Tighter Value-Function Approximations for POMDPs

Merlijn Krale
Radboud University
Nijmegen, The Netherlands
merlijn.krale@ru.nl

Wietze Koops
Lund University, Sweden
University of Copenhagen, Denmark
wietze.koops@cs.lth.se

Sebastian Junges
Radboud University
Nijmegen, The Netherlands
sebastian.junges@ru.nl

Thiago D. Simão
Eindhoven University of Technology
The Netherlands
t.simao@tue.nl

Nils Jansen
Ruhr-University Bochum, Germany
Radboud University
Nijmegen, The Netherlands
n.jansen@rub.de

ABSTRACT

Solving partially observable Markov decision processes (POMDPs) typically requires reasoning about the values of exponentially many state beliefs. Towards practical performance, state-of-the-art solvers use value bounds to guide this reasoning. However, sound upper value bounds are often computationally expensive to compute, and there is a tradeoff between the tightness of such bounds and their computational cost. This paper introduces new and provably tighter upper value bounds than the commonly used *fast informed bound*. Our empirical evaluation shows that, despite their additional computational overhead, the new upper bounds accelerate state-of-the-art POMDP solvers on a wide range of benchmarks.

CCS CONCEPTS

• **Computing methodologies** → **Artificial intelligence**; **Planning under uncertainty**.

KEYWORDS

POMDPs, Heuristic Search, Value Bounds, Planning

ACM Reference Format:
Merlijn Krale, Wietze Koops, Sebastian Junges, Thiago D. Simão, and Nils Jansen. 2025. Tighter Value-Function Approximations for POMDPs. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 9 pages.

1 INTRODUCTION

Partially observable Markov decision processes (POMDPs) are a versatile modeling framework for stochastic environments where the decision maker (the agent) cannot fully observe the current state of its environment [26]. Finding optimal policies for POMDPs is generally undecidable [38]. Yet, in recent years, methods like POMCP [45], DESPOT [58], and AdaOPS [57] have been able to find policies for increasingly large POMDPs.

Although such methods often provide a (statistical) *lower bound* on the value of the policy, they are typically unable to find an *upper bound* on the optimal value. Such further certification of the quality of a policy may be essential for safety-critical problems. For example, planning medical treatments [24], scheduling infrastructure maintenance [39, 40] or computing safe flight paths [52] require us not only to know how well our policy will perform, but also that we cannot (reasonably) do any better.

So-called ϵ-optimal solvers such as SARSOP [33] and HSVI [49] compute both a policy and an upper bound. These algorithms make use of heuristic search to find good policies quickly. However, they often struggle to find upper bounds that are reasonably tight, since this requires reasoning over *all* possible policies.

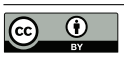
Both HSVI and SARSOP use the *fast informed bound*, or FIB [23], to initialize their upper bound computations. Intuitively, FIB computes values in a simplified POMDP, where the agent fully observes the state of the environment with a delay of one time step. However, these bounds are often loose in practice, while tighter upper bounds could improve the performance of ϵ-optimal solvers.

We contribute three different methods to obtain bounds that exhibit varying levels of tightness and computational overhead.

We first introduce the *tighter informed bound* (TIB) as an alternative for FIB. Intuitively, TIB uses a delay of two time steps rather than one time step. TIB can be computed using value iteration, as employed by [8, 44], on all *one-step beliefs*, that is, beliefs the agent can have one time step after knowing the state. These precomputations are more expensive than for FIB, but allow to compute a bound for any belief at the same computational cost as FIB. However, we show that increasing the delay further would significantly increase these computational costs.

Closer inspection of TIB shows that it expresses posterior beliefs of the agent as a convex combination of one-step beliefs. However, choosing *different* combinations may further tighten the bound. The *optimized tighter informed bound* (OTIB) uses the convex combinations that yield the tightest possible bound. However, finding this convex combination requires solving a linear program for each posterior belief in each iteration step, which is usually too expensive. Instead, the *entropy-based tighter informed bound* (ETIB) heuristically chooses a single combination for each posterior belief by maximizing the weighted entropy of the chosen one-step beliefs. This combination is reused for each iteration, thus greatly reducing computational cost.

Empirically, TIB and ETIB provide better bounds than FIB on a large range of benchmarks with reasonable computational cost. To test the practical relevance of our bounds, we adapt the offline



Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

state-of-the-art solver SARSOP [33] to use our upper bounds as initialization. With this alteration, SARSOP finds tighter optimality bounds more quickly on a wide range of benchmarks, which means the additional computational overhead of our bounds is compensated by a speedup in convergence. Moreover, this positive effect grows as the discount factor increases.

Contributions. To summarize, our **main contributions** are introducing three novel bounds for POMDPs, namely TIB, ETIB, and OTIB. These bounds both theoretically and empirically improve prior methods. Moreover, integrating these novel bounds with the state-of-the-art ϵ-optimal solver SARSOP [33] leads to significant speedups and smaller optimality gaps. Both our code [31] and appendices [30] are publically available.

2 PROBLEM SETTING

To start, we define our problem setting and provide our problem statement. We first introduce some basic notation: $\Delta(X)$ denotes the set of probability distributions over a finite set X. Given a function $F\colon X \to \Delta(Y)$ and elements $x \in X$, $y \in Y$, $F(\cdot \mid x)$ denotes the conditional probability distribution over Y given x, $F(y \mid x)$ the probability of element y given x, and $y \sim F(x)$ an element y randomly sampled from $F(x)$.

POMDPs. An (infinite-horizon, discounted) *partially observable Markov decision process* (POMDP) [26, 51] is defined as a tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, T, \mathcal{O}, O, R, \gamma \rangle$, with $\langle \mathcal{S}, \mathcal{A}, T, R, \gamma \rangle$ an MDP [44] with a finite set of *states* $\mathcal{S}$, a finite set of *actions* $\mathcal{A}$, a *transition function* $T\colon \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$, a *reward function* $R\colon \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, and a *discount factor* $\gamma \in (0, 1)$. Additionally, $\mathcal{O}$ is a finite set of *observations* and $O\colon \mathcal{A} \times \mathcal{S} \to \Delta(\mathcal{O})$ is the *observation function*.

A POMDP models the interaction between a stochastic environment and an agent. Let $b_0 \in \Delta(\mathcal{S})$ be the fixed *initial distribution* (aka *initial belief*). The *initial state* s_0 of the environment is sampled from b_0. At each time step t, the agent picks an action $a_t \in \mathcal{A}$. As a result, the environment transitions to a new state $s_{t+1} \sim T(\cdot \mid s_t, a_t)$ and returns a reward $r_t = R(s_t, a_t)$. However, unlike for MDPs, the agent does not observe the state s_{t+1}, but instead receives an observation $o_{t+1} \sim O(\cdot \mid a_t, s_{t+1})$. In general, agents make decisions based on their history $(b_0, a_0, o_1, \ldots, a_t, o_{t+1})$. As shown by Åström [4], this history can be summarized by a *belief* $b_t \in \Delta(\mathcal{S})$. Therefore, we can assume that the agent chooses actions according to a (deterministic) belief-based policy $\pi\colon \Delta(\mathcal{S}) \to \mathcal{A}$. Given a policy π and an initial belief b, we define the *value* as the expected discounted return over an infinite horizon $\mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t r_t | s_0 \sim b\right]$. The agent aims to maximize the value for the initial belief b_0.

Probabilities. We now introduce additional notation that will be used throughout this paper. Firstly, let $R_{\max} = \max_{(s,a)\in\mathcal{S}\times\mathcal{A}} R(s, a)$ and $R_{\min} = \min_{(s,a)\in\mathcal{S}\times\mathcal{A}} R(s, a)$ denote the maximal and minimal reward. For any belief $b \in \Delta(\mathcal{S})$, let $R(b, a) = \sum_{s\in\mathcal{S}} b(s) R(s, a)$ be the expected reward of action a in belief b and let $T(s' \mid b, a) = \sum_{s\in\mathcal{S}} b(s) T(s' \mid s, a)$ be the probability of transitioning to state s' when taking action a in belief b.

