Approximation Algorithms for Observer Aware MDPs

Shuwa Miura[1] **Olivier Buffet**[2] **Shlomo Zilberstein**[1]

[1]University of Massachusetts Amherst, Amherst, USA
[2] Universite de Lorrainé, INRIA, CNRS, LORIA, F-54000 Nancy, France

Abstract

We present approximation algorithms for Observer-Aware Markov Decision Processes (OAMDPs). OAMDPs model sequential decision-making problems in which rewards depend on the beliefs of an observer about the goals, intentions, or capabilities of the observed agent. The first proposed algorithm is a grid-based value iteration (Grid-VI), which discretizes the observer's belief into regular grids. Based on the same discretization, the second proposed algorithm is a variant of Real-Time Dynamic Programming (RTDP) called Grid-RTDP. Unlike Grid-Vi, Grid-RTDP focuses its updates on promising states using heuristic estimates. We provide theoretical guarantees of the proposed algorithms and demonstrate that Grid-RTDP has a good anytime performance comparable to the existing approach without performance guarantees.

1 INTRODUCTION

Effective communication of intentions, goals, and desires is crucial in our daily interactions and is equally vital for autonomous agents. For instance, consider an autonomous vehicle (AV) approaching a crosswalk with a pedestrian nearby. While the AV might optimize for travel time by approaching the crosswalk at high speed before stopping, this can be unsettling for the pedestrian. A more reassuring approach would be for the AV to slow down well before reaching the crosswalk, signaling its intention to stop. We term such actions that take into account the perspective or beliefs of an observing agent as *observer-aware* behaviors. Observer-aware behaviors range from making the agent's goal clear [Dragan and Srinivasa, 2013], demonstrating its capabilities [Kwon et al., 2018] or disguising possible intentions [Masters and Sardina, 2017, Savas et al., 2022].

The Observer-Aware Markov Decision Process (OAMDP) [Miura and Zilberstein, 2021] offers a general framework for producing observer-aware behaviors. The OAMDP framework assumes a model of how the agent's actions would be interpreted by the observer. In OAMDPs, possible goals, intentions, or capabilities of the observed agent are represented as types. After the observed agent takes an action, the observing agent updates its belief over the possible types, which determines the reward function.

While OAMDP allows modeling various observer-aware planning problems in a unified way, solving OAMDPs is shown to be intractable in the worst case [Miura and Zilberstein, 2021]. The intractability stems from the fact that rewards depend on the belief of the observer, which in turn depends on the history so far. Previous work proposed using Monte-Carlo Tree Search (MCTS) to solve OAMDPs for the finite-horizon objective [Miura and Zilberstein, 2021]. While MCTS exhibits good anytime behavior, it does not provide guarantees on the qualities of the resulting policies.

In this paper, we propose the first approximation algorithms for OAMDPs. We begin by establishing that the domain state and the observer's belief are sufficient for optimal control in OAMDPs (Proposition 1). Our first proposed algorithm is a grid-based value iteration (Grid-VI), which discretizes the belief of the observer into regular grids. We show that Grid-VI converges to the unique fixpoint both in discounted (Proposition 4) and undiscounted (Proposition 6) settings under the standard assumptions, and provide the error bounds for the discounted setting (Proposition5). A potential drawback of Grid-VI is that it can waste time updating values at irrelevant states. To address the issue, we propose a variant of Real-Time Dynamic Programming (RTDP) [Barto et al., 1995] to solve OAMDPs, called Grid-RTDP. Grid-RTDP utilizes heuristic estimates to focus updates on promising states. We demonstrate that Grid-RTDP retains RTDP's desirable property (Proposition 7). Our experimental results indicate that our proposed algorithms are capable of computing near-optimal policies. Specifically, Grid-RTDP solves problems significantly faster than Grid-VI and offers anytime performance comparable to MCTS.

Accepted for the 40th Conference on Uncertainty in Artificial Intelligence (UAI 2024).

2 BACKGROUNDS AND NOTATIONS

2.1 MARKOV DECISION PROCESSES

A finite Markov decision process (MDP) models sequential decision-making under uncertainty. An MDP is described by a tuple $M = \langle S, A, T, R, \gamma, d_0 \rangle$. S and A are finite sets of states and actions, respectively. S_t and A_t represent a state and an action at time t. $T(s_t, a_t, s_{t+1})$ is the probability of $S_{t+1}{=}s_{t+1}$ when $A_t{=}a_t$ and $S_t{=}s_t$. R is a reward for taking a_t at s_t. γ is a parameter called the discount factor. d_0 is the initial state distribution $s_0 \sim d_0$.

A solution of an MDP is called a *policy* (π). An optimal policy for an MDP is a policy that maximizes $\mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R(S_t, A_t)|d_0, \pi]$. A policy ($\pi$) induces a value function $V^\pi(s) = \mathbb{E}[\gamma^t R(S_t, A_t)|S_0 = s, \pi]$. The optimal value function V^* is a value function corresponding to an optimal policy.

2.2 STOCHASTIC SHORTEST PATH PROBLEMS

A *stochastic shortest path problem* (SSP) is an undiscounted, cost-based counterpart of an MDP. An SSP is represented by a tuple $\langle S, A, T, C, d_0, G \rangle$ where: S, A, T are the same as in an MDP. $C(s_t, a_t) : S \times A \to \mathbb{R}_+$ is the cost of performing a_t at s_t. d_0 is the initial state distribution. $G \subset S$ is a set of goal states. The goal states are absorbing and transitions out of goal states have zero costs.

A solution of an SSP is a *policy*. An *optimal policy* π^* is a policy that minimizes $\mathbb{E}[\sum_{t=0}^{\infty} C(S_t, A_t)|d_0, \pi]$. We restrict our attention to problems in which there exists at least one *proper policy*, which reaches the goal from all states with probability 1. Under this assumption, an SSP is guaranteed to have an optimal policy that is proper [Bertsekas and Tsitsiklis, 1991].

2.3 OBSERVER-AWARE MDPS

Observer-Aware Markov Decision Processes (OAMDPs) extend MDPs by allowing the reward to depend on the observer's assumed belief over the types of the observed agent [Miura and Zilberstein, 2021].

Definition. An OAMDP is a tuple[1]
$M = \langle S, A, T, \gamma, d_0, \Theta, b_0, \tau, R \rangle$ where:

- S, A, T, γ, and d_0 are the same as in MDPs. In this paper, we assume S and A are finite.
- Θ is a (finite) set of *types*, representing a characteristic of the agent such as possible goals, intentions, or capabilities.
- $b_0 \in \Delta^{|\Theta|}$ is the initial belief of the observer over the types, where $\Delta^{|\Theta|}$ is a simplex on Θ.
- $\tau : S \times A \times S \times \Theta \to [0, 1]$ is the probability of the observer witnessing a transition $\langle s, a, s' \rangle$ given s and θ. τ can represent different policies and transition functions of the observed agent depending on types.
- $R : S \times A \times \Delta^{|\Theta|} \to \mathbb{R}$ is a belief-dependent reward function. In this paper, we assume that the rewards can be represented as a linear combination of *domain* and *belief-dependent* rewards. That is, $R(s, a, b){=}w_d R_d(s, a) + w_b R_b(b)$ for $w_d, w_b{\in}\mathbb{R}_+$, where R_d and R_b represent domain and belief-dependent reward, respectively.

