Exploiting Submodular Value Functions for Scaling Up Active Perception

Satsangi, Y.; Whiteson, S.; Oliehoek, F.; Spaan, M.T.J.

DOI
10.1007/s10514-017-9666-5

Publication date
2018

Document Version
Final published version

Published in
Autonomous Robots

Auton Robot (2018) 42:209–233
https://doi.org/10.1007/s10514-017-9666-5

Exploiting submodular value functions for scaling up active perception

Yash Satsangi[1] · **Shimon Whiteson**[2] · **Frans A. Oliehoek**[1,3] · **Matthijs T. J. Spaan**[4]

Received: 29 February 2016 / Accepted: 7 July 2017 / Published online: 29 August 2017

Abstract In *active perception* tasks, an agent aims to select sensory actions that reduce its uncertainty about one or more hidden variables. For example, a mobile robot takes sensory actions to efficiently navigate in a new environment. While partially observable Markov decision processes (POMDPs) provide a natural model for such problems, *reward functions* that directly penalize uncertainty in the agent's belief can remove the piecewise-linear and convex (PWLC) property of the *value function* required by most POMDP planners. Furthermore, as the number of sensors available to the agent grows, the computational cost of POMDP planning grows exponentially with it, making POMDP planning infeasible with traditional methods. In this article, we address a twofold challenge of modeling and planning for active perception tasks. We analyze ρPOMDP and POMDP-IR, two frameworks for modeling active perception tasks, that restore the PWLC property of the value function. We show the mathematical equivalence of these two frameworks by showing that given a ρPOMDP along with a policy, they can be reduced to a POMDP-IR and an equivalent policy (and vice-versa). We prove that the value function for the given ρPOMDP (and the given policy) and the reduced POMDP-IR (and the reduced policy) is the same. To efficiently plan for active perception tasks, we identify and exploit the independence properties of POMDP-IR to reduce the computational cost of solving POMDP-IR (and ρPOMDP). We propose greedy point-based value iteration (PBVI), a new POMDP planning method that uses *greedy maximization* to greatly improve scalability in the action space of an active perception POMDP. Furthermore, we show that, under certain conditions, including *submodularity*, the value function computed using greedy PBVI is guaranteed to have bounded error with respect to the optimal value function. We establish the conditions under which the value function of an active perception POMDP is guaranteed to be *submodular*. Finally, we present a detailed empirical analysis on a dataset collected from a multi-camera tracking system employed in a shopping mall. Our method achieves similar performance to existing methods but at a fraction of the computational cost leading to better scalability for solving active perception tasks.

Keywords Sensor selection · Long-term planning · Mobile sensors · Submodularity · POMDP

1 Introduction

Multi-sensor systems are becoming increasingly prevalent in a wide-range of settings. For example, multi-camera systems are now routinely used for security, surveillance and tracking (Kreucher et al. 2005; Natarajan et al. 2012; Spaan et al. 2015). A key challenge in the design of these systems is the efficient allocation of scarce resources such as the bandwidth required to communicate the collected data to a central server, the CPU cycles required to process that data, and the energy costs of the entire system (Kreucher et al. 2005; Williams et al. 2007; Spaan and Lima 2009). For example, state of the art human activity recognition algorithms require high resolution video streams coupled with

This is one of several papers published in *Autonomous Robots* comprising the Special Issue on Active Perception.

✉ Yash Satsangi
y.satsangi@uva.nl; yashziz@gmail.com

[1] University of Amsterdam, Amsterdam, Netherlands

[2] University of Oxford, Oxford, England

[3] University of Liverpool, Liverpool, England

[4] Delft University of Technology, Delft, Netherlands

significant computational resources. When a human operator must monitor many camera streams, displaying only a small number of them can reduce the operator's cognitive load. IP-cameras connected directly to a local area network need to share the available bandwidth. Such constraints gives rise to the *dynamic sensor selection* problem where an agent at each time step must select K out of the N available sensors to allocate these resources to, where K is the maximum number of sensors allowed given the resource constraints (Satsangi et al. 2015).[1]

For example, consider the *surveillance task*, in which a mobile robot aims to minimize its future uncertainty about the state of the environment but can use only K of its N sensors at each time step. Surveillance is an example of an *active perception* task, where an agent takes actions to reduce uncertainty about one or more hidden variables, while reasoning about various resource constraints (Bajcsy 1988). When the state of the environment is static, a *myopic* approach that always selects actions that maximize the immediate expected reduction in uncertainty is typically sufficient. However, when the state changes over time, a *non-myopic* approach that reasons about the long term effects of action selection performed at each time step can be better. For example, in the surveillance task, as the robot moves and the state of the environment changes, it becomes essential to reason about the long term consequences of the robot's actions to minimize the future uncertainty.

A natural decision-theoretic model for such an approach is the *partially observable Markov decision process* (POMDP) (Sondik 1971; Kaelbling et al. 1998; Kochenderfer 2015). POMDPs provide a comprehensive and powerful framework for planning under uncertainty. They can model the dynamic and partially observable state and express the goals of the systems in terms of rewards associated with state-action pairs. This model of the world can be used to compute closed-loop, long term policies that can help the agent to decide what actions to take given a belief about the state of the environment (Burgard et al. 1997; Kurniawati et al. 2011).

In a typical POMDP reducing uncertainty about the state is only *a means to an end*. For example, a robot whose goal is to reach a particular location may take sensing actions that reduce its uncertainty about its current location because doing so helps it determine what future actions will bring it closer to its goal. By contrast, in active perception problems reducing uncertainty is *an end in itself*. For example, in the surveillance task, the system's goal is typically to ascertain the state of its environment, not use that knowledge to achieve a goal. While perception is arguably always performed to aid decision-making, in an active perception problem that decision is made by another agent such as a human, that is not modeled as a part of the POMDP. For example, in the surveillance task, the robot might be able to detect a suspicious activity but only the human users of the system may decide how to react to such an activity.

One way to formulate uncertainty reduction as an end in itself is to define a *reward function* whose additive inverse is some measure of the agent's uncertainty about the hidden state, e.g., the *entropy* of its *belief*. However this formulation leads to a reward function that conditions on the belief, rather than the state and the resulting *value function* is not PWLC, which makes many traditional POMDP solvers inapplicable. There exist *online planning methods* (Silver and Veness 2010; Bonet and Geffner 2009) that generate policies on the fly that do not require the PWLC property of the value function. However, many of these methods require multiple 'hypothetical' belief updates to compute the optimal policy, which makes them unsuitable for sensor selection where the optimal policy must be computed in a fraction of a second. There exist other online planning methods that do not require hypothetical belief updates (Silver and Veness 2010), but since we are dealing with belief based rewards, they cannot be directly applied here. Here, we address the case of *offline planning* where the policy is computed before the execution of the task.

Thus, to efficiently solve active perception problems, we must (a) model the problem with minimizing uncertainty as the objective while maintaining a PWLC value function and (b) use this model to solve the POMDP efficiently. Recently, two frameworks have been proposed, *ρPOMDP* (Araya-López et al. 2010) and *POMDP with Information Reward* (POMDP-IR) (Spaan et al. 2015) to efficiently model active perception tasks, such that the PWLC property of the value function is maintained. The idea behind ρPOMDP is to find a PWLC approximation to the "true" continuous belief-based reward function, and then solve it with the traditional solvers. POMDP-IR, on the other hand, allows the agent to make predictions about the hidden state and the agent is rewarded for accurate predictions via a state-based reward function. There is no research that examines the relationship between these two frameworks, their pros and cons, or their efficacy in realistic tasks, thus it is not clear how to choose between these two frameworks to model the active perception problems.

In this article, we address the problem of efficient modeling and planning for active perception tasks. First, we study the relationship between ρPOMDP and POMDP-IR. Specifically, we establish *equivalence* between them by showing that any ρPOMDP can be reduced to a POMDP-

[1] This article extends the research already presented by Satsangi et al. (2015) at AAAI 2015. In this article, we present additional theoretical results on equivalence of POMDP-IR and ρPOMDP, a new technique that exploits the independence properties of POMDP-IR to solve it more efficiently, and we present a detailed empirical analysis of belief-based rewards for POMDPs in active perception tasks.

IR (and vice-versa) that preserves the value function for equivalent policies. Having established the theoretical relationship between ρPOMDP and POMDP-IR, we model the surveillance task as a POMDP-IR and propose a new method to solve it efficiently by exploiting a simple insight that lets us decompose the maximization over prediction actions and normal actions while computing the value function.

Although POMDPs are computationally difficult to solve, recent methods (Littman 1996; Hauskrecht 2000; Pineau et al. 2006; Spaan and Vlassis 2005; Poupart 2005; Ji et al. 2007; Kurniawati et al. 2008; Shani et al. 2013) have proved successful in solving POMDPs with large state spaces. Solving active perception POMDPs pose a different challenge: as the number of sensors grows, the size of the action space $\binom{N}{K}$ grows exponentially with it. Current POMDP solvers fail to address the scalability in the action space of a POMDP. We propose a new *point-based* planning method that scales much better in the number of sensors for such POMDPs. The main idea is to replace the maximization operator in the Bellman optimality equation with *greedy maximization* in which a subset of sensors is constructed iteratively by adding the sensor that gives the largest marginal increase in value.

