MDPs with a State Sensing Cost

Vansh Kapoor
IIT Bombay
vanshk@cs.cmu.edu

Jayakrishnan Nair
IIT Bombay
jayakrishnan.nair@ee.iitb.ac.in

Abstract

In many practical sequential decision-making problems, tracking the state of the environment incurs a sensing/computation cost. In these settings, the agent's interaction with its environment includes the additional component of deciding *when* to sense the state, in a manner that balances the value associated with optimal (state-specific) actions and the cost of sensing. We formulate this as an expected discounted cost Markov Decision Process (MDP), wherein the agent incurs an additional cost for sensing its next state, but has the option to take actions while remaining 'blind' to the system state. We pose this problem as a classical discounted cost MDP with an expanded (countably infinite) state space. While computing the optimal policy for this MDP is intractable in general, we derive lower bounds on the optimal value function, which allow us to bound the suboptimality gap of any policy. We also propose a computationally efficient algorithm SPI, based on policy improvement, which in practice performs close to the optimal policy. Finally, we benchmark against the state-of-the-art via a numerical case study.

1 INTRODUCTION

Markov Decision Processes (MDPs) provide a powerful framework for modeling sequential interactions between an agent and an adaptive environment that 'responds' to the agent's actions. In the classical MDP formulation, at each time step t, the agent sees the environment's state, denoted X_t, and selects an action A_t. This action generates a feedback signal (cost) determined by the state-action pair (X_t, A_t), and triggers a random, action-dependent Markovian transition in the state. However, in many applications, 'seeing' the current state involves a cost. For example:

• **Healthcare:** Monitoring a patient's state during an ongoing intervention, such as white blood cell counts in anti-HIV therapy or lab tests for ICU patients, incurs monetary or delay costs (Ernst et al., 2006).
• **Mobile Applications:** Sensing a user's location, motion, or environment on mobile devices incurs energy costs, which must be balanced with user experience goals (Wang et al., 2010).
• **Wireless Sensor Networks:** Sensing the network state might require turning on battery-operated (and therefore energy-constrained) sensors.
• **Distributed Sensing:** Aggregating sensor data to determine system state involves computational costs.
• **Remote Surveillance:** Transmitting state information (in the form of images/video) to a controller incurs communication costs.
• **Robotics:** In applications where an autonomous robot is engaged in a task, sensing the surroundings might induce a latency (cost) in task completion.

This motivates integrating the cost of state sensing and a model for opportunistic state sensing into the MDP framework. However, such integration introduces technical challenges. In particular, if the agent chooses not to sense the system state (while continuing to interact with the environment), it is left with a *belief distribution* over the system state. Incorporating this belief into the MDP formulation induces a state space explosion, which makes the decision problem intractable.

In this paper, we formulate a cost-sensing MDP built on top of a finite, infinite-horizon discounted cost baseline MDP. The 'augmented' MDP, which incorporates the state sensing cost, has a countably infinite state space. At each time/epoch, the agent must, in addition to taking an action, decide whether or not to sense the *next* state of the MDP. If it decides to sense, it incurs an additional state sensing cost in that epoch. There is thus a non-trivial trade-off between the cost induced by suboptimal actions (under state uncertainty) and

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300. Copyright 2026 by the author(s).

the cost of state sensing.

While the 'augmented' MDP admits a stationary Markov policy, it is computationally intractable given the infinite state space. The goal of this paper is to design tractable, near-optimal algorithms for solving it. Our key contributions are as follows:

- We propose SPI, a policy-improvement-based algorithm that selectively searches for improving blind action sequences and is near-optimal in practice.
- We analyse a sequence of truncated (and therefore finite) MDPs that restrict the number of consecutive non-state-sensing (a.k.a., blind) actions the agent can take. We provide a sufficient condition for the optimal policy under such a truncated MDP to also be optimal for the original (infinite state space) MDP of interest.
- We utilize the truncated MDPs to derive lower bounds on the optimal value function, enabling computation of the suboptimality gap for any policy.
- We derive a computable state-sensing cost threshold, below which always sensing is optimal.
- Finally, we conduct an extensive numerical case study, including a real-world healthcare application, demonstrating that SPI consistently outperforms state-of-the-art POMDP solvers under diverse sensing costs, while maintaining reasonable compute time.

It is worth noting that our formulation can be posed as a Partially Observable Markov Decision Processes (POMDP) and indeed we benchmark our methods against state-of-the-art POMDP solvers (see Section 5). However, in doing so, one loses the specific problem structure that arises in the opportunistic state sensing formulation, which we seek to exploit here for computational tractability. Since POMDPs are known to be intractable in general (Papadimitriou and Tsitsiklis, 1987), leveraging this structure is crucial. Specifically, our algorithms and analytical results exploit the specific structure of this formulation, namely that the uncertainty in the belief distribution can be collapsed entirely by paying a sensing cost, allowing a good policy to go 'blind' until the cost induced by state uncertainty is outweighed by the sensing cost. A generic MDP/POMDP solver would not be able to exploit this structure. Moreover, our algorithmic design and proof arguments rely on classical MDP tools–policy improvement and the Bellman operator–whose POMDP analogues are significantly weaker.

Related Literature. Formulations equivalent to ours have been analysed in (Hansen, 1994; Bellinger et al., 2021; Nam et al., 2021; Krale et al., 2023); the last two references refer to this formulation as an *Action-Contingent-Noiseless-Observable MDP*, or ACNO-MDP. Hansen (1994) proposes a truncation-based approximation analogous to that in Section 4.2, except they provide no approximation guarantees. (By contrast, Chen and Liew (2025) develops a truncation-based approach with theoretical guarantees for a simpler, related setting where the state uncertainty is exogenous.) Nam et al. (2021); Bellinger et al. (2021) focus on *reinforcement learning* (RL) (as opposed to the planning problem considered here). Specifically, Nam et al. (2021) focuses on developing RL algorithms for a fixed-horizon setting using the generic POMDP solver POMCP (Silver and Veness, 2010). On the other hand, Bellinger et al. (2021) adapts Q-learning for this setting by utilizing a statistical state estimator to achieve a "higher costed return" – for every non-state-sensing action, the subsequent state is simply sampled from the belief distribution. An ϵ-greedy action is then taken based on the sampled state to update the Q-table, without leveraging any structure of the belief distribution while choosing the action. Krale et al. (2023) proposes a policy improvement heuristic referred to as ATM and devises an RL algorithm to learn this heuristic; we contrast the algorithm proposed here to the ATM heuristic in Section 3, and also in our numerical case study in Section 5. Note that none of the above-mentioned papers focuses on the planning problem in a manner that exploits the specific structure of the MDP and from the standpoint of provable optimality/suboptimality guarantees. A related formulation is considered in Armstrong-Crews and Veloso (2007), which treats "sensing" as a distinct action and applies a discount factor for its cost at each step a sensing action is taken. Aside from this distinction in the problem formulation, the JIV algorithm proposed in this paper is conceptually similar to the ATM heuristic proposed in Krale et al. (2023). A similar framework also appears in Treven et al. (2024), but for continuous-time problems with emphasis on learning rather than planning. Finally, another related formulation is analysed in Reisinger and Tam (2024); here, if the agent decides not to sense the state in any epoch, it is constrained to play the same action as in the previous epoch.

2 PROBLEM FORMULATION

In this section, we formally define our MDP formulation with a state sensing cost. We do this by first defining a 'standard' discounted cost MDP that serves as our baseline; we subsequently incorporate a state sensing cost, and a protocol for opportunistic state sensing on the part of the agent, into this baseline MDP.

Baseline MDP: Consider an infinite horizon discounted cost MDP $\mathcal{M}(\mathcal{S}, A, \mathcal{T}, \mathcal{C}, \alpha)$. Here,

- $\mathcal{S} = \{1, 2, \ldots, |\mathcal{S}|\}$ denotes the (finite) state space,
- $A = \{1, 2, \ldots, |A|\}$ denotes the (finite) action space,
- $\mathcal{T}$ denotes the transition function (i.e., $\mathcal{T}(s, a, s')$ denotes the probability of transitioning to state s' on

taking action a in state s),
• $\mathcal{C}$ denotes the cost function (i.e., $\mathcal{C}(s,a)$ is the cost associated with taking action a in state s),
• $\alpha \in (0,1)$ denotes the discount factor.

With some abuse of notation, for $a \in A$, we use $\mathcal{T}(a)$ and $\mathcal{C}(a)$ to denote, respectively, the $|\mathcal{S}| \times |\mathcal{S}|$ transition probability matrix, and the $|\mathcal{S}| \times 1$ (column) vector of costs, associated with the action a.[1] Denoting the state at time t by X_t, and the action at time t by A_t, there is a well-established theory for characterizing and computing the optimal policy that minimizes the expected discounted cost

$$\mathbb{E}\left[\sum_{t=0}^{\infty} \alpha^t \mathcal{C}(X_t, A_t)\right];$$

see Puterman (2014) and Ross (1992). We use V^* and Q^* to denote, respectively, the optimal value function and the optimal action-value function, corresponding to $\mathcal{M}$. As is convention, we treat V^* to be an $|\mathcal{S}| \times 1$ column vector, and Q^* to be a $|\mathcal{S}| \times |A|$ matrix.

MDP with state sensing cost: We now incorporate a state sensing cost $k>0$ to the above baseline MDP. The interaction protocol between the agent and environment at each time step $t \geq 0$ is as follows:

• The agent takes action A_t and commits either to sensing the state at the next step (a *sensing action*) or not (a *blind action*).
• Agent incurs cost $\mathcal{C}(X_t, A_t)$, and an additional sensing cost k in case it made a *sensing action*.
• The next state X_{t+1} is drawn according to $\mathcal{T}(X_t, A_t, \cdot)$. If the agent made a *sensing action*, then X_{t+1} is revealed; if it made a *blind action*, then X_{t+1} remains hidden (saving on the sensing cost k).

We assume the agent knows its initial state X_0. If it chooses a blind action at time t, then its next action A_{t+1} must be taken without precise knowledge of X_{t+1}. The agent's objective is to minimize expected discounted cost (including the state sensing cost), i.e.,

$$\mathbb{E}\left[\sum_{t=0}^{\infty} \alpha^t \left(\mathcal{C}(X_t, A_t) + k\mathbb{1}_{\{\text{sensing action at } t\}}\right)\right].$$

We formulate the sequential decision problem as an MDP $\mathcal{M}_k$ with a countably infinite state space. For clarity, we refer to states in the baseline MDP (elements of $\mathcal{S}$) as 'root states.' The state space of $\mathcal{M}_k$ is

$$\mathcal{S}_\infty := \mathcal{S} \cup \left[\mathcal{S} \times \left(\cup_{j=1}^{\infty} A^j\right)\right].$$

Here, the state variable corresponds to the most recently sensed (root) state, along with the string of blind actions taken thereafter. Each state $\tilde{s} \in \mathcal{S}_\infty$ is associated with a belief distribution $\mathcal{B}(\tilde{s}) \in \mathbb{R}^{1\times|\mathcal{S}|}$ over the set of root states. Specifically, for $\tilde{s} = (s, a_1, \ldots, a_n)$, where $s \in \mathcal{S}$ and $a_i \in A$ for $1 \leq i \leq n$,

$$\mathcal{B}(\tilde{s}) = e_s \mathcal{T}(a_1)\mathcal{T}(a_2)\cdots\mathcal{T}(a_n),$$

where e_s denotes the unit row vector with the sth entry being one. By convention, for $\tilde{s} = s \in \mathcal{S}$, (i.e., right after a sensing action), $\mathcal{B}(\tilde{s}) = e_s$. Next, the action space $\mathcal{A}$ for $\mathcal{M}_k$ is defined as

$$\mathcal{A} = A \times \{sense, blind\},$$

where the second component of the action captures the decision of whether or not to sense the state at the next time step. Thus $\mathcal{A}$ is finite; we also write $\mathcal{A} = \mathcal{A}_s \cup \mathcal{A}_b$, where $\mathcal{A}_s = A \times \{sense\}$ are sensing actions, and $\mathcal{A}_b = A \times \{blind\}$ blind actions.

The cost function $\mathcal{C}_\infty : \mathcal{S}_\infty \times \mathcal{A} \to \mathbb{R}$ associated with $\mathcal{M}_k$ is defined as follows:

$$\mathcal{C}_\infty(\tilde{s}, (a, sense)) = \mathcal{B}(\tilde{s})\mathcal{C}(a) + k$$
$$\mathcal{C}_\infty(\tilde{s}, (a, blind)) = \mathcal{B}(\tilde{s})\mathcal{C}(a)$$

Note that the cost has been averaged over the belief distribution over the root states.

