Perception-Aware Point-Based Value Iteration for Partially Observable Markov Decision Processes

Mahsa Ghasemi and **Ufuk Topcu**
University of Texas at Austin
{mahsa.ghasemi, utopcu}@utexas.edu

Abstract

In conventional partially observable Markov decision processes, the observations that the agent receives originate from fixed known distributions. However, in a variety of real-world scenarios, the agent has an active role in its perception by selecting which observations to receive. We avoid combinatorial expansion of the action space from integration of planning and perception decisions, through a greedy strategy for observation selection that minimizes an information-theoretic measure of the state uncertainty. We develop a novel point-based value iteration algorithm that incorporates this greedy strategy to pick perception actions for each sampled belief point in each iteration. As a result, not only the solver requires less belief points to approximate the reachable subspace of the belief simplex, but it also requires less computation per iteration. Further, we prove that the proposed algorithm achieves a near-optimal guarantee on value function with respect to an optimal perception strategy, and demonstrate its performance empirically.

1 Introduction

In the era of information explosion it is crucial to develop decision-making platforms that are able to judiciously extract useful information to accomplish a task. The importance of mining useful information from large data appears in many applications including artificial intelligence, robotics, networked systems and internet of things.

Partially observable Markov decision processes (POMDPs) provide a framework to model sequential decision-making with partial perception of the environment and stochastic outcomes. While relatively efficient algorithms for computing near-optimal policies have been developed, the majority of existing algorithms focus on either merely perception or merely planning.

In this paper, we address joint perception and planning in POMDPs. In particular, we consider an agent that decides about two sets of actions: perception actions and planning actions. The perception actions, such as activating a sensor, only affect the belief of the agent regarding the environment. The planning actions, such as choosing a navigation direction, only affect the environment evolution. To the best of our knowledge, this is the first work that considers the problem of joint active perception and planning in POMDPs.

While treating perception and planning in isolation from each other likely deteriorates performance, an integrated approach is typically intractable. Essentially, the complexity is roughly dictated by how much perception and planning rely on each other. Therefore, a trade-off between optimality and tractability is necessary.

The main contribution of this paper is establishing near-optimal and tractable solutions for a class of problems where perception is defined as picking a subset of information sources. Essentially, we prove that it is possible to decouple the perception action space and the planning action space yet still achieve near-optimal strategies. What enables this decoupling is the fact that the perception strategy aims to reduce uncertainty. Hence, one can use approximate algorithms, such as greedy methods, to optimize a quantitative measure of uncertainty. Furthermore, we develop a novel POMDP solver through which we can evaluate and hence optimize the joint effect of perception and planning actions on the overall value of a strategy, even though the action spaces are decoupled.

The class of active perception considered in this paper, i.e., picking the most useful information sources, resembles the well-established problem of subset selection [Krause and Golovin, 2014; Qian *et al.*, 2017]. This type of active perception arises in various applications in control systems, robotics, and machine learning, where the constraints on sensing stem from power, processing capability, or communication limits.

1.1 Contributions

We point to the main contributions below.

Problem formulation. We introduce a new mathematical definition of POMDPs, called AP^2-POMDP, that captures active perception and planning. The objective is to find pure belief-based policies for perception and planning such that the expected discounted cumulative reward is maximized.

Algorithm development. To solve AP^2-POMDP, we develop a novel point-based method that approximates the value function using a finite set of belief points, each associated with a pair of perception and planning actions. We exploit the

uncertainty reduction purpose of perception actions to devise a greedy perception decision that is conditioned on a belief point and a planning decision. The value iteration step then integrates the effects of a pair of perception and planning actions on the expected cumulative reward.

Theoretical guarantees. We establish theoretical guarantees on the near-optimality of the greedy perception decision with respect to an optimal perception decision. Subsequently, we prove near-optimality of the value function obtained by the proposed algorithm. We also provide complexity analysis of the algorithm to demonstrate the computational gain.[1]

1.2 Related Work

Finding exact solution to POMDPs is PSPACE-complete [Papadimitriou and Tsitsiklis, 1987]. Hence, near-optimal algorithms have been subject to extensive research. A common technique is to approximate the reachable subspace of belief by a finite set and apply value iteration over this set [Sondik, 1978; Cheng, 1988; Lovejoy, 1991; Zhang and Zhang, 2001]. Pineau *et al.* [2006] proves that the error due to belief sampling is bounded and depends on the density of the belief set. Well-established offline POMDP solvers include SARSOP [Kurniawati *et al.*, 2008] and HSVI [Smith and Simmons, 2012], that guide the belief sampling toward the reachable subspace under optimal policies. We show that the proposed greedy observation selection scheme leads to belief points that are, in expectation, close to the ones from the optimal set of observations.