We define a shorthand for four probabilities. Given a state s and an action a, the probability of transitioning to state s' and observing o is denoted by $\Pr(s', o \mid s, a) = O(o \mid a, s') T(s' \mid s, a)$, while the probability of observing o is denoted by

$$\Pr(o \mid s, a) = \sum_{s'\in\mathcal{S}} \Pr(s', o \mid s, a).$$

Given a belief b and action a, we denote the probability of transitioning to s' and observing o by

$$\Pr(s', o \mid b, a) = \sum_{s\in\mathcal{S}} [b(s) \Pr(s', o \mid s, a)],$$

while the probability of observing o is given by

$$\Pr(o \mid b, a) = \sum_{s'\in\mathcal{S}} \Pr(s', o \mid b, a).$$

Beliefs. We also define notation for specific beliefs. For any $s \in \mathcal{S}$, let the *unit belief* $\mathfrak{b}_s$ be the belief such that $\mathfrak{b}_s(s) = 1$ (and hence $\mathfrak{b}_s(s') = 0$ for $s' \neq s$). Let $\mathcal{B}_\mathcal{S} = \{\mathfrak{b}_s \mid s \in \mathcal{S}\}$ be the set of all unit beliefs. If $\Pr(o \mid b, a) > 0$, $\mathfrak{b}_{b,a,o}$ is the belief after taking action a and observing o from belief b, i.e.:[1]

$$\mathfrak{b}_{b,a,o}(s') = \Pr(s' \mid b, a, o) = \frac{\sum_{s\in\mathcal{S}} b(s) \Pr(s', o \mid s, a)}{\Pr(o \mid b, a)}. \quad (1)$$

We write $\mathfrak{b}_{s,a,o} = \mathfrak{b}_{\mathfrak{b}_s,a,o}$, which denotes the belief reached from the unit belief $\mathfrak{b}_s$ (i.e., the belief where the agent knows the state s) in a single time step after executing a and observing o. We call $\mathfrak{b}_{s,a,o}$ a *one-step belief*. We define $\mathcal{B}_1$ as the (finite) set containing all one-step beliefs and the initial belief b_0, i.e.,

$$\mathcal{B}_1 = \{\mathfrak{b}_{s,a,o} \mid s \in \mathcal{S}, a \in \mathcal{A}, o \in \mathcal{O}, \Pr(o \mid s, a) > 0\} \cup \{b_0\}. \quad (2)$$

See Example 3.1 for a concrete example of sets $\mathcal{B}_\mathcal{S}$ and $\mathcal{B}_1$. We note that every reachable belief (except possibly b_0) can be written as a convex combination of one-step beliefs. Hence, all reachable beliefs can be written as a convex combination of beliefs in $\mathcal{B}_1$.

Q-values. Lastly, to reason about the decision-making process of an agent, we define the Q-value function $Q\colon \Delta(\mathcal{S}) \times \mathcal{A} \to \mathbb{R}$ as the value for a given belief-action pair. Let $\boldsymbol{Q}$ be the set of all functions $Q\colon \Delta(\mathcal{S}) \times \mathcal{A} \to \mathbb{R}$. The Q-value function corresponding to an optimal policy can be given as the (unique) fixed point of the *Bellman operator* $H_{\text{POMDP}}\colon \boldsymbol{Q} \to \boldsymbol{Q}$ [50]:

$$H_{\text{POMDP}} Q(b, a) = R(b, a) + \gamma \sum_{o\in\mathcal{O}} \Pr(o \mid b, a) \max_{a'\in\mathcal{A}} Q(\mathfrak{b}_{b,a,o}, a'). \quad (3)$$

Problem statement. With our problem setting defined, we formalize our problem statement as follows:

Problem Statement. *Find tractable methods of computing tight overapproximations (or* bounds*) of the Q-value function for POMDPs to improve the performance of ϵ-optimal solvers.*

3 PRIOR METHODS

In this section, we describe the baseline methods of finding upper bounds for POMDPs using the notation introduced in Sect. 2. We discuss the *fast informed bound* (FIB) [23], but define it using Q-functions. Then, we recall *point set bounds* [43] and show how FIB can be interpreted as a point set bound. Finally, we briefly review how upper bounds are used in the state-of-the-art solver SARSOP.

[1] For conciseness, we assume beliefs $\mathfrak{b}_{b,a,o}$ with $\Pr(o \mid b, a) = 0$ are arbitrarily defined, and that sums over observations consider only those observations that occur with non-zero probability.

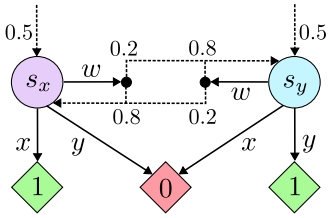


Figure 1: Visualisation of the Guessing POMDP.

First, we introduce the Guessing POMDP (Fig. 1), which we will use as a running example to illustrate the various upper bounds.

Example 3.1 (Guessing). In the Guessing POMDP (Fig. 1), the agent starts in an initial belief b_0, with $b_0(s_x) = b_0(s_y) = 0.5$. From here, the agent can *guess* in which state it is by taking actions x or y, which both lead to a terminal state s_{sink} (not depicted), which yields a reward of 1 if the state is guessed correctly and 0 otherwise. Alternatively, the agent can execute the *waiting* action w, which has a probability of 0.2 to transition to the other state and 0.8 to stay in the same state. All state-action pairs yield the same observation (denoted $\perp$), and we assume a discount factor $\gamma \in [0.9, 1)$. Since taking the waiting action w does not change the agent's belief and yields no reward, it is intuitively easy to see that an optimal policy is to pick action x (or action y), yielding an expected reward of 0.5.

In Guessing, the sets of unit- and one-step beliefs are given by:

$$\mathcal{B}_{\mathcal{S}} = \{\mathfrak{b}_{s_x},\ \mathfrak{b}_{s_y},\ \mathfrak{b}_{s_{\text{sink}}}\}$$
$$\mathcal{B}_1 = \{b_0\} \cup \{\mathfrak{b}_{s_x,w,\perp},\ \mathfrak{b}_{s_y,w,\perp},\ \mathfrak{b}_{s_x,x,\perp},\ \mathfrak{b}_{s_y,x,\perp},\ \mathfrak{b}_{s_x,y,\perp},\ \mathfrak{b}_{s_y,y,\perp},\ \mathfrak{b}_{s_{\text{sink}},w,\perp}\}.$$

We note that many beliefs in $\mathcal{B}_1$ describe the same state distribution: in fact, the last 5 elements are all equal to $\mathfrak{b}_{s_{\text{sink}}}$. However, throughout this paper, we will regard such beliefs as distinct members of this set for notational simplicity.

3.1 Fast Informed Bound (FIB)

A common method of over-approximating the value of a POMDP is to (partially) ignore the effect of partial observability. The most straightforward example of this is the *QMDP bound* [35], which intuitively corresponds to the assumption that agents can fully observe their state in the future. We can define this as follows:

Definition 3.2. Q_{MDP} is *the* fixed point of the operator H_{MDP}:

$$H_{\text{MDP}}Q(b,a) = R(b,a) + \gamma \cdot \sum_{s' \in \mathcal{S}} [\Pr(s'|b,a) \max_{a' \in \mathcal{A}} Q(\mathfrak{b}_{s'}, a')]. \quad (4)$$

To further tighten this bound, **the *fast informed bound* (FIB) [23] assumes an agent fully observes the current and future states with a delay of 1 time step**. More precisely, we define Q_{FIB}, the Q-value function for this bound, as follows:

Definition 3.3. Q_{FIB} is *the* fixed point of the operator H_{FIB}:

$$H_{\text{FIB}}Q(b,a) = R(b,a) + \gamma \cdot \sum_{o \in O} \max_{a' \in \mathcal{A}} \sum_{s' \in \mathcal{S}} \left[\Pr(o, s' \mid b, a) Q(\mathfrak{b}_{s'}, a')\right]. \quad (5)$$

App. B.2 [30] provides a proof that this fixed point exists and is unique (based on the original proof from Hauskrecht [23]).