Finally, we define the transition probability function for $\mathcal{M}_k$ as $\mathcal{T}_\infty : \mathcal{S}_\infty \times \mathcal{A} \times \mathcal{S}_\infty \to \mathbb{R}$ as follows:

$$\mathcal{T}_\infty(\tilde{s}_1, (a, blind), \tilde{s}_2) = \begin{cases} 1, & \text{for } \tilde{s}_2 = (\tilde{s}_1, a) \\ 0, & \text{otherwise} \end{cases}$$

$$\mathcal{T}_\infty(\tilde{s}_1, (a, sense), \tilde{s}_2) = \begin{cases} 0, & \text{for } \tilde{s}_2 \notin \mathcal{S} \\ \mathcal{B}(\tilde{s})\mathcal{T}(a)e_{\tilde{s}_2}^T, & \text{for } \tilde{s}_2 \in \mathcal{S} \end{cases}$$

The MDP $\mathcal{M}_k$, modeling opportunistic state sensing with a sensing cost, is defined using $(\mathcal{S}_\infty, \mathcal{A}, \mathcal{T}_\infty, \mathcal{C}_\infty, \alpha)$. Since the state space is countable and the action space finite, there exists an optimal stationary policy (Puterman, 2014; Ross, 1992); however, the exact computation of the optimal policy is infeasible due to the infinite state space. In Section 3, we propose an algorithm based on selective policy improvement, and in Section 4, iterative schemes for computing an optimal (or near-optimal) policy via state-space truncation together with a lower bound on the optimal value function, allowing us to quantify the suboptimality gap of any policy.

It is worth noting that, while POMDPs can be reformulated as MDPs over belief states (i.e., belief MDPs), such a transformation leads to an uncountably infinite state space, as the belief space forms a continuous $(n-1)$-simplex. In contrast, our formulation—an MDP with sensing costs—can be viewed as a special case of a POMDP that admits a countable state-space representation. This structural property makes our problem significantly more tractable than general POMDPs.

[1] Implicit in this notation is the assumption that any action $a \in A$ can be taken in any state $s \in \mathcal{S}$.

Algorithm 1 Selective Policy Improvement (**SPI**)

Input: Initial policy π_{init} , *maxsteps*, δ

Output: $\pi' \overset{\mathcal{S}}{\succeq} \pi_{init}$

$\pi' \leftarrow \pi_{init}$
$\pi_{\text{improv}} \leftarrow \textsc{PolicyUpdate}(\pi', maxsteps)$
while max $(V^{\pi'}_{\mathcal{M}_k} - V^{\pi_{\text{improv}}}_{\mathcal{M}_k}) > \delta$ **do**
  $\pi' \leftarrow \pi_{\text{improv}}$
  $\pi_{\text{improv}} \leftarrow \textsc{PolicyUpdate}(\pi', maxsteps)$
end while

3 SELECTIVE POLICY IMPROVEMENT

In this section, we introduce the Selective Policy Improvement (SPI) algorithm for the opportunistic state sensing MDP. The algorithm is stated formally as Algorithm 1. For our notation, let $V^{\pi}_{\mathcal{M}_k} \in \mathbb{R}^{|\mathcal{S}|\times 1}$ denote the value function column vector of policy π for the root states. The notation $\pi' \overset{\mathcal{S}}{\succeq} \pi$ indicates that the vector $V^{\pi'}_{\mathcal{M}_k}$ is element-wise less than or equal to $V^{\pi}_{\mathcal{M}_k}$. Finally, **max** x (respectively, **min** x) for a vector x denotes its maximum (respectively, minimum) entry.

SPI is initialized with a policy π_{init} and hyperparameters δ and *maxsteps*. It iteratively applies the PolicyUpdate routine (Algorithm 2) until the improvement in the value function falls below δ. The PolicyUpdate routine improves a reference policy π_{ref} as follows:

For each root state s, it searches for an 'improving' sequence of blind actions of length at most *maxsteps*, which lowers the action-value function at s, relative to the reference policy π_{ref}. While the set of all such blind action sequences can be quite large, PolicyUpdate searches this space *selectively* for computational tractability. Specifically, for a root state s, π' considers the candidate blind action sequence $(a_1, \ldots, a_n)$ only when for each i, $(a_1, \ldots, a_i)$ is an improvement over $(a_1, \ldots, a_{i-1})$, followed by the optimal sensing action. This allows for an efficient (and greedy) search for an improving blind action trajectory.

The preceding check is performed using the following functions: for any vector $\bar{V} \in \mathbb{R}^{|\mathcal{S}|\times 1}$, define

$$V_{MS}(\mathcal{B}(\tilde{s}), \bar{V}) = \min_{a\in A}\big(B(\tilde{s})\mathcal{C}(a) + \alpha B(\tilde{s})\mathcal{T}(a)\bar{V}\big) + k$$
$$\pi_{MS}(\mathcal{B}(\tilde{s}), \bar{V}) = \arg\min_{a\in A}\big(B(\tilde{s})\mathcal{C}(a) + \alpha B(\tilde{s})\mathcal{T}(a)\bar{V}\big)$$

Here, $V_{MS}(\mathcal{B}(\tilde{s}), \bar{V})$ denotes the value associated with playing the optimal sensing action under belief $\mathcal{B}(\tilde{s})$ with terminal values $\bar{V}$ (MS stands for *myopic sensing*). π_{MS} denotes the corresponding optimal sensing action. Importantly, note that SPI only performs policy evaluations over the *root states* of $\mathcal{M}_k$.

Finally, in our numerical case studies, we initialize

Algorithm 2 PolicyUpdate

Input: $\pi_{ref}, maxsteps$

Output: $\pi_o \overset{\mathcal{S}}{\succeq} \pi_{ref}$

$\pi_o \leftarrow \pi_{ref}$
for $s \in \mathcal{S}$ **do**
  $\tilde{s} \leftarrow s$
  $steps \leftarrow 0$
  $\pi' \leftarrow \pi_{ref}$
  $exploredstates \leftarrow \emptyset$
  while $steps \leq maxsteps$ **do**
    $exploredstates \leftarrow exploredstates \cup \{\tilde{s}\}$
    $V_{blind} \leftarrow \min_{a\in\mathcal{A}}\left(\mathcal{B}(\tilde{s})\mathcal{C}(a) + \alpha V_{MS}(\mathcal{B}(\tilde{s})\mathcal{T}(a), V^{\pi_{ref}}_{\mathcal{M}_k})\right)$
    $a_{blind} \leftarrow \arg\min_{a\in\mathcal{A}}\left(\mathcal{B}(\tilde{s})\mathcal{C}(a) + \alpha V_{MS}(\mathcal{B}(\tilde{s})\mathcal{T}(a), V^{\pi_{ref}}_{\mathcal{M}_k})\right)$
    if $V_{MS}(\mathcal{B}(\tilde{s}), V^{\pi_{ref}}_{\mathcal{M}_k}) \leq V_{blind}$ **then**
      $\pi'(\tilde{s}) \leftarrow (\pi_{MS}(\mathcal{B}(\tilde{s}), V^{\pi_{ref}}_{\mathcal{M}_k}(\mathcal{S})), sense)$
      break *Exit the while loop*
    end if
    $\pi'(\tilde{s}) \leftarrow (a_{blind}, blind)$
    $steps \leftarrow steps + 1$
    $\tilde{s} \leftarrow (\tilde{s}, a_{blind})$
  end while
  if $V^{\pi'}_{\mathcal{M}_k}(s) < V^{\pi_{ref}}_{\mathcal{M}_k}(s)$ **then**
    for $state \in exploredstates$ **do**
      $\pi_o(state) \leftarrow \pi'(state)$
    end for
  end if
end for

the SPI algorithm with the **A**lways **S**ense (AS) policy, which selects the optimal sensing action at each belief state. The corresponding value function of the AS policy, for $\tilde{s} \in \mathcal{S}_\infty$, is given by

$$V_{AS}(B(\tilde{s})) = \min_{a\in A}\big(B(\tilde{s})\mathcal{C}(a) + \alpha B(\tilde{s})\mathcal{T}(a)V^*\big) + \frac{k}{1-\alpha}$$
$$= \textbf{min}\ B(\tilde{s})Q^* + \frac{k}{1-\alpha}.$$

Here, with some abuse of notation, we parameterize the value function V_{AS} by the belief vector of state $\tilde{s}$, rather than by $\tilde{s}$ directly. The action corresponding to the AS policy is thus $(\pi_{AS}(\tilde{s}), sense)$, where

$$\pi_{AS}(\tilde{s}) = \textbf{argmin}\ B(\tilde{s})Q^*.$$

Here, **argmin** x for a row vector x denotes the column index of its minimum entry. It is important to note that AS policy π_{AS} agrees with the optimal policy π^* associated with the baseline MDP $\mathcal{M}$ over root states. In Section 4.1, we show that the AS policy is also optimal for $\mathcal{M}_k$ when the sensing cost k is small.

3.1 Step-by-Step Walkthrough of Policy Update

Starting at every root state s, PolicyUpdate initializes the candidate policy π' with π_{ref}. It then searches for

a sequence of actions to update π' and improve upon π_{ref} at the specific root state s. For the given root state, the output policy π_o plays the sequence of action that performs better at that root state.

Lines 9–10. For a given belief state $\tilde{s}$ (initialized by the root state s), the algorithm identifies the blind action a_{blind} as follows: For each blind action $(a, blind) \in \mathcal{A}_b$, it computes the cumulative cost, which consists of:

- Executing the blind action $(a, blind)$, incurring an immediate cost $\mathcal{B}(\tilde{s})\mathcal{C}(a)$.
- Transitioning to the belief state $(\tilde{s}, a_{blind})$ and obtaining the subsequent return $V_{MS}(\mathcal{B}(\tilde{s})\mathcal{T}(a), V^{\pi_{ref}}_{\mathcal{M}_k})$.

Note that this subsequent cumulative cost of $V_{MS}(\mathcal{B}(\tilde{s})\mathcal{T}(a), V^{\pi_{ref}}_{\mathcal{M}_k})$is obtained by first executing the sensing action given by $\pi_{MS}(\mathcal{B}(\tilde{s})\mathcal{T}(a), V^{\pi_{ref}}_{\mathcal{M}_k})$ at the belief state $(\tilde{s}, a_{blind})$, and then following the reference policy π_{ref} thereafter. The minimum cumulative cost over all blind actions is denoted V_{blind}, with the corresponding minimizing action denoted a_{blind}.

Lines 11–17. If the cumulative cost of playing the optimal sensing action at $\tilde{s}$ given by $V_{MS}(\mathcal{B}(\tilde{s}), V^{\pi_{ref}}_{\mathcal{M}_k})$, is lower than V_{blind}, the algorithm terminates the search and updates the candidate policy π' with the obtained sequence of actions, including the final sensing action. Otherwise (lines 15-17), the process repeats (up to a maximum of *maxsteps*) for the subsequent belief state $(\tilde{s}, a_{blind})$ obtained by executing the action $(a_{blind}, blind)$ on state s.

Lines 19–23. The value function $V^{\pi'}_{\mathcal{M}_k}(s)$ of the policy π' at s, which evaluates the cumulative cost of the candidate action sequence, is compared to that of the reference policy π_{ref}. The output policy π_o then adopts the sequence of actions starting from s that corresponds to the policy with the lower value function.

This process is repeated for all root states, constructing the final output policy π_o by aggregating the improved action sequences for each root state where the value of π' is lower than that of the reference policy π_{ref}.

It is instructive to compare SPI with the ATM heuristic proposed in Krale et al. (2023). The latter is more restricted in its search for good blind actions; in any belief state $\tilde{s}$, it seeks to improve upon π_{AS} by comparing the actions $(\pi_{AS}(\tilde{s}), blind)$ and $(\pi_{AS}(\tilde{s}), sense)$. For a detailed analysis of the ATM heuristic, we refer the reader to Section D.

4 SOLVING $\mathcal{M}_k$ WITH PROVABLE GUARANTEES

In this section, we establish conditions and methods that yield provable optimality and suboptimality guarantees for any policy π of MDP $\mathcal{M}_k$. We first present a sufficient condition under which always sensing is optimal. We then analyze a truncated version of $\mathcal{M}_k$ to derive a lower bound on its optimal value function, which in turn provides a computable bound on the suboptimality gap of any policy.

4.1 Optimality of always sensing

The following theorem shows that if the state sensing cost k is smaller than a certain threshold, then it is optimal to always sense the state.

Theorem 4.1. *If*

$$k < \alpha \min \min_{a_1, a_2 \in A} \left[\mathcal{T}(a_1)\left(Q^*(a_2) - V^*\right)\right],$$

then the AS policy defined in Section 3 is optimal for the MDP $\mathcal{M}_k$.

The threshold on state sensing cost in Theorem 4.1 is strictly positive if and only if, for any action a_1 taken in any root state j, there does not exist an action a_2 that is optimal in the baseline MDP on all states that lie in the belief support.

