_t)$ be the theoretical value function by following a policy π, and let $\bar{V}^\pi(b_t)$ be the simplified value function, as defined in (7), by following the same policy. Then, for any policy π, the difference between the theoretical and simplified value functions is bounded as follows,*

$$\left|V^\pi(b_t) - \bar{V}^\pi(b_t)\right| \leq \mathcal{R}_{\max} \sum_{\tau=t+1}^{T} \left[1 - \sum_{z_{t+1:\tau}} \sum_{x_{t:\tau}} b(x_t) \prod_{k=t+1}^{\tau} \bar{\mathbb{P}}(z_k \mid x_k) \mathbb{P}(x_k \mid x_{k-1}, \pi_{k-1})\right] \triangleq \epsilon^\pi(b_t). \tag{10}$$

Proof. The proof is provided in Appendix A.1. □

Similarly, the action-dependent bound on the value difference, denoted $\epsilon^\pi(b_t, a_t)$, is the bound of taking action a_t in belief b_t and following policy π thereafter,

$$\left|Q^\pi(b_t, a_t) - \bar{Q}^\pi(b_t, a_t)\right| \leq \epsilon^\pi(b_t, a_t), \tag{11}$$

where,

$$\epsilon^\pi(b_t, a_t) \triangleq \mathcal{R}_{\max} \sum_{\tau=t+1}^{T} \Big[1 - \sum_{z_{t+1:\tau}} \sum_{x_{t:\tau}} b(x_t) \bar{\mathbb{P}}(z_{t+1} \mid x_{t+1}) \mathbb{P}(x_{t+1} \mid x_t, a_t) \cdot \prod_{k=t+2}^{\tau} \bar{\mathbb{P}}(z_k \mid x_k) \mathbb{P}(x_k \mid x_{k-1}, \pi_{k-1})\Big]. \tag{12}$$

Importantly, $\epsilon^\pi(b_t)$ and $\epsilon^\pi(b_t, a_t)$ only contain terms which depend on observations that are within the simplified space, $z \in \bar{\mathcal{Z}}$. This is an essential property of the bounds, as it is a value that can easily be calculated during the planning process and provides a certification of the policy quality at any given node along the tree. Furthermore, it is apparent from (10) that as the number of observations included in the simplified set, $\bar{\mathcal{Z}}$, increases, the values of $\epsilon^\pi(b_t)$ and $\epsilon^\pi(b_t, a_t)$ consequently diminishes,

$$\sum_{z_{1:\tau}} \sum_{x_{0:\tau}} b(x_0) \prod_{k=1}^{\tau} \bar{\mathbb{P}}(z_k \mid x_k) \mathbb{P}(x_k \mid x_{k-1}, \pi_{k-1}) \xrightarrow{\bar{\mathcal{Z}} \to \mathcal{Z}} 1$$

leading to a convergence towards the theoretical value function, i.e. $\epsilon^\pi(b_t) \to 0$ and $\epsilon^\pi(b_t, a_t) \to 0$.

5.1.2. Optimality guarantees for simplified observation spaces

Theorem 1 provides both lower and upper bounds for the theoretical value function, assuming a fixed policy. Using this theorem, we can derive upper and lower bounds for any policy, including the optimal one. This is achieved by applying the Bellman optimality operator to the upper bound in a repeated manner, instead of the estimated value function; In the context of tree search algorithms, our algorithm explores only a subset of the decision tree due to pruned observations. However, at every belief node encountered during this exploration, all potential actions are expanded. The action-value function of these expanded actions is bounded using the Upper Deterministic Bound, which we now define as

$$\mathrm{UDB}^\pi(b_t, a_t) \triangleq \bar{Q}^\pi(b_t, a_t) + \epsilon^\pi(b_t, a_t) = r(b_t, a_t) + \bar{\mathbb{E}}_{z_{t+1}}[\bar{V}^\pi(b_{t+1})] + \epsilon^\pi(b_t, a_t). \tag{13}$$

In the event that no subsequent observations are chosen for a given history, the value of $\bar{Q}^\pi(b_t, a_t)$ simplifies to the immediate reward plus an upper bound for any subsequent policy, given by $\mathcal{R}_{\max} \cdot (T - t - 1)$. Then, we make the following claim,

Lemma 1. *The optimal value function can be bounded by,*

$$V^{\pi*}(b_t) \leq UDB^{\pi^\dagger}(b_t), \tag{14}$$

where the policy $\pi^\dagger$ is determined according to Bellman optimality over the UDB, i.e.

$$\pi^\dagger(b_t) = \arg\max_{a_t \in \mathcal{A}} [\bar{Q}^{\pi^\dagger}(b_t, a_t) + \epsilon^{\pi^\dagger}(b_t, a_t)] = \arg\max_{a_t \in \mathcal{A}} UDB^{\pi^\dagger}(b_t, a_t) \tag{15}$$

$$UDB^{\pi^\dagger}(b_t) \triangleq \max_{a_t \in \mathcal{A}} UDB^{\pi^\dagger}(b_t, a_t). \tag{16}$$

Proof. The proof is provided in Appendix A.1.2. □

Notably, using UDB to find the optimal policy does not require a recovery of all the observations in the theoretical belief tree, but only a subset which depends on the definition and complexity of the POMDP. Each action-value is bounded by a lower and upper bound, which can be represented as an interval enclosing the theoretical value. When the bound intervals of two candidate actions do not overlap, one can clearly discern which action is suboptimal, rendering its subtree redundant for further exploration. This distinction sets UDB apart from current state-of-the-art online POMDP algorithms. In those methods, any finite-time stopping condition fails to ensure optimality since the bounds used are either heuristic or probabilistic in nature. Note, however, that previous *offline* algorithms did utilize similar pruning, such as SARSOP.

In addition to certifying the obtained policy with Bellman optimality criteria, one can utilize UDB as an exploration criteria,

$$a_t = \arg\max_{a_t \in \mathcal{A}} [\mathrm{UDB}^{\pi^\dagger}(b_t, a_t)], \tag{17}$$

which ensures convergence to the optimal value function, as the number of visited posterior nodes increases.

Corollary 1.1. *By utilizing Lemma 1 and the exploration criteria defined in (17), an increasing number of explored belief nodes guarantees convergence to the optimal value function.*

Proof. The proof is provided in Proof 2.3. □

5.2. Simplified state and observation spaces

In most scenarios, a complete evaluation of posterior beliefs during the planning stage may pose significant computational challenges. To tackle this issue, we propose the use of a simplified state space in addition to the simplified observation space considered thus far. Specifically, we derive deterministic guarantees of the value function that allow for the selection of a subset from both the states and observations.

We start the analysis of simplifying the state-and-observation spaces by fixing a policy and derive upper and lower bounds for the theoretical, yet unknown, value function at the root node, hereafter referred to as the 'root-value'. This process involves the use of a simplified value function and an additional bonus term, which are easier to compute than the theoretical value function. Considering that various segments of the decision tree contribute differently to the upper bound, we then examine each subtree's contribution separately, which leads to a recursive formulation of the bound. Importantly, these bounds are exclusively derived in relation to, and hold only with respect to, the root node. This is in contrast to the bounds shown in Theorem 1, which bound the value function of each node in the belief tree.

Using the deterministic bounds at the root allows us to certify the performance of following a particular policy starting from the root of the planning tree. Based on these bounds we extend previous results, shown in theorem 2, and show that, (1) exploring the tree with a bound that is formulated with respect to the root node leads to an optimistic estimation of the optimal value function with respect to that root node. (2) Utilizing the bounds for action exploration leads to convergence to the optimal solution of the entire tree.

5.2.1. Fixed policy guarantees

We begin by stating the core theorem of our paper, which sets forth the upper and lower bounds of a root-value function, with a simplified value function,

Theorem 2.

Let b_0 and $\bar{b}_0$ be the theoretical and simplified belief states, respectively, at time $t = 0$, and T be the last time step of the POMDP. Let $V^{\pi}(b_0)$ be the theoretical value function by following a policy π, and let $\bar{V}^{\pi}(\bar{b}_0)$ be the simplified value function by following the same policy, as defined in (7). Then, for any policy π, the theoretical value function and at the root is bounded as follows,

$$\mathcal{L}_0^{\pi}(H_0) \leq V^{\pi}(b_0) \leq \mathcal{U}_0^{\pi}(H_0). \tag{18}$$

where,

$$\mathcal{U}_0^{\pi}(H_0) \equiv \bar{V}^{\pi}(\bar{b}_0) + \mathcal{V}_{max,0}\left[1 - \sum_{\tau_0} \bar{\mathbb{P}}(\tau_0)\right] + \sum_{t=0}^{T-1} \mathcal{V}_{max,t+1}\left[\sum_{\tau_t} \bar{\mathbb{P}}^{\pi}(\tau_t) - \sum_{\tau_{t+1}} \bar{\mathbb{P}}^{\pi}(\tau_{t+1})\right] \tag{19}$$

$$\mathcal{L}_0^{\pi}(H_0) \equiv \bar{V}^{\pi}(\bar{b}_0) + \mathcal{V}_{min,0}\left[1 - \sum_{\tau_0} \bar{\mathbb{P}}(\tau_0)\right] + \sum_{t=0}^{T-1} \mathcal{V}_{min,t+1}\left[\sum_{\tau_t} \bar{\mathbb{P}}^{\pi}(\tau_t) - \sum_{\tau_{t+1}} \bar{\mathbb{P}}^{\pi}(\tau_{t+1})\right] \tag{20}$$

Proof. The proof is provided in Proof A.1.4. □

In this theorem, we introduced a minor modification to the theorem presented in the conference version of this paper, [6]. We replaced the term $\mathcal{R}_{\max} \cdot (T - t)$ with the more general $\mathcal{V}_{max,t}$ by performing simple algebraic transitions; the principles and conclusions of both remain the same. A key aspect of Theorem 2 is that the bounds it establishes are exclusively dependent on the simplified state and observation spaces. This characteristic is vital in order to compute them during the planning phase.

The intuition behind the result of the derivation can be interpreted as follows; it takes a conservative approach to the value estimation by assuming that every trajectory not observed may obtain an extremum value. Moreover, it allows flexibility in how the trajectories are selected, which are allowed to be chosen arbitrarily in terms of the simplified state space, observation space and the horizon of each trajectory.

The theorem provides bounds for the theoretical value function at the root node of the search tree, given a policy. Using Bellman-like equations, one can restructure the formulation to compute the bounds recursively, which is crucial for making computations in online planning computationally efficient,

$$\mathcal{U}_0^{\pi}(H_t) \triangleq \sum_{\tau_t \in \mathcal{T}(H_t)} \bar{\mathbb{P}}(\tau_t) r(x_t, \pi_t) + \sum_{\tau_t \in \mathcal{T}(H_t)} \bar{\mathbb{P}}(\tau_t)\mathcal{V}_{\max,t} + \sum_{z_{t+1} \in \bar{\mathcal{Z}}(H_t, \pi_t)} \left[\mathcal{U}_0^{\pi}(H_{t+1}) - \sum_{\tau_{t+1} \in \mathcal{T}(H_{t+1})} \bar{\mathbb{P}}(\tau_{t+1})\mathcal{V}_{\max,t}\right] \tag{21}$$

$$\mathcal{L}_0^{\pi}(H_t) \triangleq \sum_{\tau_t \in \mathcal{T}(H_t)} \bar{\mathbb{P}}(\tau_t) r(x_t, \pi_t) + \sum_{\tau_t \in \mathcal{T}(H_t)} \bar{\mathbb{P}}(\tau_t)\mathcal{V}_{\min,t} + \sum_{z_{t+1} \in \bar{\mathcal{Z}}(H_t, \pi_t)} \left[\mathcal{L}_0^{\pi}(H_{t+1}) - \sum_{\tau_{t+1} \in \mathcal{T}(H_{t+1})} \bar{\mathbb{P}}(\tau_{t+1})\mathcal{V}_{\min,t}\right] \tag{22}$$

and,

$$\mathcal{U}_0^{\pi}(H_T) \triangleq \sum_{\tau_T \in \mathcal{T}(H_T)} \bar{\mathbb{P}}(\tau_T) r(x_T), \quad \mathcal{L}_0^{\pi}(H_T) \triangleq \sum_{\tau_T \in \mathcal{T}(H_T)} \bar{\mathbb{P}}(\tau_T) r(x_T). \tag{23}$$

where $\mathcal{T}(H_t)$ represent the set of trajectories that consist history H_t, i.e., all trajectories $\mathcal{T}(H_t) = \{\left(x_{0:t}, a_{0:t-1}, z_{1:t}\right) \mid (a_{0:t-1}, z_{1:t}) = H_t\}$. The values $\mathcal{U}_0^{\pi}(H_t)$ and $\mathcal{L}_0^{\pi}(H_t)$, represent the relative upper and lower bounds of node H_t with respect to the value function at the root, H_0. In other words, they do not represent the bounds of a policy starting from node H_t. The first two summands have a similar structure to the standard Bellman update operator used in POMDPs, with two main differences. First, the state dependent reward is multiplied by the probability of the entire trajectory from the root node, and not the density value of the belief. Notably, the value of $\sum_{\tau_t \in \mathcal{T}(H_t)} \mathbb{P}(\tau_t)$ will generally not sum to one, due to the dependence of the summed trajectories on the history. Second, there is no expectation operator over the values of the next time step. This is a result of using a distribution over the trajectories, instead of the belief itself. The last summand assigns an optimistic value for the set of trajectories reached to node H_t but not to H_{t+1}.