An instance of active perception is dynamic sensor selection. Kreucher *et al.* [2005] use Rènyi divergence to compute the utility of sensing actions. In a setting of Kalman filtering, Shamaiah *et al.* [2010] and Hashemi *et al.* [2018] develop greedy selection schemes, with near-optimality guarantees, to minimize scalarizations of the error covariance matrix. Prior work such as [Spaan and Lima, 2009; Natarajan *et al.*, 2015] model active perception as a POMDP. However, the most relevant work to ours is that of [Araya *et al.*, 2010; Spaan *et al.*, 2015; Satsangi *et al.*, 2018]. Araya *et al.* [2010] proposed ρPOMDP framework where the reward depends on the entropy of the belief. Spaan *et al.* [2015] introduced POMDP-IR where the reward depends on an accurate prediction about the state. Satsangi *et al.* [2018] established an equivalence property between ρPOMDP and POMDP-IR. Furthermore, they employed the submodularity of the underlying value function, under some conditions, to use greedy scheme for sensor selection. The main difference of our work is that we consider active perception as a means to accomplishing the original task while in these work, active perception is the task itself and hence the POMDP rewards are metrics to capture perception quality.

2 Problem Formulation

This section starts by giving an overview of the related concepts and then stating the problem formulation.

[1] See [Ghasemi and Topcu, 2019] for the extended version.

2.1 Preliminaries

We introduce a new class of POMDP models, called AP^2-POMDP, that are suitable for problems with both elements of active perception and planning.

POMDP with Perception Action

We formally define an AP^2-POMDP below.

Definition 1. *An AP^2-POMDP is a tuple $\mathcal{P} = (S, A, k, T, \Omega, O, R, \gamma)$. S is the finite set of states. $A = A^{pl} \times A^{pr}$ denotes the finite set of paired actions with A^{pl} being the set of planning actions and A^{pr} being the set of perception actions. $A^{pr} = \{\delta \in \{0,1\}^n : \|\delta\|_0 \leq k\}$ constructs an n-dimensional lattice where k is the maximum number of information sources to be activated. Each component of an action $\delta \in A^{pr}$ determines whether to activate the corresponding information source, e.g. sensor. $T : S \times A^{pl} \times S \to [0,1]$ denotes the probabilistic transition function. $\Omega = \Omega^1 \times \Omega^2 \times \ldots \times \Omega^n$ is the partitioned set of observations, where each Ω_i corresponds to the set of measurements observable by information source i. $O : S \times A \times \Omega \to [0,1]$ denotes the probabilistic observation function. $R : S \times A^{pl} \to \mathbb{R}$ is the reward function, and $\gamma \in [0,1]$ is the discount factor.*

At each time step, the environment is in a state $s \in S$. The agent takes an action $\beta \in A^{pl}$ that causes a transition to a state $s' \in S$ with probability $Pr(s'|s,\beta) = T(s,\beta,s')$. At the same time step, the agent also picks k information sources by $\delta \in A^{pr}$. Then it receives an observation $\omega \in \Omega$ with probability $Pr(\omega|s',\beta,\delta) = O(s',\beta,\delta,\omega)$, and a scalar reward $R(s,\beta)$. Note that conventional POMDP models are a special case of AP^2-POMDP where $k = n = 1$.

Remark 1. *The constrained active perception defined above, i.e., selecting a subset of available information sources under cardinality constraint, arises in settings that the cost of sensing actions are uniform, e.g., for homogeneous sensors. It is possible to consider variations of constraints for different settings, leading to slightly modified solutions. For example, a more general case with nonuniform cost falls into the category of subset selection with linear cost constraints [Qian* et al.*, 2017].*

In many practical settings, the measurements from information sources only depend on the state and the previous action, as formally stated below.

Assumption 1. *We assume that given the current state and the previous action, the observations from information sources are mutually independent, i.e., $\forall I_1, I_2 \subseteq \{1,2,\ldots,n\}$, $I_1 \cap I_2 = \emptyset$: $Pr(\bigcup_{i_1 \in I_1} \omega^{i_1}, \bigcup_{i_2 \in I_2} \omega^{i_2} | \mathbf{s}, \beta) = Pr(\bigcup_{i_1 \in I_1} \omega^{i_1} | \mathbf{s}, \beta) Pr(\bigcup_{i_2 \in I_2} \omega^{i_2} | \mathbf{s}, \beta)$.*

Let $\zeta(\delta) = \{i | \delta(i) = 1\}$ to denote the subset of information sources that are selected by δ. Assumption 1 yields:

$$Pr\left(\omega|s',\beta,\delta\right) = Pr\left(\bigcup_{i\in\zeta(\delta)} \omega^i | s',\beta,\delta\right) = \prod_{i\in\zeta(\delta)} O_i(s',\beta,\omega^i), \tag{1}$$

where $Pr(\omega^i|s',\beta) = O_i(s',\beta,\omega^i)$.

The belief of the agent at each time step, denoted by b_t is the posterior probability distribution of state given

the history of previous actions and observations, i.e., $h_t = (a_0, \omega_1, a_1, \ldots, a_{t-1}, \omega_t)$. A well-known fact is that due to Markovian property, a sufficient statistics to represent history of actions and observations is belief [Åström, 1965; Smallwood and Sondik, 1973]. Given the initial belief b_0, the following update equation holds between previous belief b and the belief $b_b^{'a,\omega}$ after taking action $a = (\beta, \delta)$ and receiving observation ω:

$$b_b^{'a,\omega}(s') = \frac{Pr(\omega|s', \beta, \delta) \sum_s Pr(s'|s, \beta) b(s)}{Pr(\omega|\beta, \delta)} = \frac{\prod_{i \in \zeta(\delta)} O_i(s', \beta, \omega^i) \sum_s T(s, \beta, s') b(s)}{\sum_{s'} \prod_{i \in \zeta(\delta)} O_i(s', \beta, \omega^i) \sum_s T(s, \beta, s') b(s)}. \tag{2}$$

The goal is to learn a pure policy to maximize $\mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R(s_t, \beta_t)|b_0]$. A pure policy is a mapping from beliefs to actions $\pi : B \to A$, where B is the set of beliefs that constructs a $(|S| - 1)$-dimensional probability simplex.