We present theoretical results bounding the error in the value functions computed by this method. We prove that, under certain conditions including *submodularity*, the value function computed using POMDP backups based on greedy maximization has bounded error. We achieve this by extending the existing results (Nemhauser et al. 1978) for the greedy algorithm, which are valid only for a single time step, to a full sequential decision making setting where the greedy operator is employed multiple times over multiple time steps. In addition, we show that the conditions required for such a guarantee to hold are met, or approximately met, if the reward is defined using negative belief entropy.

Finally, we present a detailed empirical analysis on a real-life dataset from a multi-camera tracking system installed in a shopping mall. We identify and study the critical factors relevant to the performance and behavior of the agent in active perception tasks. We show that our proposed planner outperforms a myopic baseline and nearly matches the performance of existing point-based methods while incurring only a fraction of the computational cost, leading to much better scalability in the number of cameras.

2 Related work

Sensor selection as an active perception task has been studied in many contexts. Most work focus on either open-loop or myopic solutions, e.g., Kreucher et al. (2005), Spaan and Lima (2009), Williams et al. (2007), Joshi and Boyd (2009). Kreucher et al. (2005) proposes a Monte-Carlo approach that mainly focuses on a myopic solution. Williams et al. (2007) and Joshi and Boyd (2009) developed planning methods that can provide long-term but open-loop policies. By contrast, a POMDP-based approach enables a closed-loop, non-myopic approach that can lead to a better performance when the underlying state of the world changes over time. Spaan (2008), Spaan and Lima (2009), Spaan et al. (2010) and Natarajan et al. (2012) also consider a POMDP-based approach to active perception and cooperative active perception. However, they consider an objective function that conditions on the state and not on the belief, as the belief-dependent rewards in POMDP break the PWLC property of the value function. They use point-based methods (Spaan and Vlassis 2005) for solving the POMDPs. While recent point-based methods (Shani et al. 2013) for solving POMDPs scale reasonably in the state space of the POMDPs, they do not address the scalability in the action and observation space of a POMDP.

In recent years, applying greedy maximization to submodular functions has become a popular and effective approach to sensor placement/selection (Krause and Guestrin 2005, 2007; Kumar and Zilberstein 2009; Satsangi et al. 2016). However, such work focuses on myopic or fully observable settings and thus does not enable the long-term planning required to cope with the dynamic state in a POMDP.

Adaptive submodularity (Golovin and Krause 2011) is a recently developed extension that addresses these limitations by allowing action selection to condition on previous observations. However, it assumes a static state and thus cannot model the dynamics of a POMDP across timesteps. Therefore, in a POMDP, adaptive submodularity is only applicable *within* a timestep, during which state does not change but the agent can sequentially add sensors to a set. In principle, adaptive submodularity could enable this intra-timestep sequential process to be adaptive, i.e., the choice of later sensors could condition on the observations generated by earlier sensors. However, this is not possible in our setting because (a) we assume that, due to computational costs, all sensors must be selected simultaneously; (b) information gain is not known to be adaptive submodular (Chen et al. 2015). Consequently, our analysis considers only classic, non-adaptive submodularity.

To our knowledge, our work is the first to establish the sufficient conditions for the submodularity of POMDP value functions for active perception POMDPs and thus leverage greedy maximization to scalably compute bounded approximate policies for dynamic sensor selection modeled as a full POMDP.

3 Background

In this section, we provide background on POMDPs, active perception POMDPs and solution methods for POMDPs.

3.1 Partially observable Markov decision processes

POMDPs provide a decision-theoretic framework for modeling partial observability and dynamic environments. Formally, a POMDP is defined by a tuple $\langle S, A, \Omega, T, O, R, b_0, h\rangle$. At each time step, the environment is in a state $s \in S$, the agent takes an action $a \in A$ and receives a reward whose expected value is $R(s,a)$, and the system transitions to a new state $s' \in S$ according to the transition function $T(s,a,s') = \Pr(s'|s,a)$. Then, the agent receives an observation $z \in \Omega$ according to the observation function $O(s',a,z) = \Pr(z|s',a)$. Starting from an initial belief b_0, the agent maintains a *belief* $b(s)$ about the state which is a probability distribution over all the possible states. The number of time steps for which the decision process lasts, i.e., the horizon is denoted by h. If the agent takes an action a in belief b and gets an observation z, then then the updated belief $b^{a,z}(s)$ can be computed using Bayes rule. A policy π specifies how the agent acts in each belief. Given $b(s)$ and $R(s,a)$, one can compute a belief-based reward, $\rho(b,a)$ as:

$$\rho(b,a) = \sum_s b(s)R(s,a). \tag{1}$$

The t-step *value function* of a policy V_t^π is defined as the expected future discounted reward the agent can gather by following π for next t steps. V_t^π can be characterized recursively using the *Bellman equation*:

$$V_t^\pi(b) \triangleq \left[\rho(b,a_\pi) + \sum_{z\in\Omega} \Pr(z|a_\pi,b) V_{t-1}^\pi(b^{a_\pi,z})\right], \tag{2}$$

where $a_\pi = \pi(b)$ and $V_0^\pi(b) = 0$. The action-value function $Q_t^\pi(b,a)$ is the value of taking action a and following π thereafter:

$$Q_t^\pi(b,a) \triangleq \rho(b,a) + \sum_{z\in\Omega} \Pr(z|a,b) V_{t-1}^\pi(b^{a,z}). \tag{3}$$

The policy that maximizes V_t^π is called the *optimal policy* π^* and the corresponding value function is called the *optimal value function* V_t^*. The *optimal value function* $V_t^*(b)$ can be characterized recursively as:

$$V_t^*(b) = \max_a \left[\rho(b,a) + \sum_{z\in\Omega} \Pr(z|a,b) V_{t-1}^*(b^{a,z})\right]. \tag{4}$$

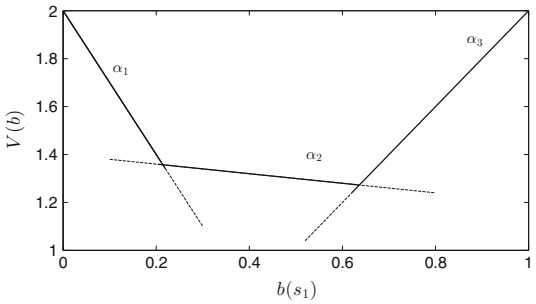

Fig. 1 Illustration of the PWLC property of the value function. The value function is the upper surface indicated by the *solid lines*

We can also define *Bellman optimality operator* $\mathfrak{B}^*$:

$$(\mathfrak{B}^* V_{t-1})(b) = \max_a \left[\rho(b,a) + \sum_{z\in\Omega} \Pr(z|a,b) V_{t-1}(b^{z,a})\right],$$

and write (4) as $V_t^*(b) = (\mathfrak{B}^* V_{t-1}^*)(b)$.

An important consequence of these equations is that the value function is *piecewise-linear and convex* (PWLC), as shown in Fig. 1, a property exploited by most POMDP planners. Sondik (1971) showed that a PWLC value function at any finite time step t can be expressed as a set of vectors: $\Gamma_t = \{\alpha_0, \alpha_1, \ldots, \alpha_m\}$. Each α_i represents an $|S|$-dimensional hyperplane defining the value function over a bounded region of the belief space. The value of a given belief point can be computed from the vectors as: $V_t^*(b) = \max_{\alpha_i\in\Gamma_t} \sum_s b(s)\alpha_i(s)$.

3.2 POMDP solvers

Exact methods like Monahan's enumeration algorithm (Monahan 1982) computes the value function for all possible belief points by computing the optimal $\mathbf{\Gamma}_t$. Point-based planners (Pineau et al. 2006; Shani et al. 2013; Spaan and Vlassis 2005), on the other hand, avoid the expense of solving for all belief points by computing Γ_t only for a set of sampled beliefs B. Since the exact POMDP solvers (Sondik 1971; Monahan 1982) are intractable for all but the smallest POMDPs, we focus on point-based methods here. Point-based methods compute Γ_t using the following recursive algorithm.

At each iteration (starting from t = 1), for each action a and observation z, an intermediate $\Gamma_t^{a,z}$ is computed from Γ_{t-1}:

$$\Gamma_t^{a,z} = \left\{\alpha_i^{a,z} : \alpha_i \in \Gamma_{t-1}\right\}, \tag{5}$$

Next, Γ_t^a is computed only for the sampled beliefs, i.e., $\Gamma_t^a = \{\alpha_b^a : b \in B\}$, where:

$$\alpha_b^a = \Gamma^a + \sum_{z\in\Omega} \operatorname*{argmax}_{\alpha\in\Gamma_t^{a,z}} \sum_{s'} b(s')\alpha(s'). \tag{6}$$

Finally, the best α-vector for each $b \in B$ is selected:

$$\alpha_b = \underset{\alpha_b^a}{\text{argmax}} \sum_{s'} b(s')\alpha_b^a(s'), \tag{7}$$

$$\Gamma_t = \cup_{b \in B} \alpha_b. \tag{8}$$

The above algorithm at each timestep t generates $|A_n||\Omega||\Gamma_{t-1}|$ alpha vectors in $\mathcal{O}(|S|^2|A||\Omega||\Gamma_{t-1}|)$ time and then reduces them to $|B|$ vectors in $\mathcal{O}(|S||B||A||\Omega||\Gamma_{t-1}|)$ (Pineau et al. 2006).