4.2 Analysis via state space truncation

We now consider a class of finite MDPs obtained via state space truncation of $\mathcal{M}_k$. Specifically, the truncation is parameterized by $n \geq 0$, the maximum number of consecutive blind actions the agent is permitted to take. Formally, the truncated MDP, denoted by $\mathcal{M}_{k,n}$ is defined as follows. The state space is given by

$$\mathcal{S}_n := \mathcal{S} \cup \left[\mathcal{S} \times \left(\cup_{j=1}^{n} A^j\right)\right].$$

For $n \geq 1$, this yields $\mathcal{S} = \mathcal{S}_0 \subset \mathcal{S}_n \subset \mathcal{S}_{n+1} \subset \mathcal{S}_\infty$. It is convenient to categorize the states of $\mathcal{S}_n$ into 'layers' as follows: Let $\mathcal{L}_0$ be the set of root states (the 0th layer), where the agent knows its current state precisely. Next, we define $\mathcal{L}^j_m$ as the set of 'descendants' of the root state j in the mth layer, i.e., the set of states corresponding to playing m successive blind steps starting from the root state j. More formally,

$$\mathcal{L}^j_m = \{\tilde{s} \in \mathcal{S}_\infty \,|\, \tilde{s} = (j, a_1, \ldots, a_m), \text{where } a_1, \ldots, a_m \in A\}.$$

Finally, $\mathcal{L}_m$ defines the mth layer, defined as the union of sets $\mathcal{L}^j_m$ over all root states j in $\mathcal{L}_0$, i.e., $\mathcal{L}_m = \bigcup_{j \in \mathcal{L}_0} \mathcal{L}^j_m$. Note that $\mathcal{S}_n = \cup_{m=0}^{n} \mathcal{L}_m$. Figure 1 provides an illustration of this layered view of the state space $\mathcal{S}_n$ for the special case of a two-state ($\mathcal{S} = \{0, 1\}$), two-action ($A = \{L, R\}$) baseline MDP.

The action space $\mathcal{A}_n$ of $\mathcal{M}_{k,n}$ is simply $\mathcal{A}_\infty$, whereas the transition function $\mathcal{T}_n : \mathcal{S}_n \times \mathcal{A}_n \times \mathcal{S}_n \to \mathbb{R}$ and the cost function $\mathcal{C}_n : \mathcal{S}_n \times \mathcal{A}_n \to \mathbb{R}$ are given by:

$$\mathcal{T}_n(\tilde{s}_1, (a, \cdot), \tilde{s}_2) = \begin{cases} \mathcal{T}_\infty(\tilde{s}_1, (a, \cdot), \tilde{s}_2), & \text{if } \tilde{s}_1 \notin \mathcal{L}_n \\ \mathcal{T}_\infty(\tilde{s}_1, (a, sense), \tilde{s}_2), & \text{otherwise} \end{cases}$$

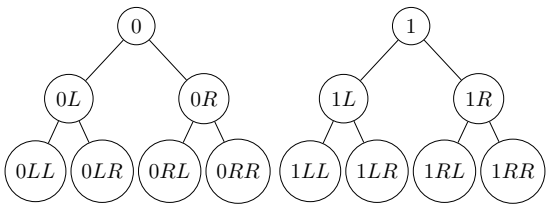


Figure 1: **State space of** $\mathcal{M}_{k,2}$ for a 2-state 2-action baseline MDP $\mathcal{M}$

$$\mathcal{C}_n(\tilde{s},(a,\cdot)) = \begin{cases} \mathcal{C}_\infty(\tilde{s},(a,\cdot)), & \text{if } \tilde{s} \notin \mathcal{L}_n \\ \mathcal{C}_\infty(\tilde{s},(a,sense)), & \text{otherwise} \end{cases}$$

Note that the transition and cost functions in $\mathcal{M}_{k,n}$ agree with those in $\mathcal{M}_k$, except on states at the nth layer, where state sensing is enforced.

Since $\mathcal{M}_{k,n}$ is a finite MDP, it admits an exact computation of its optimal policy $\pi^*_{\mathcal{M}_{k,n}}$ and optimal value function $V^*_{\mathcal{M}_{k,n}}$. Of course, the complexity of this computation grows exponentially in n, so this is only feasible for small values of n. In the remainder of this section, we relate the solutions of the (finite, and therefore 'tractable') truncated MDPs $\{\mathcal{M}_{k,n}\}$ to one another, and to the solution of $\mathcal{M}_k$.

Our first result bounds the suboptimality induced by the aforementioned state space truncation.

Theorem 4.2.

$$V^*_{\mathcal{M}_{k,N}}(j) - V^*_{\mathcal{M}_k}(j) \le \frac{\alpha^N k}{1-\alpha} \quad \forall j \in \mathcal{S}, N \ge 0.$$

Note that using the above result, one can determine a suitable truncation depth N given a suboptimality tolerance, *without having to solve* $\mathcal{M}_{k,N}$ *first*. Moreover, solving $\mathcal{M}_{k,N}$ yields a computable lower bound on the optimal value function $V^*_{\mathcal{M}_k}$ of $\mathcal{M}_k$, which can then be used to bound the suboptimality gap of any policy. (A sharper bound, expressed in terms of the solution of $\mathcal{M}_{k,N}$ is provided later in Theorem 4.4.)

Our next result gives a necessary and sufficient condition for the optimal policy for $\mathcal{M}_{k,n}$ to also be optimal for $\mathcal{M}_{k,n+1}$. For $s \in \mathcal{S}$ and $a_1, a_2, \ldots, a_m \in A$, define

$$Z((s,a_1,a_2,\ldots,a_m)) := \mathcal{C}(s,a_1) + \sum_{i=1}^{m-1} \alpha^i \mathcal{B}((s,a_1,a_2,\ldots,a_i))\mathcal{C}(s,a_{i+1})$$

as the average cumulative discounted cost incurred in reaching the state $\tilde{s} = (s,a_1,a_2,\ldots,a_m)$ from its root state s (by taking a sequence of blind steps).

Lemma 4.3. *Fix* $N \ge 0$. $V^*_{\mathcal{M}_{k,N}}(j) = V^*_{\mathcal{M}_{k,N+1}}(j)$ *for all root states* $j \in \mathcal{S}$ *if and only if*

$$Z(i) + \alpha^{N+1} \min_a \Big(\mathcal{B}(i)\mathcal{C}(a) + k + \alpha \mathcal{B}(i)\mathcal{T}(a) V^*_{\mathcal{M}_{k,N}} \Big) \ge V^*_{\mathcal{M}_{k,N}}(j) \quad \forall j \in \mathcal{S}, i \in \mathcal{L}^j_{N+1}.$$

Note that $V^*_{\mathcal{M}_{k,N}}(j) = V^*_{\mathcal{M}_{k,N+1}}(j)$ for all $j \in \mathcal{S}$ implies that the optimal stationary policy $\pi^*_{M_{k,N}}$ for $\mathcal{M}_{k,N}$ is also optimal for $\mathcal{M}_{k,N+1}$ when starting in a root state (i.e., with knowledge of the starting state). However, this condition *does not* imply that $\pi^*_{\mathcal{M}_{k,N}}$ is optimal for $\mathcal{M}_k$ when starting in a root state, as shown by a counterexample in Appendix E.2. One needs a stricter condition on $\mathcal{M}_{k,N}$ to conclude that $\pi^*_{\mathcal{M}_{k,N}}$ is optimal for $\mathcal{M}_k$; this is the focus of our next result.

Define

$$V_{AS;0}(\tilde{s}) := \mathbf{min}\ B(\tilde{s})Q^*.$$

This is simply the value function corresponding to the AS policy introduced in Section 3, assuming zero sensing cost. This means $V_{AS;0}$ provides a *lower bound* on the optimal value function for $\mathcal{M}_k$.

Theorem 4.4. *Fix* $N \ge 0$. *If*

$$Z(i) + \alpha^{N+1} V_{AS;0}(i) \ge V^*_{\mathcal{M}_{k,N}}(j) \quad \forall j \in \mathcal{S},\ i \in \mathcal{L}^j_{N+1}, \tag{1}$$

then the optimal stationary policy $\pi^*_{\mathcal{M}_{k,N}}$ *of* $\mathcal{M}_{k,N}$ *is optimal for* $\mathcal{M}_k$ *when starting at any root state. If* (1) *does not hold,*

$$V^*_{\mathcal{M}_{k,N}}(s) - V^*_{\mathcal{M}_k}(s) \le \epsilon_N \quad \forall s \in \mathcal{S}, \tag{2}$$

where

$$\epsilon_N := \max_{j \in S} \left[V^*_{\mathcal{M}_{k,N}}(j) - \min_{i \in \mathcal{L}^j_{N+1}} \big(Z(i) + \alpha^{N+1} V_{AS;0}(i) \big) \right].$$

Theorem 4.4 provides a sufficient condition (1) for the optimal policy for $\mathcal{M}_{k,n}$ to also be optimal for $\mathcal{M}_k$ (assuming the starting state is a root state). Even if this condition is violated, Theorem 4.4 provides a computable upper bound on the suboptimality of the policy $\pi^*_{\mathcal{M}_{k,N}}$ on $\mathcal{M}_k$. Thus, Theorem 4.4 suggests a recipe for computing an optimal/near-optimal policy for $\mathcal{M}_k$: Iteratively solve $\mathcal{M}_{k,N}$ for increasing N, until either (i) the condition (4.4) is satisfied, in which case the optimal policy just computed is also optimal for $\mathcal{M}_k$, or (ii) the suboptimality bound ϵ_N is acceptably small. Moreover, analogous to Theorem 4.2, one can also derive a (stronger) lower bound on the optimal value function $V^*_{\mathcal{M}_k}$ of $\mathcal{M}_k$.

As shown in Appendix E.1, even if (1) is satisfied only at certain root states, one does not need to explore depths $N+1$ and beyond at these root states. Furthermore, Lemma E.1 establishes that the suboptimality bound ϵ_N decreases monotonically in N.

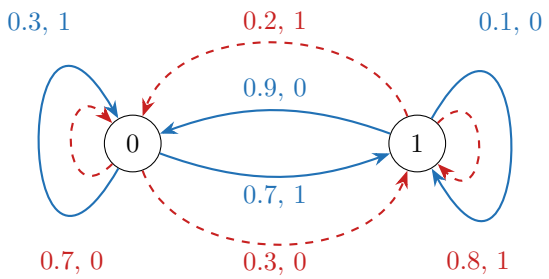


Figure 2: **A two-state two-action baseline MDP** with actions $\{Red, Blue\}$ and $\alpha = 0.5$.

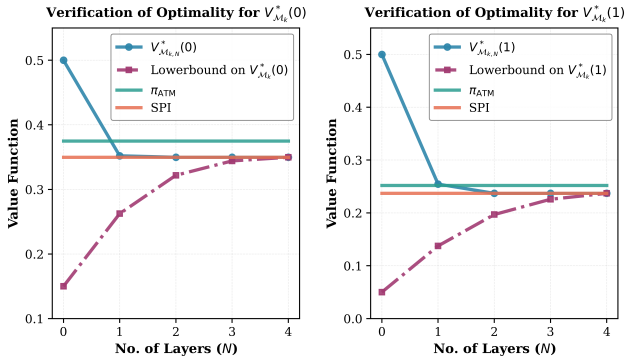


Figure 3: **Applying Thm 4.4** on MDP in Fig 2; $k = 0.25$

5 NUMERICAL CASE STUDIES

In this section, we present numerical experiments that validate and complement the analytical results from prior sections. We also benchmark the proposed approaches against the state of the art.

2-State 2-Action example: We consider a two-state, two-action baseline MDP, shown in Figure 2, and evaluate it under sensing costs (a) $k = 0.01$ and (b) $k = 0.25$.

• Applying Theorem 4.1 yields a sensing cost threshold of 0.05, making always sensing optimal in case(a). Thus, the optimal policy for state 0 is $(R, sense)$, and for 1 is $(B, sense)$.
• For $k = 0.25$, the optimal policy for both root states remains unchanged after $N \geq 2$. Indeed, the condition of Lemma 4.3 is satisfied at $N = 2$. However, the condition of Theorem 4.4 is satisfied for $N = 4$ which suggests that the optimal policy should remain unchanged for $N \geq 4$; see Figure 3.
(The stronger lower bound on $V^*_{\mathcal{M}_k}$ used in Figure 3 is detailed in Appendix E.1.)

Experimental Setup. We benchmark the proposed SPI algorithm against the ATM heuristic (Krale et al., 2023), alongside widely used general-purpose POMDP planning algorithms, including SARSOP (Kurniawati et al., 2008), Fast Informed Bound (FIB) (Hauskrecht, 2000), using the `POMDPs.jl` framework (Egorov et al., 2017). Experiments are conducted across the ICU-Sepsis benchmark (Choudhary et al., 2024) and gridworld-based environments from Gymnasium (Towers et al., 2024), including Taxi and Frozen Lake. Algorithms such as Incremental Pruning (Cassandra et al., 1997) and PBVI (Pineau et al., 2003) exhibited prohibitive runtimes, failing to produce effective policies even for small state spaces, while other methods like QMDP (Littman et al., 1995) were consistently outperformed by our approach. Even popular online planning algorithms like PO-UCT (Silver and Veness, 2010) required significantly higher planning times to generate competitive policies.

These POMDP planners were initialized with a belief distribution corresponding to the starting state distribution, rather than a uniform distribution over all states. The experimental design, along with an analysis of POMCP's performance, is provided in Appendix B.1. For SARSOP, the reward and sensing costs were scaled by 10^3 during policy computation to match performance on Frozen Lake.