The POMDP solvers apply value iteration [Sondik, 1978], a dynamic programming technique, to find an optimal policy. Let V be a value function that maps beliefs to values in $\mathbb{R}$. The following recursive expression holds for V:

$$V_t(b) = \max_a \left(\sum_{s \in S} b(s) R(s, a) + \gamma \sum_{\omega \in \Omega} Pr(\omega|b, a) V_{t-1}(b_b^{'a,\omega}) \right). \tag{3}$$

The value iteration converges to the optimal value function V^* which satisfies the Bellman's optimality equation [Bellman, 1957]. Once the optimal value function is learned, an optimal policy can be derived. An important outcome of (3) is that at any horizon, the value function is piecewise-linear and convex [Smallwood and Sondik, 1973] and hence, can be represented by a finite set of hyperplanes. Each hyperplane is associated with an action. Let α's to denote the corresponding vectors of the hyperplanes and let Γ_t to be the set of α vectors at horizon t. Then,

$$V_t(b) = \max_{\alpha \in \Gamma_t} \alpha \cdot b. \tag{4}$$

This fact has motivated approximate point based solvers that try to approximate the value function by updating the hyperplanes over a finite set of belief points.

Submodularity

Since the theoretical guarantee of the proposed algorithm is founded upon the theoretical results from the field of submodular optimization, here, we overview the necessary definitions. Let $\mathcal{X}$ to denote a ground set and f a set function that maps an input set to a real number.

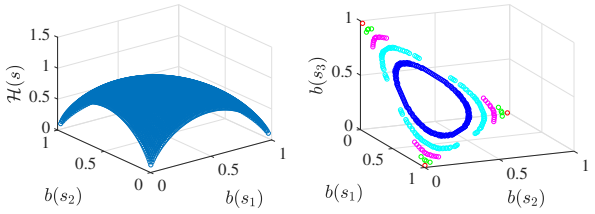
Definition 2. *Set function $f : 2^{\mathcal{X}} \to \mathbb{R}$ is monotone nondecreasing if $f(T_1) \leq f(T_2)$ for all $T_1 \subseteq T_2 \subseteq \mathcal{X}$.*

Definition 3. *Set function $f : 2^{\mathcal{X}} \to \mathbb{R}$ is submodular if*

$$f(T_1 \cup \{i\}) - f(T_1) \geq f(T_2 \cup \{i\}) - f(T_2)$$

for all subsets $T_1 \subseteq T_2 \subset \mathcal{X}$ and $i \in \mathcal{X} \backslash T_2$. The term $f_i(T) = f(T \cup \{i\}) - f(T)$ is the marginal value of adding element i to set T.

Monotonicity states that adding elements to a set increases the function value while submodularity refers to diminishing returns property.



(a) Entropy (b) Level sets of entropy

Figure 1: Entropy of belief for a 3-state POMDP.

2.2 Problem Definition

Having stated the required background, next, we formulate the joint perception and planning problem.

Problem 1. *Let $\mathcal{P} = (S, A, k, T, \Omega, O, R, \gamma)$ to denote an AP^2-POMDP and b_0 to be an initial belief. The goal is to learn a pure belief-based policy $\pi(b) = (\beta, \delta)$ such that the expected discounted cumulative reward is maximized, i.e,*

$$\pi^* = \underset{\pi}{\operatorname{argmax}} \ \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R(s_t, \pi(b_t))|b_0]. \tag{5}$$

3 Active Perception with Greedy Scheme

For variety of performance metrics, finding an optimal subset of information sources poses a computationally challenging combinatorial optimization problem that is NP-hard [Williamson and Shmoys, 2011]. Augmenting POMDP planning actions with $\binom{n}{k}$ active perception actions results in a combinatorial expansion of the action space. Thereupon, it is infeasible to directly apply existing POMDP solvers to Problem 1. Instead of concatenating both sets of actions and treating them similarly, we propose a greedy strategy for selecting perception actions that aims to pick the information sources that result in minimal uncertainty about the state. The key enabling factor is that the perception actions does not affect the transition, consequently, we can decompose the single-step belief update in (2) into two steps:

$$\tilde{b}_b^{\beta}(s') = \sum^{s} T(s, \beta, s') b(s), \tag{6a}$$

$$b_{\tilde{b}}^{'\delta,\omega}(s'') = \frac{\prod_{i \in \zeta(\delta)} O_i(s'', \beta, \omega^i) \tilde{b}(s'')}{\sum_{s'} \prod_{i \in \zeta(\delta)} O_i(s', \beta, \omega^i) \tilde{b}(s')}. \tag{6b}$$

This in turn implies that after a transition is made, the agent should pick a subset of observations that lead to minimal uncertainty in $b_{\tilde{b}}^{'\delta,\omega}$.