4 Active perception POMDP

The goal in an active perception POMDP is to reduce uncertainty about a *feature of interest* that is not directly observable. In general, the feature of interest may be only a part of the state, e.g., if a surveillance system cares only about people's positions, not their velocities, or higher-level features derived from the state. However, for simplicity, we focus on the case where the feature of interest is just the state s[2] of the POMDP. For simplicity, we also focus on *pure* active perception tasks in which the agent's only goal is to reduce uncertainty about the state, as opposed to hybrid tasks where the agent may also have other goals. For such cases, *hybrid* rewards (Eck and Soh 2012), which combine the advantage of belief-based and state-based rewards, are appropriate. Although not covered in this article, it is straightforward to extend our results to hybrid tasks (Spaan et al. 2015).

We model the active perception task as a POMDP in which an agent must choose a subset of available sensors at each time step. We assume that all selected sensors must be chosen simultaneously, i.e. it is not possible within a timestep to condition the choice of one sensor on the observations generated by another sensor. This corresponds to the common setting where generating each sensor's observation is time consuming, e.g., in the surveillance task, because it requires applying expensive computer vision algorithms, and thus all the observations from the selected cameras must be generated in parallel. Formally, an active perception POMDP has the following components:

- Actions $\mathbf{a} = \langle a_1 \dots a_N \rangle$ are vectors of N binary *action features*, each of which specifies whether a given sensor is selected or not. For each $\mathbf{a}$, we also define its set equivalent $\mathfrak{a} = \{i : a_i = 1\}$, i.e., the set of indices of the selected sensors. Due to the resource constraints, the set of all actions $A = \{\mathfrak{a} : |\mathfrak{a}| \leq K\}$ contains only sensor subsets of size K or less. $A^+ = \{1, \dots, N\}$ indicates the set of all sensors.

[2] We make this assumption without loss of generality. The following sections clarify that none of our results require this assumption.

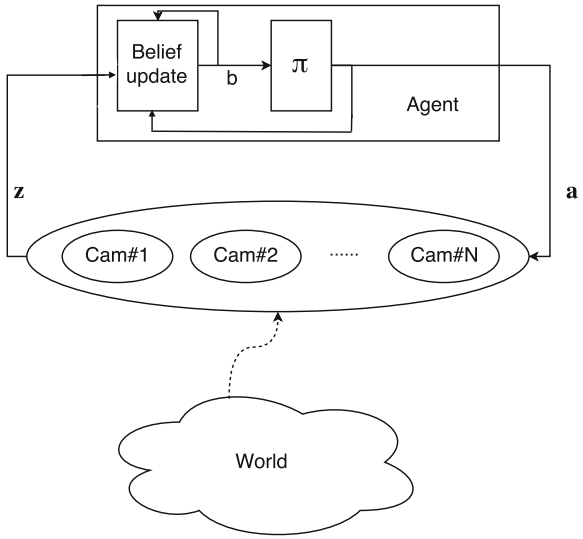

Fig. 2 Model for sensor selection problem

- Observations $\mathbf{z} = \langle z_1 \dots z_N \rangle$ are vectors of N *observation features*, each of which specifies the sensor reading obtained by the given sensor. If sensor i is not selected, then $z_i = \emptyset$. The set equivalent of $\mathbf{z}$ is $\mathfrak{z} = \{z_i : z_i \neq \emptyset\}$. To prevent ambiguity about which sensor generated which observation in $\mathfrak{z}$, we assume that, for all i and j, the domains of z_i and z_j share only $\emptyset$. This assumption is only made for notational convenience and does not restrict the applicability of our methods in any way.

For example, in the surveillance task, $\mathfrak{a}$ indicates the set of cameras that are active and $\mathfrak{z}$ are the observations received from the cameras in $\mathfrak{a}$. The model for the sensor selection problem for surveillance task is shown in Fig. 2. Here, we assume that the actions involve only selecting K out of N sensors. The transition function is thus independent of the actions, as selecting sensors cannot change the state. However, as we outline in Sect. 7.4, it is possible to extend our results to general active perception POMDPs with arbitrary transition functions, that can model, e.g., mobile sensors that, by moving, change the state.

A challenge in these settings is properly formalizing the reward function. Because the goal is to reduce the uncertainty, reward is a direct function of the belief, not the state, i.e., the agent has no preference for one state over another, so long as it knows what that state is. Hence, there is no meaningful way to define a state-based reward function $R(s, \mathbf{a})$. Directly defining $\rho(b, \mathbf{a})$ using, e.g., negative *belief entropy*: $-H_b(s) = \sum_s b(s) \log(b(s))$ results in a value function that is not piecewise-linear. Since $\rho(b, \mathbf{a})$ is no longer a convex combination of a state-based reward function, it is no longer guaranteed to be PWLC, a property most POMDP solvers rely on. In the following subsections, we describe

two recently proposed frameworks designed to address this problem.

4.1 ρPOMDPs

A ρPOMDP (Araya-López et al. 2010), defined by a tuple $\langle S, A, T, \Omega, O, \Gamma_\rho, b_0, h\rangle$, is a normal POMDP except that the state-based reward function $R(s, \mathbf{a})$ has been omitted and Γ_ρ has been added. Γ_ρ is a set of vectors that defines the immediate reward for ρPOMDP. Since we consider only pure active perception tasks, ρ depends only on b, not on $\mathbf{a}$ and can be written as $\rho(b)$. Given Γ_ρ, $\rho(b)$ can be computed as: $\rho(b) = \max_{\alpha\in\Gamma_\rho} \sum_s b(s)\alpha(s)$. If the true reward function is not PWLC, e.g., negative belief entropy, it can be approximated by defining Γ_ρ as a set of vectors, each of which is a tangent to the true reward function. Figure 3 illustrates approximating negative belief entropy with different numbers of tangents.

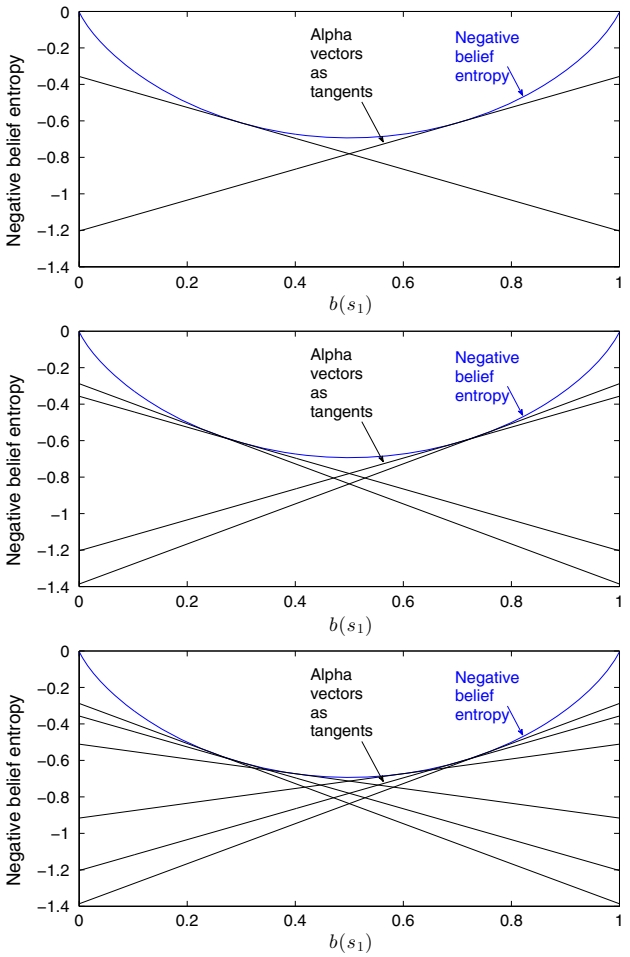

Fig. 3 Defining Γ_ρ^a with different sets of tangents to the negative belief entropy *curve* in a 2-state POMDP

Solving a ρPOMDP[3] requires a minor change to the existing algorithms. In particular, since Γ_ρ is a set of vectors, instead of a single vector, an additional cross-sum is required to compute $\Gamma_t^{\mathbf{a}}$: $\Gamma_t^{\mathbf{a}} = \Gamma_\rho \oplus \Gamma_t^{\mathbf{a},\mathbf{z}_1} \oplus \Gamma_t^{\mathbf{a},\mathbf{z}_2} \oplus \ldots$. Araya-López et al. (2010) showed that the error in the value function computed by this approach, relative to the true reward function, whose tangents were used to define Γ_ρ, is bounded. However, the additional cross-sum increases the computational complexity of computing $\Gamma_t^{\mathbf{a}}$ to $\mathcal{O}(|S||A||\Gamma_{t-1}||\Omega||B||\Gamma_\rho|)$ with point-based methods.