ICU-Sepsis ($|\mathcal{S}| = 716$ & $|A| = 25$) : Choudhary et al. (2024) is a tabular MDP modeling personalized care for sepsis patients in the ICU, derived from electronic health record (EHR) data. The *state space* $\mathcal{S}$ models a patient's health as a vector of clustered features (demographics, vital signs, body fluid levels) discretized into 716 states. The *action space* consists of discrete combinations of medication types and dosages. The agent receives 0 reward for non-terminal steps and +1 for survival ($\alpha = 0.99$).
Frozen Lake ($|\mathcal{S}| = 16/64$ & $|A| = 4$) : The agent navigates across a frozen lake and receives a reward of +1 upon reaching the goal state ($\alpha = 0.9$). In the slippery case, actions succeed with probability $\frac{1}{3}$, otherwise moving perpendicularly with equal probability $\frac{1}{3}$. We evaluate default 4x4, 8x8, and a custom, challenging 4x4 Frozen Lake grid (see Table 2). For Frozen Lake, we report expected reward over the initial state **S** and maximum computation times across sensing costs.
Stochastic Taxi ($|\mathcal{S}| = 500$ & $|A| = 6$) : The agent navigates a 5x5 grid to pick up and drop off passengers, receiving rewards of +20 for delivery, -10 for illegal "pickup" and "drop-off" actions, and -1 per step unless another reward is triggered ($\alpha = 0.95$) (Dietterich, 1999). Each of the four navigation actions moves the agent in the intended direction with a probability of 0.8, and in one of the two perpendicular directions with an equal probability of 0.1. The initial states are sampled from 300 valid states.

Environment		Sensing Costs			
		0.005	**0.01**	**0.05**	**0.1**
ICU-Sepsis					
π_{ATM}	**Value**	0.740	0.715	0.714	0.714
	Time (s)	285	782	833	916
SARSOP (3000s)		0.741	0.741	0.741	0.740
SPI (2 iter)	**Value**	**0.765**	**0.747**	**0.742**	**0.745**
	Time (s)	23	1252	2205	2757
		0.1	**0.5**	**1**	**5**
Stochastic Taxi					
π_{ATM}	**Value**	0.908	-0.359	-2.761	-19.664
	Time (s)	2	1.7	1.8	38.4
SARSOP (100s)		-22	-1.2	-2.705	-20
SPI	**Value**	**0.911**	**-0.306**	**-1.598**	**-9.778**
	Time (s)	25.2	19.7	17.8	117.8

Scenario	Value (Rewards) for Sensing Costs				Time (s)
Frozen Lake	**0.001**	**0.005**	**0.01**	**0.05**	
4x4 (Default)					
π_{ATM}	**62.42**	36.52	6.72	16.57	0.04
$V^*_{\mathcal{M}_{k,3}}$	**62.42**	**36.53**	20.47	-28.75	11.5
SARSOP	**62.42**	**36.53**	20.79	**23.08**	1.7
FIB (1000 iter)	**62.42**	31.43	**23.08**	**23.08**	8.3
SPI	**62.42**	**36.53**	20.99	**23.08**	0.4
4x4 (Hard)					
π_{ATM}	8.41	0	0	0	0.04
$V^*_{\mathcal{M}_{k,3}}$	8.92	1.36	-5.75	-36.75	12
SARSOP	**8.95**	3.66	1.34	**1.44**	1.6
FIB (1000 iter)	4.44	0	0	0	9.5
SPI	**8.95**	**3.69**	**1.47**	1.35	0.4
8x8					
π_{ATM}	3.29	3.29	3.29	3.29	0.25
$V^*_{\mathcal{M}_{k,3}}$	2.72	-4.943	-13.64	-79.09	205
SARSOP	3.36	**3.36**	**3.36**	**3.36**	2
FIB (20 iter)	3.29	3.29	3.29	3.29	475
SPI	**3.53**	3.33	3.33	3.33	3

Table 1: **Performance comparison across different scenarios** with varying sensing costs; rewards scaled by 10^3 for Frozen Lake. (Unless the no. of PolicyUpdate iter in SPI are specified, $\delta = 10^{-6}$.)

Benchmarking Results. Table 1 shows that SPI consistently outperforms π_{ATM} and FIB, and achieves the best performance on ICU-Sepsis and Taxi across all sensing costs (FIB could not scale to these domains). Results from the custom hard setup further indicate that π_{ATM} and FIB may fail to find effective policies even in small state spaces when the planning problem is challenging, underscoring SPI's robustness.

We emphasize that, although SARSOP exhibited performance comparable to SPI for Frozen-Lake, achieving this required scaling the reward and sensing values. Without such scaling, SARSOP performed poorly, even relative to π_{ATM} and FIB. Furthermore, in Stochastic Taxi, SARSOP consistently underperformed, and scaling the values did not improve its performance. Finally, we note that SARSOP's performance is highly sensitive to several hyperparameters that are difficult to interpret and tune. In contrast, SPI relies on only two hyperparameters, both of which are intuitive and easily adjustable. Moreover, SARSOP requires the starting state distribution to compute its policy and can degrade significantly under mismatch, whereas SPI operates without requiring such information.

Finally, our analysis reveals significant limitations in existing RL algorithms for this setting, notably the Observe-before-planning (Nam et al., 2021) and Dyna-ATMQ (Krale et al., 2023). Observe-before-Planning leverages the ACNO-MDP structure during exploration but resorts to the generic POMDP solver POMCP (PO-UCT) for planning, failing to utilize the specific ACNO-MDP structure during exploitation. Furthermore, PO-UCT performs significantly worse than SPI, especially in scenarios with sparse rewards. Similarly, Dyna-ATMQ employs a less effective planner π_{ATM}, which struggles even in small, challenging state spaces. We attribute these inefficiencies to the absence of a strong theoretical foundation for ACNO-MDPs, a gap our paper addresses. By reformulating this setting using an "expanded state-space setup," we provide novel insights that enable the design of effective planning algorithms like SPI.

6 DISCUSSION AND CONCLUSION

In this paper, we have analysed a class of MDPs with a state sensing cost. Here, the agent must, in a history-dependent, opportunistic manner, determine when to sense the state of the system. While these MDPs are intractable under generic planning algorithms, we exploit the special structure of these sensing-cost MDPs to devise clever algorithms and truncation approaches with provable optimality/suboptimality guarantees. Our main contributions are summarized as follows:

- We reformulate opportunistic state sensing in an expanded state-space framework and propose a novel planning algorithm, SPI. SPI outperforms state-of-the-art POMDP solvers and specialized solvers for this setting (e.g., π_{ATM}) across diverse domains for all ranges of sensing costs within reasonable compute time (Table 1). Moreover, it scales to large-state-space MDPs and naturally extends to continuous state spaces using function approximation (Appendix C.2).
- Using truncated MDP analysis, we derive lower bounds on the optimal value function, which enable explicit computation of suboptimality gaps for any policy (Theorems 4.4 and 4.2).We establish a quantitative threshold on the sensing cost (Theorem 4.1) below which always sensing is provably optimal.

To the best of our knowledge, this work presents the first set of results establishing suboptimality guarantees for policies and explicit cost thresholds in this setting.

In practice, sensing costs are often context-dependent, shaped by factors such as resource constraints or belief uncertainty (e.g., in medical decision-making or robotics). Our framework is readily extendable to non-uniform sensing costs: both SPI and the truncated MDP approach, along with the theoretical results, apply beyond the constant-k setting (see Appendix F).

At a high level, this work is related to the vast recent literature on Age of Information (AoI), where the goal is to allocate resources or incur costs so as to minimize the age (a.k.a., staleness) of the state information; see Yates et al. (2021) for a survey. However, the AoI literature does not, as such, consider the *control* aspect where the goal of state estimation is actually to influence the state evolution favourably. Additionally, the AoI literature employs a universal (state-independent) age/staleness penalty; in practice, one would expect that the agent would be more tolerant of delayed state information in certain states than others. The present formulation seeks to formally capture this trade-off.

References

Nicholas Armstrong-Crews and Manuela Veloso. Oracular partially observable markov decision processes: A very special case. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 2477–2482, 2007. doi: 10.1109/ROBOT.2007.363691.

Colin Bellinger, Rory Coles, Mark Crowley, and Isaac Tamblyn. Active measure reinforcement learning for observation cost minimization. In *Canadian AI*, 2021.

S.P. Bradley, A.C. Hax, and T.L. Magnanti. *Applied Mathematical Programming*. Addison-Wesley Publishing Company, 1977. ISBN 9780201004649. URL https://books.google.co.in/books?id=MSWdWv3Gn5cC.

Anthony Cassandra, Michael L Littman, and Nevin L Zhang. Incremental pruning: a simple, fast, exact method for partially observable markov decision processes. In *Proceedings of the Thirteenth conference on Uncertainty in artificial intelligence*, pages 54–61, 1997.

Gongpu Chen and Soung Chang Liew. Intermittently observable markov decision processes. *IEEE Transactions on Automatic Control*, 2025.

Kartik Choudhary, Dhawal Gupta, and Philip S Thomas. Icu-sepsis: A benchmark mdp built from real medical data. *arXiv preprint arXiv:2406.05646*, 2024.

Thomas G. Dietterich. Hierarchical reinforcement learning with the MAXQ value function decomposition. *CoRR*, cs.LG/9905014, 1999. URL https://arxiv.org/abs/cs/9905014.

Maxim Egorov, Zachary N. Sunberg, Edward Balaban, Tim A. Wheeler, Jayesh K. Gupta, and Mykel J. Kochenderfer. POMDPs.jl: A framework for sequential decision making under uncertainty. *Journal of Machine Learning Research*, 18(26):1–5, 2017. URL http://jmlr.org/papers/v18/16-300.html.

Damien Ernst, Guy-Bart Stan, Jorge Goncalves, and Louis Wehenkel. Clinical data based optimal sti strategies for hiv: a reinforcement learning approach. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 667–672. IEEE, 2006.

Eric A Hansen. Cost-effective sensing during plan execution. In *AAAI*, pages 1029–1035, 1994.

Milos Hauskrecht. Value-function approximations for partially observable markov decision processes. *Journal of artificial intelligence research*, 13:33–94, 2000.

Merlijn Krale, Thiago D Simao, and Nils Jansen. Act-then-measure: reinforcement learning for partially observable environments with active measuring. In

Proceedings of the International Conference on Automated Planning and Scheduling, volume 33, pages 212–220, 2023.

Hanna Kurniawati, David Hsu, and Wee Sun Lee. Sarsop: Efficient point-based pomdp planning by approximating optimally reachable belief spaces. In *Robotics: Science and systems*, volume 2008. Citeseer, 2008.

Michael L Littman, Anthony R Cassandra, and Leslie Pack Kaelbling. Learning policies for partially observable environments: Scaling up. In *Machine Learning Proceedings 1995*, pages 362–370. Elsevier, 1995.

HyunJi Alex Nam, Scott Fleming, and Emma Brunskill. Reinforcement learning with state observation costs in action-contingent noiselessly observable markov decision processes. *Advances in Neural Information Processing Systems*, 34:15650–15666, 2021.

Christos H. Papadimitriou and John N. Tsitsiklis. The complexity of markov decision processes. *Math. Oper. Res.*, 12(3):441–450, August 1987. ISSN 0364-765X.

Joelle Pineau, Geoff Gordon, Sebastian Thrun, et al. Point-based value iteration: An anytime algorithm for pomdps. In *Ijcai*, volume 3, pages 1025–1032, 2003.

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Christoph Reisinger and Jonathan Tam. Markov decision processes with observation costs: framework and computation with a penalty scheme. *Mathematics of Operations Research*, 2024.

Sheldon M Ross. *Applied probability models with optimization applications*. Courier Corporation, 1992.

David Silver and Joel Veness. Monte-carlo planning in large pomdps. *Advances in neural information processing systems*, 23, 2010.

Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, et al. Gymnasium: A standard interface for reinforcement learning environments. *arXiv preprint arXiv:2407.17032*, 2024.

Lenart Treven, Yarden As, Florian Dorfler, and Andreas Krause. When to sense and control? a time-adaptive approach for continuous-time rl. *Advances in Neural Information Processing Systems*, 37:63654–63685, 2024.

Yi Wang, Bhaskar Krishnamachari, Qing Zhao, and Murali Annavaram. Markov-optimal sensing policy for user state estimation in mobile devices. In *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, pages 268–278, 2010.

Roy D Yates, Yin Sun, D Richard Brown, Sanjit K Kaul, Eytan Modiano, and Sennur Ulukus. Age of information: An introduction and survey. *IEEE Journal on Selected Areas in Communications*, 39(5): 1183–1210, 2021.

Checklist

The checklist follows the references. For each question, choose your answer from the three possible options: Yes, No, Not Applicable. You are encouraged to include a justification to your answer, either by referencing the appropriate section of your paper or providing a brief inline description (1-2 sentences). Please do not modify the questions. Note that the Checklist section does not count towards the page limit. Not including the checklist in the first submission won't result in desk rejection, although in such case we will ask you to upload it during the author response period and include it in camera ready (if accepted).

In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes] We provide a clear mathematical formulation of the problem and the proposed algorithms.
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [No] We do not provide a formal complexity analysis, but we present an in-depth analysis of the algorithms and demonstrate their empirical efficiency.
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes] The implementation code is provided in the supplementary material.

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes] All assumptions underlying our theoretical results are stated explicitly.

(b) Complete proofs of all theoretical results. [Yes] The proof for the theoretical results is detailed in the Appendix.

(c) Clear explanations of any assumptions. [Yes] We provide clear explanations of all assumptions alongside the results.