To quantify state uncertainty, we use Shannon entropy of the belief. For a discrete random variable $\boldsymbol{x}$, the entropy is defined as $\mathcal{H}(\boldsymbol{x}) = -\sum_i p(x_i) \log p(x_i)$. An important property of entropy is its strict concavity over the simplex of belief points, denoted by Δ_B [Cover and Thomas, 2012]. Further, the entropy is zero at the vertices of Δ_B and achieves its maximum, $\log |S|$, at the center of Δ_B that corresponds to uniform distribution, i.e., when the uncertainty about the state is the highest. Figure 1 demonstrates the entropy and its level sets for $|S| = 3$. Since the observation values are unknown before selecting the sensors, we optimize conditional

Algorithm 1 Greedy policy for perception action

1: **Input:** AP^2-POMDP $\mathcal{P} = (S, A, k, T, \Omega, O, R, \gamma)$, Current belief b, Planning action β.
2: **Output:** Perception action δ.
3: Initialize $\mathcal{X} = \{1, 2, \ldots, n\}$, $\zeta = \emptyset$.
4: **for** $l = 1, \ldots, k$ **do**
5: $\quad j^* = \text{argmax}_{j \in \mathcal{X} \setminus \zeta} -\mathcal{H}(\mathbf{s}|\tilde{b}_b^\beta, \bigcup_{i \in \zeta \cup \{j\}} \boldsymbol{\omega}^i)$
6: $\quad \zeta \leftarrow \zeta \cup \{j^*\}$
7: **end for**
8: **return** δ corresponding to ζ.

entropy that yields the expected value of entropy. For discrete random variables $\boldsymbol{x}$ and $\boldsymbol{y}$, conditional entropy is defined as $\mathcal{H}(\boldsymbol{x}|\boldsymbol{y}) = \mathbb{E}_{\boldsymbol{y}}[\mathcal{H}(\boldsymbol{x}|y)] = \sum_i p(y_i)\mathcal{H}(\boldsymbol{x}|y_i)$. Subsequently, with some algebraic manipulation, one obtains the conditional entropy of state given current belief with respect to δ as:

$$\mathcal{H}(\mathbf{s}|b, \boldsymbol{\omega}) = - \sum_{\omega^{i_1} \in \Omega^{i_1}} \cdots \sum_{\omega^{i_k} \in \Omega^{i_k}} \sum_{s \in S} \left(b(s) \prod_{i_j \in \zeta(\delta)} O_{i_j}(s, \beta, \omega^{i_j}) \log \left(\frac{b(s) \prod_{i_j \in \zeta(\delta)} O_{i_j}(s, \beta, \omega^{i_j})}{\sum_{s' \in S} b(s') \prod_{i_j \in \zeta(\delta)} O_{i_j}(s', \beta, \omega^{i_j})} \right) \right), \tag{7}$$

where $\zeta(\delta) = \{i_1, i_2, \ldots, i_k\}$. It is worth mentioning that b is the current distribution of $\mathbf{s}$ and is explicitly written only for the purpose of clarity, otherwise, $\mathcal{H}(\mathbf{s}|b, \delta) = \mathcal{H}(\mathbf{s}|\delta)$.

To minimize entropy, we define the objective function as the following set function:

$$f(\zeta) = \mathcal{H}(\mathbf{s}|\tilde{b}_b^\beta) - \mathcal{H}(\mathbf{s}|\tilde{b}_b^\beta, \bigcup_{i \in \zeta} \boldsymbol{\omega}^i) \tag{8}$$

and the optimization problem as:

$$\delta^* = \arg\max_{\delta \in A^{pr}} f(\zeta(\delta)). \tag{9}$$

We propose a greedy algorithm, outlined in Algorithm 1 to find a near-optimal, yet efficient solution to (9). The algorithm takes as input the agent's belief and planning action. Then it iteratively adds elements from the ground set (set of all information sources) whose marginal gain with respect to f is maximal and terminates after k selection.

Next, we derive a theoretical guarantee for the performance of the proposed greedy algorithm. The following lemma states the required properties to prove the theorem. The proof of the lemma follows from monotonicity and submodularity of conditional entropy [Ko *et al.*, 1995].

Lemma 1. *Let $\Omega = \{\boldsymbol{\omega}^1, \boldsymbol{\omega}^2, \ldots, \boldsymbol{\omega}^n\}$ to represent a set of observations of the state $\mathbf{s}$ such that Assumption 1 holds. Then, $f(\zeta)$, defined in* (8)*, is normalized, monotone nondecreasing, and submodular.*

The above lemma enables us to establish the approximation factor using the classical analysis in [Nemhauser *et al.*, 1978].

Theorem 1. *Let ζ^* to denote the optimal subset of observations with regard to objective function $f(\zeta)$, and ζ^g to denote the output of the greedy algorithm in Algorithm 1. Then, the following performance guarantee holds:*

$$\mathcal{H}(\mathbf{s}|\tilde{b}_b^\beta, \bigcup_{i \in \zeta^g} \boldsymbol{\omega}^i) \leq \frac{1}{e}\mathcal{H}(\mathbf{s}|\tilde{b}_b^\beta) + \left(1 - \frac{1}{e}\right) \mathcal{H}(\mathbf{s}|\tilde{b}_b^\beta, \bigcup_{i \in \zeta^*} \boldsymbol{\omega}^i). \tag{10}$$

Remark 2. *One can interpret the minimization of conditional entropy as pushing the agent's belief toward the boundary of the probability simplex Δ_B. This implies that the belief is moving toward regions of belief space that have higher value.*

Although Theorem 1 proves that the entropy of the belief point achieved by the greedy algorithm is close to the entropy of the belief point from the optimal solution, the key question is whether the value of these points are close. We assess this question in the following and show that at each time step, in expectation, the value from greedy scheme is close to the value from optimal selection with regard to (9). To that end, we first show that the distance between the two belief points is upper-bounded. Thereafter, we prove that the difference between value function at these two points is upper-bounded.