Though ρPOMDP does not put any constraints on the definition of ρ, we restrict the definition of ρ for an active perception POMDP to be a set of vectors ensuring that ρ is PWLC, which in turn ensures that the value function is PWLC. This is not a severe restriction because solving a ρPOMDP using *offline planning* requires a PWLC approximation of ρ anyway.

4.2 POMDPs with information rewards

Spaan et al. proposed *POMDPs with information rewards* (POMDP-IR), an alternative framework for modeling active perception tasks that relies only on the standard POMDP. Instead of directly rewarding low uncertainty in the belief, the agent is given the chance to make predictions about the hidden state and rewarded, via a standard state-based reward function, for making accurate predictions. Formally, a POMDP-IR is a POMDP in which each action $a \in A$ is a tuple $\langle \mathbf{a}_n, a_p\rangle$ where $\mathbf{a}_n \in A_n$ is a *normal action*, e.g., moving a robot or turning on a camera (in our case $\mathbf{a}_n$ is $\mathbf{a}$), and $a_p \in A_p$ is a *prediction action*, which expresses predictions about the state. The joint action space is thus the Cartesian product of A_n and A_p, i.e., $A = A_n \times A_p$.

Prediction actions have no effect on the states or observations but can trigger rewards via the standard state-based reward function $R(s, \langle \mathbf{a}_n, a_p\rangle)$. While there are many ways to define A_p and R, a simple approach is to create one prediction action for each state, i.e., $A_p = S$, and give the agent positive reward if and only if it correctly predicts the true state:

$$R(s, \langle \mathbf{a}_n, a_p\rangle) = \begin{cases} 1, & \text{if } s = a_p \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

[3] Arguably, there is a counter-intuitive relation between the general class of POMDPs and the sub-class of pure active perception problems: on the one hand, the class of POMDPs is a more general set of problems, and it is intuitive to assume that there might be harder problems in the class. On the other hand, many POMDP problems admit a representation of the value function using a finite set of vectors. In contrast, the use of entropy would require an infinite number of vectors to merely represent the reward function. Therefore, even though we consider a specific subclass of POMDPs, this class has properties that make it difficult to address using existing methods.

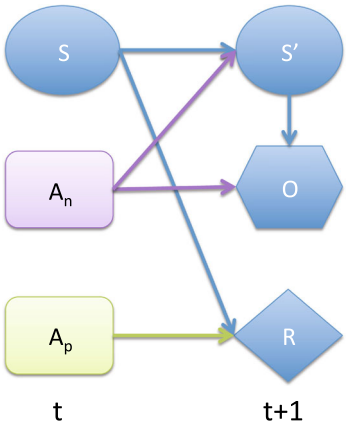

Fig. 4 Influence diagram for POMDP-IR

Thus, POMDP-IR indirectly rewards beliefs with low uncertainty, since these enable more accurate predictions and thus more expected reward. Furthermore, since a state-based reward function is explicitly defined, ρ can be defined as a convex combination of R, as in (1), guaranteeing a PWLC value function, as in a regular POMDP. Thus, a POMDP-IR can be solved with standard POMDP planners. However, the introduction of prediction actions leads to a blowup in the size of the joint action space $|A| = |A_n||A_p|$ of POMDP-IR. Replacing $|A|$ with $|A_n||A_p|$ in the analysis yields a complexity of computing $\Gamma_t^{\mathbf{a}}$ for POMDP-IR of $\mathcal{O}(|S||A_n||\Gamma_{t-1}||\Omega||B||A_p|)$ for point-based methods.

Note that, though not made explicit by Spaan et al. (2015), several independence properties are inherent to the POMDP-IR framework, as shown in Fig. 4. Specifically, the two important properties are (a) in our setting the reward function is independent of the normal actions; (b) the transition and the observation function are independent of the normal actions. Although POMDP-IR can model *hybrid rewards*, where in addition to prediction actions, normal actions can reward the agent as well (Spaan et al. 2015), in this article, because we focus on pure active perception, the reward function R is independent of the normal actions. Furthermore, state transitions and observations are independent of the prediction actions. In Sect. 6, we introduce a new technique to show that these independence properties can be exploited to solve a POMDP-IR much more efficiently and thus avoid the blowup in the size of the action space caused by the introduction of the prediction actions. Although the reward function in our setting is independent of the normal actions, the main results we present in this article are not dependent on this property and can be easily extended or applied to cases where the reward is dependent on the normal actions.

5 ρPOMDP and POMDP-IR equivalence

ρPOMDP and POMDP-IR offer two perspectives on modeling active perception tasks. ρPOMDP starts from a "true" belief-based reward function such as the negative entropy and then seeks to find a PWLC approximation via a set of tangents to the curve. In contrast, POMDP-IR starts from the queries that the user of the system will pose, e.g., "What is the position of everyone in the room?" or "How many people are in the room?" and creates prediction actions that reward the agent correctly for answering such queries. In this section we establish the relationship between these two frameworks by proving the *equivalence* of ρPOMDP and POMDP-IR. By equivalence of ρPOMDP and POMDP-IR, we mean that given a ρPOMDP and a policy, we can construct a corresponding POMDP-IR and a policy such that the value function for both the policies is exactly the same. We show this equivalence by starting with a ρPOMDP and a policy and introducing a *reduction* procedure for both ρPOMDP and the policy (and vice-versa). Using the reduction procedure, we reduce the ρPOMDP to a POMDP-IR and the policy for ρPOMDP to an equivalent policy for POMDP-IR. We then show that the value function, V_t^{π} for the ρPOMDP we started with and the reduced POMDP-IR is the same for the given and the reduced policy. To complete our proof, we repeat the same process by starting with a POMDP-IR and then reducing it to a ρPOMDP. We show that the value function V_t^{π} for the POMDP-IR and the corresponding ρPOMDP is the same.

Definition 1 Given a ρPOMDP $\mathbf{M}_\rho = \langle S, A_\rho, \Omega, T_\rho, O_\rho, \Gamma_\rho, b_0, h\rangle$ the REDUCE-POMDP-ρ-IR($\mathbf{M}_\rho$) produces a POMDP-IR $\mathbf{M}_{IR} = \langle S, A_{IR}, \Omega, T_{IR}, O_{IR}, R_{IR}, b_0, h\rangle$ via the following procedure.

- The set of states, set of observations, initial belief and horizon remain unchanged. Since the set of states remain unchanged, the set of all possible beliefs is also the same for $\mathbf{M}_{IR}$ and $\mathbf{M}_\rho$.
- The set of normal actions in $\mathbf{M}_{IR}$ is equal to the set of actions in $\mathbf{M}_\rho$, i.e., $A_{n,IR} = A_\rho$.
- The set of prediction actions $A_{p,IR}$ in $\mathbf{M}_{IR}$ contains one prediction action for each $\alpha_\rho^{a_p} \in \Gamma_\rho$.
- The transition and observation functions in $\mathbf{M}_{IR}$ behave the same as in $\mathbf{M}_\rho$ for each $\mathbf{a}_n$ and ignore the a_p, i.e., for all $\mathbf{a}_n \in A_{n,IR}$: $T_{IR}(s, \mathbf{a}_n, s') = T_\rho(s, \mathbf{a}, s')$ and $O_{IR}(s', \mathbf{a}_n, \mathbf{z}) = O_\rho(s', \mathbf{a}, \mathbf{z})$, where $\mathbf{a} \in A_\rho$ corresponds to $\mathbf{a}_n$.
- The reward function in $\mathbf{M}_{IR}$ is defined such that $\forall a_p \in A_p$, $R_{IR}(s, a_p) = \alpha_\rho^{a_p}(s)$, where $\alpha_\rho^{a_p}$ is the α-vector corresponding to a_p.

For example, consider a ρPOMDP with 2 states, if ρ is defined using tangents to belief entropy at $b(s_1) = 0.3$ and $b(s_1) = 0.7$. When reduced to a POMDP-IR, the resulting reward function gives a small negative reward for correct predictions and a larger one for incorrect predictions, with

the magnitudes determined by the value of the tangents when $b(s_1) = 0$ and $b(s_1) = 1$:

$$R_{IR}(s, a_p) = \begin{cases} -0.35, & \text{if } s = a_p \\ -1.21, & \text{otherwise.} \end{cases} \quad (10)$$

This is illustrated in Fig. 3 (top).

Definition 2 Given a policy π_ρ for a ρPOMDP, $\mathbf{M}_\rho$, the REDUCE- POLICY- ρ- IR(π_ρ) procedure produces a policy π_{IR} for a POMDP-IR as follows. For all b,

$$\pi_{IR}(b) = \left\langle \pi_\rho(b), \operatorname*{argmax}_{a_p} \sum_s b(s) R(s, a_p) \right\rangle. \quad (11)$$

That is, π_{IR} selects the same normal action as π_ρ and the prediction action that maximizes expected immediate reward.

Using these definitions, we prove that solving $\mathbf{M}_\rho$ is the same as solving $\mathbf{M}_{IR}$.