After observing a transition $\langle s, a, s' \rangle$, the observer is assumed to update its belief (b_t) using Bayes' rule:

$$b_{t+1}^{s,a,s'}(\theta) = \frac{\tau(a, s'|s, \theta) \cdot b_t(\theta)}{\sum_{\theta' \in \Theta} \tau(a, s'|s, \theta') \cdot b_t(\theta')}. \quad (1)$$

A solution to an OAMDP is a policy that maximizes the expected discounted return:

$$\mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R(S_t, A_t, B_t)|d_0, \pi]. \quad (2)$$

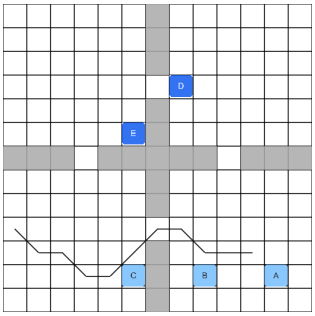
For example, Figure 1 shows an example of an OAMDP with $\Theta = \{\theta_A, \theta_B, \theta_C, \theta_D, \theta_E\}$, where each type corresponds to the observed agent's goal. $\tau(a, s'|s, \theta)$ is typically set to $T_\theta(s, a, s')\pi_\theta(s, a)$, where π_θ is an assumed policy of the observed agent given a type θ and T_θ is a transition function given a type θ. For example, π_{θ_A} represents a policy given the observed agent is going to the goal A. When Θ represents different capabilities of the observed agent, T_θ represents transition functions corresponding to different capabilities. When T_θ is the same for all $\theta \in \Theta$, $\tau(a, s'|s, \theta)$ simplifies to $\pi_\theta(s, a)$ in Equation 1.

Noisy Rational Model A common approach in modeling the observer involves using inverse planning. This assumes that the observed agent behaves approximately rationally given its type. Baker et al. [2009] explored the connection between Bayesian reasoning and human understanding of goals. A model presented in their work presumes noisy rationality:

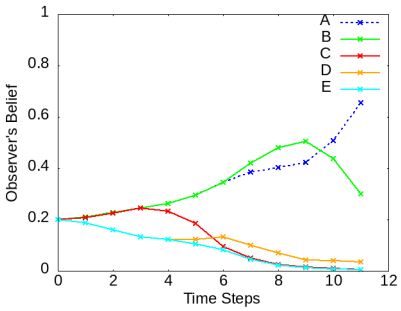
$$\pi_\theta(s, a) \propto \exp^{\beta Q_\theta^*(s,a)}, \quad (3)$$

where Q_θ^* is the optimal Q-value representing how good a is given s and θ. Note that, Q_θ^* is computed with respect to T_θ and R_θ (the reward function corresponding to θ), $\beta \in \mathbb{R}$ serves as a hyper-parameter representing the agent's rationality level. Intuitively, it is assumed that the observed agent selects an action with a probability exponentially proportional to the quality of the action at the current state.

[1]The original work [Miura and Zilberstein, 2021] allowed an arbitrary function from H^* to $\Delta^{|\Theta|}$ to update the observer's belief. Here, we restrict our attention to a case where the observer updates its belief in a Bayesian fashion.



(a) Environment



(b) Observer's belief ($\beta = 0.3$)

Figure 1: MazeWorld Domain

Figure 1b shows the observer's belief changes according to Equation 3.

Belief-Dependent Rewards OAMDP can produce various observer-aware behaviors by changing R_b. For instance, to clarify intentions, R_b might be defined as the negative total variation (TV) or the Euclidean distance between the current and target beliefs, where the target belief is $b(\theta) = 1$ for the intended type $\theta \in \Theta$. On the other hand, if the observed agent wants to obscure its intention, rewards could be the entropy of the observer's belief.

3 PROPERTIES OF OAMDPS

In this section, we discuss properties of OAMDPs necessary for developing proposed algorithms.

3.1 SUFFICIENT STATISTICS FOR OPTIMAL CONTROL

To compute policies for OAMDPs, previous work [Miura and Zilberstein, 2021] used a general-purpose method such as UCT [Kocsis and Szepesvári, 2006] to compute history-dependent policies. However, we show that the current state and the belief of the observer contain sufficient information to choose the best action to take:

Proposition 1. *The current state and the belief of the observer are sufficient for optimal control for OAMDPs.*

Proof. For all $s_t, s_{t+1} \in S$, $a_t \in A$, $b_t \in \Delta^{|\Theta|}$, $h_t \in H_t$:

$$\Pr(s_{t+1}, b_{t+1} | s_t, a_t, b_t, h_t) \tag{4}$$
$$= \Pr(b_{t+1} | s_t, a_t, s_{t+1}, b_t, h_t) \Pr(s_{t+1} | s_t, a_t, b_t, h_t) \tag{5}$$
$$= [b_{t+1} = b_t^{s_t, a_t, s_{t+1}}] T(s_t, a_t, s_{t+1}) \text{ by definition} \tag{6}$$
$$= \Pr(s_{t+1}, b_{t+1} | s_t, a_t, b_t) \tag{7}$$

where $[\cdot]$ is the Iverson bracket. Moreover, R only depends on S_t, A_t, and B_t by definition. □

With Proposition 1 in place, we can look for policies of the forms $\pi : S \times \Delta^{|\Theta|} \times A \to [0, 1]$. In other words, we can look for policies to *belief MDP*, whose state space is $S \times \Delta^{|\Theta|}$ instead of S. Note that, while the original OAMDP has a finite number of states, the belief MDP has a continuous state space. Proposition 1 is analogous to how beliefs over states (belief states) are sufficient for optimal control for POMDPs [Kaelbling et al., 1998]. However, while most solution methods for POMDPs [Monahan, 1982, Pineau et al., 2003] rely on piecewise linear convexity (PWLC) of the value function, we see that the value functions for OAMDPs are not necessarily PWLC. For example, consider using the negative Euclidean distance from the intended type as R_b. R_b is not PWLC on $\Delta^{|\Theta|}$. Therefore, solution methods for POMDPs are not directly applicable to OAMDPs.

3.2 DISCONTINUITY IN VALUE FUNCTIONS

Before delving into our proposed algorithms, we address a potential issue in developing an approximation algorithm for OAMDPs. Both of our proposed algorithms approximate values by grouping similar beliefs. This approach operates under the implicit assumption that nearby beliefs should yield similar values. However, we demonstrate that, in a general OAMDP, the rate at which the observer's belief changes can be unbounded, thus invalidating this assumption. To illustrate this issue, consider the following example:

Example. *Let us assume that we have an OAMDP with:*

- $\Theta = \{\theta_0, \theta_1, \theta_2\}$,
- $b_1 = (1 - \epsilon, \epsilon, 0) \in \Delta^3$,
- $b_2 = (1 - \epsilon, 0, \epsilon) \in \Delta^3$, *and*
- $\tau^{s,a,s'} = (\tau_0 = 0, \tau_1 > 0, \tau_2 > 0)$.

Then, $b_1^{s,a,s'} = (0, 1, 0)$ *and* $b_2^{s,a,s'} = (0, 0, 1)$. *Thus,*

$$\frac{\|b_1^{s,a,s'} - b_2^{s,a,s'}\|_\infty}{\|b_1 - b_2\|_\infty} = \frac{\|(0, 1, -1)\|_\infty}{\|(0, \epsilon, -\epsilon)\|_\infty} = \frac{1}{\epsilon}. \tag{8}$$

$\frac{\|b_1^{s,a,s'} - b_2^{s,a,s'}\|_\infty}{\|b_1 - b_2\|_\infty}$ *diverges as* $\epsilon \to 0$.