In Eq. (5), the next action a' is picked independently of the next state s' but must depend only on the current belief b and received observation o. However, for all future time steps, we use Q-values computed for the unit belief $\mathfrak{b}_{s'}$, i.e., as if s' is revealed. Thus, the formula matches the intuitive description of full observability delayed by one time step. In contrast to Eq. (3), H_{FIB} depends only on the Q-values of the (finite) set of beliefs $\mathfrak{b}_s \in \mathcal{B}_{\mathcal{S}}$. Thus, the value of Q_{FIB} for *any* belief b can be computed efficiently by (approximately) computing the fixed point for beliefs in $\mathcal{B}_{\mathcal{S}}$. Both the QMDP bound and FIB are commonly used in POMDP literature due to their tractability but tend to be loose.

Running example. Recall the Guessing POMDP. Under the QMDP assumption, taking action w would fully reveal the agent's state. In that case, an agent can always guess correctly after taking action w, which yields an expected value of γ. Similarly, under the FIB assumption, taking action w would fully reveal the agent's previous state. The probability of still being in this state after this action is 0.8. Thus, taking action w and guessing the revealed initial state yields an expected return of 0.8γ. Both are strict overapproximations of the optimal value 0.5 of the POMDP, and both incorrectly give higher Q-values for action w than for x or y.

3.2 Point Set Bounds

To compute tighter approximations than FIB, we consider a general value bound that uses *point sets* [43]: sets of beliefs with known upper bounds. To make the connection with our own method more clear, we define them using our own (non-standard) notation. We start by defining a *weight function* as follows:

Definition 3.4. Let $b \in \Delta(\mathcal{S})$ be a belief and let $\mathcal{B} \subseteq \Delta(\mathcal{S})$ be a point set. A *weight function* $w\colon \mathcal{B} \to \mathbb{R}_{\geq 0}$ is any function satisfying $b(s) = \sum_{b' \in \mathcal{B}} w(b')b'(s)$ for all $s \in \mathcal{S}$. $\mathcal{W}_{\mathcal{B},b}$ denotes the set of all possible weight functions for belief b given point set $\mathcal{B}$.[2]

Intuitively, a weight function expresses a belief b as a convex combination of beliefs $b' \in \mathcal{B}$. We can use weight functions to compute upper bounds as follows:

Theorem 3.5 (Point set bound). *Given a belief b, a point set $\mathcal{B}$, and a function $Q\colon \mathcal{B} \times \mathcal{A} \to \mathbb{R}$ which over-approximates the Q_{POMDP}-values of all beliefs-action pairs $(b', a) \in \mathcal{B} \times \mathcal{A}$. Then, any weight function $w \in \mathcal{W}_{\mathcal{B},b}$ gives an upper bound on the value of b:*

$$Q_{\text{POMDP}}(b,a) \leq \widetilde{Q}(w,a) := \sum_{b' \in \mathcal{B}} w(b')Q(b',a). \quad (6)$$

This theorem follows directly from the convexity of the value function for POMDPs [50]. To understand how Theorem 3.5 is used implicitly by FIB, consider using point set $\mathcal{B}_{\mathcal{S}}$ and the weight functions $w_{b,a,o}(\mathfrak{b}_{s'}) = \frac{\Pr(o,s' \mid b,a)}{\Pr(o \mid b,a)}$. In that case, we find:

$$H_{\text{FIB}}Q(b,a) = R(b,a) + \gamma \sum_{o \in O} \max_{a' \in \mathcal{A}} \left[\Pr(o \mid b,a)\widetilde{Q}(w_{b,a,o}, a')\right], \quad (7)$$

with $\widetilde{Q}$ the weighted sum over values of Q_{FIB} as defined in Eq. (6).

[2] $\mathcal{W}_{\mathcal{B},b}$ is empty if b does not lie in the convex hull of $\mathcal{B}$.

Algorithm 1 Precomputations for TIB, OTIB and ETIB

Compute all (unique) beliefs in $\mathcal{B}_1$ ▷ *Eq.* (2)
for $b, a \in \mathcal{B}_1 \times \mathcal{A}$ **do**
  $Q'(b,a) \leftarrow Q_{\text{FIB}}(b,a)$ ▷ *Eq.* (5)
  $\forall o \in O$, precompute $\hat{w}_{b,a,o}$ ▷ *Eq.* (10), *only for ETIB*
for h iterations **do**
  $Q \leftarrow Q'$
  for $b \in \mathcal{B}_1, a \in \mathcal{A}$ **do**
    $Q'(b,a) \leftarrow HQ(b,a)$ ▷ *Eq.* (8), *Eq.* (9) *or Eq.* (11)
  if $\frac{\gamma}{1-\gamma} \max_{(b,a)\in\mathcal{B}_1\times\mathcal{A}} \frac{Q'(b,a)-Q(b,a)}{Q'(b,a)} < \epsilon$ **then** ▷ *Precision reached* [3]
    break
return Q'

Given a point set $\mathcal{B}$ and belief b, the *tightest* upper bound we can compute using Theorem 3.5 is found using the linear program (LP): $\min_{w\in\mathcal{W}_{\mathcal{B},b}} \widetilde{Q}(w,a)$. However, POMDP solvers often need to compute upper bounds for many beliefs using large point sets, in which case this method is computationally expensive. Thus, instead of solving these LPs exactly, solvers may approximate their outcome instead. One such approximation method is the *sawtooth bound* [23]. This bound is based on the observation that, for point sets of the form $\{b'\} \cup \mathcal{B}_{\mathcal{S}}$, an upper bound can be computed in only $O(|\mathcal{S}|)$ time. Thus, if $\mathcal{B}_{\mathcal{S}} \subseteq \mathcal{B}$, we can compute such a bound for all beliefs $b' \in \mathcal{B}$ and take their minimum. This takes only $O(|\mathcal{S}||\mathcal{B}|)$ time, but still yields tight bounds in practice. We refer to Kochenderfer et al. [29] for a detailed implementation of the sawtooth bound.

3.3 Using Bounds in Point-Based Solvers

Point set bounds are an important component of *point-based solvers*, a type of algorithm that uses a finite set of beliefs to compute both upper- and lower bounds on the value of a POMDP. Early methods use predefined sets of beliefs to cover the entire belief space evenly [9, 36, 60], but these methods typically scale poorly to large POMDPs. Instead, state-of-the-art algorithms such as HSVI [48, 49] and SARSOP [33] use *heuristic search* to find beliefs that closely resemble those encountered by an optimal policy, which is sufficient for finding ϵ-optimal solutions [33].

SARSOP [33] is a state-of-the-art point-based solver that uses a variant of value iteration [47] to compute lower bounds and the sawtooth bound for upper bounds. The latter requires precomputing value bounds for the set of unit beliefs $\mathcal{B}_{\mathcal{S}}$, which is traditionally done using FIB. The next section proposes methods of computing tighter bounds for this set in tractable time.

4 INTRODUCING TIGHTER BOUNDS

In this section, we introduce three novel bounds on the value function Q_{POMDP}, which are tighter than FIB.

4.1 Tighter Informed Bound (TIB)

Firstly, we propose an extension of FIB that extends the delay at which the full state is observed. More precisely, we define the ***tighter informed bound* (TIB), which assumes an agent fully observes the current and future states with a delay of 2 time steps.** We define the corresponding Q-value function, Q_{TIB}, as:

Definition 4.1. Q_{TIB} is *the* fixed point of the operator H_{TIB}:

$$H_{\text{TIB}}Q(b,a) = R(b,a) + \gamma \sum_{o\in O} \max_{a'\in\mathcal{A}} \sum_{s\in\mathcal{S}} \left[b(s) \Pr(o \mid s,a) Q(\mathfrak{b}_{s,a,o}, a') \right]. \quad (8)$$

We prove that this unique fixed point exists in App. B.3 [30].[4]

Recall from Eq. (5) that for FIB, we can compute Q-values for any belief using only Q-values of unit beliefs $\mathfrak{b}_{s'}$. This intuitively corresponds to state s' being observed with a delay of 1 time step. In contrast, in Eq. (8), we use Q-values of one-step beliefs $\mathfrak{b}_{s,a,o}$, which corresponds to state s being revealed with a delay of 2 time steps and thus aligns with the intuitive definition of TIB.