5.2.2. Optimality guarantees

We have shown in Theorem 2 how to calculate bounds for the difference in value functions between the original and the simplified POMDP, given a fixed policy. In this section, we show that by applying Bellman-like optimality operator on $\mathcal{U}_0(H_t)$, the obtained value at the root node is an upper bound for the optimal value function. More formally,

Corollary 1.1. *Let $\mathcal{A}$ be the set of actions and $\mathcal{U}_0^{\star}(H_t)$, $\mathcal{L}_0^{\star}(H_t)$ be the upper and lower bounds of node H_t chosen according to,*

$$\mathcal{U}_0^{\star}(H_t) \triangleq \max_{a_t \in \mathcal{A}} \sum_{\tau_t \in \mathcal{T}(H_t)} \bar{\mathbb{P}}(\tau_t)\left[r(x_t, a_t) + \mathcal{V}_{\max,t}\right] + \sum_{z_{t+1} \in \bar{\mathcal{Z}}(H_t, a_t)} \left[\mathcal{U}_0^{\star}(H_{t+1}) - \sum_{\tau_{t+1} \in \mathcal{T}(H_{t+1})} \bar{\mathbb{P}}(\tau_{t+1})\mathcal{V}_{\max,t}\right] \tag{24}$$

$$\mathcal{L}_0^{\star}(H_t) \triangleq \max_{a_t \in \mathcal{A}} \sum_{\tau_t \in \mathcal{T}(H_t)} \bar{\mathbb{P}}(\tau_t)\left[r(x_t, a_t) + \mathcal{V}_{\min,t}\right] + \sum_{z_{t+1} \in \bar{\mathcal{Z}}(H_t, a_t)} \left[\mathcal{L}_0^{\star}(H_{t+1}) - \sum_{\tau_{t+1} \in \mathcal{T}(H_{t+1})} \bar{\mathbb{P}}(\tau_{t+1})\mathcal{V}_{\min,t}\right] \tag{25}$$

and,

$$\mathcal{U}_0^{\star}(H_T) \triangleq \sum_{\tau_T \in \mathcal{T}(H_T)} \bar{\mathbb{P}}(\tau_T) r(x_T), \quad \mathcal{L}_0^{\star}(H_T) \triangleq \sum_{\tau_T \in \mathcal{T}(H_T)} \bar{\mathbb{P}}(\tau_T) r(x_T). \tag{26}$$

Then, the optimal root-value is bounded by,

$$\mathcal{L}_0^{\star}(H_0) \leq V^{\pi^*}(H_0) \leq \mathcal{U}_0^{\star}(H_0). \tag{27}$$

Proof. The proof is provided in Proof A.1.5. □

In this corollary, we establish that employing the 'partial' root-bound is sufficient for ensuring both upper and lower bounds in relation to the optimal value function at the root node. This approach differs from that presented in the previous section (see Lemma 1). There, each node in the tree was associated with its unique upper bound based on its value function. In contrast, the current corollary demonstrates that using the 'partial' bound across all nodes in the tree, which is valid only at the root, still guarantees bounded value for the optimal root-value function, while avoiding the requirement to maintain a complete belief at each node of the tree.

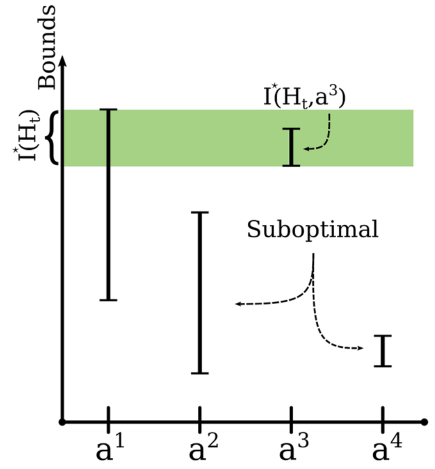


Fig. 2. Bound intervals for different actions. The optimal value function is guaranteed to be between the maximal lower and upper bounds. As a result, actions a^2 and a^4 are suboptimal and can be pruned safely.

5.2.3. *Early stopping criteria*

Corollary 1.1 establishes that the recursive Bellman-like optimality operator, can be used to bound the optimal value function at the root. Since the bounds are deterministic, these bounds can be used for eliminating suboptimal actions with full certainty while planning.

Then, we define the interval for each action at the root as,

$$I^\star(H_0, a_0) \in \left[\mathcal{L}_0^\star(H_t, a_0), \mathcal{U}_0^\star(H_0, a_0)\right], \tag{28}$$

and use it as a tool for pruning suboptimal actions once an upper bound of an action falls below the best lower bounds amongst other actions within that node, see Fig. 2 for an illustration.

State-of-the-art algorithms such as POMCP and DESPOT employ probabilistic and asymptotic reasoning to approximate the optimal policy, and lack a mechanism to conclusively determine the suboptimality of an action, leading to infinite exploration of suboptimal actions. In contrast, utilizing (28) guarantees that once an action is identified suboptimal, it can be safely excluded from further consideration. Since the bounds can be integrated with arbitrary exploration methods, it provides a novel mechanism for pruning with contemporary SOTA algorithms.

Importantly, this approach introduces a practical stopping criterion for the online tree search process. When the exploration results in only one viable action remaining at the root, it signifies the identification of the optimal action. Note that this does note necessitate exhaustive exploration of the entire tree or complete convergence of the bounds.

5.2.4. *Exploration strategies*

One can further utilize the root upper bound to determine the exploration of actions, the simplified state and observation spaces at run time, which guarantees convergences to the optimal value function in finite time, which is novel for online tree search POMDPs solvers to the best of our knowledge. We define the following deterministic exploration strategy,

$$a_t = \arg\max_{a\in\mathcal{A}}\Big\{ \sum_{\tau_t\in\mathcal{T}(H_t)} \bar{\mathbb{P}}(\tau_t) r(x_t, a) + \sum_{z_{t+1}\in\bar{\mathcal{Z}}(H_t,a)} \mathcal{U}_0^\star(H_{t+1}) \tag{29}$$

$$+ \mathcal{V}_{\max,t}\left[\sum_{\tau_t\in\mathcal{T}(H_t)} \bar{\mathbb{P}}(\tau_t) - \sum_{\tau_{t+1}\in\mathcal{T}(H_t,a)} \bar{\mathbb{P}}(\tau_{t+1})\right]\Big\}$$

$$z_{t+1} = \arg\max_{o_{t+1}\in\mathcal{Z}(H_t,a_t)} \{\mathcal{U}_0^\star((H_t, a_t, o_{t+1})) - \mathcal{L}_0^\star((H_t, a_t, o_{t+1}))\} \tag{30}$$

$$x_{t+1} = \arg\max_{x\in\mathcal{X}(H_{t+1})} \{\bar{\mathbb{P}}^\star((\tau_t, a_t, z_{t+1}, x)) - \sum_{\tau_T} \bar{\mathbb{P}}^\star(\tau_T \mid \tau_t, a_t, z_{t+1}, x)\}, \tag{31}$$

where the actions are chosen by the highest upper bound, sometimes referred to as an *"optimism in face of uncertainty"*, which offers a balance between exploration and exploitation of actions that are possibly optimal or have high uncertainty in their value. Observations are chosen based on the maximum gap between the upper and lower bounds, which results in observations with high uncertainty in their value. Last, we define $\bar{\mathbb{P}}^\star(\tau_t)$ as the probability of a trajectory τ_t under a policy derived from recursive action selection as per (29). Subsequently, the selection of states effectively maximizes the difference in probability between the individual trajectory density and the aggregate probability of all sampled trajectories that begin with that particular trajectory.

Corollary 2.2. *Performing exploration based on* (29)–(31) *ensures that the algorithm converges to the optimal value function within a finite number of planning iterations.*

Proof. The proof is provided in Proof A.1.6. □

Importantly, alternative methods for the state-action-observation exploration are viable and, if given limited planning time, may offer improved performance in practice. Corollary 2.2 suggests one way that is guaranteed to converge in finite time. We leave the investigation of other approaches for finite-time convergence using the deterministic bounds for future research.

Moreover, the bounds suggested in this chapter can be integrated with established algorithms like POMCP or DESPOT [1,2], an approach which offers several advantages over the existing algorithms. First, The quality of their solutions with respect to the optimal value can be assessed and validated. Second, whenever the bounds at the root of the solver do not overlap, the planning session can be terminated early with a guarantee of identifying the optimal action.

5.3. *Impact of POMDP characteristics on deterministic bounds*

To make explicit how POMDP characteristics affect bound tightness, we recast the upper–lower gap in terms of simplified-space coverage and local value spread.

The analytical bounds depend on trajectory-probability terms such as $\sum_\tau \bar{\mathbb{P}}(\tau)$ and on the auxiliary bounds $\mathcal{V}_{\max,t}, \mathcal{V}_{\min,t}$; starting from the recursive definitions in (21), let

$$\Delta V(H_t) \triangleq \mathcal{V}_{\max}(H_t) - \mathcal{V}_{\min}(H_t),$$

$$\delta(H_t) \triangleq \sum_{\tau_t\in\mathcal{T}(H_t)} \bar{\mathbb{P}}(\tau_t) - \sum_{z_{t+1}\in\bar{\mathcal{Z}}(H_t,\pi_t)} \sum_{\tau_{t+1}\in\mathcal{T}(H_{t+1})} \bar{\mathbb{P}}(\tau_{t+1}),$$

then following the derivation in Section A.1.7,

$$\mathcal{U}_0^\pi(H_t) - \mathcal{L}_0^\pi(H_t) = \Delta V(H_t)\,\delta(H_t) + \sum_{z_{t+1}\in\bar{\mathcal{Z}}(H_t,\pi_t)} \left[\mathcal{U}_0^\pi(H_{t+1}) - \mathcal{L}_0^\pi(H_{t+1})\right]. \tag{32}$$

Eq. (32) indicates that the gap accumulates the value function spread $\Delta V(H_k)$ only on the time-steps where the trajectory is not part of the simplified set. The attributes govern the magnitude of the bounds are twofold:

1. **Coverage probability of the simplified spaces.** An important quantity is the probability mass of trajectories not part of the simplified sets, $\bar{\mathcal{X}}, \bar{\mathcal{Z}}$, rather than the cardinality of the omitted sets. A large POMDP whose belief is highly concentrated within the simplified spaces can therefore yield a smaller gap than a small, nearly uniform domain.
2. **Magnitude of the local value spread, $\Delta V(H_k)$.** Each time a trajectory leaves the simplified spaces, its contribution to the gap is scaled by $\Delta V(H_k) = \mathcal{V}_{\max}(H_k) - \mathcal{V}_{\min}(H_k)$, the policy-independent spread between the upper and lower *cumulative* value bounds at node H_k. Sharpening these auxiliary bounds, e.g. by using informed rollouts or MDP-optimal surrogates, reduces $\Delta V(H_k)$ and proportionally narrows the overall gap.

Practical implication. Maximizing the probability coverage of the simplified spaces, followed by tightening the auxiliary bounds to reduce $\Delta V(H_k)$, yields a more favourable gap-complexity trade-off than a uniform expansion of the simplified spaces.

6. Algorithms

In this section we aim to describe how to fit our bounds to a blueprint of a general algorithm, named ALGORITHM—$\mathcal{A}$, which serves as an abstraction to many existing algorithms. Then, we explicitly describe two algorithms, DB-POMCP, an adaptation to POMCP that uses UCB for exploration, and our deterministic bounds for decision-making, and RB-POMCP, a particle-based solver that utilizes the bounds both for decision-making and exploration.

To compute the deterministic bounds, we utilize Bellman's update and optimality criteria. This approach naturally fits dynamic programming approaches such as DESPOT [11] and AdaOPS [5]. However, it may also be attached with algorithms that rely on Monte-Carlo estimation, such as POMCP [1], by viewing the search tree as a policy tree.

While the analysis presented in Section 5 is general and independent of the selection mechanism of the states or observations, we focus on sampling as a way to choose the simplified states at each belief node and the observations to branch from. Furthermore, the selection of the subspaces $\bar{\mathcal{X}}, \bar{\mathcal{Z}}$ need not be fixed, and may change over the course of time, similar to state-of-the-art algorithms, such as [1–5]. Alternative selection methods may also be feasible, as sampling from the correct distribution is not required for the bounds to hold. Importantly, attaching our bounds to arbitrary exploration mechanism certifies the algorithm solution with deterministic bounds to the optimal solution, and may result in an improved decision making, as will be shown in the experimental section.

ALGORITHM—$\mathcal{A}$ is outlined in Algorithm 1. For clarity of exposition, we assume the following; at each iteration a single state particle is propagated from the root node to the leaf (Line 2 of function SEARCH). The selection of the next state and observations are done by sampling from the observation and transition models (Line 26), and each iteration ends with the full horizon of the POMDP (Line 23). However, none of these are a restriction of our approach and may be replaced with arbitrary number of particles, arbitrary state and observation selection mechanism and a single or multiple expansions of new belief nodes at each iteration.