3.3 LIPSCHITZ OAMDPS

Given the potential discontinuity in values, we discuss special cases of OAMDPs with Lipschitz-continuous reward and belief transitions.

Definition. An OAMDP is (L_r, L_p)-Lipschitz if for all $s, s' \in S$, $a \in A$, and $b_1, b_2 \in \Delta^{|\Theta|}$:

$$|R(s,a,b) - R(s,a,b')| \leq L_r \|b_1 - b_2\|_\infty, \tag{9}$$

$$\|b_1^{s,a,s'} - b_2^{s,a,s'}\|_\infty \leq L_p \|b_1 - b_2\|_\infty. \tag{10}$$

Intuitively, in Lipschitz OAMDPs, beliefs close to each other have similar rewards and update to close beliefs. The definition is analogous to Lipschitz continuity of continuous MDPs in general [Rachelson and Lagoudakis, 2010].

Lipschitz continuity of reward and belief transitions can be related to Lipschitz continuity of the value function under a favorable assumption:

Proposition 2. *For a (L_r, L_p)-Lipschitz OAMDP, if $\gamma L_p < 1$, then V^* is L_{V^*}-Lipschitz continuous where:*

$$L_{V^*} = \frac{L_r}{1 - \gamma L_p}. \tag{11}$$

Proof. See Appendix A □

As we will see later, Lipschitz continuity enables us to provide the error bound for discretization (Proposition 5).

Moreover, in OAMDPs, belief transitions are assumed to be the Bayesian update using Equation 1. We can establish a relationship between the Lipschitz continuity of belief transitions and τ as follows:

Proposition 3. *If $\tau^{s,a,s'}(\theta) > 0$ for $\forall \theta \in \Theta$, $s, s' \in S$, and $a \in A$, belief transitions are Lipschitz continuous.*

Proof. See Appendix A □

For example, using the noisy rational model (Equation 3) ensures that $\tau^{s,a,s'}(\theta) > 0$, which guarantees the Lipschitz continuity of belief transitions.

3.4 OASSPS

We define an undiscounted, cost-based version of OAMDPs called OASSPs. An OASSP is a tuple $\langle S, A, T, d_0, \Theta, b_0, \tau, C, G \rangle$ where $C : S \times A \times \Delta^{|\Theta|} \to \mathbb{R}_+$ is a belief-dependent cost function, and G is a set of goal states. The other components are the same as in OAMDPs. An optimal policy for an OASSP is a policy that minimizes $\mathbb{E}[\sum_{t=0}^{\infty} C(S_t, A_t, B_t) | d_0, \pi]$. As in OAMDPs, we assume that C is a linear combination of the domain cost (C_d) and belief-dependent cost (C_b). That is, $C(s,a,b) = w_d C_d(s,a) + w_b C_b(s,a)$. A domain SSP corresponding to an OASSP is an SSP defined as $M_d = \langle S, A, T, d_0, C_d, G \rangle$.

4 APPROXIMATION ALGORITHMS

In this section, we propose approximation algorithms for OAMDP/SSPs. Our first proposed algorithm is a grid-based value iteration (Grid-VI), which discretizes the observer's belief into regular grids. Our second proposed algorithm is a variant of Real-Time Dynamic Programming (RTDP), called Grid-RTDP. Grid-RTDP relies on the same grid-based discretization scheme as Grid-VI, but focuses its updates on promising states using heuristic estimates.

4.1 GRID-BASED VALUE ITERATION FOR OAMDP/SSPS

We first describe a grid-based value iteration algorithm for OAMDP/SSPs. Grid-VI uses a set of regular grid points to approximate value functions. A regular grid with the resolution K is defined as:

$$P_K = \Big\{ b = (\frac{1}{K})k | k \in I_+^{|\Theta|}, \sum_{i=1}^{|\Theta|} k(i) = K \Big\}, \tag{12}$$

where $I_+^{|\Theta|}$ is the set of $|\Theta|$-vectors of non-negative integers. P_K divides $\Delta^{|\Theta|}$ into a set of equal-size sub-simplices. Figure 2 shows an example of a regular grid on Δ^3 with $K = 2$.

As in Lovejoy [1991], the value at a given belief point $b \in \Delta^{|\Theta|}$ is interpolated as using the barycentric coordinates of b with respect to $P_K(b)$:

$$V_K(s,b) = \sum_{b_i \in P_K(b)} \lambda_i V_K(s, b_i), \tag{13}$$

where $P_K(b)$ is the corners of the sub simplex containing b, $\lambda_i \geq 0$, $\sum_{i=1}^{|\Theta|} \lambda_i = 1$, and $b = \sum_{i=1}^{|\Theta|} \lambda_i b_i$. In Figure 2, the value at b is interpolated using the values at b_4, b_5, and b_6. For each iteration, the algorithm updates values at all $s \in S$ and $b \in P_K$ using the Bellman optimality operator ($\mathcal{T}$):

$$(\mathcal{T}V_K)(s,b) = \max_{a \in A} \big[R(s,a,b) + \gamma \sum_{s' \in S} T(s,a,s') V_K(s', b^{s,a,s'}) \big], \tag{14}$$

where values at $b \notin P_K$ are interpolated using Equation 13. The resulting policy is obtained as:

$$\pi_K(s,b,a) = \sum_{b_i \in P_K(b)} \lambda_i [a = \arg\max_{a_i \in A} Q_K(s, b_i, a_i)], \tag{15}$$

where $Q_K(s, b_i, a_i) = R(s, a_i, b_i) + \gamma \sum_{s' \in S} T(s,a,s') V_K(s', b^{s,a,s'})$. That is, we take the optimal actions at the corners of sub-simplices proportional to the corresponding weights λ_i.

For problems with undiscounted objectives (OASSPs), Equation 14 is replaced with minimizing costs without the discount factor.

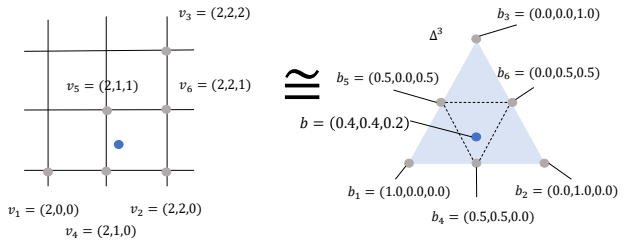


Figure 2: An example of discretized belief points P_K (right) with $K = 2$ and $|\Theta| = 3$. The left is the corresponding integer points (P'_K).

Efficient Interpolation

One key advantage of using a regular grid is that finding λ is quite efficient. To efficiently find barycentric coordinates of $b \in \Delta^{|\Theta|}$ with respect to ($P_K(b) \subset \Delta^{\Theta}$), we use a Freudenthal triangulation [Freudenthal, 1942]:

$$P'_K = \left\{ q \in I_+^{|\Theta|} | K = q_1 \geq q_2 \geq \cdots \geq q_{|\Theta|} \right\}. \quad (16)$$

Note that, we have $|P'_K| = |P_K| = \frac{(K+|\Theta|-1)!}{K!(|\Theta|-1)!}$. Due to one-to-one correspondence between points in P_K and P'_K, we can find a barycentric coordinate for $b \in \Delta^{|\Theta|}$ using a barycentric coordinate for the corresponding $v \in I_+^{|\Theta|}$ [Lovejoy, 1991]. As discussed by Zhou and Hansen [2001], finding a sub-simplex can be done in $\mathcal{O}(|\Theta| \log |\Theta|)$ time.