Next, we highlight some important properties of Q_{TIB}:

Theorem 4.2. Q_{TIB} *has the following properties:*
(1) ***Soundness:*** $\forall b \in \Delta(\mathcal{S}), a \in \mathcal{A}\colon Q_{\text{TIB}}(b,a) \geq Q_{\text{POMDP}}(b,a)$*;*
(2) ***Tightness:*** $\forall b \in \Delta(\mathcal{S}), a \in \mathcal{A}\colon Q_{\text{FIB}}(b,a) \geq Q_{\text{TIB}}(b,a)$.

Full proofs of Theorem 4.2 are provided in App. B.3 [30]. Intuitively, we recall that TIB and FIB correspond with the agent observing the state with a delay, which gives them additional information. Since this information can only help the agent, Q_{TIB} is a sound upper bound, showing (1). Moreover, since our model is Markovian and the delay of FIB is lower than that of TIB, the additional information of FIB is at least as useful. Thus, for any belief-action pair, Q_{TIB} is never larger than Q_{FIB}, showing (2).

Running example. To provide some intuition on the tightness of TIB, we recall the Guessing POMDP. Under the TIB assumption, taking action w twice lets an agent observe its initial state. The probability of still being in this state after these actions is $0.8^2 + 0.2^2 = 0.68$. Thus, taking action w twice and guessing the revealed initial state yields an expected return of $0.68\gamma^2$. This is a strict overapproximation of the optimal value of the POMDP, which is 0.5, but significantly tighter than the bound of 0.8γ found by FIB.

Complexity analysis. Q_{TIB} can be computed up to an arbitrary precision using value iteration, as shown in Algorithm 1. Table 1 shows the computational complexity of computing Q_{TIB} and Q_{FIB} using such methods. The computational costs for a single Bellman operation are equal for FIB and TIB, which also means the online computational costs are the same. Precomputations for TIB are a factor $O(|\mathcal{B}_1|/|\mathcal{S}|) \in O(|\mathcal{A}||O|)$ more expensive than for FIB. The empirical evaluation (Sect. 5) shows that this is often manageable in practice.

Further increasing delays. One method of computing even tighter bounds is to consider even longer observation delays. However, increasing the observation delay also means we need to consider a larger set of beliefs. Thus, as shown in App. C [30], a delay of 3 time steps yields a computational complexity of $O(|\mathcal{S}|^2|\mathcal{A}|^5|O|^4 h)$ (with h the number of iterations), which is a factor $O(|\mathcal{A}|^2|O|^2)$ larger than for TIB. It may be possible to efficiently compute such bounds regardless, but we will not consider this line of research

[3] For an explanation of the precision parameter, see App. B.6 [30]. Alternatively, when performing h iterations, the computed bound for any belief-action pair is at most $\frac{\gamma^h}{1-\gamma}(R_{\max} - R_{\min})$ away from the fixed point.

[4] This follows from showing H_{TIB} is a contraction mapping with Lipschitz constant $\gamma < 1$ and using Banach's fixed point theorem [5].

Table 1: Computational complexities of different bounds, assuming precomputations are performed using Algorithm 1 with h iterations. L denotes the computational complexity of solving an LP with $|\mathcal{B}_1|$ variables and $|\mathcal{S}|$ constraints.

Bound	Bellman Operation	Precomputations
FIB	$O(\|\mathcal{S}\|\|\mathcal{A}\|\|O\|)$	$O(\|\mathcal{S}\|^2\|\mathcal{A}\|^2\|O\|h)$
TIB	$O(\|\mathcal{S}\|\|\mathcal{A}\|\|O\|)$	$O(\|\mathcal{B}_1\|\|\mathcal{S}\|\|\mathcal{A}\|^2\|O\|h)$
OTIB	$O(\|\mathcal{A}\|\|O\|L)$	$O(\|\mathcal{B}_1\|\|\mathcal{A}\|^2\|O\|Lh)$
ETIB	$O(\|O\|(L+\|\mathcal{B}_1\|\|\mathcal{A}\|))$	$O(\|\mathcal{B}_1\|\|\mathcal{A}\|\|O\|(L+\|\mathcal{B}_1\|\|\mathcal{A}\|h))$

here. Instead, we focus on methods that can tighten our bounds without further increasing the delay.

4.2 Optimized Tighter Informed Bound (OTIB)

Like for FIB, we notice TIB can be rewritten using Theorem 3.5 with point set $\mathcal{B}_1$, as follows:

$$H_{\text{TIB}}Q(b,a) = R(b,a) + \gamma \sum_{o\in O} \max_{a'\in\mathcal{A}} \left[\Pr(o \mid b,a)\widetilde{Q}\left(w_{b,a,o}, a\right)\right],$$

where we use the following weight function:

$$w_{b,a,o}(\mathfrak{b}_{s,a,o}) = \frac{b(s)\Pr(o \mid s,a)}{\Pr(o \mid b,a)}.$$

In contrast to FIB, however, these weights are not necessarily unique, and Theorem 3.5 tells us *any* weight that represents our belief gives a viable upper bound. Thus, we define the ***optimized tighter informed bound* (OTIB), which assumes the value for future beliefs is equal to the minimal point set bound (Theorem 3.5) using point set $\mathcal{B}_1$.** We define the corresponding Q-value function as follows:

Definition 4.3. Write $\mathcal{W}_{b,a,o} = \mathcal{W}_{\mathcal{B}_1,\mathfrak{b}_{b,a,o}}$. Then, Q_{OTIB} is *the* fixed point of the operator H_{OTIB}:

$$H_{\text{OTIB}}Q(b,a) = R(b,a) + \gamma \sum_{o\in O} \max_{a'\in\mathcal{A}} [\Pr(o \mid b,a) \min_{w\in\mathcal{W}_{b,a,o}} \widetilde{Q}\left(w,a'\right)]. \tag{9}$$

We note that $w_{b,a,o} \in \mathcal{W}_{b,a,o}$, which means the minimization in Eq. (9) always has a feasible solution. A full proof of the existence and uniqueness of Q_{OTIB} is provided in App. B.4 [30]. We highlight a number of properties of Q_{OTIB}:

Theorem 4.4. *Q_{OTIB} has the following properties:*

(1) ***Soundness:*** $\forall b \in \Delta(\mathcal{S}), a \in \mathcal{A}\colon Q_{\text{OTIB}}(b,a) \geq Q_{\text{POMDP}}(b,a)$*;*
(2) ***Tightness:*** $\forall b \in \Delta(\mathcal{S}), a \in \mathcal{A}\colon Q_{\text{TIB}}(b,a) \geq Q_{\text{OTIB}}(b,a)$.

Proofs are provided in App. B.4 [30]. Similarly to Theorem 4.2, soundness follows by the fact that the agent is provided with extra information. Namely, the convex combination of beliefs defining $\widetilde{Q}\left(w,a'\right)$ effectively splits the belief b in beliefs that are (on average) more informative. Tightness follows from the observation that $w_{b,a,o} \in \mathcal{W}_{b,a,o}$, which means the weights used in Eq. (8) are a valid solution to the minimization in Eq. (9).

Running example. We consider the Guessing POMDP (Example 3.1). Under the OTIB assumption, the belief after taking action w can be expressed using any weight function in $\mathcal{W}_{b_0,w,\perp}$. In particular, since $b_0 \in \mathcal{B}_1$, one valid choice uses weight 1 for b_0 and 0 for all others. In that case, the action is suboptimal (with value 0.5γ), and the OTIB bound corresponds with the real value 0.5.

Complexity analysis. As for TIB, Q_{OTIB} can be approximated using Algorithm 1. OTIB and TIB use the same point set and thus require the same amount of Bellman operations per iteration. However, a single Bellman operation for OTIB is significantly more expensive since it requires solving an LP with at most $|\mathcal{B}_1|$ variables and $|\mathcal{S}|$ constraints. This yields the computational complexities shown in Table 1, where L denotes the complexity of solving such an LP. These computation costs are typically too high for practical use.