Theorem 1 *Let $\mathbf{M}_\rho$ be a ρPOMDP and π_ρ an arbitrary policy for $\mathbf{M}_\rho$. Furthermore let $\mathbf{M}_{IR}$ =* REDUCE- POMDP- ρ- IR($\mathbf{M}_\rho$) *and π_{IR} =* REDUCE- POLICY- ρ- IR(π_ρ). *Then, for all b,*

$$V_t^{IR}(b) = V_t^{\rho}(b), \quad (12)$$

where V_t^{IR} is the t-step value function for π_{IR} and V_t^{ρ} is the t-step value function for π_ρ.

Proof See Appendix. □

Definition 3 Given a POMDP-IR $\mathbf{M}_{IR} = \langle S, A_{IR}, \Omega, T_{IR}, O_{IR}, R_{IR}, b_0, h \rangle$ the REDUCE- POMDP- IR- ρ($\mathbf{M}_{IR}$) produces a ρPOMDP $\mathbf{M}_\rho = \langle S, A_\rho, \Omega, T_\rho, O_\rho, \Gamma_\rho, b_0, h \rangle$ via the following procedure.

- The set of states, set of observations, initial belief and horizon remain unchanged. Since the set of states remain unchanged, the set of all possible beliefs is also the same for $\mathbf{M}_{IR}$ and $\mathbf{M}_\rho$.
- The set of actions in $\mathbf{M}_\rho$ is equal to the set of normal actions in $\mathbf{M}_{IR}$, i.e., $A_\rho = A_{n,IR}$.
- The transition and observation functions in $\mathbf{M}_\rho$ behave the same as in $\mathbf{M}_{IR}$ for each $\mathbf{a}_n$ and ignore the a_p, i.e., for all $\mathbf{a} \in A_\rho$: $T_\rho(s, \mathbf{a}, s') = T_{IR}(s, \mathbf{a}_n, s')$ and $O_\rho(s', \mathbf{a}, \mathbf{z}) = O_{IR}(s', \mathbf{a}_n, \mathbf{z})$ where $\mathbf{a}_n \in A_{n,IR}$ is the action corresponding to $\mathbf{a} \in A_\rho$.
- The Γ_ρ in $\mathbf{M}_\rho$ is defined such that, for each prediction action in $A_{p,IR}$, there is a corresponding α vector in Γ_ρ, i.e., $\Gamma_\rho = \{\alpha_\rho^{a_p}(s) : \alpha_\rho^{a_p}(s) = R(s, a_p)$ for each $a_p \in A_{p,IR}\}$. Consequently, by definition, ρ is defined as: $\rho(b) = \max_{\alpha_\rho^{a_p}} [\sum_s b(s) \alpha_\rho^{a_p}(s)]$.

Definition 4 Given a policy $\pi_{IR} = \langle \mathbf{a}_n, a_p \rangle$ for a POMDP-IR, $\mathbf{M}_{IR}$, the REDUCE- POLICY- IR- $\rho(\pi_{IR})$ procedure produces a policy π_ρ for a POMDP-IR as follows. For all b,

$$\pi_\rho(b) = \pi_{IR}^n(b), \quad (13)$$

Theorem 2 *Let $\mathbf{M}_{IR}$ be a POMDP-IR and $\pi_{IR} = \langle \mathbf{a}_n, a_p \rangle$ a policy for $\mathbf{M}_{IR}$, such that $a_p = \operatorname{argmax}_{a'_p} b(s) R(s, a'_p)$. Furthermore let $\mathbf{M}_\rho$ =* REDUCE- POMDP- IR- $\rho(\mathbf{M}_{IR})$ *and π_ρ =* REDUCE- POLICY- IR- $\rho(\pi_{IR})$. *Then, for all b,*

$$V_t^{\rho}(b) = V_t^{IR}(b), \quad (14)$$

where V_t^{IR} is the value of following π_{IR} in $\mathbf{M}_{IR}$ and V_t^{ρ} is the value of following π_ρ in $\mathbf{M}_\rho$.

Proof See Appendix. □

The main implication of these theorems is that any result that holds for either ρPOMDP or POMDP-IR also holds for the other framework. For example, the results presented in Theorem 4.3 in Araya-López et al. (2010) that bound the error in the value function of ρPOMDP also hold for POMDP-IR. Furthermore, with this equivalence, the computational complexity of solving ρPOMDP and POMDP-IR comes out to be the same, since POMDP-IR can be converted into ρPOMDP (and vice-versa) trivially, without any significant blow-up in representation. Although we have proved the equivalence of ρPOMDP and POMDP-IR only for pure active perception tasks where the reward is solely conditioned on the belief, it is straightforward to extend it to hybrid active perception tasks, where the reward is conditioned both on belief and the state. Although, the resulting active perception POMDP for dynamic sensor selection is such that the action does not affect the state, the results from this section do not use that property at all and thus are valid for active perception POMDPs where an agent might take an action which can affect the state in the next time step.

6 Decomposed maximization for POMDP-IR

The POMDP-IR framework enables us to formulate uncertainty as an objective, but it does so at the cost of additional computations, as adding prediction actions enlarges the action space. The computational complexity of performing a point-based backup for solving POMDP-IR is $\mathcal{O}(|S|^2|A_n||A_p||\Omega||\Gamma_{t-1}|) + \mathcal{O}(|S||B||A_n||\Gamma_{t-1}||\Omega||A_p|)$. In this section, we present a new technique that exploits the independence properties of POMDP-IR, mainly that the transition function and the observation function are independent of the prediction actions, to reduce the computational costs. We also show that the same principle is applicable to ρPOMDPs.

The increased computational cost of solving POMDP-IR arises from the size of the action space, $|A_n||A_p|$. However, as shown in Fig. 4, prediction actions only affect the reward function and normal actions only affect the observation and transition function. We exploit this independence to decompose the maximization in the Bellman optimality equation:

$$\begin{aligned} V_t^*(b) &= \max_{\langle \mathbf{a}_n, a_p\rangle \in A} \Big[\sum_s b(s) R(s, a_p) \\ &\quad + \sum_{\mathbf{z}\in\Omega} \Pr(\mathbf{z}|\mathbf{a}_n, b) V_{t-1}^*(b^{\mathbf{a}_n,\mathbf{z}}) \Big] \\ &= \max_{a_p \in A_p} \sum_s b(s) R(s, a_p) \\ &\quad + \max_{\mathbf{a}_n \in A_n} \sum_{\mathbf{z}\in\Omega} \Pr(\mathbf{z}|\mathbf{a}_n, b) V_{t-1}^*(b^{\mathbf{a}_n,\mathbf{z}}) \end{aligned}$$

These decomposition can be exploited by point-based methods by computing $\Gamma_t^{a,z}$ only for normal actions, $\mathbf{a}_n$ and α^{a_p} only for prediction actions. That is, (5) can be changed to:

$$\Gamma_t^{\mathbf{a}_n,\mathbf{z}} = \left\{ \alpha_i^{\mathbf{a}_n,\mathbf{z}} : \alpha_i \in \Gamma_{t-1} \right\}. \tag{15}$$

For each prediction action, we compute the vector specifying the immediate reward for performing the prediction action in each state: $\Gamma^{A_p} = \{\alpha^{a_p}\}$, where $\alpha^{a_p}(s) = R(s, a_p)\ \forall\ a_p \in A_p$. The next step is to modify (6) to separately compute the vectors maximizing expected reward induced by prediction actions and the expected return induced by the normal action:

$$\begin{aligned} \alpha_b^{\mathbf{a}_n} &= \operatorname*{argmax}_{\alpha^{a_p}\in\Gamma^{A_p}} \sum_s b(s)\alpha^{a_p}(s) \\ &\quad + \sum_{\mathbf{z}} \operatorname*{argmax}_{\alpha^{\mathbf{a}_n,\mathbf{z}}\in\Gamma_t^{\mathbf{a}_n,\mathbf{z}}} \sum_s \alpha^{\mathbf{a}_n,\mathbf{z}}(s) b(s). \end{aligned}$$

By decomposing the maximization, this approach avoids iterating over all $|A_n||A_p|$ joint actions. At each timestep t, this approach generates $|A_n||\Omega||\Gamma_{t-1}| + |A_p|$ backprojections in $\mathcal{O}(|S|^2|A_n||\Omega||\Gamma_{t-1}|+|S||A_p|)$ time and then prunes them to $|B|$ vectors, with a computational complexity of $\mathcal{O}(|S||B|(|A_p| + |A_n||\Gamma_{t-1}||\Omega|))$.

The same principle can be applied to ρPOMDP by changing (6) such that it maximizes over immediate reward independently from the future return:

$$\begin{aligned} \alpha_b^{\mathbf{a}} &= \operatorname*{argmax}_{\alpha^\rho\in\Gamma_\rho} \sum_s b(s)\alpha_\rho^{a_p}(s) \\ &\quad + \sum_{\mathbf{z}} \operatorname*{argmax}_{\alpha^{\mathbf{a},\mathbf{z}}\in\Gamma_t^{\mathbf{a},\mathbf{z}}} \sum_s \alpha^{\mathbf{a},\mathbf{z}}(s) b(s). \end{aligned}$$

The computational complexity of solving ρPOMDP with this approach is $\mathcal{O}(|S|^2|A||\Omega||\Gamma_{t-1}| + |S||\Gamma_\rho|) + \mathcal{O}(|S||B|(|\Gamma_\rho| + |A||\Gamma_{t-1}||\Omega|))$. Thus, even though both POMDP-IR and ρPOMDP use extra actions or vectors to formulate belief-based rewards, they can both be solved at only minimal additional computational cost.