To compute the bounds, we require both the state trajectory, denoted as τ, and its probability value, $\mathbb{P}_\tau$. We use the state trajectory as a mechanism to avoid duplicate summation of an already accounted for probability value and is utilized to ascertain its uniqueness at a belief node. The probability value, $\mathbb{P}_\tau$, is the likelihood of visiting a trajectory $\tau = \{x_0, a_0, x_1, z_1, \dots, a_{t-1}, x_t, z_t\}$ and is calculated as the product of the prior, transition and observation likelihoods (Line 16). If a trajectory was not previously observed in a belief node, its reward value is multiplied by the likelihood of the trajectory. Each trajectory likelihood is maintained as part of a cumulative sum of all visited trajectories in the node. This cumulative sum is then used to calculate the upper and lower bounds, which are shown in Lines 32-33. The term computed in Line 32 represents the loss of holding only a subset of the states in node ha from the set in node h, plus the loss of having only a partial set of posterior nodes and a subset of their states. $V_{\max,d}$ represents an upper bound for the value function. A simple bound on the value function can be $V_{\max,d} = \mathcal{R}_{\max} \cdot (D-d)$, but other more sophisticated bounds may also be used. In the experimental section we show that despite the additional overhead, utilizing the deterministic bounds, (20) and (19), within the actual decision-making improves the results of the respective algorithms.

Algorithm 1 ALGORITHM—$\mathcal{A}$.

```
    function SEARCH
 1: while time permits do
 2:    Generate states $x$ from $b_0$.
 3:    $\tau_0 \leftarrow x$
 4:    $\bar{\mathbb{P}}_0 \leftarrow b(x = \tau_0 \mid h_0)$
 5:    if $\tau_0 \notin \tau(h_0)$ then
 6:       $\bar{\mathbb{P}}(h_0) \leftarrow \bar{\mathbb{P}}(h_0) + \bar{\mathbb{P}}_0$
 7:    end if
 8:    SIMULATE($h_0, D, \tau_0, \bar{\mathbb{P}}_0$).
 9: end while
10: return
    function FWDUPDATE($ha, haz, \tau_d, \bar{\mathbb{P}}_\tau, x'$)
11: if $\tau_d \notin \tau(ha)$ then
12:    $\tau(ha) \leftarrow \tau(ha) \cup \{\tau_d\}$
13:    $\bar{R}(ha) \leftarrow \bar{R}(ha) + \bar{\mathbb{P}}_\tau \cdot r(x, a)$
14: end if
15: $\tau_d \leftarrow \tau_d \cup \{x'\}$
16: $\bar{\mathbb{P}}_\tau \leftarrow \bar{\mathbb{P}}_\tau \cdot Z_{z|x'} \cdot T_{x'|x,a}$
17: if $\tau_d \notin \tau(haz)$ then
18:    $\bar{\mathbb{P}}(haz) \leftarrow \bar{\mathbb{P}}(haz) + \bar{\mathbb{P}}_\tau$
19:    $\tau(haz) \leftarrow \tau(haz) \cup \{\tau_d\}$
20: end if
21: return
    function SIMULATE($h, d, \tau_d, \bar{\mathbb{P}}_d$)
22: if $d = 0$ then
23:    return
24: end if
25: Select action $a$.
26: Generate next states and observations, $x', z$.
27: $\tau_d,\ \bar{\mathbb{P}}_\tau \leftarrow$ FWDUPDATE($ha, haz, \tau_d, \bar{\mathbb{P}}_\tau, x'$)
28: Select next observation $z$.
29: SIMULATE($haz, d-1, \tau_d, \bar{\mathbb{P}}_\tau$)
30: BWDUPDATE($h, ha, d$)
31: return
    function BWDUPDATE($h, ha, d$)
32: $\epsilon(ha) = \gamma^{D-d} V_{\max,d}(\bar{\mathbb{P}}(h) - \bar{\mathbb{P}}(ha)) + \gamma^{D-d-1} \cdot V_{\max,d+1}(\bar{\mathbb{P}}(ha) - \sum_{z|ha} \bar{\mathbb{P}}(haz))$
33: $U(ha) = \bar{R}(ha) + \gamma \sum_{z|ha} U(haz) + \epsilon(ha)$
34: $L(ha) = \bar{R}(ha) + \gamma \sum_{z|ha} L(haz) - \epsilon(ha)$
35: $U(h) \leftarrow \max_{a'}\{U(ha')\}$
36: $L(h) \leftarrow \max_{a'}\{L(ha')\}$
37: return
```

6.1. DB-POMCP

DB-POMCP uses Theorem 2 for decision-making once an optimal action was found or at time-out given limited planning time. In aligning Algorithm 1 with the POMCP framework, the action exploration process determined by the Upper Confidence Bounds for Trees (UCT) criterion,

$$UCT(H_t, a_t) = \hat{Q}^{mean}(H_t, a_t) + c\sqrt{\frac{log(N(H_t))}{N(H_t, a_t)}}, \tag{33}$$

where $\hat{Q}^{mean}$ is the average of the cumulative sums obtained from sampled explorations, and c is a tunable constant that trades-off exploration and exploitation during planning. Following this criterion, each state and observation is then sampled according to their respective transition and observation models. The original POMCP method, as discussed in [1], employs Monte-Carlo rollouts for value estimation and refrains from adding new nodes during these rollouts. During our evaluations we saw a negligible difference in performance, thus we avoid presenting rollouts to Algorithm 1 for simplicity. However, DB-POMCP supports both settings.

6.2. RB-POMCP

Root-Bounded POMCP (RB-POMCP) differs from DB-POMCP in that it uses a different exploration method. We denote it RB-POMCP to emphasize that the bounds hold only in the root node, and are not valid for any node along the tree, yet unlike DB-POMCP the bounds are used for exploration in any part of the tree. The RB-POMCP methodology draws inspiration from the Monte-Carlo approach suggested the original POMCP algorithm and innovates by incorporating upper and lower bounds, as defined in (24) and (25), to guide both the exploration and the decision-making processes.

The RB-POMCP framework is constructed based on the structure outlined in Algorithm 1, which necessitates specific implementations for abstract state, action, and observation exploration functions. In our approach, we opt for an approximation to the exploration mechanism proposed in Section 5.2.4. More precisely, while we adhere to the action exploration strategy described in the lemma, we simplify the observation and state exploration components by employing basic Monte-Carlo sampling techniques, akin to those used in the standard POMCP algorithm. This modification is intended to enhance the algorithm's planning efficiency without compromising the integrity of the algorithm bounds. The remainder of the RB-POMCP algorithm adheres closely to the procedures specified in Algorithm 1. Additionally, we use pruning and stopping criteria, as described in Section 5.2.3.

6.3. *Time complexity*

The cost of updating a posterior node depends on the underlying solver. For dynamic programming methods, such as DESPOT and AdaOPS, the extra bookkeeping required by our bounds is negligible, so the original time complexity is essentially unchanged. For Monte Carlo methods, such as POMCP, the baseline complexity is $O(|\mathcal{A}|)$ related mainly to the action-selection; A naive implementation of our bounds adds another linear complexity term, making it $O(|\mathcal{A}| + |\bar{\mathcal{Z}}|)$ due to the summation over the simplified observation space described in Corollary 1.1 and Algorithm 1, Line 32 of BWDUPDATE function.

However, we show that our added bounds can still be updated in $O(|\mathcal{A}|)$ by storing two additional scalar bookkeeping values for each node. Namely, incremental visitation probability, $\Delta\mathbb{P}^{(i)}(\cdot)$, and the change to child's upper bound, $\Delta\mathcal{U}^{(i)}(\cdot)$.

Each POMCP visit touches a single observation child, so only that branch must be updated. The bounds are therefore updated incrementally,

$$\mathcal{U}_0^{(i)}(H_t, a_t) = \mathcal{U}_0^{(i-1)}(H_t, a_t) + \Delta\overline{\mathbb{P}}^{(i)}\left(\tau_t^{(i)}\right)\left[r\left(x_t^{(i)}, a_t\right) + V_{max,t}\right] \\ + \Delta\mathcal{U}_0^{(i)}\left(H_{t+1}^{(i)}\right) - \Delta\overline{\mathbb{P}}^{(i)}\left(\tau_{t+1}^{(i)}\right)V_{max,t}$$

and,

$$\mathcal{U}_0^{(i)}(H_T) = \max_{a_t \in \mathcal{A}}\left\{\mathcal{U}_0^{(i)}(H_t, a_t)\right\}. \tag{34}$$

Because only one branch is modified, the per-visit overhead remains $O(|\mathcal{A}|)$. The full algebraic derivation is provided in Section A.1.7.

During each visitation to a node, the trajectory bookkeeping $\tau_d \in \tau(ha)$ shown in Algorithm 1 Line 11, is used to determine whether a specific trajectory has already been encountered at the current node. This verification process can potentially result in an added linear complexity of $O(D)$, where D represents the planning horizon. However, this overhead can be circumvented by assigning a unique ID value to each trajectory at the previous step and subsequently checking whether a pair, comprising the ID value and the new state, has already been visited. This approach reduces the overhead to an average time complexity of $O(1)$ by utilizing hash maps efficiently.

While the asymptotic time complexity remains similar, the deterministic certificates add a constant-factor bookkeeping cost per node, due to hash lookups and updates of bounds ($O(|A|)$ worst-case plus $O(1)$ on average for Algorithm 1). Practical wall-clock effects depend on implementation details and engineering optimizations, so we confine this section to the algorithmic analysis; systematic profiling is orthogonal to our contribution and left to future work.

7. Experiments

Our primary contribution is of a theoretical nature, yet we conducted experiments to evaluate the practical applicability of our proposed methodologies. Initially, we adopted a hybrid strategy, such as DB-POMCP, by incorporating our deterministic bounds exclusively for the decision-making, while relying on existing exploration strategies such as POMCP and DESPOT. Essentially, this approach enhances the POMCP and DESPOT frameworks by equipping them with mechanisms that ensure bounded sub-optimality. In a subsequent experimental setup, we applied the deterministic bounds to both the exploration and decision-making phases, based on the methodologies outlined in Section 6.2. We then compared the empirical performance of using the deterministic bounds solely for decision-making to the baseline algorithms without the incorporation of any deterministic bounds. Our findings indicate that while the application of deterministic bounds to decision-making can enhance performance, this strategy becomes less effective in identifying the optimal action as the complexity of the POMDP increases. Conversely, when the deterministic bounds are applied to both exploration and decision-making (Section 6.2), the results demonstrate a linear increase in planning time proportional to the size of the POMDP, indicating better scalability.

7.1. *Deterministic-bounds for decision-making*

In this subsection, we focus on the application of deterministic bounds exclusively for decision-making. This approach involves using a predefined exploration strategy during the planning phase, but making the final action selection based on the deterministic bounds as shown in (24). The comparative results for the standard and deterministically-bounded versions of the POMCP and DESPOT algorithms are presented in Table 1. These versions, labeled DB-POMCP and DB-DESPOT, adhere to the original exploration criteria of their respective algorithms but select actions based on the highest lower bound, as specified in (20).

Our experimental analysis reveals that, in addition to offering a level of optimality certification for the chosen actions, utilizing deterministic bounds for action selection can enhance the expected cumulative reward. It is important to note, however, that this method does not always lead to better outcomes. Specifically, it may not be advantageous in situations where the highest lower bound is less than other available upper bounds (for instance, comparing actions a^1 and a^3 in Fig. 2). In practice, when the bounds overlap, using them for action selection provides only heuristic guidance and may be incorrect. In our experiments, we used uniform bounds accross all future belief states, which are not useful for guiding the policy towards better actions. In fact, the more symmetric the bounds are at different nodes, the more belief nodes the algorithm will have to visit to distinguish between the attractiveness of different actions. While this can be improved by considering informed bounds, e.g. MDP-optimal for upper bounds or rollout-based lower bounds, we leave it for future work. When using uninformed bounds, a large POMDP with large state, observation or action spaces, leads to more loose bounds, which in turn reduces the effectiveness of our approach. This limitation is evident in the results for the Laser Tag POMDP, a considerably larger problem compared to the other POMDPs evaluated, where the deterministic bounds did not yield performance improvements.

7.2. *Root-bounds for decision-making and exploration*

The performance outcomes presented in Table 2 reveal that the RB-POMCP algorithm typically matches or surpasses the standard POMCP in various tested environments, except for the Navigate to Goal POMDP scenario. The limited performance in this particular context can be attributed to the nature of RB-POMCP's exploration strategy, which is designed to assure optimality over extended planning periods but does not inherently guarantee enhanced results within limited planning durations. Unlike probabilistic algorithms that leverage statistical concentration inequalities-such as the Hoeffding inequality employed in the Upper Confidence Bounds for Trees (UCT) [14] exploration mechanism of POMCP-RB-POMCP adopts a more cautious strategy. This approach

Table 1
Performance comparison with and without deterministic bounds, for short horizon, $H = 5$.