3. For all figures and tables that present empirical results, check if you include:

(a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes] The implementation code is provided in the supplementary material, and the experimental setup is detailed in the Appendix B Experimental Design and Setup.

(b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes] The complete experimental setup, the choice of benchmarking domains (ICU-Sepsis, Stochastic Taxi, and Frozen Lake), and the implementation details of the baseline algorithms—including hyperparameters and initialization—are provided in Appendix B Experimental Design and Setup.

(c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes] The performance of algorithms (e.g., SPI, π_{ATM}, SARSOP) on the benchmark domains (ICU-Sepsis, Stochastic Taxi, and Frozen Lake) is evaluated using total expected return on the initial state distribution, which is deterministic and does not require statistical significance analysis or error bars.

(d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes] The computing infrastructure used is provided in the Appendix B Experimental Design and Setup.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:

(a) Citations of the creator If your work uses existing assets. [Yes] We cite all assets used in our work with proper references.

(b) The license information of the assets, if applicable. [Not Applicable] The assets used are open-source.

(c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable] We do not release new assets in this work.

(d) Information about consent from data providers/curators. [Not Applicable] No data requiring consent was collected.

(e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable] The work did not involve sensitive or personally identifiable content.

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

(a) The full text of instructions given to participants and screenshots. [Not Applicable] No human participants were involved.

(b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable] No IRB approval was required.

(c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable] No participant compensation was involved.

Appendix

Appendix Contents

A. Proofs 12
A.1 Proof of Theorem 4.1 12
A.2 Proof of Theorem 4.2 13
A.3 Proof of Lemma 4.3 14
A.4 Proof of Theorem 4.4 15
A.5 Proof of Lemma E.1 16
B. Experimental Design and Setup 17
C. Analysis of SPI 18
D. Analysis of ATM Heuristic 19
E. Additional Results 20
F. Extension to Non-Uniform Sensing Costs 21
G. Inventory Management Case Study 22

A Proofs

A.1 Proof of Theorem 4.1

Proof. We define π to be the AS policy for the MDP $\mathcal{M}_k$ and hence for each $\tilde{s} \in \mathcal{S}_\infty, \pi(\tilde{s}) = (\pi_{AS}(\tilde{s}), sense)$. It is important to note that AS policy π_{AS} takes the optimal sensing action in each belief state and agrees with the optimal policy π^* associated with the baseline MDP $\mathcal{M}$ over root states. Therefore, we have:

$$
\begin{aligned}
V^{\pi}_{\mathcal{M}_k}(\tilde{s}) &= \min_{a \in A} \left(B(\tilde{s})\mathcal{C}(a) + \alpha B(\tilde{s})\mathcal{T}(a)V^*\right) + \frac{k}{1-\alpha} \\
&= \min_{a \in A} \left(B(\tilde{s})\mathcal{C}(a) + \alpha B(\tilde{s})\mathcal{T}(a)V^{\pi}_{\mathcal{M}_k}\right) + k \quad \left(\text{Since } V^{\pi}_{\mathcal{M}_k}(s) = V^*(s) + \frac{k}{1-\alpha} \quad \forall s \in \mathcal{S}\right) \\
&= \min_{a \in A} Q^{\pi}_{\mathcal{M}_k}(\tilde{s}, (a, sense))
\end{aligned}
$$

Hence, policy π cannot be improved by any sensing action for any of the states, i.e.,

$$
Q^{\pi}_{\mathcal{M}_k}(\tilde{s}, (a, sense)) \geq V^{\pi}_{\mathcal{M}_k}(\tilde{s}) \quad \forall \tilde{s} \in \mathcal{S}_\infty,\ a \in A.
$$

$$
Q^{\pi}_{\mathcal{M}_k}(\tilde{s}, (a_1, blind)) \geq Q^{\pi}_{\mathcal{M}_k}(\tilde{s}, (a_1, sense)) \quad \forall (\tilde{s}, a_1) \in \mathcal{S}_\infty \times A. \tag{3}
$$

We claim that (3) is a sufficient condition for π to be optimal: If for every state $\tilde{s} \in \mathcal{S}_\infty$ and every action $a \in A$, the action-value function for the blind action $(a, blind)$ is greater than or equal to that for the corresponding sensing action $(a, sense)$, i.e., $Q^{\pi}_{\mathcal{M}_k}(\tilde{s}, (a, blind)) \geq Q^{\pi}_{\mathcal{M}_k}(\tilde{s}, (a, sense))$, then π is optimal. This follows from the initial claim that $Q^{\pi}_{\mathcal{M}_k}(\tilde{s}, (a, sense)) \geq V^{\pi}_{\mathcal{M}_k}(\tilde{s})$ holds (indicating that π cannot be improved by any sensing action), combined with the fact that a policy satisfying (3) cannot be improved by taking blind actions for any states during the Policy Improvement Step.

Criterion (3) holds for a state-action pair $(\tilde{s}, a_1)$ if and only if

$$
\mathcal{B}(\tilde{s})\mathcal{C}(a_1) \ + \ \alpha\left(k + \mathcal{B}(\tilde{s})\mathcal{T}(a_1)\mathcal{C}(a_2)\right) \ + \ \alpha^2\mathcal{B}(\tilde{s})\mathcal{T}(a_1)\mathcal{T}(a_2)V^{\pi}_{\mathcal{M}_k} \quad \geq \quad \mathcal{B}(\tilde{s})\mathcal{C}(a_1) \ + \ k \ + \ \alpha\mathcal{B}(\tilde{s})\mathcal{T}(a_1)V^{\pi}_{\mathcal{M}_k},
$$

where a_2 is the action taken according to π at the state reached by taking action a_1 at state $\tilde{s}$. Moreover, for any root state $s \in \mathcal{S}$, we have

$$V^{\pi}_{\mathcal{M}_k}(s) = V_{AS}(\mathcal{B}(s)) = V^*(s) + \frac{k}{1-\alpha}.$$

Substituting this and rearranging, we get

$$\frac{k}{1-\alpha} + \alpha\mathcal{B}(\tilde{s})\mathcal{T}(a_1)\mathcal{T}(a_2)V^* \geq \frac{k}{\alpha(1-\alpha)} + \mathcal{B}(\tilde{s})\mathcal{T}(a_1)V^* - \mathcal{B}(\tilde{s})\mathcal{T}(a_1)\mathcal{C}(a_2).$$

Simplifying, we get

$$\mathcal{B}(\tilde{s})\mathcal{T}(a_1)\mathcal{C}(a_2) - \mathcal{B}(\tilde{s})\mathcal{T}(a_1)V^* + \alpha\mathcal{B}(\tilde{s})\mathcal{T}(a_1)\mathcal{T}(a_2)V^* \geq \frac{k}{\alpha}.$$

Further simplifying, we obtain

$$\mathcal{B}(\tilde{s})\mathcal{T}(a_1)(\mathcal{C}(a_2) + \alpha\mathcal{T}(a_2)V^* - V^*) \geq \frac{k}{\alpha},$$

which implies

$$\frac{k}{\alpha} \leq \mathcal{B}(\tilde{s})\mathcal{T}(a_1)(Q^*(a_2) - V^*). \tag{4}$$

If condition (5) holds, then (4) is satisfied for all $(\tilde{s}, a_1) \in \mathcal{S}_\infty \times A$, and hence the AS policy π is optimal for $\mathcal{M}_k$.

$$k < \alpha \, \mathbf{min} \min_{a_1, a_2 \in A} \left[\mathcal{T}(a_1)\left(Q^*(a_2) - V^*\right)\right]. \tag{5}$$

□

A.2 Proof of Theorem 4.2

Proof. Consider any stationary policy π of $\mathcal{M}_k$ and define the set of root states G, such that starting from any state $j \in G$ and following π, we play at least $N+1$ consecutive blind steps. Also define $\bar{G} := \mathcal{S} \setminus G$.

First, we show that

$$V^*_{\mathcal{M}_{k,N}}(i) - V^{\pi}_{\mathcal{M}_k}(i) \leq \sum_{j \in \mathcal{L}_0} p_{ij}\left(V^*_{\mathcal{M}_{k,N}}(j) - V^{\pi}_{\mathcal{M}_k}(j)\right) \quad \forall i \in \bar{G}, \tag{6}$$

where p_{ij} denotes the probability of landing in root state $j \in \mathcal{S}$ after taking the first sensing step when starting from i and following π.

Next, we show that

$$V^*_{\mathcal{M}_{k,N}}(i) - V^{\pi}_{\mathcal{M}_k}(i) \leq \frac{\alpha^N k}{1-\alpha}, \quad \forall i \in G \tag{7}$$

Finally, define $G' = \{s \mid s \in \bar{G} \text{ and } s \not\to i, \forall i \in G\}$, i.e., starting from any state $s \in G'$, we never reach a state in G under policy π. It is easy to see that

$$V^*_{\mathcal{M}_{k,N}}(i) - V^{\pi}_{\mathcal{M}_k}(i) = 0, \quad \forall i \in G'. \tag{8}$$

We now show how the the statement of the lemma follows from (6)–(8). Treat $f(i) := V^*_{\mathcal{M}_{k,N}}(i) - V^{\pi}_{\mathcal{M}_k}(i)$ as the reward in state i corresponding to a Markov chain over $\mathcal{S}$ with transition probabilities $\{p_{ij}\}$, the states in $G \cup G'$ being absorbing states. Note that (6) implies that starting in any non-absorbing state, the average reward increases with time; moreover, eventual absorption is guaranteed with probability 1. Since the reward on absorbing states is at most $\frac{\alpha^N k}{1-\alpha}$ (see (7) and (8)), it follows that

$$f(i) = V^*_{\mathcal{M}_{k,N}}(i) - V^{\pi}_{\mathcal{M}_k}(i) \leq \frac{\alpha^N k}{1-\alpha} \quad \forall\, i \in \mathcal{S}.$$

This implies the statement of the lemma, taking π to be an optimal policy under $\mathcal{M}_k$. It now remains to prove (6) and (7).

To prove (6), consider the value function for any state $i \in \bar{G}$ under policy π

$$V^{\pi}_{\mathcal{M}_{k,N}}(i) = Z(s_m) + \alpha^m \mathcal{C}_N(s_m, \pi(s_m)) + \alpha^{m+1}\left(\sum_{j\in\mathcal{L}_0} p_{ij} V^{\pi}_{\mathcal{M}_{k,N}}(j)\right),$$

where $s_m \in \mathcal{L}^i_m$, $m \le N$, is the first state from which a sensing action is taken starting from i following π. Let B_π denote the Bellman operator corresponding to policy π, then

$$B^{m+1}_{\pi} V^{*}_{\mathcal{M}_{k,N}}(i) = Z(s_m) + \alpha^m \mathcal{C}_N(s_m, \pi(s_m)) + \alpha^{m+1}\left(\sum_{j\in\mathcal{L}_0} p_{ij} V^{*}_{\mathcal{M}_{k,N}}(j)\right).$$

Now observe that

$$\begin{aligned} B^{m+1}_{\pi} V^{*}_{\mathcal{M}_{k,N}}(i) - V^{\pi}_{\mathcal{M}_k}(i) &= \alpha^{m+1}\left(\sum_{j\in\mathcal{L}_0} p_{ij}\big(V^{*}_{\mathcal{M}_{k,N}}(j) - V^{\pi}_{\mathcal{M}_k}(j)\big)\right) \\ \implies V^{*}_{\mathcal{M}_{k,N}}(i) - V^{\pi}_{\mathcal{M}_k}(i) &\le \alpha^{m+1}\left(\sum_{j\in\mathcal{L}_0} p_{ij}\big(V^{*}_{\mathcal{M}_{k,N}}(j) - V^{\pi}_{\mathcal{M}_k}(j)\big)\right) \\ &\text{(since } V^{*}_{\mathcal{M}_{k,N}} \le B^{m+1}_{\pi} V^{*}_{\mathcal{M}_{k,N}} \text{ elementwise).} \end{aligned}$$

Note that the above inequality implies (6).

To prove (7), first note that for any state $j \in \mathcal{S}_n$ we have

$$V^{*}_{\mathcal{M}_{k,N}}(j) \le V^{\pi_{AS}}_{\mathcal{M}_{k,N}}(j) = V^{\pi_{AS}}_{\mathcal{M}_k}(j) \le V^{*}_{\mathcal{M}_k}(j) + \frac{k}{1-\alpha}.$$

Following similar steps as in the proof of (6), for any root state $j \in G$,

$$\begin{aligned} B^{N}_{\pi} V^{*}_{\mathcal{M}_{k,N}}(j) - V^{\pi}_{\mathcal{M}_k}(j) &\le \alpha^N (V^{*}_{\mathcal{M}_{k,N}}(s_N) - V^{\pi}_{\mathcal{M}_k}(s_N)) \le \frac{\alpha^N k}{1-\alpha}, \\ \implies V^{*}_{\mathcal{M}_{k,N}}(j) - V^{\pi}_{\mathcal{M}_k}(j) &\le \frac{\alpha^N k}{1-\alpha}, \end{aligned}$$

where $s_N \in \mathcal{L}^j_N$, is the state reached after playing N blind steps starting from j following π. This establishes (7). □

A.3 Proof of Lemma 4.3

Define π_{N+1} as an extension of the policy $\pi^{*}_{\mathcal{M}_{k,N}}$ for $\mathcal{M}_{k,N+1}$. Without loss of generality (W.L.O.G.), assume that $\pi^{*}_{\mathcal{M}_{k,N}}(\tilde{s}) \in A_s$ for all states $\tilde{s} \in \mathcal{L}_N$. Under π_{N+1}, states $\tilde{s} \in \bigcup_{l=0,1,\ldots,N} \mathcal{L}_l$ (i.e., all states from layers 0 to N) are mapped to actions provided by the policy $\pi^{*}_{\mathcal{M}_{k,N}}$, while states $\tilde{s} \in \mathcal{L}_{N+1}$ (i.e., states in the $N+1$ layer) are assigned arbitrary actions.