4.3 Entropy-based Tighter Informed Bound (ETIB): A Heuristic Approach

To reduce the complexity of the precomputations of OTIB, we consider using a single weight for each belief that we reuse for all iterations. More precisely, we approximate the worst-case weights by those that maximize the *weighted entropy*. This gives higher weights to more uncertain beliefs, which should intuitively give a tighter bound. To formalize this, we first define the *maximal entropy weight function* $\hat{w}_{b,a,o}$ for a belief $\mathfrak{b}_{b,a,o}$ as follows:

$$\hat{w}_{b,a,o} \in \arg\max_{w\in\mathcal{W}_{b,a,o}} \sum_{b'\in\mathcal{B}_\mathcal{S}} H(b')w(b'). \tag{10}$$

This equation always has a feasible solution, since $w_{b,a,o} \in \mathcal{W}_{b,a,o}$. Then, **the entropy-based tighter informed bound (ETIB) assumes the value for future beliefs is equal to the point set bound (Theorem 3.5) using point set $\mathcal{B}_1$ and maximal entropy weight functions (Eq. (10))**. We define the corresponding Q-value function as follows:

Definition 4.5. Q_{ETIB} is *the* fixed point of the operator H_{ETIB}:

$$H_{\text{ETIB}}Q(b,a) = R(b,a) + \gamma \sum_{o\in O} \max_{a'\in\mathcal{A}} [\Pr(o \mid b,a)\widetilde{Q}\left(\hat{w}_{b,a,o}, a'\right)]. \tag{11}$$

As for the other bounds, we provide a proof that this unique fixed point exists in App. B.5 [30]. We highlight the following properties of Q_{ETIB}:

Theorem 4.6. *Q_{ETIB} has the following properties:*

(1) ***Soundness:*** $\forall b \in \Delta(\mathcal{S}), a \in \mathcal{A}\colon Q_{\text{ETIB}}(b,a) \geq Q_{\text{POMDP}}(b,a)$*;*
(2) ***Tightness:*** $\forall b \in \Delta(\mathcal{S}), a \in \mathcal{A}\colon Q_{\text{FIB}}(b,a) \geq Q_{\text{ETIB}}(b,a)$.

The proof for Theorem 4.6 is provided in App. B.5 [30] and follows the same intuition as the proof of Theorem 4.2. In contrast to OTIB, we note that *ETIB does not necessarily provide tighter bounds than TIB*, since there is no guarantee $\hat{w}_{b,a,o}$ yields tighter bounds than $w_{b,a,o}$. In practice, however, we find ETIB is at least comparably tight as TIB, and sometimes (significantly) tighter.

Running example. Consider the Guessing environment. Under the ETIB assumption, the value of taking action w is approximated using the maximal entropy weight function $\hat{w}_{b_0,w,\perp}$. Since b_0 is the belief with the largest entropy in $\mathcal{B}_1$, this weight function is

Table 2: Upper bounds and computation times of different methods on a number of POMDP benchmarks. If a bound has not converged within 1200s, we report the last computed bound and denote computation time as TO. The tightest bounds are bolded. We include lower bounds computed by SARSOP as a proxy for the optimal value, with ϵ-optimal values underlined.

	Environment Properties					Baselines				Our methods					
Environment	$\lvert\mathcal{S}\rvert$	$\lvert\mathcal{A}\rvert$	$\lvert\mathcal{O}\rvert$	$\lvert\mathcal{B}_1\rvert$	$\lvert\mathcal{B}_2\rvert$	SARSOP		FIB		TIB		ETIB		OTIB	
Guessing	3	3	1	6	8	<u>0.50</u>	<1s	0.76	<1s	0.61	<1s	**0.50**	<1s	**0.50**	<1s
Tiger	2	3	2	3	5	<u>19.4</u>	<1s	87.2	<1s	49.6	<1s	**40.5**	<1s	**40.5**	<1s
Grid6x6	36	5	36	152	722	6.42	TO	8.31	<1s	8.15	<1s	7.25	<1s	**7.19**	16s
RockSample (5,3)	201	8	3	202	210	<u>16.9</u>	1s	**18.3**	<1s	**18.3**	<1s	**18.3**	<1s	**18.3**	<1s
RockSample (7,8)	13k	13	3	13k	13k	20.9	TO	28.5	105s	**27.2**	407s	**27.2**	411s	**27.2**	424s
K-out-of-N (2)	16	9	16	61	230	-1.75	TO	-1.24	<1s	**-1.52**	<1s	**-1.52**	<1s	**-1.52**	5s
K-out-of-N (3)	64	27	64	499	4.8k	-2.63	TO	-1.89	<1s	-2.28	5s	-2.28	10s	**-2.29**	386s
Aloha (30)	90	29	90	2.5k	202k	389	TO	394	4s	**392**	20s	**392**	935s	**392**	TO
Tag	842	5	30	2.4k	6.3k	-10.8	TO	-4.75	5s	-5.58	17s	-5.57	21s	**-5.64**	519s
TigerGrid	36	5	36	1.4k	100k	2.28	TO	2.73	<1s	2.58	28s	**2.57**	266s	**2.57**	TO
Hallway1	60	5	60	2.2k	147k	1.00	TO	1.29	2s	1.19	24s	**1.17**	395s	**1.17**	TO
Hallway2	92	5	92	3.4k	229k	0.34	TO	0.98	4s	0.89	50s	**0.88**	780s	**0.88**	TO
Pentagon	212	4	212	6.0k	447k	0.33	TO	0.38	3s	**0.38**	20s	**0.38**	655s	**0.38**	TO
Fourth	1.1k	4	1.1k	29k	2125k	0.06	TO	0.09	100s	**0.09**	434s	**0.09**	TO	**0.09**	TO

the weight function defined by $w(b) = 1$ if $b = b_0$, and $w(b) = 0$ otherwise. Thus, ETIB and OTIB find the same optimal bound.

Complexity analysis. As for our other proposed bounds, Q_{ETIB} can be approximated using Algorithm 1, with corresponding complexities shown in Table 1. The computational complexity of a single Bellman operation is a factor $O(|\mathcal{A}|)$ smaller for ETIB as compared to OTIB, since the same weight can be used for each next action a'. Moreover, since these weights can be reused at each iteration, the complexity for precomputations is significantly lower as well. In practice, the number of beliefs b' with $w_{b,a,o}(b') > 0$ is often much smaller than $|\mathcal{B}_1|$, in which case the iterations take significantly less time than the complexity bound suggests.

5 EMPIRICAL EVALUATION

In this section, we empirically evaluate the proposed bounds: TIB, ETIB, and OTIB. We address the following questions:

- **(Q1) Bounds tightness.** How do the proposed bounds compare to each other and prior bounds such as FIB? How close are these bounds to the optimal value?
- **(Q2) Computational cost.** What is the computational cost of these bounds? How do they scale with the POMDP size?
- **(Q3) Benefits for SARSOP.** Can these bounds improve the performance of POMDP solvers such as SARSOP?
- **(Q4) Discount dependency.** How does the effect of using these bounds in SARSOP depend on the discount factor?

Implementation & Baselines. We implement Algorithm 1 within the *POMDPs.jl* framework [16], and extend the Julia implementation of SARSOP [33] to use our bounds as initialization. As discussed in Sect. 3.3, we replace the bounds for $\mathcal{B}_{\mathcal{S}}$ with those computed by our methods. Unless stated otherwise, we use discount factor $\gamma = 0.95$ and compute bounds using relative precision $\epsilon = 10^{-3}$ with $h = 250$ maximum iterations. To ensure that our results are valid upper bounds, and to decrease computational costs, we use (looser) bounds as initializations. In particular, we initialize OTIB with ETIB, ETIB with TIB, TIB with FIB, FIB with QMDP, and QMDP with $\frac{1}{1-\gamma}R_{\max}$. App. A [30] provides further details, and all code and data is publically available [31].

Environments. For our experiments, we use several standard POMDP benchmarks: Tiger [12], RockSample [48], Aloha [25], Tag [43], TigerGrid [35], Hallway1 [35], Hallway2 [35], Pentagon [13] and Fourth [13].[5] These environments have diverse characteristics, varying from 2 to 12545 states, from 3 to 29 actions, and from 1 to 1052 observations. Table 2 shows these characteristics, as well as the number of one- and two-step beliefs, for each environment. Additionally, we consider the Guessing environment (Fig. 1) and two new environments inspired by problems in the literature. Firstly, we consider a 6×6 grid where an agent needs to navigate from the bottom left to the top right corner. The observation function is as in Amato et al. [3]: the agent observes in which column it is, but not in which row. Secondly, we consider a maintenance environment called K-out-of-N with the goal of keeping a number of components from breaking down [27]. We add partial observability using a *measuring action*, which gives a negative reward but reveals the current state, and assume the agent gets no observations otherwise (similar to, e.g., [7, 32, 41]). App. A.2 [30] provides a complete description of both new environments.