 Cardinalities ($\|S\|, \|A\|, \|O\|$)	Tiger POMDP (2, 3, 2)	Laser Tag (5930, 5, $\sim 1.5\times10^6$)	Discrete Light Dark (122, 5, 21)	Baby POMDP (2, 2, 2)
DB-DESPOT (ours)	3.74 ± 0.48	−5.29 ± 0.14	−5.29 ± 0.01	−3.92 ± 0.56
AR-DESPOT	2.82 ± 0.55	−5.10 ± 0.14	−61.53 ± 5.80	−5.40 ± 0.85
DB-POMCP (ours)	3.01 ± 0.21	−3.97 ± 0.24	−3.70 ± 0.82	−4.48 ± 0.57
POMCP	2.18 ± 0.76	−3.92 ± 0.27	−4.51 ± 1.15	−5.39 ± 0.63

Table 2
Performance comparison with and without deterministic bounds, for medium horizon, $H = 15$.

Algorithm *Cardinalities* ($\|S\|, \|A\|, \|O\|$)	Tiger POMDP (2, 3, 2)	Rock Sample (1801, 8, 3)	Navigate to Goal (25, 5, 25)	Baby POMDP (2, 2, 2)
RB-POMCP (ours)	1.53 ± 0.76	8.50± 0.22	61.21 ± 0.71	−11.97 ± 0.27
DB-POMCP (ours)	−1.05 ± 0.15	7.86 ± 0.21	62.37 ± 0.75	−12.13 ± 0.22
POMCP	−5.59 ± 0.24	5.69 ± 0.20	68.45 ± 0.69	−12.49 ± 0.27

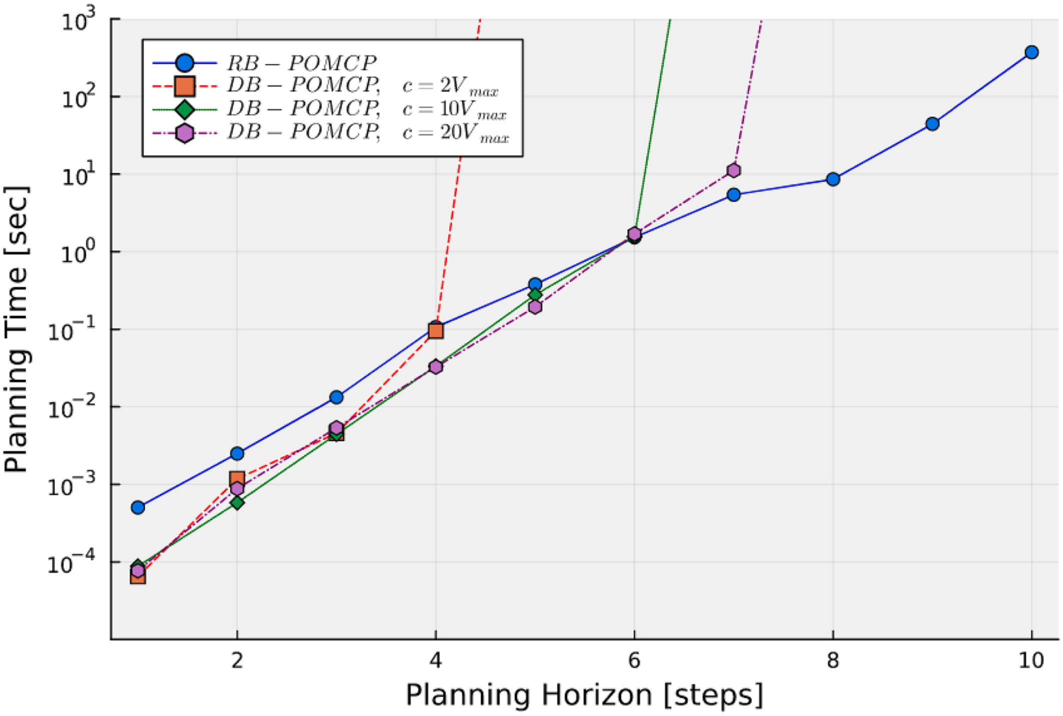


Fig. 3. The graphs show the measured planning time for RB-POMCP and DB-POMCP to find the optimal action for Rock Sample under different UCT coefficient values. Guaranteeing the optimal action made possible by using the bounds in Corollary 1.1. All simulation runs were capped at 3,600 s.

entails considering both worst-case and best-case scenarios to establish a deterministic link with the optimal value may not always translate to superior immediate performance due to its conservative nature.

7.3. Planning for optimal action

To highlight the differences between RB-POMCP and DB-POMCP, we examined each algorithm's planning time to deterministically identify the optimal value, as depicted in Fig. 3. Notably, conventional state-of-the-art algorithms, such as POMCP and DESPOT, cannot deterministically identify the optimal action within a finite timeframe and are thus not considered in this analysis.

DB-POMCP incorporates the Upper Confidence Bounds for Trees (UCT) method for exploration. However, its exploration strategy lacks awareness of the deterministic bounds of the optimal value function, leading to insufficient guidance toward actions that may be optimal. Despite significantly increasing the exploration coefficient beyond the values suggested in previous works [1,3], our findings, as presented in Fig. 3, demonstrate that the exploration bonus diminishes too rapidly, effectively limiting further exploration of potentially optimal actions. While UCT, in theory, explores the belief tree indefinitely, in practical scenarios, the exploration rate of new branches diminishes exponentially over time, making it less effective in environments where identifying the optimal action in a reasonable time is crucial. Conversely, RB-POMCP directly utilizes upper and lower bounds information, facilitating a more targeted search for the optimal value. This approach leads to a planning duration that scales linearly with the problem size, as evidenced in Fig. 3, highlighting its efficiency in identifying optimal actions within a finite timeframe.

7.4. Technical details

The implementation of our algorithm written in the Julia programming language, using the Julia POMDPs package for evaluation and the vanilla POMDP versions, provided by [15]. This package primarily supports infinite horizon problems; however, we modified it to also handle finite-horizon POMDPs. The experiments were conducted on a computing platform consisting of an Intel(R) Core(TM) i7-7700 processor with 8

CPUs operating at 3.60GHz and 15.6 GHz. The hyper-parameters for the POMCP and AR-DESPOT solvers, and further details about the POMDPs used for our experiments are detailed in the appendix.

8. Discussion and future work

The principal advantage of the proposed algorithm is its explicit and deterministic upper and lower bounds on the value function, quantifying the maximum deviation of the current policy from optimality. These bounds support a principled early termination by choosing a user-specified tolerance ϵ, guaranteeing an ϵ-optimal policy and pruning of provably irrelevant action branches - capabilities that, to our knowledge, existing online POMDP planners lack. Moreover, they allow one to utilize the bounds for deterministic limits on the probability of catastrophic outcomes, an essential feature for safety-critical tasks.

However, the present implementation has few shortcomings. First, while not limited by the theoretical derivations, our implementation considers naive bounds on the value function, that is, best-or-worst possible rewards at each unvisited history or state. This results in relatively loose bounds which hinders scalability to large POMDPs. Tighter alternatives, such as history-dependent relaxations or bounds derived from the MDP-optimal value function commonly used in other POMDP solvers would alleviate this issue. We believe that this opens a new avenue of research, that focuses on general, efficient value function bounds which are easily applicable to the present deterministic-bounds algorithm (see $V_{max,d}$ in ALGORITHM-$\mathcal{A}$ Line 32). Second, it adds implementation complexity, especially apparent compared simple algorithms like POMCP. Last, the bounds add bookkeeping overhead in both time and memory when attached to existing algorithms, although Section 6.3 shows that the asymptotic complexity can be reduced to match existing algorithms, and pruning suboptimal actions may further reduce the time efficiency.

9. Conclusions

This work addresses the computational challenges of decision-making under uncertainty, typically formalized as Partially Observable Markov Decision Processes (POMDPs). Our objective is to bridge the theoretical gap between the quality of solutions obtained from approximate solvers and the generally intractable optimal solutions. We present a novel methodology that guarantees anytime, deterministic bounds for approximate POMDP solvers. We achieve this by defining a simplified POMDP, that utilizes only a subset of the state and observation spaces to alleviate the computational burden. We establish a theoretical relationship between the optimal value function, which is computationally intensive, and a more tractable value function obtained using the simplified POMDP. Based on the theoretical derivation, we suggest the use of the deterministic bounds to govern the exploration, while being theoretically guaranteed to converge to the optimal value in finite time. Building upon this theoretical framework, we show how to integrate the bounds with a general structure of common state-of-the-art algorithms. Additionally, we leverage our deterministic bounds to develop an early stopping criterion that identifies convergence to the optimal value, a novel capability that is not possible with existing probabilistic bounds. We introduce two algorithms that incorporate the suggested bounds, named DB-POMCP and RB-POMCP. DB-POMCP exploits the deterministic relationship for decision-making, while RB-POMCP uses the bounds for both decision-making and exploration. Finally, we evaluate the practical use of our approach by comparing the suggested algorithms to state-of-the-art algorithms, demonstrating their effectiveness.

CRediT authorship contribution statement

Moran Barenboim: Writing – review & editing, Writing – original draft, Software, Methodology, Investigation, Formal analysis, Conceptualization; **Vadim Indelman:** Writing – review & editing, Validation, Supervision, Methodology, Investigation, Formal analysis.

Data availability

No data was used for the research described in the article.

Declaration of interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported by the Israel Science Foundation (ISF) and by US NSF/US-Israel BSF.

Appendix A.

A.1. Mathematical analysis

We start by restating the definition of the simplified value function,

$$\bar{V}^{\pi}(\bar{b}_t) \triangleq r(\bar{b}_t, \pi_t) + \bar{\mathbb{E}}\left[\bar{V}(b_t)\right] \tag{A.36}$$

$$= \sum_{x_t} \bar{b}(x_t) r(x_t, \pi_t) + \sum_{z_t} \bar{\mathbb{P}}(z_{t+1} \mid H^-_{t+1}) \bar{V}(\bar{b}(z_{t+1})), \tag{A.37}$$

A.1.1. Theorem 1

Let b_t belief state at time t, and T be the last time step of the POMDP. Let $V^{\pi}(b_t)$ be the theoretical value function by following a policy π, and let $\bar{V}^{\pi}(b_t)$ be the simplified value function, as defined in (7), by following the same policy. Then, for any policy π, the difference between the theoretical and simplified value functions is bounded as follows,

$$\left|V^{\pi}(b_t) - \bar{V}^{\pi}(b_t)\right| \\ \leq \mathcal{R}_{\max} \sum_{\tau=t+1}^{T} \left[1 - \sum_{z_{t+1:\tau}} \sum_{x_{t:\tau}} b(x_t) \prod_{k=t+1}^{\tau} \bar{\mathbb{P}}(z_k \mid x_k) \mathbb{P}(x_k \mid x_{k-1}, \pi_{k-1})\right] \triangleq \epsilon_z^{\pi}(b_t). \tag{A.38}$$

Proof. For notational convenience, we derive the bounds for the value function by denoting the prior belief as b_0,

$$V_0^{\pi}(b_0) = \mathbb{E}_{z_{1:T}}\left[\sum_{t=0}^{T} r(b_t, a_t)\right] \tag{A.39}$$

applying the belief update equation,

$$V_0^{\pi}(b_0) = \sum_{z_{1:T}} \prod_{\tau=1}^{T} \mathbb{P}\left(z_\tau \mid H^-_\tau\right) \\ \sum_{t=0}^{T} \left[\sum_{x_t} \frac{\mathbb{P}(z_t \mid x_t) \sum_{x_{t-1}} \mathbb{P}(x_t \mid x_{t-1}, \pi_{t-1}) b_{t-1}}{\mathbb{P}\left(z_t \mid H^-_t\right)} r(x_t, a_t)\right] \tag{A.40}$$

$$= \sum_{z_{1:T}} \prod_{\tau=1}^{T} \mathbb{P}\left(z_\tau \mid H^-_\tau\right) \\ \sum_{t=0}^{T} \left[\sum_{x_{0:t}} \frac{\prod_{k=1}^{t} \mathbb{P}(z_k \mid x_k) \mathbb{P}(x_k \mid x_{k-1}, \pi_{k-1}) b(x_0)}{\prod_{\tau=1}^{t} \mathbb{P}\left(z_\tau \mid H^-_\tau\right)} r(x_t, a_t)\right] \tag{A.41}$$

$$= \sum_{t=0}^{T} \sum_{z_{1:T}} \sum_{x_{0:T}} \prod_{k=1}^{t} \mathbb{P}(z_k \mid x_k) \mathbb{P}(x_k \mid x_{k-1}, \pi_{k-1}) b(x_0) r(x_t, a_t) \tag{A.42}$$

which applies similarly to the simplified value function,

$$\bar{V}_0^{\pi}(b_0) = \sum_{t=0}^{T} \sum_{z_{1:T}} \sum_{x_{0:T}} \prod_{k=1}^{t} \bar{\mathbb{P}}(z_k \mid x_k) \mathbb{P}(x_k \mid x_{k-1}, \pi_{k-1}) b(x_0) r(x_t, a_t). \tag{A.43}$$

We begin the derivation by focusing on a single time step, t, and later generalize to the complete value function.