Let S^j_{exp} denote the sequence of states $s \in \mathcal{L}^j_m$ for $m \ge 0$ that are visited under $\pi^{*}_{\mathcal{M}_{k,N}}$ starting from the root state j (inclusive of j). Define $S_{exp} := \cup_j S^j_{exp}$. Furthermore, define

$$Z_{s_m}(s_T) = \sum_{i=m}^{T-1} \alpha^{i-m} \mathcal{B}(s_i)\mathcal{C}(a_i)$$

for $0 \le m \le T-1$, where the only difference from $Z(s_T)$ is that we start from state s_m at $t = 0$ and calculate the cumulative cost to reach s_T.

$$Z(i) + \alpha^{N+1} \min_a \Big(\mathcal{B}(i)\mathcal{C}(a) + k + \alpha\mathcal{B}(i)\mathcal{T}(a)V^{*}_{\mathcal{M}_{k,N}}\Big) \ge V^{*}_{\mathcal{M}_{k,N}}(j) \quad \forall j \in \mathcal{S},\ i \in \mathcal{L}^j_{N+1}. \tag{9}$$

We claim that (9) is a necessary and sufficient condition for the optimal actions to remain unchanged for all states $\tilde{s} \in S_{exp}$ in every step of policy iteration. This follows from the fact that there exists an improvable action at some state $\tilde{s} \in S_{exp}$ in $\mathcal{L}_m$ for the improved policy π'_{N+1} at some step of the policy iteration algorithm if and only if (10) is satisfied for some $i \in \mathcal{L}_{N+1}$ and $a \in A$.

$$Z_{\tilde{s}}(i) + \alpha^{N+1-m}\left(\mathcal{B}(i)\mathcal{C}(a) + k + \alpha\mathcal{B}(i)\mathcal{T}(a)V^*_{\mathcal{M}_{k,N}}\right) < V^{\pi'_{N+1}}_{\mathcal{M}_{k,N}}(\tilde{s}) \tag{10}$$

By inequality (9), we have

$$Z(\tilde{s}) + Z_{\tilde{s}}(i) + \alpha^{N+1-m}(\mathcal{B}(i)\mathcal{C}(a) + k + \alpha\mathcal{B}(i)\mathcal{T}(a)V^*_{\mathcal{M}_{k,N}}) \geq Z(\tilde{s}) + V^*_{\mathcal{M}_{k,N}}(\tilde{s}).$$

Simplifying, we get

$$Z_{\tilde{s}}(i) + \alpha^{N+1-m}(\mathcal{B}(i)\mathcal{C}(a) + k + \alpha\mathcal{B}(i)\mathcal{T}(a)V^*_{\mathcal{M}_{k,N}}) \geq V^*_{\mathcal{M}_{k,N}}(\tilde{s}),$$

which is the necessary and sufficient condition for the optimal value function of all root states to remain unchanged even when evaluated on $\mathcal{M}_{k,N+1}$.

A.4 Proof of Theorem 4.4

Proof. Suppose that (1) holds. Fix a state $i \in \mathcal{L}^j_{N+1}$ and consider a policy π^{ji} such that we traverse i starting from root state j by following this policy. Let $\mathcal{M}_0$ denote the corresponding MDP with no sensing cost. Then,

$$\begin{aligned}
&V^{\pi^{ji}}_{\mathcal{M}_0}(j) \geq Z(i) + \alpha^{N+1}V^*_{\mathcal{M}_0}(i),\\
&V^{\pi^{ji}}_{\mathcal{M}_0}(j) \geq Z(i) + \alpha^{N+1}\min_a\left(\mathcal{B}(i)\mathcal{C}(a) + \alpha\mathcal{B}(i)\mathcal{T}(a)V^*\right),\\
&V^{\pi^{ji}}_{\mathcal{M}_0}(j) \geq Z(i) + \alpha^{N+1}V_{AS;0}(i).
\end{aligned}$$

Let π^j_M be a policy such that, starting from root state j and following π^j_M, we take $M > N$ consecutive blind steps. Note that

$$V^{\pi^j_M}_{\mathcal{M}_0}(j) \geq \min_{i\in\mathcal{L}^j_{N+1}}\left(Z(i) + \alpha^{N+1}V_{AS;0}(i)\right). \tag{11}$$

W.L.O.G., assume $\pi^*_{\mathcal{M}_{k,N}}(\tilde{s}) \in A_s$ for all $\tilde{s} \in \mathcal{L}_N$. If condition (1) holds, then

$$\begin{aligned}
&V^*_{\mathcal{M}_{k,N}}(j) \leq \min_{i\in\mathcal{L}^j_{N+1}}\left(Z(i) + \alpha^{N+1}V_{AS;0}(i)\right) \leq V^{\pi^M_j}_{\mathcal{M}_0}(j) \leq V^{\pi^M_j}_{\mathcal{M}_k}(j)\\
\implies &V^*_{\mathcal{M}_k}(j) \leq V^{\pi^*_{\mathcal{M}_{k,N}}}_{\mathcal{M}_k}(j) = V^*_{\mathcal{M}_{k,N}}(j) \leq V^{\pi^M_j}_{\mathcal{M}_k}(j)
\end{aligned}$$

Thus, the optimal policy for $\mathcal{M}_k$ takes at most N consecutive blind steps starting from j, and consequently, when (1) holds, $V_{\mathcal{M}_{k,N}}(j) = V_{\mathcal{M}_k}(j)$ for all $j \in \mathcal{S}$.

Now consider a scenario where condition (1) does not hold and notice that this condition is not satisfied if and only if $\epsilon_N > 0$. Exactly as in the proof of Theorem 4.2 (see Section A.2), for a stationary policy π of $\mathcal{M}_k$, define a set of root states G such that, starting from any state $j \in G$ and following π, at least $N+1$ consecutive blind steps are taken. Similarly, define $\bar{G} := \mathcal{S} \setminus G$. We have already proved that for root states $j \in G$,

$$\begin{aligned}
&V^{\pi}_{\mathcal{M}_k}(j) \geq \min_{i\in\mathcal{L}^j_{N+1}}\left(Z(i) + \alpha^{N+1}V_{AS;0}(i)\right)\\
\implies &V^*_{\mathcal{M}_{k,N}}(j) - V^{\pi}_{\mathcal{M}_k}(j) \leq \epsilon_N, \quad \forall i \in G.
\end{aligned} \tag{12}$$

Now, the value function for any state $l \in \bar{G}$ under policy π can be represented as

$$V^{\pi}_{\mathcal{M}_{k,N}}(l) = Z(s_m) + \alpha^m\mathcal{C}_N(s_m, \pi(s_m)) + \alpha^{m+1}\left(\sum_{j\in\mathcal{L}_0} p_{lj}V^{\pi}_{\mathcal{M}_{k,N}}(j)\right),$$

where $s_m \in \mathcal{L}_m^l$, $m \leq N$, is the first state from which a sensing action is taken starting from l following π. Let B_π is the Bellman operator corresponding to policy π, then

$$B_\pi^{m+1} V^*_{\mathcal{M}_{k,N}}(l) = Z(s_m) + \alpha^m \mathcal{C}_N(s_m, \pi(s_m)) + \alpha^{m+1}\left(\sum_{j\in\mathcal{L}_0} p_{lj} V^*_{\mathcal{M}_{k,N}}(j)\right).$$

Now observe that

$$\begin{aligned} B_\pi^{m+1} V^*_{\mathcal{M}_{k,N}}(l) - V^\pi_{\mathcal{M}_k}(l) &= \alpha^{m+1}\left(\sum_{j\in\mathcal{L}_0} p_{lj}\big(V^*_{\mathcal{M}_{k,N}}(j) - V^\pi_{\mathcal{M}_k}(j)\big)\right) \\ \implies V^*_{\mathcal{M}_{k,N}}(l) - V^\pi_{\mathcal{M}_k}(l) &\leq \alpha^{m+1}\left(\sum_{j\in\mathcal{L}_0} p_{lj}\big(V^*_{\mathcal{M}_{k,N}}(j) - V^\pi_{\mathcal{M}_k}(j)\big)\right) \\ &\text{(since } V^*_{\mathcal{M}_{k,N}} \leq B_\pi^{m+1} V^*_{\mathcal{M}_{k,N}} \text{ elementwise)} \end{aligned}$$

where p_{lj} denotes the probability of landing in root state $j \in \mathcal{S}$ after taking the first sensing step when starting from l and following π. Consider $G' = \{s \mid s \in \bar{G} \text{ and } s \not\to i, \forall i \in G\}$, i.e., starting from any state $s \in G'$ we never reach any state $i \in G$, under policy π on $\mathcal{M}_{k,N}$. Therefore, we have

$$\begin{aligned} &V^*_{\mathcal{M}_{k,N}}(i) - V^\pi_{\mathcal{M}_k}(i) \leq \epsilon_N, \quad \forall i \in G, \\ &V^*_{\mathcal{M}_{k,N}}(i) - V^\pi_{\mathcal{M}_k}(i) = 0, \quad \forall i \in G', \\ &V^*_{\mathcal{M}_{k,N}}(i) - V^\pi_{\mathcal{M}_k}(i) \leq \alpha^{a_i}\Big(\sum_{j\in\mathcal{L}_0} p_{ij}\big(V^*_{\mathcal{M}_{k,N}}(j) - V^\pi_{\mathcal{M}_k}(j)\big)\Big), \quad \forall i \in \bar{G}, \end{aligned}$$

where a_i's are policy π and root state-dependent constants, with $a_i \geq 1$. Identical to the argument made in Section A.2, treat $f(i) := V^*_{\mathcal{M}_{k,N}}(i) - V^\pi_{\mathcal{M}_k}(i)$ as the reward in state i corresponding to a Markov chain over $\mathcal{S}$ with transition probabilities $\{p_{ij}\}$, where the states in $G \cup G'$ are absorbing. Since the reward on absorbing states is at most ϵ_N (see (12)), it follows that $f(i) = V^*_{\mathcal{M}_{k,N}}(i) - V^\pi_{\mathcal{M}_k}(i) \leq \alpha\epsilon_N$ for all $i \in \bar{G} \setminus G'$. (for all non-absorbing states). This establishes (2)

NOTE:
1. Even if condition (17) is satisfied and the optimal policy for a root state j of $\mathcal{M}_k$ is restricted to having a maximum of N consecutive blind steps, it does not generally imply that $V^*_{\mathcal{M}_k}(j) = V^*_{\mathcal{M}_{k,N}}(j)$.

2. It follows from the same proof that the stronger claim below holds for any root state j:

$$V^*_{\mathcal{M}_k}(j) \geq \min\Big\{\min_{i\in\mathcal{L}^j_{N+1}}\big(Z(i) + \alpha^{N+1}V_{AS;0}(i)\big), V^*_{\mathcal{M}_{k,N}}(j) - \alpha \max_{s\in\mathcal{S}\setminus\{j\}}\Big[V^*_{\mathcal{M}_{k,N}}(s) - \min_{r\in\mathcal{L}^s_{N+1}}\big(Z(r) + \alpha^{N+1}V_{AS;0}(r)\big)\Big]^+\Big\}.$$

Here, $[x]^+$ denotes the positive part of x.