Theoretical Guarantees

We now discuss theoretical guarantees of Grid-VI.

Proposition 4. *For an OAMDP, Grid-VI converges to the unique fixpoint V_K^*.*

Proof. The interpolation (Equation 13) can be understood as an operator on the value function. Let $\mathcal{A}_K$ be the corresponding operator, then our Grid-VI can be seen as repeatedly applying ($\mathcal{T}_K = \mathcal{A}_K \circ \mathcal{T}$) to the value function. Since $\mathcal{A}_K$ is nonexpansion and $\mathcal{T}$ is contraction, $\mathcal{A}_K \circ \mathcal{T}$ is also a contraction, and Grid-VI converges to the unique fixpoint V_K^* [Gordon, 1995]. □

Lemma 1. *For an OAMDP with Lipschitz-continuous value function with the constant L_{V^*}, one-step approximation errors using a regular grid with resolution K are bounded as:*

$$\|\mathcal{T}_K V^* - V^*\|_\infty \leq \frac{L_{V^*}}{K}. \quad (17)$$

Proposition 5. *For an OAMDP whose value function is L_{V^*}-Lipschitz continuous, we have:*

$$\|V^* - V_K^*\|_\infty \leq \frac{L_{V^*}}{(1-\gamma)K}. \quad (18)$$

Proof. See Section A. □

Note that the right-hand sides go to 0 as $K \to \infty$.

Next, we discuss a case where Grid-VI is applied to undiscounted problems (OASSPs). We first note that, for an OASSP $M = \langle S, A, T, d_0, \Theta, b_0, \tau, C, G \rangle$, Grid-VI for OASSPs implicitly defines an SSP $M_K = \langle S \times P_K, A, T, d_0^K, C_K, G_K \rangle$ where

$$T_K(\langle s, b \rangle, a, \langle s', b_i \rangle) = \begin{cases} 0 & b_i \neq P_K(b^{s,a,s'}), \\ \lambda_i T(s,a,s') & b^{s,a,s'} = \sum_i \lambda_i b_i, \end{cases} \quad (19)$$

$$d_0^K(\langle s, b_i \rangle) = \begin{cases} 0 & b_i \neq P_K(b_0), \\ \lambda_i d_0(s) & b_0 = \sum_i \lambda_i b_i, \end{cases} \quad (20)$$

$$C_K(s,a,b) = C(s,a,b), \quad (21)$$

$$G_K = G \times P_K. \quad (22)$$

The states in M_K consist only of the corners of sub-simplices. The transitions in M_K are the same as in the original OASSP, except that, after the belief update, $b^{s,a,s'}$ is transitioned to one of the belief points $b_i \in P_K(b^{s,a,s'})$ surrounding it. Note that, unlike the original OASSP, the number of belief states in M_K is finite.

Since all M, M_d and M_K have the same dynamics in terms of domain state transitions, we have:

Lemma 2. *If M_d has a proper policy, M and M_K also have at least one proper policy.*

Proof. Let π_d be a proper policy for M_d. Then $\pi(\langle s, b \rangle, a) = \pi_K(\langle s, b \rangle, a) = \pi_d(s, a)$ are proper policies for M and M_K, respectively. □

For an SSP with a finite number of states, value iteration converges to the unique fixpoint as long as there is a proper policy [Bertsekas and Tsitsiklis, 1991]. Thus, we get:

Proposition 6. *If M_d has a proper policy, Grid-VI for OASSPs converges to the unique fixpoint V_K^*.*

Our algorithm shares similarities with grid-based approximations for POMDPs [Lovejoy, 1991, Brafman, 1997, Hauskrecht, 2000, Zhou and Hansen, 2001, Bonet, 2002]. The main difference is that the belief is over Θ in OAMDP/SSPs instead of over S as in POMDPs. Approximation using regular grids requires the number of points exponential to the dimension of belief vectors. However, in most scenarios, it is reasonable to assume that the number of possible intentions ($|\Theta|$) is much smaller than the number of states. Thus, having grid points exponential to the dimension of belief vectors is less of a constraint for OAMDP/SSPs.

Our Grid-VI for OAMDP/SSPs is a special case of grid-based value iteration for continuous MDPs [Chow and Tsitsiklis, 1991, Munos and Moore, 2002] in general. One main difference is that, in OAMDP/SSPs, the continuous part of the state space ($\Delta^{|\Theta|}$) is guaranteed to be a simplex, which enables the efficient interpolation method. Another difference is that, due to the structure of OAMDP/SSPs, discretization preserves the existence of a proper policy (Lemma 2).

4.2 GRID-BASD REAL-TIME DYNAMIC PROGRAMMING FOR OAMDP/SSPS

We now propose an extension of Real-Time Dynamic Programming (RTDP) [Barto et al., 1995] to OAMDP/SSPs, called Grid-RTDP. The potential issue for Grid-VI is that it needs to update values at every state and grid points. However, many of these points could be irrelevant in computing an optimal policy. RTDP is an asynchronous value iteration algorithm that can converge to the optimal solution without having to consider the entire state space. RTDP avoids exploring a portion of the state space by utilizing an admissible heuristic. Our presentation in this section will be based on OASSPs.

Grid-RTDP discretizes beliefs into regular grids as in Grid-VI. The value at a belief $b \in \Delta^{|\Theta|}$ is interpolated using Equation 13. Algorithm 1 shows a pseudocode for Grid-RTDP. The algorithm consists of repeated trials, where each trial starts from the initial state and belief of the observer. During each trial, the algorithm first maps the current belief b to one of the surrounding grid points $b_i \in P_K(b)$ randomly, where $b = \sum_{i=1}^{|\Theta|} \lambda_i b_i$. Each b_i has probability λ_i of transitioning into (line 11). Then the algorithm selects an action that minimizes the current cost estimate to the goal $Q_K(s, b_i, a)$ (line 12):

$$Q_K(s,b,a) \tag{23}$$
$$= C(s,a,b) + \sum_{s' \in S} T(s,a,s') V_K(s', b^{s,a,s'}) \tag{24}$$
$$= C(s,a,b) + \sum_{s' \in S} T(s,a,s') \sum_{b_i \in P_K(b^{s,a,s'})} \lambda_i V_K(s', b_i), \tag{25}$$

where V_K is initialized with a given heuristic function h. In this paper, we consider the following two heuristic functions:

- h_0: which always returns 0 (in other words, no heuristics), and
- h_d: which returns the scaled optimal cost to go for the underlying domain cost ($w_d \cdot V_d^*(s)$).