$$|\mathbb{E}_{z_{1:t}}[r(b_t)] - \bar{\mathbb{E}}_{z_{1:t}}[r(\bar{b}_t)]| \tag{A.44}$$

$$=| \sum_{z_{1:t}} \sum_{x_{0:t}} [\prod_{k=1}^{t} \mathbb{P}(z_k \mid x_k)\mathbb{P}(x_k \mid x_{k-1}, \pi_{k-1}) b(x_0) r(x_t) \tag{A.45}$$

$$- \prod_{k'=1}^{t} \bar{\mathbb{P}}(z_{k'} \mid x_{k'})\mathbb{P}(x_{k'} \mid x_{k'-1}, \pi_{k'-1}) b(x_0) r(x_t)]|$$

$$\leq \sum_{z_{1:t}} \sum_{x_{0:t}} \left| r(x_t) \left[\prod_{k=1}^{t} \mathbb{P}(z_k \mid x_k)\mathbb{P}(x_k \mid x_{k-1}, \pi_{k-1}) b(x_0) \right.\right.$$

$$\left.\left. - \prod_{k'=1}^{t} b(x_0)\, \bar{\mathbb{P}}(z_{k'} \mid x_{k'})\mathbb{P}(x_{k'} \mid x_{k'-1}, \pi_{k'-1}) \right] \right| \tag{A.46}$$

$$= \sum_{z_{1:t}} \sum_{x_{0:t}} |r(x_t)| \left[\prod_{k=1}^{t} \mathbb{P}(z_k \mid x_k)\mathbb{P}(x_k \mid x_{k-1}, \pi_{k-1}) b(x_0) \right.$$

$$\left. - \prod_{k'=1}^{t} b(x_0)\, \bar{\mathbb{P}}(z_{k'} \mid x_{k'})\, \mathbb{P}(x_{k'} \mid x_{k'-1}, \pi_{k'-1}) \right] \tag{A.47}$$

where the second transition is due to triangle inequality, the third transition is equality by the construction, i.e. using the simplified observation models imply that the difference is nonnegative. We add and subtract, followed by rearranging terms,

$$= \sum_{z_{1:t}} \sum_{x_{0:t}} |r(x_t)| \left[\prod_{k=1}^{t} \mathbb{P}(z_k, x_k \mid x_{k-1}, \pi_{k-1}) b(x_0) \right. \tag{A.48}$$

$$- \prod_{k=1}^{t-1} b(x_0)\bar{\mathbb{P}}(z_k, x_k \mid x_{k-1}, \pi_{k-1})\mathbb{P}(z_t, x_t \mid x_{t-1}, \pi_{t-1})$$

$$+ \prod_{k=1}^{t-1} b(x_0)\bar{\mathbb{P}}(z_k, x_k \mid x_{k-1}, \pi_{k-1})\mathbb{P}(z_t, x_t \mid x_{t-1}, \pi_{t-1})$$

$$\left. - \prod_{k'=1}^{t} b(x_0)\bar{\mathbb{P}}(z_{k'}, x_{k'} \mid x_{k'-1}, \pi_{k'-1}) \right]$$

$$= \sum_{z_{1:t}} \sum_{x_{0:t}} |r(x_t)| \Big\{ \tag{A.49}$$

$$\mathbb{P}(z_t, x_t \mid x_{t-1}, \pi_{t-1}) \left[\prod_{k=1}^{t-1} \mathbb{P}(z_k, x_k \mid x_{k-1}, \pi_{k-1}) b(x_0) \right.$$

$$\left. - \prod_{k=1}^{t-1} b(x_0)\bar{\mathbb{P}}(z_k, x_k \mid x_{k-1}, \pi_{k-1}) \right]$$

$$+ \prod_{k=1}^{t-1} b(x_0)\bar{\mathbb{P}}(z_k, x_k \mid x_{k-1}, \pi_{k-1})[\mathbb{P}(z_t, x_t \mid x_{t-1}, \pi_{t-1})$$

$$- \ \bar{\mathbb{P}}(z_t, x_t \mid x_{t-1}, \pi_{t-1})] \Big\}$$

applying Holder’s inequality,

$$\leq \mathcal{R}_{\max} \sum_{z_{1:t}} \sum_{x_{0:t}} \mathbb{P}(z_t, x_t \mid x_{t-1}, \pi_{t-1}) \left[b(x_0) \prod_{k=1}^{t-1} \mathbb{P}(z_k, x_k \mid x_{k-1}, \pi_{k-1}) \right.$$

$$\left. - b(x_0) \prod_{k=1}^{t-1} \bar{\mathbb{P}}(z_k, x_k \mid x_{k-1}, \pi_{k-1}) \right] \tag{A.50}$$

$$+ \mathcal{R}_{\max} \sum_{z_{1:t}} \sum_{x_{0:t}} \prod_{k=1}^{t-1} \bar{\mathbb{P}}(z_k, x_k \mid x_{k-1}, \pi_{k-1}) b(x_0)[\mathbb{P}(z_t, x_t \mid x_{t-1}, \pi_{t-1})$$

$$- \bar{\mathbb{P}}(z_t, x_t \mid x_{t-1}, \pi_{t-1})]$$

$$= \mathcal{R}_{\max} \sum_{z_{1:t}} \sum_{x_{0:t}} \mathbb{P}(z_t, x_t \mid x_{t-1}, \pi_{t-1}) \cdot \tag{A.51}$$

$$\left[b(x_0) \prod_{k=1}^{t-1} \mathbb{P}(z_k, x_k \mid x_{k-1}, \pi_{k-1}) - b(x_0) \prod_{k=1}^{t-1} \bar{\mathbb{P}}(z_k, x_k \mid x_{k-1}, \pi_{k-1}) \right] + \mathcal{R}_{\max}\delta_t$$

$$= \mathcal{R}_{\max} \sum_{z_{1:t-1}} \sum_{x_{0:t-1}} \left[b(x_0) \prod_{k=1}^{t-1} \mathbb{P}(z_k, x_k \mid x_{k-1}, \pi_{k-1}) \right. \tag{A.52}$$

$$\left. - b(x_0) \prod_{k=1}^{t-1} \bar{\mathbb{P}}(z_k, x_k \mid x_{k-1}, \pi_{k-1}) \right] + \mathcal{R}_{\max}\delta_t,$$

following similar steps recursively,

$$= \ldots = \mathcal{R}_{\max} \sum_{\tau=1}^{t} \delta_{\tau}. \tag{A.53}$$

Finally, applying similar steps for every time step $t \in [1, T]$ results in,

$$\left| V^{\pi}(b_t) - \bar{V}^{\pi}(b_t) \right| \leq \mathcal{R}_{\max} \sum_{t=1}^{T} \sum_{\tau=1}^{t} \delta_{\tau} \tag{A.54}$$

where,

$$\delta_{\tau} = \sum_{z_{1:\tau}} \sum_{x_{0:\tau}} \prod_{k=1}^{\tau-1} \bar{\mathbb{P}}(z_k, x_k \mid x_{k-1}, \pi_{k-1}) b(x_0)[\mathbb{P}(z_{\tau}, x_{\tau} \mid x_{\tau-1}, \pi_{\tau-1})$$

$$- \ \bar{\mathbb{P}}(z_{\tau}, x_{\tau} \mid x_{\tau-1}, \pi_{\tau-1})]$$

$$= \sum_{z_{1:\tau-1}} \sum_{x_{0:\tau-1}} \prod_{k=1}^{\tau-1} \bar{\mathbb{P}}(z_k, x_k \mid x_{k-1}, \pi_{k-1}) b(x_0)[1$$

$$- \sum_{z_{\tau}} \sum_{x_{\tau}} \bar{\mathbb{P}}(z_{\tau}, x_{\tau} \mid x_{\tau-1}, \pi_{\tau-1})] \tag{A.55}$$

plugging the term in (A.55) to (A.54) and expanding the terms results in the desired bound,

$$\left| V^{\pi}(b_t) - \bar{V}^{\pi}(b_t) \right|$$

$$\leq \mathcal{R}_{\max} \sum_{\tau=t+1}^{T} \left[1 - \sum_{z_{t+1:\tau}} \sum_{x_{t:\tau}} b(x_t) \prod_{k=t+1}^{\tau} \bar{\mathbb{P}}(z_k \mid x_k)\mathbb{P}(x_k \mid x_{k-1}, \pi_{k-1}) \right] \tag{A.56}$$

which concludes our derivation. □

A.1.2. Lemma

Lemma 1 The optimal value function can be bounded as

$$V^{\pi*}(b_t) \leq \mathrm{UDB}^{\pi}(b_t), \tag{A.57}$$

where the policy π is determined according to Bellman optimality over the UDB, i.e.

$$\mathrm{UDB}^{\pi}(b_t) \triangleq \max_{a_t \in \mathcal{A}} [\bar{Q}^{\pi}(b_t, a_t) + \epsilon_z^{\pi}(b_t, a_t)] \tag{A.58}$$

$$= \max_{a_t \in \mathcal{A}} [r(b_t, a_t) + \bar{\mathbb{E}}_{z_{t+1} | b_t, a_t} [\bar{V}^{\pi}(b_{t+1})] + \epsilon_z^{\pi}(b_t, a_t)]. \tag{A.59}$$

Proof. In the following, we prove by induction that applying the Bellman optimality operator on upper bounds to the value function in finite-horizon POMDPs will result in an upper bound on the optimal value function. The notations are the same as the ones presented in the main body of the paper. We restate some of the definitions from the paper for convenience.

The policy $\pi_t(b_t)$ determined by applying Bellman optimality at belief b_t, i.e.,

$$\pi_t(b_t) = \arg\max_{a_t \in \mathcal{A}} [\bar{Q}^{\pi}(b_t, a_t) + \epsilon_z^{\pi}(b_t, a_t)]. \tag{A.60}$$

As it will be needed in the following proof, we also define the value of a belief which includes in its history at least one observation out of the simplified set, e.g. $H_t = \{a_0, z_1, \ldots, z_k \notin \bar{\mathcal{Z}}, \ldots, z_t\}$ as being equal to zero. Explicitly,

$$\bar{V}_t^{\pi}(\mathbb{P}(x_t \mid a_0, z_1, \ldots, z_k \notin \bar{\mathcal{Z}}, \ldots, z_t)) \equiv 0 \ \ \forall k \in [1, t]. \tag{A.61}$$

We also use the following simple bound,

$$V_{t,\max} \triangleq \mathcal{R}_{\max} \cdot (T - t - 1) \tag{A.62}$$

Base case $(t = T)$ - At the final time step T, for each belief we set the value function to be equal to the reward value at that belief state, b_T and taking the action that maximizes the immediate reward,

$$\mathrm{UDB}^{\pi}(b_T) = \max_{a_T} \{ r(b_T, a_T) + \epsilon_z(b_T, a_T) \} \equiv \arg\max_{a_T} \{ r(b_T, a_T) \} \tag{A.63}$$

which provides an upper bound for the optimal value function for the final time step, $V^{\star}_{T}(b_T) \leq \mathrm{UDB}^{\pi}(b_T)$.

Induction hypothesis - Assume that for a given time step, t, for all belief states the following holds,

$$V^{\star}_{t}(b_t) \leq \mathrm{UDB}^{\pi}(b_t). \tag{A.64}$$

Induction step - We will show that the hypothesis holds for time step $t-1$. By the induction hypothesis,

$$V^{\star}_{t}(b_t) \leq \mathrm{UDB}^{\pi}(b_t) \ \ \forall b_t, \tag{A.65}$$

thus,

$$Q^{\star}(b_{t-1}, a_{t-1}) = r(b_{t-1}, a_{t-1}) + \sum_{z_t \in \mathcal{Z}} \mathbb{P}\left(z_t \mid H_t^-\right) V^{\star}_{t}(b(z_t)) \tag{A.66}$$

$$\leq r(b_{t-1}, a_{t-1}) + \sum_{z_t \in \mathcal{Z}} \mathbb{P}\left(z_t \mid H_t^-\right) \mathrm{UDB}^{\pi}(b(z_t)) \tag{A.67}$$

$$= r(b_{t-1}, a_{t-1}) + \sum_{z_t \in \mathcal{Z}} \mathbb{P}\left(z_t \mid H_t^-\right) \left[\bar{V}^{\pi}_{t}(b_t) + \epsilon^{\pi}_{z}(b_t)\right]. \tag{A.68}$$

For the following transition, we make use of Lemma 2,

$$= r(b_{t-1}, a_{t-1}) + \bar{\mathbb{E}}_{z_t \mid b_{t-1}, a_{t-1}} \left[\bar{V}^{\pi}_{t}(b_t)\right] + \epsilon^{\pi}_{z}(b_{t-1}, a_{t-1}) \tag{A.69}$$

$$\equiv \mathrm{UDB}^{\pi}(b_{t-1}, a_{t-1}). \tag{A.70}$$

Therefore, under the induction hypothesis, $Q^{\star}_{t-1}(b_{t-1}, a_{t-1}) \leq \mathrm{UDB}^{\pi}(b_{t-1}, a_{t-1})$. Taking the maximum over all actions a_t,

$$\mathrm{UDB}^{\pi}(b_{t-1}) = \max_{a_{t-1} \in \mathcal{A}} \left\{\mathrm{UDB}^{\pi}(b_{t-1}, a_{t-1})\right\} \tag{A.71}$$

$$\geq \max_{a_{t-1} \in \mathcal{A}} \left\{Q^{\star}_{t-1}(b_{t-1}, a_{t-1})\right\} = V^{\star}_{t-1}(b_{t-1}),$$

which completes the induction step and the required proof. □

Lemma 2. *Let b_t denote a belief state and π_t a policy at time t. Let $\bar{\mathbb{P}}(z_t \mid x_t)$ be the simplified observation model which represents the likelihood of observing z_t given x_t. Then, the following terms are equivalent,*

$$\mathbb{E}_{z_t}\left[\bar{V}^{\pi}_{t}(b_t) + \epsilon^{\pi}_{z}(b_t)\right] = \bar{\mathbb{E}}_{z_t}\left[\bar{V}^{\pi}_{t}(b_t)\right] + \epsilon^{\pi}_{z}(b_{t-1}, a_{t-1}) \tag{A.72}$$