Theorem 2. *Let the agent's current belief to be b and its planning action to be β. Consider the optimization problem in* (9)*, and let δ^* and δ^g to denote the optimal perception action and the perception action obtained by the greedy algorithm, respectively. It holds that:*

$$\mathbb{E}_{\bigcup_{i \in [n]} \boldsymbol{\omega}^i}[\|b^g - b^*\|_1] \leq \sqrt{\frac{2}{e} \mathbb{E}_{\bigcup_{i \in \delta^*} \boldsymbol{\omega}^i}[D_{\mathcal{KL}}(b^* \| b)]},$$

where b^* and b^g are the updated beliefs according to (6)*.*

Theorem 3. *Instate the notation and hypothesis of Theorem 2. Additionally, let V to be the true value function for AP^2-POMDP. The following statement holds:* [2]

$$\mathbb{E}_{\bigcup_{i \in [n]} \boldsymbol{\omega}^i}[V(b^g) - V(b^*)] \leq \sqrt{\frac{2}{e} \mathbb{E}_{\bigcup_{i \in \delta^*} \boldsymbol{\omega}^i}[D_{\mathcal{KL}}(b^* \| b)]} \frac{\max\{|R_{max}|, |R_{min}|\}}{1 - \gamma}.$$

4 Perception-Aware Point-Based Value Iteration

In this section, we propose a novel point-based value iteration algorithm to approximate the value function for AP^2-POMDPs. The algorithm relies on the performance guarantee of the proposed greedy observation selection in previous section. Algorithm 2 outlines the general procedure for a point-based solver. It starts with an initial set of belief points B_0 and their corresponding α vectors. Then it performs a Bellman backup for each point to update α vectors. Next, it prunes α vectors to remove dominated ones. Afterwards, it samples a new set of belief points and repeats these steps until convergence or other termination criteria is met. The difference between solvers is in how they apply sampling and pruning. The sampling step usually depends on the reachability tree of belief space, see Figure 2. The state-of-the-art point-based methods do not traverse the whole reachability tree, but they try to have enough sample points to provide a good coverage of the reachable space.

[2] See [Ghasemi and Topcu, 2019] for the proofs.

Algorithm 2 Generic algorithm for point-based solvers [Araya *et al.*, 2010]

1: **Input:** POMDP.
2: **Output:** Approximate value function V.
3: Initialize $B = B_0$ and Γ_0.
4: **while** $\sim$ (termination condition) **do**
5: $B \leftarrow \text{Sample}(B)$
6: $\Gamma \leftarrow \text{BackUp}(B, \Gamma)$
7: $\Gamma \leftarrow \text{Prune}(B, \Gamma)$
8: **end while**
9: **return** $V(b) = \max_{\alpha \in \Gamma} \alpha \cdot b$.

Note that the combinatorial number of actions due to perception decisions highly expand the size of the reachability tree. However, since the perception decisions aim to reduce the state uncertainty, we apply the greedy scheme for entropy minimization to make the choice of δ deterministically dependent on β and previous belief. To that end, we modify the BackUp step of point-based value iteration. The proposed BackUp step can be combined with any sampling and pruning method in other solvers, such as the ones developed by Spaan and Vlassis [2005], Kurniawati *et al.* [2008], and Smith and Simmons [2012].

4.1 Proposed Point-Based Solver

In point-based solvers each witness belief point is associated with an α vector and an action. Nevertheless, for AP^2-POMDPs, each witness point is associated with two actions, β and δ. We compute δ based on greedy maximization of (9) so that given b and β, δ is uniquely determined. Henceforth, we can rewrite (3) using (4) to obtain:

$$
\begin{aligned}
V_t(b) =& \max_{(\beta,\delta)} \Bigg(\sum_{s\in S} b(s) R(s,\beta) + \\
& \gamma \sum_{\omega \in \Omega} Pr(\omega|b,\beta,\delta) \max_{\alpha \in \Gamma_{t-1}} \alpha \cdot b_b^{'\beta,\delta,\omega} \Bigg) \\
=& \max_{\beta} \Bigg(\sum_{s\in S} b(s) R(s,\beta) + \\
& \gamma \sum_{\substack{\omega \in \Omega_{i_1} \times \ldots \times \Omega_{i_k} \\ i_j \in \zeta(\bar{\delta})}} \max_{\alpha \in \Gamma_{t-1}} \sum_{s' \in S} \alpha(s') \times \\
& \prod_{i_j \in \zeta(\bar{\delta})} O_i(s', \beta, \omega^{i_j}) \sum_{s\in S} T(s,\beta,s') b(s) \Bigg) \\
=& \max_{\beta} \Bigg(\sum_{s\in S} b(s) R(s,\beta) + \\
& \gamma \sum_{\substack{\omega \in \Omega_{i_1} \times \ldots \times \Omega_{i_k} \\ i_j \in \zeta(\bar{\delta})}} \max_{\alpha \in \Gamma_{t-1}} \sum_{s\in S} \sum_{s' \in S} \alpha(s') \times \\
& \prod_{i_j \in \zeta(\bar{\delta})} O_i(s', \beta, \omega^{i_j}) T(s,\beta,s') b(s) \Bigg).
\end{aligned}
\tag{11}
$$

where $\bar{\delta} = \text{argmax}_{\delta \in A^{pr}} f(\zeta(\delta))$ and f is computed at $\tilde{b}_b^{\beta}$.