Note that both h_0 and h_d are admissible heuristics. After selecting the best action a^*, the cost estimate for the current state ($V_K(s, b_i)$) is updated to $Q_K(s, b_i, a^*)$ (line 34), the values are updated only at beliefs in P_k. The next state is

Algorithm 1 Grid-RTDP

```
1:  function GRID-RTDP
2:      while within computational budget do
3:          TRIAL(d_0, b_0)
4:      end while
5:  end function
6:
7:  function TRIAL(d_0, b_0)
8:      s ~ d_0
9:      b ← b_0
10:     while episode continues do
11:         sample b_i ∈ P_K(b) with the weight λ_i
12:         a* ← min_a Q_K(s, b_i, a)
13:         V_K(s, b_i) ← Q_K(s, b_i, a*)
14:         s' ~ Pr(·|s, a*)
15:         b ← b_i^{s,a,s'}
16:     end while
17: end function
```

then sampled according to the dynamics of the environment (line 14) and the belief of the observer is updated accordingly (line 15). The resulting policy is obtained as:

$$\pi_K(s,b,a) = \sum_{b_i \in P_K(b)} \lambda_i [a = \arg\min_{a_i \in A} Q_K(s, b_i, a_i)]. \tag{26}$$

That is, we take the optimal actions at corners of subsimplices proportional to the corresponding weights λ_i.

The algorithm is akin to RTDP-Bel [Bonet and Geffner, 2009], a version of RTDP developed for POMDPs. Similar to Grid-RTDP, RTDP-Bel is based on discretizing beliefs. Let $d(b)$ be a discretization of b. Unlike Grid-RTDP that updates the value at $d(b)$ using Q-values at $d(b)$, RTDP-Bel updates the value at $d(b)$ using Q-values at b. This can be a problem when two different belief points b_1 and b_2 discretizes to the same point ($d(b_1) = d(b_2)$), resulting in RTDP-Bel's oscillating behavior.

Properties We discuss some properties of Grid-RTDP. When applied to SSPs, RTDP has the following guarantee:

Theorem 1 (Barto et al. [1995])**.** *If there exists a proper policy for an SSP, the initial value is admissible, RTDP converges to the optimal value at relevant states.*

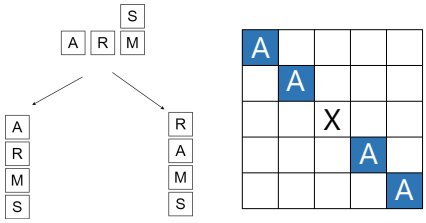
We will now show that Grid-RTDP inherits the properties analogous to Theorems 1 under the following conditions:

A1 The domain SSP M_d has a proper policy.

A2 The initial value estimates are admissible.

Combining Lemma 2 with Theorem 1, we get:

Proposition 7. *Under A1-2, Grid-RTDP converges to the optimal values (V_K^*) at relevant states.*



(a) BlocksWorld (b) Acronym

Figure 3: Problems

4.2.1 Grid-Based Labeled RTDP for OASSPs

We now propose labeled RTDP (LRTDP) [Bonet and Geffner, 2002] for OASSPs, called Grid-LRTDP. The original RTDP does not explicitly check for convergence, and can keep visiting states that are already solved, resulting in its slow convergence behavior. LRTDP alleviates the issue by labeling those states as solved. The algorithm labels states as solved if residuals of Bellman updates in the states that could be visited under the current best policy are smaller than a given threshold. Alternatively, Grid-LRTDP can be understood as applying LRTDP to M_K. The pseudocode for the algorithm is available in the appendix (Appendix B).

5 EXPERIMENTS

We present experimental results solving OASSPs using the proposed algorithms.

5.1 DOMAINS

We briefly describe the problem domains used in the experiments.

MazeWorld Figure 1a shows an example of MazeWorld. The agent's goal is to reach either one of the possible goals $\{A, B, C, D, E\}$. The domain costs are proportional to the distance traveled. To encourage being clear about the intention, C_b is the TV distance from the target belief. To make the problem more challenging, the agent can get transported to the initial state with the probability 0.1 at each time step. $w_d = 0.1$ and $w_b = 1.0$.

BlocksWorld Figure 3a shows an example of Blocksworld from Miura and Zilberstein [2021], where the goal is to stack blocks to spell "ARMS". Picking up a block always succeeds with probability 1, while putting down a block fails with probability 0.3 (the block falls on the table). Each domain action has a cost of 1. C_b is the TV distance from the target belief. $w_d = 0.1$ and $w_b = 1.0$. The optimal policy first stacks "R" on top of "S". This is not optimal in terms of task progression, but tells the observer that the goal "ARMS" is more likely than "RAMS".

Acronym Figure 3b illustrates the Acronym domain. There are four locations with letters. The agent can move in eight different directions. Once the agent is in the locations with letters, it can toggle the letters among $A \to M \to R \to S \to A$. The potential goals are to spell "ARMS", "RAMS", or "MARS" from top left to bottom right. When toggling among letters, there is 0.3 probability of accidentally toggling too much. The objective is spelling "ARMS" while being ambiguous about the intention. $C_b(b) = H_{max} - H(b)$ where H_{max} is the entropy of the uniform distribution and $H(b)$ represents the entropy of b. $w_d = 0.5$ and $w_b = 1.0$.

5.2 OFFLINE CONVERGENCE

We compare the following algorithms on the time before the maximum residual is smaller than $\epsilon = 10^{-3}$:

- Grid-VI with $K = 1, 4, 16$;
- Grid-LRTDP with $K = 1, 4, 16$ using h_0 and h_d.

Each run has time limit 10m and memory limit 2Gbytes.

Table 1 shows the results. Grid-LRTDP using h_d was overall the best algorithm, generating fewer belief states to solve problems. The exception was the MazeWorld domain, where, due to the random transition back to the initial state, Grid-(L)RTDP had to generate most of the belief points. While some problems required only coarse discretization of beliefs, other problems required finer discretization to compute near optimal policies.

5.3 ANYTIME PERFORMANCE

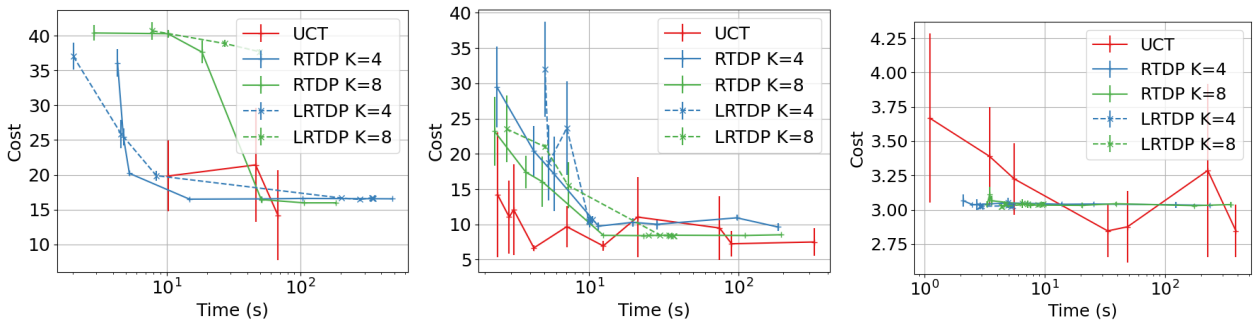
We compare the following algorithms in terms of the anytime behaviors:

- Grid-(L)RTDP with $K = 4, 8$ using h_d;
- UCT where the rollout policy π^*_d is an optimal policy for the domain SSP.