Proof.

$$\mathbb{E}_{z_t}\left[\bar{V}^{\pi}_{t}(b_t) + \epsilon^{\pi}_{z}(b_t)\right] = \tag{A.73}$$

$$\mathbb{E}_{z_t}\left[\bar{V}^{\pi}_{t}(b_t)\right] + \mathbb{E}_{z_t}\left[\mathcal{R}_{\max} \sum_{\tau=t+1}^{T} \left[1 - \sum_{z_{t+1:\tau}} \sum_{x_{t:\tau}} b_t \prod_{k=t+1}^{\tau} \bar{\mathbb{P}}(z_k \mid x_k) \right.\right.$$

$$\left.\left. \mathbb{P}(x_k \mid x_{k-1}, \pi_{k-1})\right]\right] \tag{A.74}$$

focusing on the second summand,

$$\sum_{z_t \in \mathcal{Z}} \mathbb{P}\left(z_t \mid H_t^-\right) \mathcal{R}_{\max} \sum_{\tau=t+1}^{T} \left[1 - \sum_{z_{t+1:\tau}} \sum_{x_{t:\tau}} b_t \prod_{k=t+1}^{\tau} \bar{\mathbb{P}}(z_k \mid x_k) \mathbb{P}(x_k \mid x_{k-1}, \pi_{k-1})\right] \tag{A.75}$$

$$= \mathcal{R}_{\max} \sum_{\tau=t+1}^{T} \left[1 - \sum_{z_t} \mathbb{P}\left(z_t \mid H_t^-\right) \sum_{z_{t+1:\tau}} \sum_{x_{t:\tau}} b(x_t) \right.$$

$$\left. \prod_{k=t+1}^{\tau} \bar{\mathbb{P}}(z_k \mid x_k) \mathbb{P}(x_k \mid x_{k-1}, \pi_{k-1})\right] \tag{A.76}$$

by marginalizing over x_{t-1},

$$= \mathcal{R}_{\max} \sum_{\tau=t+1}^{T} [1 - \sum_{z_t} \mathbb{P}\left(z_t \mid H_t^-\right) \sum_{z_{t+1:\tau}} \sum_{x_{t-1:\tau}} \frac{\bar{\mathbb{P}}(z_t \mid x_t) \mathbb{P}(x_t \mid x_{t-1}, \pi_{t-1}) b(x_{t-1})}{\mathbb{P}\left(z_t \middle| H_t^-\right)} \cdot \tag{A.77}$$

$$\prod_{k=t+1}^{\tau} \bar{\mathbb{P}}(z_k \mid x_k) \mathbb{P}(x_k \mid x_{k-1}, \pi_{k-1})]$$

canceling out the denominator,

$$= \mathcal{R}_{\max} \sum_{\tau=t+1}^{T} [1 - \sum_{z_{t:\tau}} \sum_{x_{t-1:\tau}} \bar{\mathbb{P}}(z_t \mid x_t) \mathbb{P}(x_t \mid x_{t-1}, a_{t-1}) b(x_{t-1}) \cdot \tag{A.78}$$

$$\prod_{k=t+1}^{\tau} \bar{\mathbb{P}}(z_k \mid x_k) \mathbb{P}(x_k \mid x_{k-1}, \pi_{k-1})] \equiv \epsilon^{\pi}_{z}(b_{t-1}, a_{t-1})$$

it is left to show that $\mathbb{E}_{z_t \mid b_{t-1}, a_{t-1}}\left[\bar{V}^{\pi}_{t}(b_t)\right] = \bar{\mathbb{E}}_{z_t \mid b_{t-1}, a_{t-1}}\left[\bar{V}^{\pi}_{t}(b_t)\right]$. By the definition of a value function of a belief not included in the simplified set, we have that,

$$\mathbb{E}_{z_t \mid b_{t-1}, a_{t-1}}\left[\bar{V}^{\pi}_{t}(b_t)\right] = \sum_{z_t \in \mathcal{Z}} \mathbb{P}\left(z_t \mid H_t^-\right) \bar{V}^{\pi}_{t}(b_t) \tag{A.79}$$

$$= \sum_{z_t \in \bar{\mathcal{Z}}} \mathbb{P}\left(z_t \mid H_t^-\right) \bar{V}^{\pi}_{t}(b_t) + \sum_{z_t \in \mathcal{Z} \setminus \bar{\mathcal{Z}}} \mathbb{P}\left(z_t \mid H_t^-\right) \bar{V}^{\pi}_{t}(b_t) \tag{A.80}$$

$$= \sum_{z_t \in \bar{\mathcal{Z}}} \bar{\mathbb{P}}\left(z_t \mid H_t^-\right) \cdot \bar{V}^{\pi}_{t}(b_t) + \sum_{z_t \in \mathcal{Z} \setminus \bar{\mathcal{Z}}} \mathbb{P}\left(z_t \mid H_t^-\right) \cdot 0 \tag{A.81}$$

$$= \bar{\mathbb{E}}_{z_t \mid b_{t-1}, a_{t-1}}\left[\bar{V}^{\pi}_{t}(b_t)\right], \tag{A.82}$$

which concludes the derivation. □

A.1.3. Lemma

Corollary 1.1 We restate the definition of UDB exploration criteria,

$$a_t = \arg\max_{a_t \in \mathcal{A}} [\mathrm{UDB}^{\pi}(b_t, a_t)] = \arg\max_{a_t \in \mathcal{A}} [\bar{Q}^{\pi}(b_t, a_t) + \epsilon^{\pi}_{z}(b_t, a_t)]. \tag{A.83}$$

Corollary 2.3. *Using Lemma 1 and the exploration criteria defined in (17) guarantees convergence to the optimal value function.*

Proof. Let us define a sequence of bounds, $\mathrm{UDB}^{\pi}_{n}(b_t)$ and a corresponding difference value between UDB_n and the simplified value function,

$$\mathrm{UDB}^{\pi}_{n}(b_t) - \bar{V}^{\pi}_{n}(b_t) = \epsilon^{\pi}_{n,z}(b_t), \tag{A.84}$$

where $n \in [0, |\mathcal{Z}|]$ corresponds to the number of unique observation instances within the simplified observation set, $\bar{\mathcal{Z}}_n$, and $|\mathcal{Z}|$ denotes the cardinality of the complete observation space. Additionally, for the clarity of the proof and notations, assume that by construction the simplified set is chosen such that $\bar{\mathcal{Z}}_n(H_t) \equiv \bar{\mathcal{Z}}_n$ remains identical for all time steps t and history sequences, H_t given n. By the definition of $\epsilon^{\pi}_{n,z}(b_t)$,

$$\epsilon^{\pi}_{n,z}(b_t) = \mathcal{R}_{\max} \sum_{\tau=t+1}^{T} \left[1 - \sum_{z_{t+1:\tau} \in \bar{\mathcal{Z}}_n} \sum_{x_{t:\tau}} b(x_t) \prod_{k=t+1}^{\tau} \bar{\mathbb{P}}(z_k \mid x_k) \mathbb{P}(x_k \mid x_{k-1}, \pi_{k-1})\right], \tag{A.85}$$

we have that $\epsilon^{\pi}_{n,z}(b_t) \rightarrow 0$ as $n \rightarrow |\mathcal{Z}|$, since

$$\sum_{z_{t+1:\tau} \in \bar{\mathcal{Z}}_n} \sum_{x_{t:\tau}} b(x_t) \prod_{k=t+1}^{\tau} \bar{\mathbb{P}}(z_k \mid x_k) \mathbb{P}(x_k \mid x_{k-1}, \pi_{k-1}) \rightarrow 1 \tag{A.86}$$

as more unique observation elements are added to the simplified observation space, $\bar{\mathcal{Z}}_n$, eventually recovering the entire support of the discrete observation distribution.

From Lemma 1 we have that, for all $n \in [0, |\mathcal{Z}|]$ the following holds,

$$V^{\pi *}(b_t) \leq \mathrm{UDB}^{\pi}_{n}(b_t) = \bar{V}^{\pi}_{n}(b_t) + \epsilon^{\pi}_{n,z}(b_t). \tag{A.87}$$

Additionally, from Theorem 1 we have that,

$$\left|V^{\pi}(b_t) - \bar{V}^{\pi}_{n}(b_t)\right| \leq \epsilon^{\pi}_{n,z}(b_t), \tag{A.88}$$

for any policy π and subset $\bar{\mathcal{Z}}_n \subseteq \mathcal{Z}$, thus,

$$\bar{V}^{\pi}_{n}(b_t) - \epsilon^{\pi}_{n,z}(b_t) \leq V^{\pi}(b_t) \leq V^{\pi *}(b_t) \leq \bar{V}^{\pi}_{n}(b_t) + \epsilon^{\pi}_{n,z}(b_t). \tag{A.89}$$

Since $\epsilon^{\pi}_{n,z}(b_t) \rightarrow 0$ as $n \rightarrow |\mathcal{Z}|$, and $|\mathcal{Z}|$ is finite, it is guaranteed that $\mathrm{UDB}^{\pi}_{n}(b_t) \xrightarrow{n \rightarrow |\mathcal{Z}|} V^{\pi *}(b_t)$ which completes our proof. □

Moreover, depending on the algorithm implementation, the number of iterations can be finite (e.g. by directly choosing actions and observations to minimize the bound). A stopping criteria can also be verified by calculating the difference between the upper and lower bounds. The optimal solution is obtained once the upper bound equals the lower bound.

A.1.4. Theorem 2

Let b_t belief state at time t, and T be the last time step of the POMDP. Let $V^\pi(b_t)$ be the theoretical value function by following a policy π, and let $\bar{V}^\pi(b_t)$ be the simplified value function, as defined in (7), by following the same policy. Then, for any policy π, the difference between the theoretical and simplified value functions is bounded as follows,

$$\left|V^\pi(b_t) - \bar{V}^\pi(b_t)\right|$$

$$\leq \mathcal{R}_{\max} \sum_{\tau=t+1}^{T} \left[1 - \sum_{z_{t+1:\tau}} \sum_{x_{t:\tau}} b(x_t) \prod_{k=t+1}^{\tau} \bar{\mathbb{P}}(z_k \mid x_k) \mathbb{P}(x_k \mid x_{k-1}, \pi_{k-1})\right] \triangleq \epsilon^\pi(b_t). \tag{A.90}$$

Proof. Recall that we define $\tau_t = \{x_0, a_0, z_1, x_1, a_1, \dots, a_{T-1}, x_t, z_t\}$. Then the value function is defined as,

$$V^\pi(b_0) = \sum_{\tau_T} \mathbb{P}^\pi(\tau_T) \left[\sum_{t=0}^{T} r(x_t, a_t)\right] \tag{A.91}$$

applying chain rule and rearranging terms,

$$= \sum_{\tau_T} \mathbb{P}^\pi(x_{1:T}, z_{1:T}, a_{1:T} \mid \tau_0) \mathbb{P}^\pi(\tau_0) \left[\sum_{t=0}^{T} r(x_t, a_t)\right] \tag{A.92}$$

$$= \sum_{\tau_0} \mathbb{P}^\pi(\tau_0) \sum_{x_{1:T}, z_{1:T}, a_{1:T}} \mathbb{P}^\pi(x_{1:T}, z_{1:T}, a_{1:T} \mid \tau_0) \left[\sum_{t=0}^{T} r(x_t, a_t)\right] \tag{A.93}$$

$$= \sum_{\tau_0} \mathbb{P}^\pi(\tau_0) \Bigg[r(x_0, a_0)$$

$$+ \sum_{x_{1:T}, z_{1:T}, a_{1:T}} \mathbb{P}^\pi(x_{1:T}, z_{1:T}, a_{1:T} \mid \tau_0) \left[\sum_{t=1}^{T} r(x_t, a_t)\right] \Bigg] \tag{A.94}$$

nullifying instances of the complete probability distribution, $\mathbb{P}^\pi(\cdot)$, is denoted as a simplified distribution, $\bar{\mathbb{P}}^\pi(\cdot)$. We can then split and bound from above the value function, such that the simplified value function consideres only a subset of the trajectories at time $t = 0$,

$$\leq \sum_{\tau_0} \bar{\mathbb{P}}^\pi(\tau_0) \Bigg[r(x_0, a_0) + \sum_{x_{1:T}, z_{1:T}, a_{1:T}}$$

$$\mathbb{P}^\pi(x_{1:T}, z_{1:T}, a_{1:T} \mid \tau_0) \left[\sum_{t=1}^{T} r(x_t, a_t)\right] \Bigg] \tag{A.95}$$

$$+ \left[1 - \sum_{\tau_0} \bar{\mathbb{P}}^\pi(\tau_0)\right] \mathcal{V}_{max,0} \tag{A.96}$$