7 Greedy PBVI

The previous sections allow us to model the active perception task efficiently, such that the PWLC property of the value function is maintained. Thus, we can now directly employ traditional POMDP solvers that exploit this property to compute the optimal value function V_t^*.While point-based methods scale better in the size of the state space, they are still not practical for our needs as they do not scale in the size of the action space of active perception POMDPs.

While the computational complexity of one iteration of PBVI is linear in the size of the action space $|A|$ of a POMDP, for an active perception POMDP, the action space is modeled as selecting K out of the N available sensors, yielding $|A| = \binom{N}{K}$. For fixed K, as the number of sensors N grows, the size of the action space and the computational cost of PBVI grows exponentially with it, making use of traditional POMDP solvers infeasible for solving active perception POMDPs.

In this section, we propose *greedy PBVI*, a new point-based planner for solving active perception POMDPs which scales much better in the size of the action space. To facilitate the explication of greedy PBVI, we now present the final step of PBVI, described earlier in (7) and (8), in a different way. For each $b \in B$, and $\mathfrak{a} \in A$, we must find the best $\alpha_b^{\mathfrak{a}} \in \Gamma_t^{\mathfrak{a}}$,

$$\alpha_b^{\mathfrak{a},*} = \operatorname*{argmax}_{\alpha_b^{\mathfrak{a}}\in\Gamma_t^{\mathfrak{a}}} \sum_s \alpha_b^{\mathfrak{a}}(s) b(s), \tag{16}$$

and simultaneously record its value $Q(b, \mathfrak{a}) = \sum_s \alpha_b^{\mathfrak{a},*} b(s)$. Then, for each b we find the best vector across all actions: $\alpha_b = \alpha_b^{\mathfrak{a}^*}$, where

$$\mathfrak{a}^* = \operatorname*{argmax}_{\mathfrak{a}\in A} Q(b, \mathfrak{a}). \tag{17}$$

The main idea of greedy PBVI is to exploit *greedy maximization* (Nemhauser et al. 1978), an algorithm that operates on a set function $Q : 2^X \to \mathbb{R}$. Greedy maximization is much faster than full maximization as it avoids going over the $\binom{N}{K}$ choices and instead constructs a subset of K elements iteratively. Thus, we replace the maximization operator in the Bellman optimality equation with greedy maximization. Algorithm 1 shows the argmax variant, which constructs a subset $Y \subseteq X$ of size K by iteratively adding elements of X to Y. At each iteration, it adds the element that maximally increases *marginal gain* $\Delta_Q(e|\mathfrak{a})$ of adding a sensor e to a subset of sensors $\mathfrak{a}$:

$$\Delta_Q(e|\mathfrak{a}) = Q(b, e \cup \mathfrak{a}) - Q(b, \mathfrak{a}). \tag{18}$$

Algorithm 1 `greedy-argmax`(Q, X, K)

$Y \leftarrow \emptyset$
for $m = 1$ *to* K **do**
  $Y \leftarrow Y \cup \{\text{argmax}_{e \in X \setminus Y} \Delta_Q(e|Y)\}$
end for
return Y

To exploit greedy maximization in PBVI, we need to replace an argmax over A with `greedy-argmax`. Our alternative description of PBVI above makes this straightforward: (17) contains such an argmax and $Q(b, .)$ has been intentionally formulated to be a set function over A^+. Thus, implementing greedy PBVI requires only replacing (17) with

$$\mathfrak{a}^G = \texttt{greedy-argmax}(Q(b, \cdot), A^+, K). \tag{19}$$

Since the complexity of `greedy-argmax` is only $\mathcal{O}(|N||K|)$, the complexity of greedy PBVI is only $\mathcal{O}(|S||B||N||K||\Gamma_{t-1}|)$ (as compared to $\mathcal{O}(|S||B|\binom{n}{k})$ for traditional PBVI for computing $\Gamma_t^{\mathbf{a}}$).

Using point-based methods as a starting point is essential to our approach. Algorithms like Monahan's enumeration algorithm (Monahan 1982) that rely on pruning operations to compute V^* instead of performing an explicit argmax, cannot directly use `greedy-argmax`. Thus, it is precisely because PBVI operates on a finite set of beliefs that an explicit argmax is performed, opening the door to using `greedy-argmax` instead.

7.1 Bounds given submodular value function

In the following subsections, we present the highlights of the theoretical guarantees associated with greedy PBVI. The detailed analysis can be found in the appendix. Specifically, we show that a value function computed by greedy PBVI is guaranteed to have bounded error with respect to the optimal value function under *submodularity*, a property of set functions that formalizes the notion of diminishing returns. Then, we establish the conditions under which the value function of a POMDP is guaranteed to be submodular. We define $\rho(b)$ as negative belief entropy, $\rho(b) = -H_b(s)$ to establish the submodularity of value function. Both ρPOMDP and POMDP-IR approximate $\rho(b)$ with tangents. Thus, in the last subsection, we show that even if belief entropy is approximated using tangents, the value function computed by greedy PBVI is guaranteed to have bounded error with respect to the optimal value function.

Submodularity is a property of set functions that corresponds to diminishing returns, i.e., adding an element to a set increases the value of the set function by a smaller or equal amount than adding that same element to a subset. In our notation, this is formalized as follows. Given a policy π, the set function $Q_t^\pi(b, \mathfrak{a})$ is submodular in $\mathfrak{a}$, if for every $\mathfrak{a}_M \subseteq \mathfrak{a}_N \subseteq A^+$ and $a_e \in A^+ \setminus \mathfrak{a}_N$,

$$\Delta_{Q_b}(a_e|\mathfrak{a}_M) \geq \Delta_{Q_b}(a_e|\mathfrak{a}_N), \tag{20}$$

Equivalently, $Q_t^\pi(b, \mathfrak{a})$ is submodular if for every $\mathfrak{a}_M$, $\mathfrak{a}_N \subseteq A^+$,

$$Q_t^\pi(b, \mathfrak{a}_M \cap \mathfrak{a}_N) + Q_t^\pi(b, \mathfrak{a}_M \cup \mathfrak{a}_N) \leq Q_t^\pi(b, \mathfrak{a}_M) + Q_t^\pi(b, \mathfrak{a}_N).$$

Submodularity is an important property because of the following result:

Theorem 3 (Nemhauser et al. 1978) *If $Q_t^\pi(b, \mathfrak{a})$ is non-negative, monotone and submodular in $\mathfrak{a}$, then for all b,*

$$Q_t^\pi(b, \mathfrak{a}^G) \geq (1 - e^{-1}) Q_t^\pi(b, \mathfrak{a}^*), \tag{21}$$

where $\mathfrak{a}^G = \texttt{greedy-argmax}(Q_t^\pi(b, \cdot), A^+, K)$ and $\mathfrak{a}^* = \text{argmax}_{\mathfrak{a} \in A} Q_t^\pi(b, \mathfrak{a})$.

Theorem 3 gives a bound only for a single application of `greedy-argmax`, not for applying it within each backup, as greedy PBVI does.

In this subsection, we establish such a bound. Let the *greedy Bellman operator* $\mathfrak{B}^G$ be:

$$\left(\mathfrak{B}^G V_{t-1}^\pi\right)(b) = \max_{\mathfrak{a}}^G \left[\rho(b, \mathfrak{a}) + \sum_{\mathbf{z} \in \Omega} \Pr(\mathbf{z}|\mathfrak{a}, b) V_{t-1}^\pi(b^{\mathfrak{a}, \mathbf{z}})\right],$$

where $\max_{\mathfrak{a}}^G$ refers to greedy maximization. This immediately implies the following corollary to Theorem 3:

Corollary 1 *Given any policy π, if $Q_t^\pi(b, \mathfrak{a})$ is non-negative, monotone, and submodular in $\mathfrak{a}$, then for all b,*

$$\left(\mathfrak{B}^G V_{t-1}^\pi\right)(b) \geq (1 - e^{-1}) \left(\mathfrak{B}^* V_{t-1}^\pi\right)(b). \tag{22}$$

Proof From Theorem 3 since $(\mathfrak{B}^G V_{t-1}^\pi)(b) = Q_t^\pi(b, \mathfrak{a}^G)$ and $(\mathfrak{B}^* V_{t-1}^\pi)(b) = Q_t^\pi(b, \mathfrak{a}^*)$. □

Next, we define the *greedy Bellman equation*: $V_t^G(b) = (\mathfrak{B}^G V_{t-1}^G)(b)$, where $V_0^G = \rho(b)$. Note that V_t^G is the true value function obtained by greedy maximization, without any point-based approximations. Using Corollary 1, we can bound the error of V^G with respect to V^*.