Each algorithm was run for 10^2, 10^3, $5{\cdot}10^3$, 10^4, $5{\cdot}10^4$, 10^5, $5 \cdot 10^5$, 10^6 Grid-(L)RTDP/UCT trials. For UCT, the specified number of trials are performed at each timestep online. For Grid-(L)RTDP, the trials are performed offline. Each run has time limit 10m and memory limit 2Gbytes. Figure 4 shows the results. UCT and Grid-(L)RTDP exhibited performances that complement each other. While UCT showed better anytime performance in Acronym, it took some time to achieve good performance in Blocks World, a small problem instance with $|\Theta| = 2$. Comparing Grid-(L)RTDP with different resolutions (K), using coarser grids generally resulted in better anytime behaviors as long as the resolution is sufficient. Between Grid-RTDP and Grid-LRTDP, they exhibited comparable anytime behaviors.

Domain	$\lvert\Theta\rvert$	K	Grid-VI				Grid-LRTDP(h_0)				Grid-LRTDP(h_d)			
			V	t(s)	$\lvert S\rvert$	$\lvert P_K\rvert$	V	t(s)	$\lvert S\rvert$	$\lvert P_K\rvert$	V	t(s)	$\lvert S\rvert$	$\lvert P_K\rvert$
MazeWorld	5	1	19.15	5.32	148	740	18.9	2.65	148	740	19.05	3.28	148	606
		4	16.69	167.41	148	10360	16.60	155.22	148	10157	16.67	198.60	148	9419
		16	-	-	-	-	-	-	-	-	-	-	-	-
Acronym	3	1	15.69	13.36	6379	19137	15.71	4.62	6379	19137	15.86	5.03	6379	19137
		4	8.41	121.28	6379	95685	10.27	39.38	6379	89116	10.49	19.04	6379	40480
		16	-	-	-	-	8.38	208.23	6379	973053	8.37	10.02	6292	43476
BlocksWorld	2	1	3.57	2.2	125	250	3.57	2.8	125	250	3.57	1.1	125	134
		4	3.04	4.60	125	625	3.36	3.52	125	542	3.03	2.73	124	387
		16	3.03	15.48	125	2125	3.03	16.076	125	1692	3.03	11.45	124	1103

Table 1: Time until convergence for different algorithms. V represents the value when the policy is evaluated under the true environment (M). $t(s)$ is the running time in seconds. $|S|$ and $|P_K|$ represent the number of generated domain and belief states, respectively.



(a) MazeWorld (b) Acronym (c) BlocksWorld

Figure 4: Anytime behaviors for different algorithms.

6 RELATED WORK

OAMDP is a framework unifying different kinds of observer-aware behaviors. *Legible* behavior [Dragan and Srinivasa, 2013, Miura et al., 2021], which implicitly conveys intentions via the choice of actions. Similarly, *explicable* behaviors [Zhang et al., 2017] conform to observers' expectations. *Deceptive* behaviors [Dragan et al., 2015, Masters and Sardina, 2017] hide agents' intentions or actively deceive observers. *Predictable* behaviors enable observers to predict future actions [Fisac et al., 2020]. Agents can also express their *(in)capability* via the choice of their actions [Kwon et al., 2018].

OAMDP could be regarded as a special case of Decision Process with non-Markovian Reward (NMRDP) [Bacchus et al., 1996, Thiébaux et al., 2006]. Unlike OAMDPs, existing works on NMRDPs Bacchus et al. [1996], Thiébaux et al. [2006], Brafman et al. [2018] utilize temporal logic to describe rewards over histories. OAMDP, on the other hand, employs the belief of the observer to capture the non-Markovian nature of rewards.

OAMDPs are related to the line of work that reasons about the belief of other agents. In particular, OAMDPs can be seen as a restricted subset of Interactive POMDPs [Gmytrasiewicz and Prashant, 2005], where agents act by recursively modeling the other agents' beliefs [Miura and Zilberstein, 2021]. In game theory, psychological games deal with utility that depends on the belief of the other agent [Battigalli and Dufwenberg, 2022]. Epistemic game theory [Perea, 2012] also explicitly reasons about the belief of the other agent.

7 CONCLUSION

In this paper, we propose the first approximation algorithms for solving OAMDP/SSPs, Grid-VI and Grid-(L)RTDP. Both of the algorithms are based on discretizing the observer's beliefs into regular grids. To justify the proposed algorithms, we show that the domain state and the belief of the observer constitute a sufficient statistics for OAMDPs (Proposition 1). Furthermore, we show that both algorithms converge to the unique value (Proposition 4, 6, and 7) and provide performance guarantees under the standard assumptions (Propositions 5 and 7). Our experimental results show that the proposed algorithms can compute near-optimal policies for OAMDP/SSPs. In particular, Grid-(L)RTDP can converge to a solution faster than Grid-VI and has anytime performance competitive with UCT.

8 ACKNOWLEDGEMENTS

This research was supported in part by the NSF grant number IIS-2205153 and by the Alliance Innovation Lab Silicon Valley.

References

Fahiem Bacchus, Craig Boutilier, and Adam Grove. Rewarding behaviors. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, volume 2, pages 1160–1167, 1996.

Chris L. Baker, Rebecca Saxe, and Joshua B. Tenenbaum. Action understanding as inverse planning. *Cognition*, 113: 329–349, 2009.

Andrew G. Barto, Steven J. Bradtke, and Satinder P. Singh. Learning to act using real-time dynamic programming. *Artificial Intelligence*, 72(1):81–138, 1995.

Pierpaolo Battigalli and Martin Dufwenberg. Belief-dependent motivations and psychological game theory. *Journal of Economic Literature*, 60(3):833–882, 2022.

Dimitri P. Bertsekas and John N. Tsitsiklis. An analysis of stochastic shortest path problems. *Mathematics of Operations Research*, 16(3):580–595, 1991.

Blai Bonet. An epsilon-optimal grid-based algorithm for partially observable Markov decision processes. In *Proceedings of the Nineteenth International Conference on Machine Learning (ICML 02)*, pages 51–58, 2002.

Blai Bonet and Hector Geffner. Solving stochastic shortest-path problems with RTDP. Technical report, Universidad Simon Bolivar, 2002.

Blai Bonet and Héctor Geffner. Solving POMDPs: RTDP-bel vs. point-based algorithms. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, pages 1641–1646, 2009.

R. Brafman, Giuseppe De Giacomo, and F. Patrizi. LTLf/LDLf non-Markovian rewards. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 1771–1778, 2018.

Ronen I. Brafman. A heuristic variable grid solution method for POMDPs. In *Proceedings of the fourteenth national conference on artificial intelligence and ninth conference on Innovative applications of artificial intelligence*, pages 727–733, 1997.

C.-S. Chow and J.N. Tsitsiklis. An optimal one-way multigrid algorithm for discrete-time stochastic control. *IEEE Transactions on Automatic Control*, 36(8):898–914, August 1991. Conference Name: IEEE Transactions on Automatic Control.

Anca Dragan and Siddhartha Srinivasa. Generating legible motion. In *Proceedings of Robotics: Science and Systems*, Berlin, Germany, 2013.

Anca Dragan, Rachel Holladay, and Siddhartha Srinivasa. Deceptive robot motion: Synthesis, analysis and experiments. *Auton. Robots*, 39(3):331–345, 2015.

Jaime F. Fisac, Chang Liu, Jessica B. Hamrick, Shankar Sastry, J. Karl Hedrick, Thomas L. Griffiths, and Anca D. Dragan. Generating plans that predict themselves. In *Algorithmic Foundations of Robotics XII: Proceedings of the Twelfth Workshop on the Algorithmic Foundations of Robotics*, Springer Proceedings in Advanced Robotics, pages 144–159, 2020.