We then apply similar steps on the next time step, $t = 1$,

$$= \sum_{\tau_0} \bar{\mathbb{P}}^\pi(\tau_0) \Bigg[r(x_0, a_0) + \sum_{x_{1:T}, z_{1:T}, a_{1:T}} \mathbb{P}^\pi(x_{2:T}, z_{2:T}, a_{2:T} \mid \tau_1)$$

$$\mathbb{P}^\pi(x_1, z_1, a_1 \mid \tau_0) \left[\sum_{t=1}^{T} r(x_t, a_t)\right] \Bigg] \tag{A.97}$$

$$+ \left[1 - \sum_{\tau_0} \bar{\mathbb{P}}^\pi(\tau_0)\right] \mathcal{V}_{max,0} \tag{A.98}$$

$$= \sum_{\tau_0} \bar{\mathbb{P}}^\pi(\tau_0) \Bigg[r(x_0, a_0) \tag{A.99}$$

$$+ \sum_{x_1, z_1, a_1} \mathbb{P}^\pi(x_1, z_1, a_1 \mid \tau_0) \sum_{x_{2:T}, z_{2:T}, a_{2:T}}$$

$$\mathbb{P}^\pi(x_{2:T}, z_{2:T}, a_{2:T} \mid \tau_1) \left[\sum_{t=1}^{T} r(x_t, a_t)\right] \Bigg]$$

$$+ \left[1 - \sum_{\tau_0} \bar{\mathbb{P}}^\pi(\tau_0)\right] \mathcal{V}_{max,0} \tag{A.100}$$

$$= \sum_{\tau_0} \bar{\mathbb{P}}^\pi(\tau_0) \Bigg[r(x_0, a_0) \tag{A.101}$$

$$+ \sum_{x_1, z_1, a_1} \mathbb{P}^\pi(x_1, z_1, a_1 \mid \tau_0) \Bigg[r(x_1, a_1)$$

$$+ \sum_{x_{2:T}, z_{2:T}, a_{2:T}} \mathbb{P}^\pi(x_{2:T}, z_{2:T}, a_{2:T} \mid \tau_1) \left[\sum_{t=2}^{T} r(x_t, a_t)\right] \Bigg] \Bigg]$$

$$+ \left[1 - \sum_{\tau_0} \bar{\mathbb{P}}^\pi(\tau_0)\right] \mathcal{V}_{max,0}$$

$$\leq \sum_{\tau_0} \bar{\mathbb{P}}^\pi(\tau_0) \Bigg[r(x_0, a_0) \tag{A.102}$$

$$+ \sum_{x_1, z_1, a_1} \bar{\mathbb{P}}^\pi(x_1, z_1, a_1 \mid \tau_0) \Bigg[r(x_1, a_1)$$

$$+ \sum_{x_{2:T}, z_{2:T}, a_{2:T}} \mathbb{P}^\pi(x_{2:T}, z_{2:T}, a_{2:T} \mid \tau_1) \left[\sum_{t=2}^{T} r(x_t, a_t)\right] \Bigg] \Bigg]$$

$$+ \sum_{\tau_0} \bar{\mathbb{P}}^\pi(\tau_0) \left[1 - \sum_{x_1, z_1, a_1} \bar{\mathbb{P}}^\pi(x_1, z_1, a_1 \mid \tau_0)\right] \mathcal{V}_{max,1} + \left[1 - \sum_{\tau_0} \bar{\mathbb{P}}^\pi(\tau_0)\right] \mathcal{V}_{max,0} \tag{A.103}$$

which results in,

$$= \sum_{\tau_0} \bar{\mathbb{P}}^\pi(\tau_0) \Bigg[r(x_0, a_0) \tag{A.104}$$

$$+ \sum_{x_1, z_1, a_1} \bar{\mathbb{P}}^\pi(x_1, z_1, a_1 \mid x_0, a_0) \Bigg[r(x_1, a_1) + \sum_{x_{2:T}, z_{2:T}, a_{2:T}}$$

$$\mathbb{P}^\pi(x_{2:T}, z_{2:T}, a_{2:T} \mid \tau_1) \left[\sum_{t=2}^{T} r(x_t, a_t)\right] \Bigg] \Bigg]$$

$$+ \left[\sum_{\tau_0} \bar{\mathbb{P}}^\pi(\tau_0) - \sum_{\tau_1} \bar{\mathbb{P}}^\pi(\tau_1)\right] \mathcal{V}_{max,1} + \left[1 - \sum_{\tau_0} \bar{\mathbb{P}}^\pi(\tau_0)\right] \mathcal{V}_{max,0} \tag{A.105}$$

Performing the same steps iteratively up to time $t = T$, yields the desired outcome,

$$V^\pi(b_0) \leq \sum_{t=0}^{T} \sum_{\tau_t} \bar{\mathbb{P}}^\pi(\tau_t) r(x_t, a_t) + \mathcal{V}_{max,0} \left[1 - \sum_{\tau_0} \bar{\mathbb{P}}^\pi(\tau_0)\right]$$

$$+ \sum_{t=0}^{T-1} \mathcal{V}_{max,t+1} \left[\sum_{\tau_t} \bar{\mathbb{P}}^\pi(\tau_t) - \sum_{\tau_{t+1}} \bar{\mathbb{P}}^\pi(\tau_{t+1})\right] \tag{A.106}$$

□

A.1.5. Lemma

Lemma 1.1 Let $\mathcal{A}$ be the set of actions and $\mathcal{U}_0^\star(H_t)$, $\mathcal{L}_0^\star(H_t)$ be the upper and lower bounds of node H_t chosen according to,

$$\mathcal{U}_0^\star(H_t) \triangleq \sum_{\tau_t \in \mathcal{T}(H_t)} \bar{\mathbb{P}}(\tau_t) \big[r(x_t, a_t) + \mathcal{V}_{\max,t}\big]$$

$$+ \sum_{z_{t+1} \in \bar{\mathcal{Z}}(H_t, a_t)} \left[\mathcal{U}_0^\star(H_{t+1}) - \sum_{\tau_{t+1} \in \mathcal{T}(H_{t+1})} \bar{\mathbb{P}}(\tau_{t+1}) \mathcal{V}_{\max,t}\right] \tag{A.107}$$

$$\mathcal{L}_0^\star(H_t) \triangleq \sum_{\tau_t \in \mathcal{T}(H_t)} \bar{\mathbb{P}}(\tau_t) \big[r(x_t, a_t) + \mathcal{V}_{\min,t}\big]$$

$$+ \sum_{z_{t+1} \in \bar{\mathcal{Z}}(H_t, a_t)} \left[\mathcal{L}_0^\star(H_{t+1}) - \sum_{\tau_{t+1} \in \mathcal{T}(H_{t+1})} \bar{\mathbb{P}}(\tau_{t+1}) \mathcal{V}_{\min,t}\right] \tag{A.108}$$

and,

$$\mathcal{U}_0^\star(H_T) \triangleq \sum_{\tau_T \in \mathcal{T}(H_T)} \bar{\mathbb{P}}(\tau_T) r(x_T), \qquad \mathcal{L}_0^\star(H_T) \triangleq \sum_{\tau_T \in \mathcal{T}(H_T)} \bar{\mathbb{P}}(\tau_T) r(x_T). \tag{A.109}$$

Then, the optimal root-value is bounded by,

$$\mathcal{L}_0^\star(H_0) \leq V^{\pi^*}(H_0) \leq \mathcal{U}_0^\star(H_0). \tag{A.110}$$

Proof. We wish to show that $\mathcal{L}_0^\star(H_0) \leq V^{\pi^*}(b_0) \leq \mathcal{U}_0^\star(H_0)$. We derive a proof for one side of the inequality, while the other follows similarly. First note that,

$$V^{\pi^*}(b_0) \leq \mathcal{U}_0^{\pi^*}(H_0) \leq \max_{\pi \in \Pi} \mathcal{U}_0^\pi(H_0) \tag{A.111}$$

where the first inequality is due to Theorem 2, and the second inequality is true by definition. However, the claim in Corollary 1.1 is a recursive claim, while the bound provided in Theorem 2 only holds with respect to the root. Thus, for completeness, we also need to show that the best action can be chosen recursively, even though the bound is ‘partial‘ in different parts of the tree.

$$\begin{aligned}
&\max_{\pi_{0:T} \in \Pi} \mathcal{U}_0^\pi(H_0) \\
&= \max_{\pi_{0:T} \in \Pi} \sum_{\tau_0 \in \mathcal{T}(H_0)} \overline{\mathbb{P}}(\tau_0)[r(x_0, \pi_0) \\
&\quad + \mathcal{V}_{max,0}] + \sum_{z_1 \in \overline{\mathcal{Z}}(H_0,\pi_0)} \left[\mathcal{U}_0^\pi(H_1) - \sum_{\tau_1 \in \mathcal{T}(H_1)} \overline{\mathbb{P}}(\tau_1) \mathcal{V}_{max,0} \right] \\
&= \max_{\pi_0 \in \Pi} \left\{ \sum_{\tau_0 \in \mathcal{T}(H_0)} \overline{\mathbb{P}}(\tau_0)[r(x_0, \pi_0) + \mathcal{V}_{max,0}] \right. \\
&\quad \left. + \max_{\pi_{1:T} \in \Pi} \sum_{z_1 \in \overline{\mathcal{Z}}(H_0,\pi_0)} \left[\mathcal{U}_0^\pi(H_1) - \sum_{\tau_1 \in \mathcal{T}(H_1)} \overline{\mathbb{P}}(\tau_1) \mathcal{V}_{max,0} \right] \right\} \\
&= \max_{a_0} \left\{ \sum_{\tau_0 \in \mathcal{T}(H_0)} \overline{\mathbb{P}}(\tau_0)[r(x_0, a_0) + \mathcal{V}_{max,0}] \right. \\
&\quad \left. + \sum_{z_1 \in \overline{\mathcal{Z}}(H_0,a_0)} \left[\max_{\pi_{1:T} \in \Pi} \mathcal{U}_0^\pi(H_1) - \sum_{\tau_1 \in \mathcal{T}(H_1)} \overline{\mathbb{P}}(\tau_1) \mathcal{V}_{max,0} \right] \right\}
\end{aligned}$$

which continues similarly up to time $t = T$, which completes the proof,

$$V^{\pi^*}(b_0) \leq \mathcal{U}_0^{\pi^*}(H_0) \leq \max_{\pi \in \Pi} \mathcal{U}_0^\pi(H_0) = \mathcal{U}_0^\star(H_0). \quad \square \tag{A.112}$$

□

A.1.6. Lemma 2.2

Performing exploration based on (29)–(31) ensures that the algorithm converges to the optimal value function within a finite number of planning iterations.

Proof. Consider a given policy π. We claim that following the state and observation selection criteria in Eqs. (30) and (31) will lead to visiting unexplored trajectories τ_T at every iteration unless all relevant trajectories have already been explored.

To show this, note that the upper bound $\mathcal{U}_0^\star((H_t, a_t, o_{t+1}))$ and the lower bound $\mathcal{L}_0^\star((H_t, a_t, o_{t+1}))$ will converge when the bound interval is zero, i.e.,

$$\mathcal{U}_0^\star((H_t, a_t, o_{t+1})) - \mathcal{L}_0^\star((H_t, a_t, o_{t+1})) = 0. \tag{A.113}$$

This convergence occurs when all future trajectories by following policy π from node $H_{t+1} = (H_t, a_t, o_{t+1})$ until the end of the horizon were explored,

$$\begin{aligned}
&\mathcal{U}_0^\pi(H_{t+1}) - \mathcal{L}_0^\pi(H_{t+1}) = \\
&= \sum_{\tau_{t+1} \in \mathcal{T}(H_{t+1})} \bar{\mathbb{P}}(\tau_{t+1}) \mathcal{V}_{\text{max},t+1} \\
&\quad + \sum_{z_{t+2} \in \bar{\mathcal{Z}}(H_{t+1},\pi_{t+1})} \left[\mathcal{U}_0^\pi(H_{t+2}) - \sum_{\tau_{t+2} \in \mathcal{T}(H_{t+2})} \bar{\mathbb{P}}(\tau_{t+2}) \mathcal{V}_{\text{max},t+1} \right] \\
&\quad - \Bigg[\sum_{\tau_{t+1} \in \mathcal{T}(H_{t+1})} \bar{\mathbb{P}}(\tau_{t+1}) \mathcal{V}_{\text{min},t+1} + \sum_{z_{t+2} \in \bar{\mathcal{Z}}(H_{t+1},\pi_{t+1})} \Bigg[\mathcal{L}_0^\pi(H_{t+2}) \\
&\quad - \sum_{\tau_{t+2} \in \mathcal{T}(H_{t+2})} \bar{\mathbb{P}}(\tau_{t+2}) \mathcal{V}_{\text{min},t+1} \Bigg] Bigg] \\
&= \left[\sum_{\tau_{t+1} \in \mathcal{T}(H_{t+1})} \bar{\mathbb{P}}(\tau_{t+1}) - \sum_{z_{t+2} \in \bar{\mathcal{Z}}(H_{t+1},\pi_{t+1})} \sum_{\tau_{t+2} \in \mathcal{T}(H_{t+2})} \bar{\mathbb{P}}(\tau_{t+2}) \right] \Big(\mathcal{V}_{\text{max},t+1} \\
&\quad - \mathcal{V}_{\text{min},t+1} \Big) + \sum_{z_{t+2} \in \bar{\mathcal{Z}}(H_{t+1},\pi_{t+1})} \left[\mathcal{U}_0^\pi(H_{t+2}) - \mathcal{L}_0^\pi(H_{t+2}) \right]
\end{aligned}$$

since $\forall t \in [0, T-1], \mathcal{V}_{\text{max},t+1} - \mathcal{V}_{\text{min},t+1} \neq 0$, then $\mathcal{U}_0^\pi(H_{t+1}) - \mathcal{L}_0^\pi(H_{t+1}) = 0$ only if ,

$$\sum_{\tau_{t+1} \in \mathcal{T}(H_{t+1})} \bar{\mathbb{P}}(\tau_{t+1}) - \sum_{z_{t+2} \in \bar{\mathcal{Z}}(H_{t+1},\pi_{t+1})} \sum_{\tau_{t+2} \in \mathcal{T}(H_{t+2})} \bar{\mathbb{P}}(\tau_{t+2}) = 0, \ \forall t \in [0, T-2]. \tag{A.114}$$

Thus, all the simplified probability terms in the policy tree converge to 1. Similarly, the probability gap,

$$1 - \sum_{\tau_T} \bar{\mathbb{P}}^\star(\tau_T \mid \tau_t, a_t, z_{t+1}, x) = 0 \tag{A.115}$$

only when all non-zero future trajectories with a prefix $(\tau_t, a_t, z_{t+1}, x)$ have been explored. Finally, we are left to show that selecting actions based on the criteria shown in (29), results in the optimal action upon convergence. Utilizing Corollary 1.1, the proof follows similarly to the one shown in (2.3), which concludes our derivation. □

A.1.7. Gap between the upper and lower bounds.