5.1 Bound Tightness and Computational Cost

To address questions (Q1) and (Q2), we compare the upper bounds for the initial beliefs of all environments. In addition, we show the bounds computed by FIB and the best lower bound found by SARSOP within 1200s, which we consider as the closest proxy for the optimal value when evaluating the tightness of the bounds.

OTIB is tight but computationally intractable. As shown in Table 2, OTIB is always the tightest upper bound. However, its

[5] All environments are publically available within *POMDPs.jl* or on pomdps.org.

Table 3: The tightest relative value gap found by SARSOP for different computational budgets with different bounds as initialization. N/A denotes no lower bound has been found within the given budget.

	600s			1200s			3600s		
Environment	FIB	TIB	ETIB	FIB	TIB	ETIB	FIB	TIB	ETIB
Grid6x6	0.16	0.16	**0.07**	0.12	0.12	**0.06**	0.09	0.08	**0.06**
RockSample (7,8)	**0.23**	0.27	0.25	0.19	**0.18**	0.20	**0.14**	0.15	**0.14**
K-out-of-N (3)	0.21	**0.12**	**0.12**	0.20	**0.11**	**0.11**	0.18	**0.10**	**0.10**
Tag	0.44	**0.40**	**0.40**	0.43	**0.39**	**0.39**	0.41	**0.37**	**0.37**
TigerGrid	0.11	**0.09**	**0.09**	0.11	**0.08**	**0.08**	0.10	**0.08**	**0.08**
Hallway1	0.22	**0.16**	0.17	0.21	**0.16**	**0.16**	0.21	**0.15**	**0.15**
Hallway2	1.77	**1.53**	N/A	1.66	**1.47**	1.57	1.56	**1.35**	1.37
Pentagon	0.19	**0.14**	N/A	0.15	**0.10**	0.16	0.12	**0.08**	0.11
Fourth	**0.66**	0.92	N/A	**0.57**	0.62	N/A	0.44	**0.41**	1.38

Table 4: Computation times of SARSOP for different environments, using different heuristics as initialization.

Environment	FIB	TIB	ETIB
Guessing	<1s	<1s	<1s
Tiger	<1s	<1s	<1s
RockSample (5,3)	<1s	<1s	<1s
K-out-of-N (2)	612s	101s	100s
Aloha (30)	45s	40s	945s

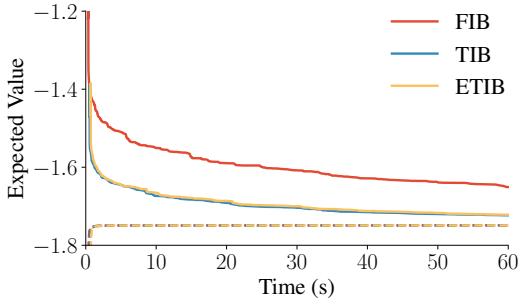


Figure 2: Upper- and lower bounds on the initial value of the K-out-of-N (2) environments as computed by SARSOP in the first 60s, using different bounds as initialization. Solid lines show upper bounds, and dashed lines lower bounds.

computation times are significantly higher than of the other tested bounds, and for larger environments it often does not converge within the given time.

TIB and ETIB are tighter than FIB, with tractable overhead. TIB and ETIB are tighter than FIB in all environments at the cost of longer, but mostly tractable, computation times. The differences in the bounds are the largest for Guessing, Tiger, and Grid6x6, where ETIB performs significantly better than TIB and about on par with OTIB. However, the results for Tag show that ETIB is not guaranteed to outperform TIB, and for most other environments the difference between TIB and ETIB is minimal. The difference between FIB and our proposed bound is small in environments where all uncertainty is contained in the initial state, as is the case for RockSample.[6]

5.2 Improvement of SARSOP

Next, we investigate question (Q3) by comparing the performance of SARSOP when using FIB, TIB, and ETIB as initialization. For our evaluation, we split the environments into two groups. For the smaller environments where SARSOP finds an ϵ-optimal policy within one hour, we consider convergence times. For the larger environments, where SARSOP does not converge within an hour, we instead consider the relative value gap

$$V_{\text{gap}} = \frac{\overline{V}(b_0) - \underline{V}(b_0)}{\left|\underline{V}(b_0)\right|}$$

[6]For RockSample (7,8), the difference in computation times between TIB, ETIB, and OTIB is almost exclusively caused by system variance (as caused by different memory allocations, garbage collection timing, etc.).

after 600s, 1200s and 3600s. We also provide the upper- and lower bounds in App. B.4 [30]. All running times include the precomputation times of the bounds.

In smaller environments, using TIB or ETIB to initialize SARSOP yields mixed results. As shown in Table 4, for most small environments, SARSOP is already sufficiently fast that the initialization has little effect on computation times. The exceptions are K-out-of-N (2), where using TIB and ETIB is significantly quicker than using FIB, and Aloha (30), where ETIB is significantly slower. For K-out-of-N (2), Fig. 2 shows the upper- and lower bounds as computed by SARSOP over time. We see that when using FIB, an initial upper bound is computed slightly quicker, but the convergence speed is worse than when using TIB or ETIB.

In larger environments, using TIB improves the bounds computed by SARSOP. Table 3 shows the tightest relative value gap found with different computational budgets for our larger environments. We see that using TIB typically improves the performance of SARSOP given a sufficiently large computational budget. However, for environments where TIB is computationally expensive (such as RockSample (7,8) and Fourth), we find that using FIB (initially) yields better results. In contrast, using ETIB yields better results for Grid6x6, but otherwise performs similar or worse then TIB due to its higher computational cost.

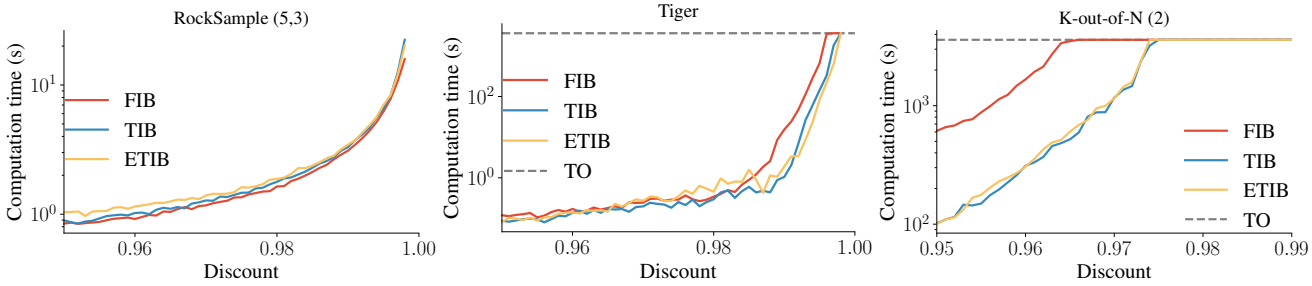


Figure 3: Computation times of SARSOP against the discount factor, using different bounds as initialization.

5.3 Discount Dependency

Lastly, we investigate question (Q4) by testing how the discount factor affects the computation times of SARSOP, given different initialization bounds. Due to the computational cost, we only test on smaller environments, but we expect this behavior to translate to larger environments as well.

SARSOP profits more from tighter initial bounds for high discount factors. As shown in Fig. 3, the effect of different initialization is minimal for goal-oriented environments, such as RockSample (5,3). However, for non-goal-oriented problems (such as Tiger and K-out-of-N), we find that the absolute speedup of using TIB and ETIB increases with the discount factor.