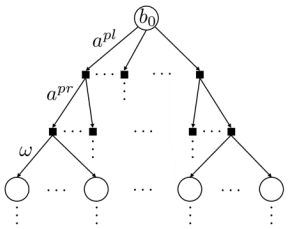
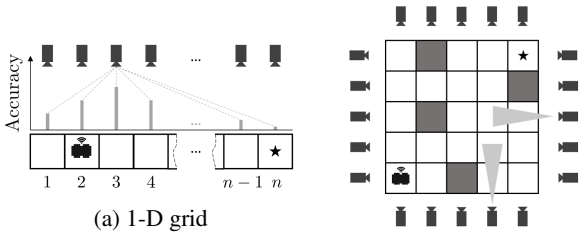


Figure 2: The belief reachability tree. The circles are belief points while squares depict branchings based on actions. Addition of perception actions leads to combinatorial expansion of number of belief points in each layer.

Based on the derivation in (11), we develop the BackUp step detailed in Algorithm 3 to compute the new set of α vectors from the previous ones using Bellman backup operation. What distinguishes this algorithm from conventional Bellman backup step is the inclusion of perception actions. Basically, we need to compute the greedy perception action for each belief point and each action (Line 7). This in turn affects computation of $\Gamma_t^{b,\beta,\omega}$ as it represents a different set for each belief point (Lines 9-13). However, notice that this added complexity is significantly lower than concatenating the combinatorial perception actions with the planning actions and using conventional point-based solvers. See [Ghasemi and Topcu, 2019] for detailed complexity analysis.

Algorithm 3 BackUp step for AP^2-POMDP

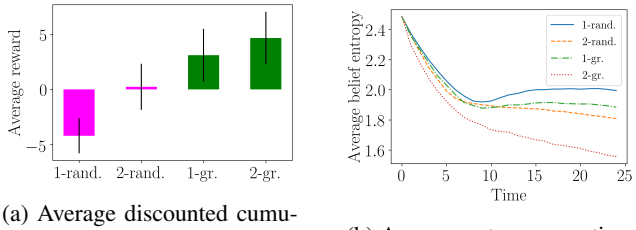
1: **Input:** AP^2-POMDP $\mathcal{P} = (S, A, k, T, \Omega, O, R, \gamma)$, Current set of belief points B_t, Current set of α vectors Γ_{t-1}.
2: **Output:** Next set of α vectors Γ_t.
3: Initialize $\Gamma_t = \emptyset$, $\Gamma_t^{b,\beta} = \emptyset$ for all $b \in B_t$ and $\beta \in A^{pl}$.
4: **for** $\beta \in A^{pl}$ **do**
5: $\Gamma_t^{\beta,*} \leftarrow \alpha^{\beta,*}(s) = R(s,\beta)$
6: **for** $b \in B_t$ **do**
7: $\bar{\delta} = \text{Greedy\_argmax}_{\delta \in A^{pr}} f(\zeta(\delta))$
8: $\Gamma_t^{b,\beta,\omega} = \emptyset$
9: **for** $\omega \in \Omega_{i_1} \times \ldots \times \Omega_{i_k}, i_j \in \zeta(\bar{\delta})$ **do**
10: **for** $\alpha \in \Gamma_{t-1}$ **do**
11: $\alpha^{b,\beta,\omega}(s) = \gamma \sum_{s' \in S} \prod_{i_j \in \zeta(\bar{\delta})} O_i(s', \beta, \omega^{i_j}) T(s,\beta,s') \alpha(s')$
12: $\Gamma_t^{b,\beta,\omega} \leftarrow \Gamma_t^{b,\beta,\omega} \cup \alpha^{b,\beta,\omega}$
13: **end for**
14: **end for**
15: $\alpha^{b,\beta} = \alpha^{\beta,*} + \sum_{\substack{\omega \in \Omega_{i_1} \times \ldots \times \Omega_{i_k} \\ i_j \in \zeta(\bar{\delta})}} \text{argmax}_{\alpha \in \Gamma_t^{b,\beta,\omega}} \alpha \cdot b$
16: $\Gamma_t^{b,\beta} \leftarrow \Gamma_t^{b,\beta} \cup \alpha^{b,\beta}$
17: **end for**
18: **end for**
19: **for** $b \in B_t$ **do**
20: $\alpha^b = \text{argmax}_{\alpha \in \Gamma_t^{b,\beta}, \beta \in A^{pl}} \alpha \cdot b$
21: $\Gamma_t = \Gamma_t \cup \alpha^b$
22: **end for**
23: **return** Γ_t.