Hans Freudenthal. Simplizialzerlegungen von Beschrankter Flachheit. *Annals of Mathematics*, 43(3):580–582, 1942. Publisher: Annals of Mathematics.

Piotr J. Gmytrasiewicz and Doshi Prashant. A framework for sequential planning in multi-agent settings. *Journal of Artificial Intelligence Research*, 24:49–79, 2005.

Geoffrey J. Gordon. Stable function approximation in dynamic programming. In *Machine Learning Proceedings 1995*, pages 261–268. Morgan Kaufmann, 1995.

Milos Hauskrecht. Value-function approximations for partially observable Markov decision processes. *Journal of Artificial Intelligence Research*, 13(1):33–94, 2000.

Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1-2):99–134, 1998.

Levente Kocsis and Csaba Szepesvári. Bandit based Monte-Carlo planning. In *European Conference on Machine Learning*, Lecture Notes in Computer Science, pages 282–293, 2006.

Minae Kwon, Sandy H. Huang, and Anca D. Dragan. Expressing robot incapability. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, pages 87–95, 2018.

William S. Lovejoy. Computationally feasible bounds for partially observed markov decision processes. *Operations Research*, 39(1):162–175, 1991.

Peta Masters and Sebastian Sardina. Deceptive path-planning. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pages 4368–4375, August 2017.

Shuwa Miura and Shlomo Zilberstein. A unifying framework for observer-aware planning and its complexity. In *Uncertainty in Artificial Intelligence*, pages 610–620, 2021.

Shuwa Miura, Andrew L. Cohen, and Shlomo Zilberstein. Maximizing legibility in stochastic environments. In *2021 30th IEEE International Conference on Robot & Human Interactive Communication*, pages 1053–1059, 2021.

George E. Monahan. A survey of partially observable Markov decision processes: Theory, models, and algorithms. *Management Science*, 28(1):1–16, 1982.

Rémi Munos and Andrew Moore. Variable resolution discretization in optimal control. *Machine Learning*, 49(2): 291–323, 2002.

Andrés Perea. *Epistemic Game Theory: Reasoning and Choice*. Cambridge University Press, 2012.

Joelle Pineau, Geoff Gordon, and Sebastian Thrun. Point-based Value Iteration: An Anytime Algorithm for POMDPs. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, pages 1025–1030, 2003.

E. Rachelson and M. Lagoudakis. On the locality of action domination in sequential decision making. In *Proceedings of the International Symposium on Artificial Intelligence and Mathematics*, 2010.

Yagiz Savas, Christos K. Verginis, and Ufuk Topcu. Deceptive decision-making under uncertainty. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(5): 5332–5340, 2022. ISSN 2374-3468, 2159-5399.

Sylvie Thiébaux, Charles Gretton, John Slaney, David Price, and F. Kabanza. Decision-theoretic planning with non-Markovian rewards. *Journal of Artificial Intelligence Research*, 25:17–74, 2006.

Yu Zhang, Sarath Sreedharan, Anagha Kulkarni, Tathagata Chakraborti, Hankz Hankui Zhuo, and Subbarao Kambhampati. Plan explicability and predictability for robot task planning. In *International Conference on Robotics and Automation*, pages 1313–1320, 2017.

Rong Zhou and Eric A. Hansen. An improved grid-based approximation algorithm for POMDPs. In *Proceedings of the 17th international joint conference on Artificial intelligence - Volume 1*, pages 707–714, 2001.

A PROOFS

To prove Proposition 2, we first prove the Lipschitz continuity of n-step value function. Let $V^{(0)}(s,b) = 0$ and $V^{(n+1)}(s,b) = \max_a R(s,a,b) + \gamma \sum_{s'} T(s,a,s') V^{(n)}(s', b^{s,a,s'})$. Then we have:

Lemma 3. *For a (L_r, L_p)-Lipschitz OAMDP, $V^{(n)}$ is $L_{V^{(n)}}$-Lipschitz continuous, where $L_{V^{(n)}}$ satisfies:*

$$L_{V^{(n+1)}} = L_r + \gamma L_p L_{V^{(n)}} \tag{27}$$

Proof. Proof by induction on n. For the base case with $n = 1$,

$$|V^{(1)}(s,b_1) - V^{(1)}(s,b_2)| \tag{28}$$
$$= |\max_a R(s,a,b_1) - \max_a R(s,a,b_2)| \tag{29}$$
$$\leq \max_a |R(s,a,b_1) - R(s,a,b_2)| \tag{30}$$
$$\leq L_r \|b_1 - b_2\|_\infty \tag{31}$$

For the induction step,

$$|V^{(n+1)}(s,b_1) - V^{(n+1)}(s,b_2)| \tag{32}$$
$$= |\max_a R(s,a,b_1) + \gamma \sum_{s'} T(s,a,s') V^{(n)}(s', b_1^{s,a,s'}) \tag{33}$$
$$- \max_a R(s,a,b_2) + \gamma \sum_{s'} T(s,a,s') V^{(n)}(s', b_2^{s,a,s'})| \tag{34}$$
$$\leq \max_a |R(s,a,b_1) + \gamma \sum_{s'} T(s,a,s') V^{(n)}(s', b_1^{s,a,s'}) \tag{35}$$
$$- R(s,a,b_2) + \gamma \sum_{s'} T(s,a,s') V^{(n)}(s', b_2^{s,a,s'})| \tag{36}$$
$$\leq \max_a |R(s,a,b_1) - R(s,a,b_2)| \tag{37}$$
$$+ \gamma \sum_{s'} T(s,a,s') |V^{(n)}(s', b_1^{s,a,s'}) - V^{(n)}(s', b_2^{s,a,s'})| \tag{38}$$
$$\leq (L_r + \gamma L_p L_{V^{(n)}}) \|b_1 - b_2\|_\infty \tag{39}$$

□

Proposition 2. *For a (L_r, L_p)-Lipschitz OAMDP, if $\gamma L_p < 1$, then V^* is L_{V^*}-Lipschitz continuous where:*

$$L_{V^*} = \frac{L_r}{1 - \gamma L_p}. \tag{11}$$

Proof. Consider a sequence $\{L_n\}_{n \geq 1}$ where $L_1 = L_r$ and:

$$L_{n+1} = L_r + \gamma L_p L_n \tag{40}$$

Then,

$$L_n = L_r + \gamma L_p L_r + (\gamma L_p)^2 L_r + \cdots + (\gamma L_p)^{n-1} L_r \tag{41}$$
$$= \frac{1 - (\gamma L_p)^n}{1 - \gamma L_p} L_r \tag{42}$$

By our assumption, $\gamma L_p < 1$, so the sequence converges. Let $L_{V^*} = \lim_{n \to \infty} L_n$. L_{V^*} must satisfy $L_{V^*} = L_r + \gamma L_p L_{V^*}$. Thus, we get Equation 11. □

Proposition 3. *If $\tau^{s,a,s'}(\theta) > 0$ for $\forall \theta \in \Theta$, $s, s' \in S$, and $a \in A$, belief transitions are Lipschitz continuous.*