Define

$$\Delta(H_t) \triangleq \mathcal{U}_0^\pi(H_t) - \mathcal{L}_0^\pi(H_t), \qquad \Delta V(H_t) \triangleq \mathcal{V}_{\text{max}}(H_t) - \mathcal{V}_{\text{min}}(H_t).$$

Step 1: Recursion. Subtract the two expressions in (21):

$$\begin{aligned}
\Delta(H_t) = &\sum_{\tau_t \in \mathcal{T}(H_t)} \bar{\mathbb{P}}(\tau_t)\, \Delta V(H_t) + \sum_{z_{t+1} \in \bar{\mathcal{Z}}(H_t,\pi_t)} \Big[\Delta(H_{t+1}) \\
&- \sum_{\tau_{t+1} \in \mathcal{T}(H_{t+1})} \bar{\mathbb{P}}(\tau_{t+1})\, \Delta V(H_t) \Big].
\end{aligned} \tag{A.116}$$

Because $\mathcal{T}(H_{t+1})$ contains *only* those trajectories whose next observation is *inside* $\bar{\mathcal{Z}}$ *and* whose next state is *inside* the current simplified state set $\bar{\mathcal{X}}(H_{t+1})$, the probability that *drops out* between the two sums in (A.116) is precisely the mass of trajectories that *leave either* simplified space at step $t+1$.

Step 2: Coverage-gap term. Let

$$\delta(H_t) \triangleq \sum_{\tau_t \in \mathcal{T}(H_t)} \bar{\mathbb{P}}(\tau_t) \; - \sum_{z_{t+1} \in \bar{\mathcal{Z}}(H_t,\pi_t)} \; \sum_{\tau_{t+1} \in \mathcal{T}(H_{t+1})} \bar{\mathbb{P}}(\tau_{t+1}),$$

i.e. the probability that either (i) $z_{t+1} \notin \bar{\mathcal{Z}}(H_t, \pi_t)$ *or* (ii) $z_{t+1} \in \bar{\mathcal{Z}}$ but $x_{t+1} \notin \bar{\mathcal{X}}(H_{t+1})$ given τ_t. Then (A.116) rewrites as

$$\Delta(H_t) \;=\; \Delta V(H_t)\, \delta(H_t) + \sum_{z_{t+1} \in \bar{\mathcal{Z}}(H_t,\pi_t)} \Delta(H_{t+1}),$$

or,

$$\mathcal{U}_0^\pi(H_t) - \mathcal{L}_0^\pi(H_t) \;=\; \Delta V(H_t)\, \delta(H_t) + \sum_{z_{t+1} \in \bar{\mathcal{Z}}(H_t,\pi_t)} \left[\mathcal{U}_0^\pi(H_{t+1}) - \mathcal{L}_0^\pi(H_{t+1}) \right].$$

Hence the upper-lower gap at node H_t accumulates the local value spread $\Delta V(H_k)$ *only* on steps where the trajectory exits *either* simplified space. If all future observations and states remain in $\bar{\mathcal{Z}}$ and $\bar{\mathcal{X}}$, the indicator is zero and the gap collapses; conversely, frequent exits slow down the tightening of the bounds.

B. Experiments

B.1. POMDP scenarios

We begin with a brief description of the Partially Observable Markov Decision Process (POMDP) scenarios implemented for the experiments. each scenario was bounded by a finite number of time steps used for every episode, where each action taken by the agent led to a decrement in the number of time steps left. After the allowable time steps ended, the simulation was reset to its initial state.

B.1.1. Tiger POMDP

The Tiger is a classic POMDP problem [16], involves an agent making decisions between two doors, one concealing a tiger and the other a reward. The agent needs to choose among three actions, either open each one of the doors or listen to receive an observation about the tiger position. In our experiments, the POMDP was limited horizon of 5 steps. The problem consists of 3 actions, 2 observations and 2 states.

B.1.2. Discrete light dark

Is an adaptation from [3]. In this setting the agent needs to travel on a 1D grid to reach a target location. The grid is divided into a dark region, which offers noisy observations, and a light region, which offers accurate localization observations. The agent receives a penalty for every step and a reward for reaching the target location. The key challenge is to balance between information gathering by traveling towards the light area, and moving towards the goal region.

B.1.3. Laser tag POMDP

In the Laser Tag problem, [2], an agent has to navigate through a grid world, shoot and tag opponents by using a laser gun. The main goal is to tag as many opponents as possible within a given time frame. The grid is segmented into various sections that have varying visibility, characterized by obstacles that block the line of sight, and open areas. There are five possible actions, moving in four cardinal directions (North, South, East, West) and shooting the laser. The observation space cardinality is $|\mathcal{Z}| \approx 1.5 \times 10^6$, which is described as a discretized normal distribution and reflect the distance measured by the laser. The states reflect the agent's current position and the opponents' positions. The agent receives a reward for tagging an opponent and a penalty for every movement, encouraging the agent to make strategic moves and shots.

B.1.4. Baby POMDP

The Baby POMDP is a classic problem that represents the scenario of a baby and a caregiver. The agent, playing the role of the caregiver, needs to infer the baby's needs based on its state, which can be either crying or quiet. The states in this problem represent the baby's needs, which could be hunger, discomfort or no need. The agent has three actions to choose from: feeding, changing the diaper, or doing nothing. The observations are binary, either the baby is crying or not. The crying observation does not uniquely identify the baby's state, as the baby may cry due to hunger or discomfort, which makes this a partially observable problem. The agent receives a reward when it correctly addresses the baby's needs and a penalty when the wrong action is taken.

B.1.5. Rock sample

In the Rock Sample problem [7] a rover explores a $k \times k$ grid containing n rocks whose quality (good / bad) is initially unknown; where $n = 15$ and $k = 3$ in the experiments section. A state is the rover cell together with an n-bit vector of rock qualities. The agent can move deterministically in the four cardinal directions, execute a *Sample* action at its current cell, or invoke one of n sensing actions. The sensing action returns the observation good, bad with an accuracy level whose determined as a function of the Manhattan distance to rock i; all other actions yield the null observation. Each step incurs a penalty $r_{\text{step}} < 0$, sensing adds $r_{\text{sensor}} < 0$, sampling yields $r_{\text{good}} > 0$ if the rock is good and $r_{\text{bad}} < 0$ otherwise, and reaching the exit column grants $r_{\text{exit}} > 0$. These dynamics create a canonical exploration-exploitation trade-off in which the rover must decide whether to spend time and energy gathering information or head directly for the exit.

B.1.6. Navigate to goal

This environment is a $k \times n$ grid (default 5×5) in which a state is the agent's coordinate pair $s = (x, y)$. Five actions are available, *Stay*, *East*, *West*, *North*, and *South*. An intended move succeeds with probability 0.7; the remaining 0.3 is shared uniformly among the other adjacent cardinal neighbors (attempts to leave the grid leave the agent in place). Rewards are $+10$ for entering a goal cell, -10 for stepping on the a trap cell, and 0 elsewhere.

B.2. Hyperparameters

The hyperparameters for both DB-DESPOT and AR-DESPOT algorithms were selected through a grid search. We explored an array of parameters for AR-DESPOT, choosing the highest-performing configuration. Specifically, the hyperparameter K was varied across $\{10, 50, 500, 5000\}$, while λ was evaluated at $\{0, 0.01, 0.1\}$. Similarly, DB-POMCP and POMCP were examined three different values for the exploration-exploitation weight, $c = \{0.1, 1.0, 10.0\}$ multiplied by $V_{\max}$, which denotes an upper bound for the value function.

For the initialization of the upper and lower bounds used by the algorithms, we used the maximal reward, multiplied by the remaining time steps of the episode, $\mathcal{R}_{\max} \cdot (\mathcal{T} - t - 1)$.

References

[1] D. Silver, J. Veness, Monte-Carlo planning in large POMDPs, in: Advances in Neural Information Processing Systems (NIPS), 2010, pp. 2164–2172.

[2] A. Somani, N. Ye, D. Hsu, W.S. Lee, DESPOT: online POMDP planning with regularization, in: NIPS, 13, 2013, pp. 1772–1780.

[3] Z. Sunberg, M. Kochenderfer, Online algorithms for POMDPs with continuous state, action, and observation spaces, in: Proceedings of the International Conference on Automated Planning and Scheduling, 28, 2018.

[4] M. Hoerger, H. Kurniawati, An on-line POMDP solver for continuous observation spaces, in: IEEE Intl. Conf. on Robotics and Automation (ICRA), IEEE, 2021, pp. 7643–7649.

[5] C. Wu, G. Yang, Z. Zhang, Y. Yu, D. Li, W. Liu, J. Hao, Adaptive online packing-guided earch for POMDPs, in: M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, J.W. Vaughan (Eds.), Advances in Neural Information Processing Systems (NIPS), 34, Curran Associates, Inc., 2021, pp. 28419–28430. https://proceedings.neurips.cc/paper_files/paper/2021/file/ef41d4887553673 16f04fc0e0e9dc9fc-Paper.pdf.

[6] M. Barenboim, V. Indelman, Online POMDP planning with anytime deterministic guarantees, in: Advances in Neural Information Processing Systems, 2023.

[7] T. Smith, R. Simmons, Heuristic search value iteration for POMDPs, in: Conf. on Uncertainty in Artificial Intelligence (UAI), 2004, pp. 520–527.

[8] H. Kurniawati, D. Hsu, W.S. Lee, SARSOP: efficient point-based POMDP planning by approximating optimally reachable bspaces, in: Robotics: Science and Systems (RSS), 2008.

[9] A. Couëtoux, J.-B. Hoock, N. Sokolovska, O. Teytaud, N. Bonnard, Continuous upper confidence trees, in: Learning and Intelligent Optimization: 5th International Conference, LION 5, Rome, Italy, January 17-21, 2011. Selected Papers 5, Springer, 2011, pp. 433–445.

[10] P. Auer, N. Cesa-Bianchi, P. Fischer, Finite-time analysis of the multiarmed bandit problem, Mach. Learn. 47 (2) (2002) 235–256.

[11] N. Ye, A. Somani, D. Hsu, W.S. Lee, DESPOT: online POMDP planning with regularization, JAIR 58 (2017) 231–266.

[12] M.H. Lim, C. Tomlin, Z.N. Sunberg, Sparse tree search optimality guarantees in POMDPs with continuous observation spaces, in: Intl. Joint Conf. on AI (IJCAI), 2020, pp. 4135–4142.

[13] M.H. Lim, T.J. Becker, M.J. Kochenderfer, C.J. Tomlin, Z.N. Sunberg, Optimality guarantees for particle belief approximation of pomdps, J. Artif. Intell. Res. 77 (2023) 1591–1636.

[14] L. Kocsis, C. Szepesvári, Bandit based Monte-Carlo planning, in: European Conference on Machine Learning, Springer, 2006, pp. 282–293.

[15] M. Egorov, Z.N. Sunberg, E. Balaban, T.A. Wheeler, J.K. Gupta, M.J. Kochenderfer, POMDPs.jl: a framework for sequential decision making under uncertainty, J. Mach. Learn. Res. 18 (26) (2017) 1–5. http://jmlr.org/papers/v18/16-300.html.

[16] L.P. Kaelbling, M.L. Littman, A.R. Cassandra, Planning and acting in partially observable stochastic domains, Artif. Intell. 101 (1) (1998) 99–134.