Idea: Define a separate terminal value for each of the states $j \in G$, given by

$$V^*_{\mathcal{M}_{k,N}}(j) - V^\pi_{\mathcal{M}_k}(j) \leq V^*_{\mathcal{M}_{k,N}}(j) - \min_{i\in\mathcal{L}^j_{N+1}}\big(Z(i) + \alpha^{N+1}V_{AS;0}(i)\big).$$

□

A.5 Proof of Lemma E.1

Proof. Let the baseline MDP $\mathcal{M}$ be defined according to the conditions specified in the lemma. For any state $\tilde{s} \in \mathcal{S}_\infty$, we have

$$V_{AS;0}(\tilde{s}) = \mathcal{B}(\tilde{s})\mathcal{C}(\pi_{AS}(\tilde{s})) + \alpha\mathcal{B}(\tilde{s})\mathcal{T}(\pi_{AS}(\tilde{s}))V^*, \tag{13}$$

where $\mathcal{B}(\tilde{s}) = \mathcal{B}(\tilde{s})\mathcal{T}(\pi_{AS}(\tilde{s}))$. We claim that for the above non-trivial MDP, for each $a \in A$, there exists a root state r such that

$$V^*(r) < \mathcal{C}(r, a) + \alpha e_r \mathcal{T}(a)V^*. \tag{14}$$

Also note that $\exists\ N^*$ s.t. $\forall N \geq |\mathcal{S}| - 1$, every element of $\mathcal{B}(\tilde{s})$ is non-zero $\forall \tilde{s} \in \mathcal{L}_N$. Hence, by applying the inequality from (14), we obtain

$$\begin{aligned} &\mathcal{B}(\tilde{s})V^* < \mathcal{B}(\tilde{s})\mathcal{C}(a) + \alpha\mathcal{B}(\tilde{s})\mathcal{T}(a)V^* \quad \forall a \in A, \\ \implies &\mathcal{B}(\tilde{s})V^* < V_{AS;0}(\tilde{s}). \end{aligned} \tag{15}$$

Thus, for all $\tilde{s} \in \mathcal{L}_N$, where $N \geq |\mathcal{S}| - 2$, applying (15) to the definition of $V_{AS;0}(\tilde{s})$ in (13), we obtain

$$V_{AS;0}(\tilde{s}) < \mathcal{B}(\tilde{s})\mathcal{C}(a) + \alpha V_{AS;0}(\tilde{s}_a) \quad \forall a \in A \tag{16}$$

Now let i' be the state reached after taking a blind step with action a from state $i \in \mathcal{L}_N$. Then, it immediately follows from (16) that

$$Z(i) + \alpha^N V_{AS;0}(i) < Z(i) + \alpha^N \left(\mathcal{B}(i)\mathcal{C}(a) + \alpha V_{AS;0}(i')\right) = Z(i') + \alpha^{N+1} V_{AS;0}(i').$$

Thus

$$\min_{i \in \mathcal{L}_N^j} \left(Z(i) + \alpha^N V_{AS;0}(i)\right) < \min_{i \in \mathcal{L}_{N+1}^j} \left(Z(i) + \alpha^{N+1} V_{AS;0}(i)\right)$$
$$\epsilon_{N+1} < \epsilon_N.$$

To prove $\epsilon_{N+1} \leq \epsilon_N$, for a general MDP, we simply replace the strict inequalities in (15) and (16) with non-strict ones. □

B Experimental Design and Setup

Experimental Setup: Experiments were conducted on a MacBook Air with an Apple M3 chip and 16GB of memory. Hyperparameters were set according to the defaults in `POMDPs.jl` Egorov et al. (2017), with adjustments clearly stated and made to ensure comparable performance or runtime. For SARSOP, the reward and sensing values were scaled by a factor of 1000 to match performance on the Frozen Lake task, and the policy computation time was increased from 1s to 100s for the Taxi task. It is important to note that this scaling was applied only during the policy computation phase using SARSOP with its default hyperparameters. The resulting policy was then evaluated in the original, unscaled environment, consistent with the evaluation of SPI and other methods, which used unscaled values throughout. Without this scaling, SARSOP performed poorly on the task—even relative to π_{ATM} and FIB.

Initialization of POMDP Baselines: The POMDP planning algorithms have been initialized with a belief distribution corresponding to the starting state distribution, rather than a uniform distribution over all states. For example, in Frozen Lake, where the starting state is deterministic, the initialized belief is a degenerate distribution concentrated entirely on the initial state. In contrast, for Stochastic Taxi, the initialized belief is uniform over the 300 valid states from which the initial state is uniformly sampled.

Choice of Domain: Our choice of the ICU-Sepsis benchmark environment Choudhary et al. (2024) is motivated by Nam et al. (2021), who employ a similar tabular sepsis simulator to model personalized treatment strategies for ICU patients under state observation costs. ICU-Sepsis is constructed from real medical data (MIMIC-III database) and serves as a standardized benchmark for evaluating RL algorithms.

We also incorporate gridworld-based domains inspired by Bellinger et al. (2021), who utilize standard RL Gym environments such as FrozenLake 8×8 (see Table 2c) and Taxi to benchmark RL algorithms for MDPs with state-sensing costs. While Nam et al. (2021) & Bellinger et al. (2021) focused on learning-based RL algorithms, we adapted these domains to evaluate planning algorithms.

To evaluate performance in smaller state spaces, we conducted experiments on both the standard Frozen Lake 4×4 environment and a custom-designed "hard" variant, as shown in Tables 2a and 2b, respectively. The results from the custom hard setup revealed that π_{ATM} and FIB may struggle to find effective policies even in small state spaces when faced with challenging planning problems, thus highlighting the robustness of SPI in such scenarios.

We also conducted experimental analysis on other non-grid world-based domains, such as an Inventory Management. This case study, detailed in the Appendix G, features two distinct sensing cost scenarios. We benchmarked our

SFFF FHFH FFFH HFFG	FHSF FGHF FHHF FFFF	SFFFFFFF FFFFFFFF FFFHFFFF FFFFFHFF FFFHFFFF FHHFFFHF FHFFHFHF FFFHFFFG	FHFF FGHF FHHF FFFS
(a) Default 4×4 grid	(b) Custom-hard 4×4 grid	(c) Default 8×8 grid	(d) Custom-hard 4×4 with start state **S** closer to goal

Table 2: **Frozen Lake grid environments** used in our experiments.

algorithm against π_{ATM}, the truncated MDP approach, and utilized 18 to derive suboptimality bounds on their performance.

We highlight that we tested nearly all offline `POMDPs.jl` planners (Incremental Pruning, PBVI, QMDP, etc) but excluded them from the experimental setup due to prohibitive runtimes/ineffective policies.

Setting hyperparameters for SPI. The SPI heuristic relies on two interpretable hyperparameters: *maxsteps*, which limits the length of blind action sequences explored by PolicyUpdate for each root state s, and δ, the improvement threshold that governs the termination of PolicyUpdate iterations. Below, we outline practical guidelines for setting these parameters.

• The choice of *maxsteps* is dependent on the desired precision of the Value function. We recommend choosing *maxsteps* such that $\alpha^{maxsteps} \cdot k$ is smaller than the desired precision (or one order of magnitude smaller). For instance, in the ICU-Sepsis environment ($\alpha = 0.99$, maximum sensing cost from Table 1 being $k = 0.1$), since we report and compare values only up to 3 decimal places, we set $maxsteps = 500$ as $0.99^{500} \cdot 0.1 \approx 6.6 \times 10^{-4} < 10^{-3}$.

• The improvement threshold δ controls the number of iterations in PolicyUpdate, terminating when the value function improvement falls below δ for all root states. We suggest setting δ to be one order of magnitude below the desired precision. Alternatively, the PolicyUpdate iterations in SPI can be capped at a fixed number (e.g., 4–5), as SPI converges rapidly. For instance, in the ICU-Sepsis task, the improvement falls below 10^{-4} after two iterations, and for other benchmark environments, it falls below 10^{-6} in 3–4 iterations.

B.1 Performance of POMCP on Frozen Lake

In the case of Frozen Lake, the belief distribution for all the POMDP solvers was solely concentrated on the single starting state **S**. Despite this initialization, the PO-UCT algorithm performed significantly worse than SPI. For instance, in the Frozen Lake 4x4 (Hard) environment, even when the starting state was positioned in the bottom-right corner, closer to the goal (see Table 2d), the PO-UCT algorithm—with over 290k rollouts ($\approx$ 500 sec) per planning step—achieved an average return of 1.91×10^{-2}. In contrast, SPI and π_{ATM} achieved average returns of 5.44×10^{-2} and 4.7×10^{-2}, respectively. Moreover, as the starting state moved farther from the goal, the expected return of PO-UCT declined sharply compared to SPI, likely due to PO-UCT's poor performance with sparse rewards, which requires significantly more planning time.

C Analysis of SPI

C.1 Performance Guarantees of SPI

PolicyUpdate guarantees that the output policy π_o is at least as "good" as the input reference policy π_{ref} for all root states ($\pi_o \overset{\mathcal{S}}{\succeq} \pi_{ref}$). Consequently, the updated policy π_{improv}, resulting from each application of PolicyUpdate within SPI, is at least as effective as the previous policy for all root states and reduces the value function by δ for at least one root state at each iteration until termination. We can, therefore, derive a theoretical upper bound on the number of PolicyUpdate steps.

SPI is guaranteed to terminate for any $\delta > 0$ due to the assumption of a bounded cost function and a finite state space. Specifically, since PolicyUpdate produces a policy π_{improv} such that $\pi_{\text{improv}} \overset{\mathcal{S}}{\succeq} \pi'$, and the value

Algorithm 3 Act-Then-Measure Heuristic (ATM)

Input: $\tilde{s} \in \mathcal{S}_\infty$ (state)
Output: Action to play $\tilde{a} \in \mathcal{A}_\infty$

$a = \pi_{AS}(\tilde{s})$
if $\alpha\Big(V_{AS}\big(\mathcal{B}(\tilde{s})\mathcal{T}(a)\big) - \mathcal{B}(\tilde{s})\mathcal{T}(a)V^*\Big) < \dfrac{k}{1-\alpha}$ **then**
    Play $(a, blind)$
else
    Play $(a, sense)$
end if

function cannot decrease by more than δ for some state indefinitely—due to the optimal value function being lower bounded—the algorithm must eventually converge.

Moreover, if the initial policy π_{init} is chosen to be the Always-Sense (AS) policy—as in our numerical experiments (Section 5)—then SPI is guaranteed to terminate within at most $\frac{k|\mathcal{S}|}{\delta(1-\alpha)}$ iterations. This follows from the fact that the AS policy is suboptimal by at most $\frac{k}{1-\alpha}$ for any state, and each PolicyUpdate step decreases the value function by at least δ for some state. In practice, however, SPI typically converges in far fewer iterations—even in large state spaces. For instance, in the ICU-Sepsis task, the improvement in return fell below 10^{-4} after just two iterations.

C.2 Adapting SPI to Continuous State Spaces

SPI can be naturally extended to continuous state spaces using function approximation, an avenue we find promising for future research. Below, we provide an informal sketch of how PolicyUpdate (and consequently SPI) can be adapted to continuous state spaces (with finite action space):

1. Our policy could be represented as a neural network with a final softmax activation layer, mapping the state features of the last sensed root state, concatenated with the sequence of blind actions played thereafter, to a probability distribution over the action space. Thus, the belief state features are defined as the root state features of the corresponding root state concatenated with the sequence of blind actions.
2. We first note that the PolicyUpdate algorithm, in all its steps, requires the value function to be evaluated for the reference policy only at the root states. Therefore, it is sufficient to have a critic (value function) neural network that is accurate at the root state features. Since we are in a planning setup, we have access to the transition probability function and reward functions, allowing us to train the critic to fit the value function corresponding to π_{ref} at the root state features by rolling out the reference policy π_{ref} at root states.
3. Since we are dealing with a continuous state space setup, we replace Line 2 of the PolicyUpdate algorithm to sample states from some distribution over the root states (e.g., root state visit distribution of π_{ref}), because it is not feasible to carry the procedure for all root states in a continuous state space.
4. An estimate of the value of $\mathcal{B}(\tilde{s})\mathcal{C}(a)$ at a belief state $\tilde{s}$ for action a, which is necessary in Lines 9 and 10 and for evaluating V_{MS}, can be easily obtained using the transition function and cost function.
5. In this setting, Lines 12 and 15 correspond to fitting the candidate policy π' by using the belief state features as input to produce the corresponding suitable action as output. Similarly, Line 21 updates the output policy π_o in the same manner. For the comparison between the value functions performed in Line 19, a critic network for π' can be trained using an approach similar to that described above for π_{ref}.

The only change required in the SPI algorithm is in Line 3, where we could simply terminate after a fixed number of iterations.

D Analysis of ATM Heuristic

In this section, we provide an alternative perspective for understanding and analyzing the ATM heuristic (Krale et al., 2023), adapted to our expanded state-space framework. The proposed heuristic, formalized in Algorithm 3,

operates as follows. Given a state $\tilde{s}$, it compares the (expected discounted) cost of following the AS policy with that of taking the one-time blind action $(\pi_{AS}(\tilde{s}), blind)$ and subsequently following the AS policy. If the former cost is less, the algorithm selects the sensing action $(\pi_{AS}(\tilde{s}), sense)$, i.e., it follows the action prescribed by AS; otherwise, it executes the blind action $(\pi_{AS}(\tilde{s}), blind)$.

We demonstrate that the ATM heuristic, in fact, arises from a policy improvement step; consequently, the following guarantee holds.

Theorem D.1. *The policy described by Algorithm 3 dominates any always-sensing policy.*

Proof. It suffices to show that Line 2 in Algorithm 3 constitutes a policy improvement step. For any $a \in A$,

$$Q_{AS}(\tilde{s}, (a, sense)) = B(\tilde{s})\mathcal{C}(a) + \alpha B(\tilde{s})\mathcal{T}(a)V^* + \frac{k}{1-\alpha},$$
$$Q_{AS}(\tilde{s}, (a, blind)) = B(\tilde{s})\mathcal{C}(a) + \alpha V_{AS}(B(\tilde{s})\mathcal{T}(a)).$$

It is therefore easy to check that

$$Q_{AS}(\tilde{s}, (a, blind)) < Q_{AS}(\tilde{s}, (a, sense))$$
$$\iff \alpha\big(V_{AS}(\mathcal{B}(\tilde{s})\mathcal{T}(a)) - \mathcal{B}(\tilde{s})\mathcal{T}(a)V^*\big) < \frac{k}{1-\alpha}.$$

□

E Additional Results

E.1 Extension of Theorem 4.4

Theorem 4.4 establishes a sufficient condition (1) for the optimal policy of $\mathcal{M}_{k,n}$ to also be optimal for $\mathcal{M}_k$. However, this condition (1) must hold for all root states for the result to apply.