Proof. Let $f^{s,a,s'}(b) = b^{s,a,s'} : \Delta^\Theta \to \Delta^\Theta$ be the belief transition after observing $\langle s, a, s' \rangle$. From the definition (Equation 1), $f^{s,a,s'}(b)(\theta_i) = \frac{\tau_i^{s,a,s'} b_i}{\sum_k \tau_k^{s,a,s'} b_k}$, where $\tau_i^{s,a,s'} = \tau^{s,a,s'}(\theta_i)$ and $b_i = b(\theta_i)$. Then we have:

$$J_{f^{s,a,s'}}(b)_{i,j} = \begin{cases} \frac{\tau_i^{s,a,s'}(\sum_{k \neq i} \tau_k^{s,a,s'} b_k)}{(\sum_k \tau_k^{s,a,s'} b_k)^2} & i = j, \\ \frac{-\tau_i^{s,a,s'} \tau_j^{s,a,s'} b_j}{(\sum_k \tau_k^{s,a,s'} b_k)^2} & i \neq j, \end{cases} \tag{43}$$

$$\|J_{f^{s,a,s'}}(b)\|_\infty = \max_{1 \leq i \leq n} \sum_{1 \leq j \leq n} |J_{f^{s,a,s'}}(b)_{i,j}|, \tag{44}$$

$$= \max_{1 \leq i \leq n} \frac{2\tau_i^{s,a,s'}(\sum_{k \neq j} \tau_k^{s,a,s'} b_k)}{(\sum_k \tau_k^{s,a,s'} b_k)^2}, \tag{45}$$

where J_f is the Jacobian of f and $\| \cdot \|_\infty$ is the induced operator norm. Let $\tau_{\min} = \min_{s,a,s',k} \tau_k^{s,a,s'}$ and $\tau_{\max} = \max_{s,a,s',k} \tau_k^{s,a,s'}$. Note that, for every $b \in \Delta^n$, $\sum_{k \neq i} \tau_k^{s,a,s'} b_k \leq \tau_{\max}$ and $\sum_k \tau_k^{s,a,s'} b_k \geq \tau_{\min} > 0$. Then we get $\|J_{f^{s,a,s'}}(b)\|_\infty \leq 2(\frac{\tau_{\max}}{\tau_{\min}})^2$. □

Lemma 1. *For an OAMDP with Lipschitz-continuous value function with the constant L_{V^*}, one-step approximation errors using a regular grid with resolution K are bounded as:*

$$\|\mathcal{T}_K V^* - V^*\|_\infty \leq \frac{L_{V^*}}{K}. \tag{17}$$

Proof. For all $n \geq 0, K \geq 1$, $s \in S$, and $b \in \Delta^{|\Theta|}$,

$$|V^*(s,b) - \mathcal{T}_K V^*(s,b)| \tag{46}$$

$$= |V^*(s,b) - \sum_{b_i \in P_K(b)} \lambda_i \mathcal{T} V^*(s,b_i)| \text{ (by definition)} \tag{47}$$

$$= |\sum_{b_i \in P_K(b)} \lambda_i (V^*(s,b) - V^*(s,b_i))| \text{ ($\mathcal{T}$ is a fixpoint of V^*)} \tag{48}$$

$$\leq \sum_{b_i \in P_K(b)} \lambda_i |V^*(s,b) - V^*(s,b_i)| \text{ (triangle inequality)} \tag{49}$$

$$\leq \sum_{b_i \in P_K(b)} \lambda_i L_{V^*} \|b - b_i\|_\infty \tag{50}$$

$$\leq L_{V^*} \frac{1}{K} \tag{51}$$

□

Proposition 5. *For an OAMDP whose value function is L_{V^*}-Lipschitz continuous, we have:*

$$\|V^* - V_K^*\|_\infty \leq \frac{L_{V^*}}{(1-\gamma)K}. \tag{18}$$

Proof.

$$\|V^* - V_K^*\|_\infty \tag{52}$$

$$\leq \|V^* - \mathcal{T}_K V^* + \mathcal{T}_K V^* - V_K^*\|_\infty \tag{53}$$

$$\leq \|V^* - \mathcal{T}_K V^*\|_\infty + \|\mathcal{T}_K V^* - \mathcal{T}_K V_K^*\|_\infty \tag{54}$$

$$\leq \frac{L_{V^*}}{K} + \gamma \|V^* - V_K^*\|_\infty \tag{55}$$

□

B PSEUDOCODE FOR GRID-LRTDP

Algorithm 2 shows the pseudocode for Grid-LRTDP. The algorithm operates identically to Grid-RTDP, except that at the end of each trial, the algorithm checks if states visited during the trial can be labeled as solved.

Algorithm 2 Grid-LRTDP

```
function GRID-LRTDP(s_0, b_0, ε, K)
    while ∃b_i ∈ P_K(b_0) ¬⟨s_0, b_0⟩.solved do
        LRTDPTRIAL(s_0, b_0, ε, K)
    end while
end function

function LRTDPTRIAL(s_0, b_0)
    visited ← Stack :: new()
    s ~ s_0
    b ← b_0
    while episode continues do
        sample b_i ∈ P_K(b) with the weight λ_i
        visited.push(⟨s, b_i⟩)
        a* ← min_a Q_K(s, b_i, a)
        V_K(s, b_i) ← Q_K(s, b_i, a*)
        s' ~ Pr(·|s, a*)
        b ← b_i^{s,a,s'}
    end while

    while ¬visited.is_empty() do
        ⟨s, b⟩ ← visited.pop()
        if ¬ CHECKSOLVED(s, b, ε, K) then
            break
        end if
    end while
end function
```

Algorithm 3 shows the procedure for labeling states. Starting from a given $\langle s, b \rangle$ the algorithm visits state that could be visited under the current best policy, and checks if the residuals of Bellman updates are smaller than a given threshold ϵ.

Algorithm 3 CHECKSOLVED

```
1: function CHECKSOLVED(s, b, ϵ, K)
2:     rv ← true
3:     open ← Stack :: new()
4:     closed ← Stack :: new()
5:     if ¬⟨s, b⟩.solved then
6:         open.push(⟨s, b⟩)
7:     end if
8:     while ¬open.is_empty() do
9:         ⟨s, b⟩ ← open.pop()
10:        closed.push(⟨s, b⟩)
11:        a* ← min_a Q_K(s, b, a)
12:        ϵ_res ← |V_K(s, b) − Q_K(s, b, a*|
13:        V_K(s, b) ← Q_K(s, b, a*)
14:        if ϵ_res < ϵ then
15:            continue
16:        end if
17:        for all s' ∈ S such that T(s, a, s') > 0 do
18:            for all b_i ∈ P_K(b^{s,a,s'}) such that λ_i > 0
    do
19:                if ¬⟨s', b_i⟩.solved ∧ ¬⟨s', b_i⟩ ∈ open ∧
    ¬⟨s', b_i⟩ ∈ closed then
20:                    open.push(⟨s', b_i⟩)
21:                end if
22:            end for
23:        end for
24:    end while
25:
26:    if rv = true then
27:        for all ⟨s, b⟩ ∈ closed do
28:            ⟨s, b⟩.solved ← true
29:        end for
30:    else
31:        while ¬closd.is_empty() do
32:            ⟨s, b⟩ ← open.pop()
33:            a* ← min_a Q_K(s, b, a)
34:            V_K(s, b) ← Q_K(s, b, a*)
35:        end while
36:    end if
37: end function
```