6 RELATED WORK

Besides QMDP [35], FIB [23] and point-based methods [9, 36, 43, 48, 49, 60], which we introduced in Sect. 2, we mention a number of other methods used for computing upper bounds. Firstly, a number of works consider simplifying the set of reachable beliefs by discretizing the belief space in a similar style as point-based solvers [10, 11, 18, 42, 56]. Next, Yoon et al. [59] introduce 'hindsight optimization', which uses deterministic planning in a number of sampled 'situations' to approximate the value of a (PO)MDP. Haugh and Lacedelli [21] use 'information relaxation' in a similar way. Barenboim and Indelman [6] consider only a subset of possible outcomes of the transition- and observation function to compute upper bounds in online solvers. However, all these methods are typically less tight (though computationally cheaper) than our proposed bounds. Lastly, some bounds are based on the properties of a particular type of POMDP. For example, Sinuany-Stern et al. [46] consider POMDPs that model maintenance, while Krale et al. [32] consider POMDPs where agents have explicit measuring actions.

Our empirical analysis focuses on SARSOP [33], but we mention a few related state-of-the-art POMDP solvers. Firstly, POMCP [45], and AdaOPS [57] are both variants of Monte Carlo tree search (MCTS) adapted for POMDPs. DESPOT [58] is also based on tree search but uses hindsight optimization to increase tractability. Lastly, many methods make use of (deep) reinforcement learning to find approximate solutions to POMDPs [19, 22, 34]. However, all these methods focus on large (continuous) state-, action- and observation spaces where our bounds are computationally intractable.

Deterministic Delay MDPs (DDMDPs) [2, 28, 54] are MDPs where the agent can fully observe its state with some constant delay, which is conceptually similar to FIB and TIB. Finding exact solutions to DDMDPs is NP-hard [54], but efficient approximate solvers exist [1, 14]. However, FIB and TIB take into account (partial) observations occurring before the state is fully revealed, while such observations do not exist in DDMDPs. This means that in POMDPs with no observations, FIB and TIB correspond to the solutions of DDMDPs with delays 1 and 2, respectively. However, solutions for DDMDPs are not sound upper bounds for POMDPs in general, so we do not compare our method with DDMDP solvers.

Lastly, we mention a number of other works related to ϵ-optimal POMDP solving. Walraven and Spaan [53] and Hansen and Bowman [20] propose methods to speed up the incremental pruning of α-vectors, which constitutes a considerable amount of the computation time of SARSOP. Relatedly, Dujardin et al. [15] propose a method that uses less α-vectors instead. Wang et al. [55] proposes to use quadratic functions instead of piecewise-linear functions to represent the upper bound.

7 CONCLUSION

To improve the performance of ϵ-optimal solvers, we introduced three novel bounds for POMDPs (TIB, OTIB, and ETIB). We prove these bounds are tighter than the commonly used FIB, and show they can be computed using value iteration. Empirically, both TIB and ETIB are computationally tractable on a large range of benchmarks. Moreover, using these bounds to initialize state-of-the-art solver SARSOP improves its performance.

Future work may focus on increasing the tractability of our bounds. For example, instead of computing bounds for all beliefs $b \in \mathcal{B}_1$, it may be quicker to use FIB for those that have a low probability of being reached. Alternatively, more research could be done on different heuristic choices for weights, particularly choices that do not require solving LPs. We consider and test one such choice in App. A.3 [30] with limited success, but using different (combinations of) heuristic(s) could potentially get tight bounds at lower computational costs than ETIB. Lastly, future work could consider how our bounds can be applied to other settings, such as finite- or indefinite horizon problems.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their useful comments. This work has been partially funded by the ERC Starting Grant DEUCE (101077178), and by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

REFERENCES

[1] Mridul Agarwal and Vaneet Aggarwal. 2021. Blind decision making: Reinforcement learning with delayed observations. *Pattern Recognit. Lett.* 150 (2021).

[2] Eitan Altman and Philippe Nain. 1992. Closed-Loop Control with Delayed Information. In *SIGMETRICS*. ACM, 193–204.

[3] Christopher Amato, Daniel S. Bernstein, and Shlomo Zilberstein. 2006. Optimal fixed-size controllers for decentralized POMDPs. In *AAMAS MSDM workshop*.

[4] Karl Johan Åström. 1965. Optimal control of Markov processes with incomplete state information. *J. Math. Anal. Appl.* 10, 1 (1965), 174–205.

[5] Stefan Banach. 1922. Sur les opérations dans les ensembles abstraits et leur application aux équations intégrales. *Fundam. math.* 3, 1 (1922), 133–181.

[6] Moran Barenboim and Vadim Indelman. 2023. Online POMDP Planning with Anytime Deterministic Guarantees. In *NeurIPS*. 79886–79902.

[7] Colin Bellinger, Rory Coles, Mark Crowley, and Isaac Tamblyn. 2021. Active Measure Reinforcement Learning for Observation Cost Minimization. In *Canadian Conference on AI*. Canadian Artificial Intelligence Association.

[8] Richard Bellman. 1957. A Markovian Decision Process. *Indiana Univ. Math. J.* 6 (1957), 679–684. Issue 4.

[9] Blai Bonet. 2002. An epsilon-Optimal Grid-Based Algorithm for Partially Observable Markov Decision Processes. In *ICML*. Morgan Kaufmann, 51–58.

[10] Alexander Bork, Sebastian Junges, Joost-Pieter Katoen, and Tim Quatmann. 2020. Verification of Indefinite-Horizon POMDPs. In *ATVA (Lecture Notes in Computer Science, Vol. 12302)*. Springer, 288–304.

[11] Alexander Bork, Joost-Pieter Katoen, and Tim Quatmann. 2022. Under-Approximating Expected Total Rewards in POMDPs. In *TACAS (2) (Lecture Notes in Computer Science, Vol. 13244)*. Springer, 22–40.

[12] Anthony R. Cassandra, Leslie P. Kaelbling, and Michael L. Littman. 1994. Acting Optimally in Partially Observable Stochastic Domains. In *AAAI*. 1023–1028.

[13] Anthony R. Cassandra, Michael L. Littman, and Nevin Lianwen Zhang. 1997. Incremental Pruning: A Simple, Fast, Exact Method for Partially Observable Markov Decision Processes. In *UAI*. 54–61.

[14] Esther Derman, Gal Dalal, and Shie Mannor. 2021. Acting in Delayed Environments with Non-Stationary Markov Policies. In *ICLR*.

[15] Yann Dujardin, Tom Dietterich, and Iadine Chades. 2017. Three New Algorithms to Solve N-POMDPs. In *AAAI*. 4495–4501.

[16] Maxim Egorov, Zachary N. Sunberg, Edward Balaban, Tim A. Wheeler, Jayesh K. Gupta, and Mykel J. Kochenderfer. 2017. POMDPs.jl: A Framework for Sequential Decision Making under Uncertainty. *JMLR* 18, 26 (2017), 1–5.

[17] John Forrest, Stefan Vigerske, Ted Ralphs, Lou Hafer, J. P. Fasano, Haroldo Gambini Santos, Jan-Willem Goossens, Matthew Saltzman, Bjarni Kristjansson, H. I. Gassmann, Alan King, Bohdan Mart, Pierre Bonami, Ruan Luies, Samuel Brito, and others. 2024. COIN-OR/Clp 1.17.10. https://doi.org/10.5281/zenodo.13347196

[18] Divya Grover and Christos Dimitrakakis. 2021. Adaptive Belief Discretization for POMDP Planning. *CoRR* abs/2104.07276 (2021).

[19] Dongqi Han, Kenji Doya, and Jun Tani. 2020. Variational Recurrent Models for Solving Partially Observable Control Tasks. In *ICLR*.

[20] Eric A. Hansen and Thomas Bowman. 2020. Improved Vector Pruning in Exact Algorithms for Solving POMDPs. In *UAI*. 1258–1267.

[21] Martin B. Haugh and Octavio Ruiz Lacedelli. 2020. Information Relaxation Bounds for Partially Observed Markov Decision Processes. *IEEE Trans. Autom. Control.* 65, 8 (2020), 3256–3271.

[22] Matthew J. Hausknecht and Peter Stone. 2015. Deep Recurrent Q-Learning for Partially Observable MDPs. In *AAAI Fall Symposia*. AAAI Press, 29–37.