Theorem 4 *If for all policies π, $Q_t^\pi(b, \mathfrak{a})$ is non-negative, monotone and submodular in $\mathfrak{a}$, then for all b,*

$$V_t^G(b) \geq (1 - e^{-1})^{2t} V_t^*(b). \tag{23}$$

Proof See Appendix. □

Theorem 4 extends Nemhauser's result to a full sequential decision making setting where multiple application of greedy maximization are employed over multiple time steps. This theorem gives a theoretical guarantee on the performance of greedy PBVI. Given a POMDP with a submodular value function, greedy PBVI is guaranteed to have bounded error with respect to the optimal value function. Moreover, this performance comes at a computational cost that is much less than that of solving the same POMDP with traditional solvers. Thus, greedy PBVI scales much better in the size of the action space of active perception POMDPs, while still retaining bounded error.

The results presented in this subsection are applicable only if the value function for a POMDP is submodular. In the following subsections, we establish the submodularity of the value function for the active perception POMDP under certain conditions.

7.2 Submodularity of value functions

The previous subsection showed that the value function computed by greedy PBVI is guaranteed to have bounded error as long as it is non-negative, monotone and submodular. In this subsection, we establish sufficient conditions for these properties to hold. Specifically, we show that, if the belief-based reward is negative entropy, i.e., $\rho(b) = -H_b(s) + \log(\frac{1}{|S|})$ then under certain conditions $Q_t^\pi(b, \mathfrak{a})$ is submodular, non-negative and monotone as required by Theorem 4. We point out that the second part, $\log(\frac{1}{|S|})$ is only required (and sufficient) to guarantee non-negativity, but is independent of the actual beliefs or actions. For the sake of conciseness, in the remainder of this paper we will omit this term.

We start by observing that $Q_t^\pi(b, \mathfrak{a}) = \rho(b) + \sum_{k=1}^{t-1} G_k^\pi(b^t, \mathfrak{a}^t)$, where $G_k^\pi(b^t, \mathfrak{a}^t)$ is the expected immediate reward with k steps to go, conditioned on the belief and action with t steps to go and assuming policy π is followed after timestep t:

$$G_k^\pi(b^t, \mathfrak{a}^t) = \sum_{\mathbf{z}^{t:k}} Pr\left(\mathbf{z}^{t:k} | b^t, \mathfrak{a}^t, \pi\right)\left(-H_{b^k}(s^k)\right),$$

where $\mathbf{z}^{t:k}$ is a vector of observations received in the interval from t steps to go to k steps to go, b^t is the belief at t steps to go, $\mathfrak{a}^t$ is the action taken at t steps to go, and $\rho(b^k) = -H_{b^k}(s^k)$, where s^k is the state at k steps to go. To show that $Q_t^\pi(b, \mathfrak{a})$ is submodular the main condition is *conditional independence* as defined below:

Definition 5 The observation set $\mathfrak{z}$ is conditionally independent given s if any pair of observation features are conditionally independent given the state, i.e.,

$$\Pr(z_i, z_j | s) = \Pr(z_i | s) \Pr(z_j | s), \quad \forall z_i, z_j \in \mathfrak{z}. \tag{24}$$

Using above definition, the submodularity of $Q(b, \mathfrak{a})$ can be established as:

Theorem 5 *If $\mathfrak{z}^{t:k}$ is conditionally independent given s^k and $\rho(b) = -H_b(s)$, then $Q_t^\pi(b, \mathfrak{a})$ is submodular in $\mathfrak{a}$, for all π.*

Proof See Appendix. □

Theorem 6 *If $\mathfrak{z}^{t:k}$ is conditionally independent given s^k and $\rho(b) = -H_b(s) + log(\frac{1}{|S|})$, then for all b,*

$$V_t^G(b) \geq (1 - e^{-1}) 2^t V_t^*(b). \tag{25}$$

Proof See Appendix. □

In this subsection we showed that if the immediate belief-based reward $\rho(b)$ is defined as negative belief entropy, then the value function of an active perception POMDP is guaranteed to be submodular under certain conditions. However, as mentioned earlier, to solve active perception POMDP, we approximate the belief entropy with vector tangents. This might interfere with the submodularity of the value function. In the next subsection, we show that, even though the PWLC approximation of belief entropy might interfere with the submodularity of the value function, the value function computed by greedy PBVI is still guaranteed to have bounded error.

7.3 Bounds given approximated belief entropy

While Theorem 6 bounds the error in $V_t^G(b)$, it does so only on the condition that $\rho(b) = -H_b(s)$. However, as discussed earlier, our definition of active perception POMDPs instead defines ρ using a set of vectors $\Gamma^\rho = \{\alpha_1^\rho, \ldots, \alpha_m^\rho\}$, each of which is a tangent to $-H_b(s)$, as suggested by Araya-López et al. (2010), in order to preserve the PWLC property. While this can interfere with the submodularity of $Q_t^\pi(b, \mathfrak{a})$, here we show that the error generated by this approximation is still bounded in this case.

Let $\tilde{\rho}(b)$ denote the PWLC approximated entropy and $\tilde{V}_t^*$ denote the optimal value function when using a PWLC approximation to negative entropy for the belief-based reward, as in an active perception POMDP, i.e.,

$$\tilde{V}_t^*(b) = \max_{\mathfrak{a}} \left[\tilde{\rho}(b) + \sum_{\mathbf{z} \in \Omega} \Pr(\mathbf{z} | b, \mathfrak{a}) \tilde{V}_{t-1}^*(b^{\mathfrak{a}, \mathbf{z}}) \right]. \tag{26}$$

Araya-López et al. (2010) showed that if $\rho(b)$ verifies the α-Hölder condition (Gilbarg and Trudinger 2001), a generalization of the Lipschitz condition, then the following relation holds between V_t^* and $\tilde{V}_t^*$:

$$||V_t^* - \tilde{V}_t^*||_\infty \leq C\delta^\alpha, \tag{27}$$

where V_t^* is the optimal value function with $\rho(b) = -H_b(s)$, δ is the *density* of the set of belief points at which tangent are drawn to the belief entropy, and C is a constant.

Let $\tilde{V}_t^G(b)$ be the value function computed by greedy PBVI when immediate belief-based reward is $\tilde{\rho}(b)$:

$$\tilde{V}_t^G(b) = \max_{\mathfrak{a}}^{G} \left[\tilde{\rho}(b) + \sum_{\mathbf{z}\in\Omega} \Pr(\mathbf{z}|b,\mathfrak{a}) \tilde{V}_{t-1}^G(b^{\mathfrak{a},\mathbf{z}}) \right], \tag{28}$$

then the error between $\tilde{V}_t^G(b)$ and $V_t^*(b)$ is bounded as stated in the following theorem.

Theorem 7 *For all beliefs, the error between $\tilde{V}_t^G(b)$ and $\tilde{V}_t^*(b)$ is bounded, if $\rho(b) = -H_b(s)$, and $\mathfrak{z}^{t:k}$ is conditionally independent given s^k.*

Proof See Appendix. □

In this subsection we showed that if the negative entropy is approximated using tangent vectors, greedy PBVI still computes a value function that has bounded error. In the next subsection we outline how greedy PBVI can be extended to general active perception tasks.

7.4 General active perception POMDPs

The results presented in this section apply to the active perception POMDP in which the evolution of the state over time is independent of the actions of the agent. Here, we outline how these results can be extended to general active perception POMDPs without many changes. The main application for such an extension is in tasks involving a mobile robot coordinating with sensors to intelligently take actions to perceive its environment. In such cases, the robot's actions, by causing it to move, can change the state of the world.

The algorithms we proposed can be extended to such settings by making small modifications to the greedy maximization operator. The greedy algorithm can be run for $K+1$ iterations and in each iteration the algorithm would choose to add either a sensor (only if fewer than K sensors have been selected), or a movement action (if none has been selected so far). Formally, using the work of Fisher et al. (1978), which extends that of Nemhauser et al. (1978) on submodularity to combinatorial structures such as *matroids*, the action space of a POMDP involving a mobile robot can be modeled as a *partition matroid* and greedy maximization subject to matroid constraints (Fisher et al. 1978) can be used to maximize the value function approximately.

The guarantees associated with greedy maximization subject to matroid constraints (Fisher et al. 1978) can then be used to bound the error of greedy PBVI. However, deriving exact theoretical guarantees for greedy PBVI for such tasks is beyond the scope of this article. Assuming that the reward function is still defined as the negative belief entropy, the submodularity of such POMDPs still holds under the conditions mentioned in Sect. 7.2.

In this section, we presented greedy PBVI, which uses greedy maximization to improve the scalability in the action space of an active perception POMDP. We also showed that, if the value function of an active perception POMDP is submodular, then greedy PBVI computes a value function that is guaranteed to have bounded error. We established that if the belief-based reward is defined as the negative belief entropy, then the value function of an active perception POMDP is guaranteed to be submodular. We showed that if the negative belief entropy is approximated by tangent vectors, as is required to solve active perception POMDPs efficiently, greedy PBVI still computes a value function that has bounded error. Finally, we outlined how greedy PBVI and the associated theoretical bounds can be extended to general active perception POMDPs.