(a) 1-D grid

(b) 2-D grid

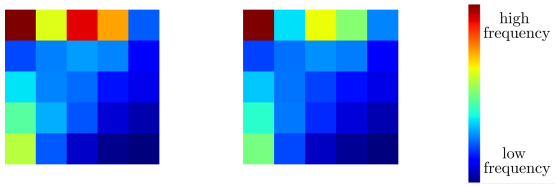
Figure 3: The robot moves in a grid while communicating with the cameras to localize itself. The accuracy of cameras' measurements depends on their distance from a state. The robot's objective is to reach the goal state, labeled by star, while avoiding the obstacles.



(a) Average discounted cumulative reward

(b) Average entropy over time

Figure 4: Results of 1-D simulation for a map of size 12. Left: The average discounted cumulative reward along its standard deviation. Right: The average belief entropy over a time horizon of 25 steps.



(a) Random

(b) Greedy

Figure 5: The frequency of visiting states when using different perception methods for a 2-D map of size 5*5 according to Figure 3-(b).

5 Simulation Results

To evaluate the proposed algorithm for active perception and planning, we implement the point-based value iteration solver for AP^2-POMDPs. We initialize the belief set by uniform sampling from Δ_B [Devroye, 1986]. To focus on the effect of perception, we keep the belief set fixed throughout the iterations. However, one can incorporate any sampling method such as the ones proposed by Kurniawati *et al.* [2008], and Smith and Simmons [2012]. The α vectors are initialized by $\frac{1}{1-\gamma}\min_{s,a} R(s,a).Ones(|S|)$ [Shani *et al.*, 2013]. Furthermore, to speedup the solver, one can employ a randomized backup step, as suggested by Spaan and Vlassis [2005]. The solver terminates once the difference between value functions in two consecutive iterations falls below a predefined threshold. We also implemented a random perception policy that selects a subset of information sources, uniformly at random, at each backup step.

5.1 Robotic Navigation in 1-D Grid

The first scenario is similar to that of [Satsangi *et al.*, 2018] and models a robot that is moving in a 1-D discrete environment (Figure 3-(a)). The robot can move to adjacent cells by its navigation actions $A^{pl} = \{left, right, stop\}$. The robot's transitions are probabilistic due to possible actuation errors. The robot does not have any sensor and it relies on a set of cameras for localization. There is one camera at each cell that outputs a probability distribution over the position of the robot. To model the effect of robot's position on the accuracy of cameras' measurements, we use a binomial distribution with its mean at the cell that camera is on. The robot's objective is to reach n specific cell in the map. For that purpose, at each time step, the robot picks a navigation action and selects k cameras from the set of n cameras.

We evaluate the computed policy by running 1000 Monte Carlo simulations. The robot starts at the origin and its initial belief is uniform. Figure 4-(a) demonstrates the discounted cumulative reward, averaged over 1000 runs, for random selection of 1 and 2 cameras, and greedy selection of 1 and 2 cameras. It shows that the greedy perception policy significantly outperforms the random perception. Figure 4-(b) depicts the belief entropy over the time. The lower entropy of greedy perception, compared to random perception, shows less uncertainty of the robot when taking planning actions.

5.2 Robotic Navigation in 2-D Grid

The second setting is a 2-D variant of the first scenario (Figure 3-(b)). The navigation actions of the robot are $A^{pl} = \{up, right, down, left, stop\}$. The rest of the setting is similar to 1-D case, except the cameras' positions, as they are now placed around the perimeter of the map. Also, now the robot must avoid the obstacles in the map. The reward is 10 at the goal state, -4 at the obstacles, and -1 in other states.

We applied the proposed solver with both random perception and greedy perception on the 2-D example. Next, we let the robot to run for a horizon of 25 steps and terminated the simulations once the robot reached the goal. Figure 5 illustrates the normalized frequency of visiting each state for each perception algorithm. It can be seen that the policy learned by greedy active perception leads to better obstacle avoidance. See [Ghasemi and Topcu, 2019] for further results.

6 Conclusion

We introduced AP^2-POMDPs as a modeling framework for joint active perception and planning in POMDPs. To tackle the computational challenge of adding perception actions, we proposed a greedy scheme for observation selection that aims to minimize the state uncertainty. Founded upon the theoretical guarantee of greedy active perception, we developed and empirically evaluated a point-based value iteration solver for AP^2-POMDPs. The idea introduced in the solver to address active perception is general and can be applied to state-of-the-art point-based solvers.

Acknowledgments

This work was supported in part by DARPA grant D19AP00004 and ONR grants N00014-18-1-2829 and N00014-19-1-2054.

References

[Araya *et al.*, 2010] Mauricio Araya, Olivier Buffet, Vincent Thomas, and Françcois Charpillet. A POMDP extension with belief-dependent rewards. In *Advances in Neural Information Processing Systems*, pages 64–72, 2010.

[Åström, 1965] Karl J Åström. Optimal control of Markov processes with incomplete state information. *J. of Mathematical Analysis and Applications*, 10(1):174–205, 1965.

[Bellman, 1957] Richard Bellman. A Markovian decision process. *J. of Mathematics and Mechanics*, pages 679–684, 1957.