[23] Milos Hauskrecht. 2000. Value-Function Approximations for Partially Observable Markov Decision Processes. *JAIR* 13 (2000), 33–94.

[24] Milos Hauskrecht and Hamish S. F. Fraser. 2000. Planning treatment of ischemic heart disease with partially observable Markov decision processes. *Artif. Intell. Medicine* 18, 3 (2000), 221–244.

[25] Wha Sook Jeon, Seung Beom Seo, and Dong Geun Jeong. 2022. POMDP-Based Contention Resolution for Framed Slotted-ALOHA Protocol in Machine-Type Communications. *IEEE Internet Things J.* 9, 15 (2022), 13511–13523.

[26] Leslie P. Kaelbling, Michael L. Littman, and Anthony R. Cassandra. 1998. Planning and Acting in Partially Observable Stochastic Domains. *Artif. Intell.* 101, 1-2 (1998), 99–134.

[27] Kailash Kapur and Michael Pecht. 2014. *Reliability Engineering*. John Wiley.

[28] Konstantinos V. Katsikopoulos and Sascha E. Engelbrecht. 2003. Markov decision processes with delays and asynchronous cost collection. *IEEE Trans. Autom. Control.* 48, 4 (2003), 568–574.

[29] Mykel J. Kochenderfer, Tim A. Wheeler, and Kyle H. Wray. 2022. *Algorithms for Decision Making*. MIT Press.

[30] Merlijn Krale, Wietze Koops, Sebastian Junges, Thiago D. Simão, and Nils Jansen. 2025. Tighter Value-Function Approximations for POMDPs. arXiv:2502.06523 https://arxiv.org/abs/2502.06523

[31] Merlijn Krale, Wietze Koops, Sebastian Junges, Thiago D. Simão, and Nils Jansen. 2025. *Tighter Value-Function Approximations for POMDPs: Code and Data*. https://doi.org/10.5281/zenodo.14848997

[32] Merlijn Krale, Thiago D. Simão, and Nils Jansen. 2023. Act-Then-Measure: Reinforcement Learning for Partially Observable Environments with Active Measuring. In *ICAPS*. AAAI Press, 212–220.

[33] Hanna Kurniawati, David Hsu, and Wee Sun Lee. 2008. SARSOP: Efficient Point-Based POMDP Planning by Approximating Optimally Reachable Belief Spaces. In *Robotics: Science and Systems*. The MIT Press.

[34] Alex X. Lee, Anusha Nagabandi, Pieter Abbeel, and Sergey Levine. 2020. Stochastic Latent Actor-Critic: Deep Reinforcement Learning with a Latent Variable Model. In *NeurIPS*. 741–752.

[35] Michael L. Littman, Anthony R. Cassandra, and Leslie P. Kaelbling. 1995. Learning Policies for Partially Observable Environments: Scaling Up. In *ICML*. Morgan Kaufmann, 362–370.

[36] William S. Lovejoy. 1991. Computationally Feasible Bounds for Partially Observed Markov Decision Processes. *Oper. Res.* 39, 1 (1991), 162–175.

[37] Miles Lubin, Oscar Dowson, Joaquim Dias Garcia, Joey Huchette, Benoît Legat, and Juan Pablo Vielma. 2023. JuMP 1.0: Recent improvements to a modeling language for mathematical optimization. *Mathematical Programming Computation* (2023). https://doi.org/10.1007/s12532-023-00239-3

[38] Omid Madani, Steve Hanks, and Anne Condon. 1999. On the Undecidability of Probabilistic Planning and Infinite-Horizon Partially Observable Markov Decision Problems. In *AAAI/IAAI*. AAAI Press / The MIT Press, 541–548.

[39] Pablo G. Morato, Konstantinos G. Papakonstantinou, Charalampos P. Andriotis, Jannie Sønderkær Nielsen, and Philippe Rigo. 2022. Optimal inspection and maintenance planning for deteriorating structural components through dynamic Bayesian networks and Markov decision processes. *Structural Safety* 94 (2022), 102140.

[40] Pablo G. Morato, Konstantinos G. Papakonstantinou, Charalampos P. Andriotis, and Philippe Rigo. 2022. Managing offshore wind turbines through Markov decision processes and dynamic Bayesian networks. In *13th International Conference on Structural Safety & Reliability (ICOSSAR)*.

[41] Hyunji Alex Nam, Scott L. Fleming, and Emma Brunskill. 2021. Reinforcement Learning with State Observation Costs in Action-Contingent Noiselessly Observable Markov Decision Processes. In *NeurIPS*. 15650–15666.

[42] Gethin Norman, David Parker, and Xueyi Zou. 2017. Verification and control of partially observable probabilistic systems. *Real Time Syst.* 53, 3 (2017), 354–402.

[43] Joelle Pineau, Geoffrey J. Gordon, and Sebastian Thrun. 2003. Point-based value iteration: An anytime algorithm for POMDPs. In *IJCAI*. 1025–1032.

[44] Martin L. Puterman. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley.

[45] David Silver and Joel Veness. 2010. Monte-Carlo Planning in Large POMDPs. In *NIPS*. Curran Associates, Inc., 2164–2172.

[46] Zilla Sinuany-Stern, Israel David, and Sigal Biran. 1997. An efficient heuristic for a partially observable Markov decision process of machine replacement. *Comput. Oper. Res.* 24, 2 (1997), 117–126.

[47] Richard D. Smallwood and Edward J. Sondik. 1973. The Optimal Control of Partially Observable Markov Processes over a Finite Horizon. *Oper. Res.* 21, 5 (1973), 1071–1088.

[48] Trey Smith and Reid G. Simmons. 2004. Heuristic Search Value Iteration for POMDPs. In *UAI*. 520–527.

[49] Trey Smith and Reid G. Simmons. 2005. Point-Based POMDP Algorithms: Improved Analysis and Implementation. In *UAI*. 542–547.

[50] Edward J. Sondik. 1978. The Optimal Control of Partially Observable Markov Processes over the Infinite Horizon: Discounted Costs. *Oper. Res.* 26, 2 (1978).

[51] Matthijs T. J. Spaan. 2012. Partially Observable Markov Decision Processes. In *Reinforcement Learning*, Marco A. Wiering and Martijn van Otterlo (Eds.). Adaptation, Learning, and Optimization, Vol. 12. Springer, 387–414.

[52] Selim Temizer, Mykel Kochenderfer, Leslie Kaelbling, Tomas Lozano-Pérez, and James Kuchar. 2010. Collision avoidance for unmanned aircraft using Markov decision processes. In *AIAA guidance, navigation, and control conference*.

[53] Erwin Walraven and Matthijs T. J. Spaan. 2017. Accelerated Vector Pruning for Optimal POMDP Solvers. In *AAAI*. 3672–3678.

[54] Thomas J. Walsh, Ali Nouri, Lihong Li, and Michael L. Littman. 2009. Learning and planning in environments with delayed feedback. *Auton. Agents Multi Agent Syst.* 18, 1 (2009), 83–105.

[55] Tao Wang, Pascal Poupart, Michael H. Bowling, and Dale Schuurmans. 2006. Compact, Convex Upper Bound Iteration for Approximate POMDP Planning. In *AAAI*. AAAI Press, 1245–1252.

[56] Kyle Hollins Wray and Shlomo Zilberstein. 2017. Approximating reachable belief points in POMDPs. In *IROS*. IEEE, 117–122.

[57] Chenyang Wu, Guoyu Yang, Zongzhang Zhang, Yang Yu, Dong Li, Wulong Liu, and Jianye Hao. 2021. Adaptive Online Packing-guided Search for POMDPs. In *NeurIPS*. 28419–28430.

[58] Nan Ye, Adhiraj Somani, David Hsu, and Wee Sun Lee. 2017. DESPOT: Online POMDP Planning with Regularization. *J. Artif. Intell. Res.* 58 (2017), 231–266.

[59] Sung Wook Yoon, Alan Fern, Robert Givan, and Subbarao Kambhampati. 2008. Probabilistic Planning via Determinization in Hindsight. AAAI Press.

[60] Rong Zhou and Eric A. Hansen. 2001. An Improved Grid-Based Approximation Algorithm for POMDPs. In *IJCAI*. 707–716.