A stronger result follows from the proof of Theorem 4.4: If, for any root state j, it holds that

$$Z(i) + \alpha^{N+1}V_{AS;0}(i) \geq V^*_{\mathcal{M}_{k,N}}(j) \quad \forall\, i \in \mathcal{L}^j_{N+1}, \tag{17}$$

then the optimal policy for $\mathcal{M}_k$ takes at most N consecutive blind steps starting from j. Thus, if (17) is satisfied at certain root states, one does not need to explore depths $N+1$ and beyond at these root states.

Finally, as the following lemma shows, the suboptimality bound ϵ_N is decreasing in N.

Lemma E.1. *It always holds that $\epsilon_{N+1} \leq \epsilon_N$, where ϵ_N is as defined in the statement of Theorem 4.4. Furthermore, if $\mathcal{M}$ is irreducible, and there exists no action that is optimal for all states in the baseline MDP, then $\epsilon_{N+1} < \epsilon_N$ for all $N \geq |S| - 2$.*

Additionally, Theorem 4.4 provides a computable upper bound on the suboptimality of the policy $\pi^*_{\mathcal{M}_{k,N}}$ for $\mathcal{M}_k$. Building on its proof in Appendix A.4, we derive a stronger result for computing a lower bound on $V^*_{\mathcal{M}_k}$: For any root state j,

$$V^*_{\mathcal{M}_k}(j) \geq \min\Big\{ \min_{i \in \mathcal{L}^j_{N+1}} \big(Z(i) + \alpha^{N+1}V_{AS;0}(i)\big), V^*_{\mathcal{M}_{k,N}}(j) - \alpha \max_{s \in \mathcal{S}\setminus\{j\}} \Big[V^*_{\mathcal{M}_{k,N}}(s) - \min_{r \in \mathcal{L}^s_{N+1}} \big(Z(r) + \alpha^{N+1}V_{AS;0}(r)\big)\Big]^+ \Big\}. \tag{18}$$

Here, $[x]^+$ denotes the positive part of x.

Note: The lower bound on $V^*_{\mathcal{M}_k}$ in Figure 3 was computed using the stronger bound provided above.

E.2 Counter-example related to Lemma 4.3

We begin with an example that demonstrates that starting at a root state, if a certain policy π is optimal for $\mathcal{M}_{k,N}$ as well as $\mathcal{M}_{k,N+1}$, that does not guaranteee that the π is also optimal for $\mathcal{M}_k$. Consider the two-state two-action baseline MDP shown in Figure 4, with sensing cost $k = 0.005$ and discount factor $\alpha = 1/2$.

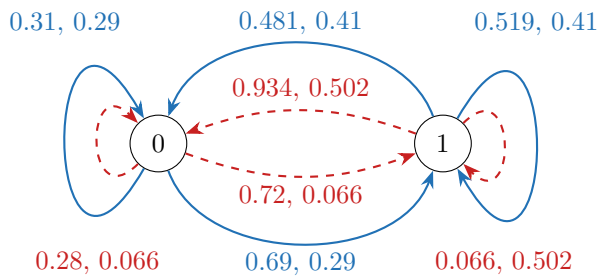


Figure 4: **Two-state two-action MDP** with actions $\{Red, Blue\}$, $k = 0.005$ and $\alpha = 0.5$

N	$\pi^*_{\mathcal{M}_{k,N}}(0)$	$V^*_{\mathcal{M}_{k,N}}(0)$	$\pi^*_{\mathcal{M}_{k,N}}(1)$	$V^*_{\mathcal{M}_{k,N}}(1)$
0	R	0.367061	B	0.6796465
1	R	0.367061	B	0.6796465
2	R	0.367061	B	0.6796465
3	R	0.367061	B	0.6796465
4	R	0.36703456	BRRRR	0.67958256
5	R	0.367029	BRRRRR	0.6795691
6	R	0.3670226	BRRRRRR	0.6795541

Table 3: **Change in optimal policy and value function** with the number of blind steps for baseline MDP in Figure 4

The optimal policy and value function corresponding to the MDPs $\mathcal{M}_{k,N}$ for different choices of N are tabulated in Table 3. The optimal policy is shown as the sequence of actions to take starting at any root state, terminating in a sensing action; for example, 'BRRRR' means to take the sequence of blind actions 'BRRR' and then the sensing action 'R.' Note that for $N \leq 3$, the optimal policy at root states for $\mathcal{M}_{k,N}$ is to take a sensing action. However, for $N \geq 4$, is optimal to make a sequence of blind steps in root state 1.

Interestingly, we find in this example that the criterion for Theorem 4.4 (see E.1) is satisfied for root state 0 at $N = 2$. Therefore, it is clear at that point that the optimal policy for root state 0 will take a maximal of 2 blind steps, and we can restrict our search for the optimal policy starting at 0 until the 2nd layer. The same criterion is *not* satisfied for root state 1 for $N \leq 6$.

F Extension to Non-Uniform Sensing Costs

Sensing costs can often be context-dependent and may depend on resource constraints or belief uncertainty. The constraint of non-uniform sensing costs can be easily incorporated in our setup (for both the truncated MDP approach as well as the SPI heuristic along with the theoretical results) by modelling the sensing cost as state-action-dependent for the augmented MDP, i.e., $k' : \mathcal{S}_\infty \times A \to \mathbb{R}$. For example, in the case where the sensing cost depends on belief uncertainty, the sensing cost can simply be treated as a function of the state in the augmented MDP since a state (of the augmented MDP) directly relates to the belief distribution over the root states. More formally, with such constraints,

1. **Modification to Truncated MDP approach:** The cost function $\mathcal{C}_\infty$ associated with $\mathcal{M}_{k'}$ will be suitably modified as follows

$$\mathcal{C}_\infty(\tilde{s}, (a, sense)) = \mathcal{B}(\tilde{s})\mathcal{C}(a) + k'(\tilde{s}, a)$$
$$\mathcal{C}_\infty(\tilde{s}, (a, blind)) = \mathcal{B}(\tilde{s})\mathcal{C}(a)$$

   This will, in turn, lead to the modification of the cost function $\mathcal{C}_n : \mathcal{S}_n \times \mathcal{A}_n \to \mathbb{R}$ associated with the truncated MDP $\mathcal{M}_{k',n}$. The rest of the truncated MDP parameters remain unchanged, and we can directly compute its optimal policy and value function for the truncated MDP.

2. **Modification to SPI:** The myopic sensing value function V_{MS} and its corresponding policy π_{MS} will also depend on the state and will be modified as follows:

$$V_{MS}(\mathcal{B}(\tilde{s}), \bar{V}) = \min_{a \in A}\big(B(\tilde{s})\mathcal{C}(a) + \alpha B(\tilde{s})\mathcal{T}(a)\bar{V} + k'(s,a)\big)$$
$$\pi_{MS}(\mathcal{B}(\tilde{s}), \bar{V}) = \arg\min_{a \in A}\big(B(\tilde{s})\mathcal{C}(a) + \alpha B(\tilde{s})\mathcal{T}(a)\bar{V} + k'(s,a)\big)$$

   With this simple modification, SPI extends to scenarios with non-uniform sensing costs.

3. **Modification to Theoretical Results:** As with the truncated MDP approach and SPI, our results can be similarly extended to scenarios with non-uniform sensing costs, with simple modifications to its proof. For instance, Theorem 4.2 is modified as follows:

$$V^*_{\mathcal{M}_{k',N}}(j) - V^*_{\mathcal{M}_{k'}}(j) \leq \frac{\alpha^N \tilde{k}}{1-\alpha} \quad \forall j \in \mathcal{S}, N \geq 0,$$

   where

$$\tilde{k} = \sup_{a \in A, \tilde{s} \in \mathcal{S}_\infty \setminus \mathcal{L}_{N-1}} k'(\tilde{s}, a) \quad \text{for } N \geq 1,$$
$$\tilde{k} = \sup_{a \in A, \tilde{s} \in \mathcal{S}_\infty} k'(\tilde{s}, a) \quad \text{for } N = 0.$$

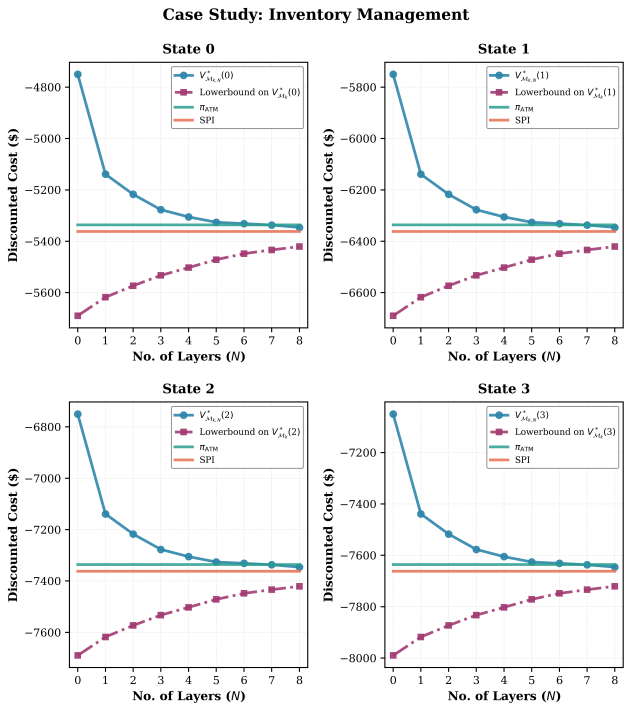
G Inventory Management Case Study

This example is adapted from Bradley et al. (1977). We consider an inventory with a capacity of 3 units. The demand for items is either 1 or 2 units, each with probability 1/2 at every step (month). The production cost for an item is \$1000 per unit, while the selling price stands at \$2000 per unit, ensuring a profit of \$1000 units per sale. We consider a holding cost of \$500 on each month for each remaining item in the inventory by month-end.[2] Furthermore, consider a sensing cost of either \$200 or \$64 for observing the remaining items in the inventory (the state), and we aim to maximize the discounted profit with the discounting factor $\alpha = 0.8$.

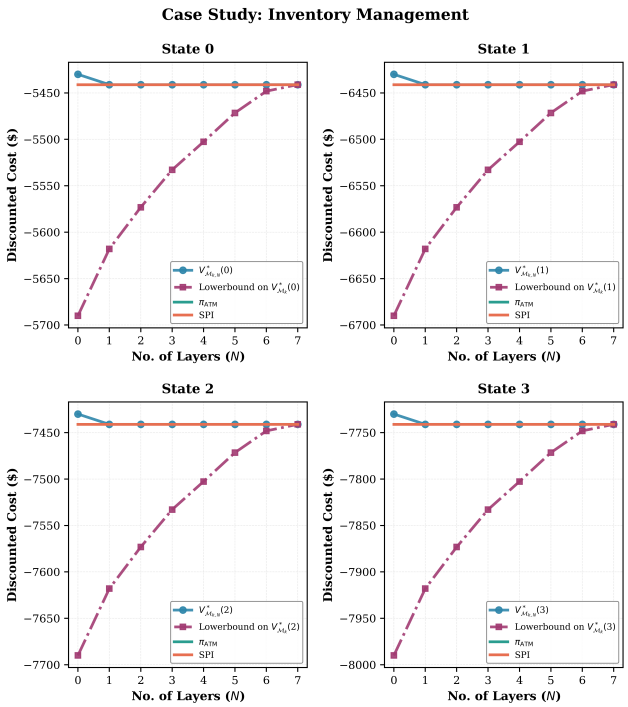
Our main takeaways are as follows. For sensing cost \$200, our results are shown in Figure 5a. In this case, we see that SPI (Section 3) performs quite close to the optimal policy (judging by the bound on sub-optimality gap) and the optimal policy for the truncated MDP $\mathcal{M}_{k,N}$ outperforms it only after $N \geq 7$. However, the conditions of Theorem 4.4 are not satisfied over the depths N we were able to compute for (recall that the computational complexity of solving $\mathcal{M}_{k,N}$ grows exponentially in N). This is consistent with the results in Figure 5a; we continue to see small cost benefits from increasing the threshold on the number of blind actions allowed.

For the lower sensing cost of \$64, our results are shown in Figure 5b. In this case, we see that the heuristic policy (Section 3), which does provide an improvement over always sensing, is in fact optimal for $\mathcal{M}_k$. Moreover, the optimal policy for $\mathcal{M}_{k,1}$ is also found to be optimal for $\mathcal{M}_k$. However, the condition of Theorem 4.4 is only satisfied at $N = 7$.

[2] Holding cost is evaluated based on the no. of remaining items of the inventory at the end of the month after meeting the demand.



(a) Inventory management with sensing cost $200



(b) Inventory management with sensing cost $64

Figure 5: **Applying Thm 4.4** to Inventory Management