[Cheng, 1988] Hsien-Te Cheng. *Algorithms for partially observable Markov decision processes*. PhD thesis, University of British Columbia, 1988.

[Cover and Thomas, 2012] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[Devroye, 1986] Luc Devroye. Sample-based non-uniform random variate generation. In *Conf. on Winter Simulation*, pages 260–265. ACM, 1986.

[Ghasemi and Topcu, 2019] Mahsa Ghasemi and Ufuk Topcu. Perception-aware point-based value iteration for partially observable Markov decision processes. *arXiv preprint*, 2019.

[Hashemi *et al.*, 2018] Abolfazl Hashemi, Mahsa Ghasemi, Haris Vikalo, and Ufuk Topcu. A randomized greedy algorithm for near-optimal sensor scheduling in large-scale sensor networks. In *American Control Conf.*, pages 1027–1032. IEEE, 2018.

[Ko *et al.*, 1995] Chun-Wa Ko, Jon Lee, and Maurice Queyranne. An exact algorithm for maximum entropy sampling. *Operations Research*, 43(4):684–691, 1995.

[Krause and Golovin, 2014] Andreas Krause and Daniel Golovin. Submodular function maximization. In *Tractability: Practical Approaches to Hard Problems*, pages 71–104. Cambridge University Press, 2014.

[Kreucher *et al.*, 2005] Chris Kreucher, Keith Kastella, and Alfred O Hero Iii. Sensor management using an active sensing approach. *Signal Process.*, 85(3):607–624, 2005.

[Kurniawati *et al.*, 2008] 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*, 2008.

[Lovejoy, 1991] William S Lovejoy. A survey of algorithmic methods for partially observed Markov decision processes. *Annals of Operations Research*, 28(1):47–65, 1991.

[Natarajan *et al.*, 2015] Prabhu Natarajan, Pradeep K Atrey, and Mohan Kankanhalli. Multi-camera coordination and control in surveillance systems: A survey. *ACM Trans. on Multimedia Computing, Communications, and Applications*, 11(4):57, 2015.

[Nemhauser *et al.*, 1978] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, 14(1):265–294, 1978.

[Papadimitriou and Tsitsiklis, 1987] Christos H Papadimitriou and John N Tsitsiklis. The complexity of Markov decision processes. *Mathematics of Operations Research*, 12(3):441–450, 1987.

[Pineau *et al.*, 2006] Joelle Pineau, Geoffrey Gordon, and Sebastian Thrun. Anytime point-based approximations for large POMDPs. *J. of Artificial Intelligence Research*, 27:335–380, 2006.

[Qian *et al.*, 2017] Chao Qian, Jing-Cheng Shi, Yang Yu, and Ke Tang. On subset selection with general cost constraints. In *Proc. Int. Joint Conf. on Artificial Intelligence*, volume 17, pages 2613–2619, 2017.

[Satsangi *et al.*, 2018] Yash Satsangi, Shimon Whiteson, Frans A Oliehoek, and Matthijs TJ Spaan. Exploiting submodular value functions for scaling up active perception. *Autonomous Robots*, 42(2):209–233, 2018.

[Shamaiah *et al.*, 2010] Manohar Shamaiah, Siddhartha Banerjee, and Haris Vikalo. Greedy sensor selection: Leveraging submodularity. In *Proc. IEEE Conf. on Decision and Control*, pages 2572–2577. IEEE, 2010.

[Shani *et al.*, 2013] Guy Shani, Joelle Pineau, and Robert Kaplow. A survey of point-based POMDP solvers. *Autonomous Agents and Multi-Agent Systems*, 27(1):1–51, 2013.

[Smallwood and Sondik, 1973] Richard D Smallwood and Edward J Sondik. The optimal control of partially observable Markov processes over a finite horizon. *Operations Research*, 21(5):1071–1088, 1973.

[Smith and Simmons, 2012] Trey Smith and Reid Simmons. Point-based POMDP algorithms: Improved analysis and implementation. *arXiv preprint arXiv:1207.1412*, 2012.

[Sondik, 1978] Edward J Sondik. The optimal control of partially observable Markov processes over the infinite horizon: Discounted costs. *Operations Research*, 26(2):282–304, 1978.

[Spaan and Lima, 2009] Matthijs TJ Spaan and Pedro U Lima. A decision-theoretic approach to dynamic sensor selection in camera networks. In *Proc. Int. Conf. on Automated Planning and Scheduling*, pages 279—-304, 2009.

[Spaan and Vlassis, 2005] Matthijs TJ Spaan and Nikos Vlassis. Perseus: Randomized point-based value iteration for POMDPs. *J. of Artificial Intelligence Research*, 24:195–220, 2005.

[Spaan *et al.*, 2015] Matthijs TJ Spaan, Tiago S Veiga, and Pedro U Lima. Decision-theoretic planning under uncertainty with information rewards for active cooperative perception. *Autonomous Agents and Multi-Agent Systems*, 29(6):1157–1185, 2015.

[Williamson and Shmoys, 2011] David P Williamson and David B Shmoys. *The design of approximation algorithms*. Cambridge University Press, 2011.

[Zhang and Zhang, 2001] Nevin Lianwen Zhang and Weihong Zhang. Speeding up the convergence of value iteration in partially observable Markov decision processes. *J. of Artificial Intelligence Research*, 14:29–51, 2001.