8 Experiments

In this section, we present an analysis of the behavior and performance of belief-based rewards for active perception tasks, which is the main motivation of our work. We present the results of experiments designed to study the effect on the performance of the choice of prediction actions/tangents, and compare the costs and benefits of myopic versus non-myopic planning. We consider the task of tracking people in a surveillance area with a multi-camera tracking system. The goal of the system is to select a subset of cameras to correctly predict the position of people in the surveillance area, based on the observations received from the selected cameras. In the following subsections, we present results on real-data collected from a multi-camera system in a shopping mall and we present the experiments comparing performance of greedy PBVI to PBVI.

We compare the performance of POMDP-IR with decomposed maximization to a naive POMDP-IR that does not decompose the maximization. Thanks to Theorems 1 and 2, these approaches have performance equivalent to their ρPOMDP counterparts. We also compare against two baselines. The first is a weak baseline we call the *rotate policy* in which the agent simply keeps switching between cameras on a turn-by-turn basis. The second is a stronger baseline we call the *coverage policy*, which was developed in earlier work on active perception (Spaan 2008; Spaan and Lima 2009). The coverage policy is obtained after solving a POMDP that rewards the agent for observing the person, i.e., the agent is encouraged to select the cameras that are most likely to generate positive observations. Thanks to the decomposed

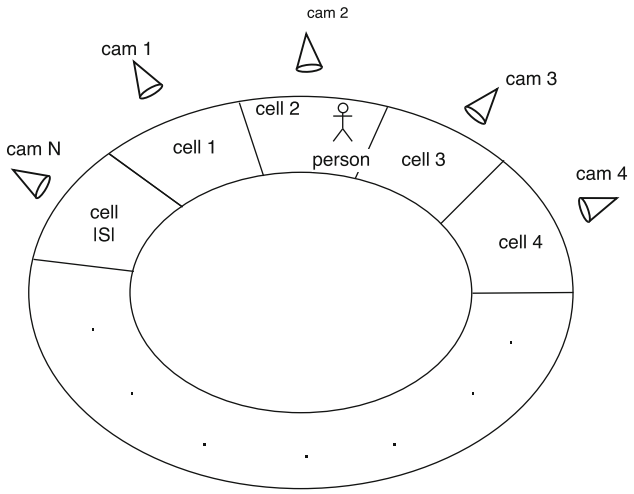

Fig. 5 Problem setup for the task of tracking one person. We model this task as a POMDP with one state for each cell. Thus the person can move among $|S|$ cells. Each cell is adjacent to two other cells and each cell is monitored by a single camera. Thus, in this case there are $N = |S|$ cameras. At each time step, the person can stay in the same cell as she was in the previous time step with probability p or she can move to one of the neighboring cells with equal probability. The agent must select K out of N cameras and the task is to predict the state of the person correctly using noisy observations from the K cameras. There is one prediction action for each state and the agent gets a reward of $+1$ if it correctly predicts the state and 0 otherwise. An observation is a vector of N *observation features*, each of which specifies the person's position as estimated by the given camera. If a camera is not selected, then the corresponding observation feature has a value of null

maximization, the computational cost of solving for the coverage policy and belief-based rewards is the same.

8.1 Simulated setting

We start with experiments conducted in a simulated setting, first considering the task of tracking a single person with a multi-camera system and then considering the more challenging task of tracking multiple people.

8.1.1 Single-person tracking

We start by considering the task of tracking one person walking in a grid-world composed of $|S|$ cells and N cameras as shown in Fig. 5. At each timestep, the agent can select only K cameras, where $K \leq N$. Each selected camera generates a noisy observation of the person's location. The agent's goal is to minimize its uncertainty about the person's state. In the experiments in this section, we fixed $K = 1$ and $N = 10$. The problem setup and the POMDP model is shown and described in Fig. 5.

To compare the performance of POMDP-IR to the baselines, 100 trajectories were simulated from the POMDP. The agent was asked to guess the person's position at each time step. Figure 6a shows the cumulative reward collected by all four methods. POMDP-IR with decomposed maximization

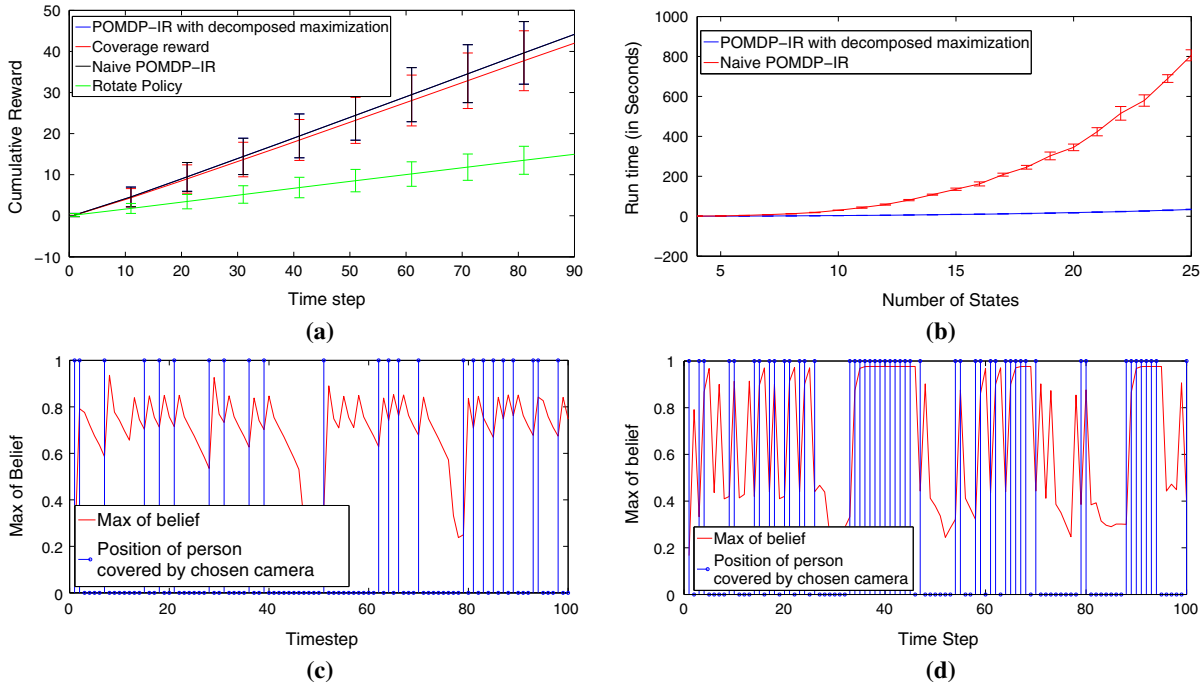

Fig. 6 **a** Performance comparison between POMDP-IR with decomposed maximization, naive POMDP-IR, coverage policy, and rotate policy; **b** runtime comparison between POMDP-IR with decomposed maximization and naive POMDP-IR; **c** behavior of POMDP-IR policy; **d** behavior of coverage policy (Color figure online)

and naive POMDP-IR perform identically as the lines indicating their respective performance lie on top of each other in Fig. 6a. However, Fig. 6b, which compares the runtimes of POMDP-IR with decomposed maximization and naive POMDP-IR, shows that decomposed maximization yields a large computational savings. Figure 6a also shows that POMDP-IR greatly outperforms the rotate policy and modestly outperforms the coverage policy.

Figures 6c, d illustrate the qualitative difference between POMDP-IR and the coverage policy. The blue lines mark the points in trajectory when the agent selected the camera that observes the person's location. If the agent selected a camera such that the person's location is not covered then the blue vertical line is not there at that point in the trajectory in the figure. The agent has to select one out of N cameras and does not have an option of not selecting any camera. The red line plots the max of the agent's belief. The main difference between the two policies is that once POMDP-IR gets a good estimate of the state, it proactively observes neighboring cells to which the person might transition. This helps it to more quickly find the person when she moves. By contrast, the coverage policy always looks at the cell where it believes her to be. Hence, it takes longer to find her again when she moves. This is evidenced by the fluctuations in the max of the belief, which often drops below 0.5 for the coverage policy but rarely does so for POMDP-IR.

Next, we examine the effect of approximating a true reward function like belief entropy with more and more tangents. Figure 3 illustrates how adding more tangents can better approximate negative belief entropy. To test the effects of this, we measured the cumulative reward when using between one and four tangents per state. Figure 7 shows the results and demonstrates that, as more tangents are added, the performance improves. However, performance also quickly saturates, as four tangents perform no better than three.

Next, we compare the performance of POMDP-IR to a myopic variant that seeks only to maximize immediate reward, i.e., $h = 1$. We perform this comparison in three variants of the task. In the *highly static* variant, the state changes very slowly: the probability of staying is the same state is 0.9. In the *moderately dynamic* variant, the state changes more frequently, with a same-state transition probability of 0.7. In the *highly dynamic* variant, the state changes rapidly (with a same-state transition probability of 0.5). Figure 8 (top) shows the results of these comparisons. In each setting, non-myopic POMDP-IR outperforms myopic POMDP-IR. In the highly static variant, the difference is marginal. However, as the task becomes more dynamic, the importance of look-ahead planning grows. Because the myopic planner focuses only on immediate reward, it ignores what might happen to its belief when the state changes, which happens more often in dynamic settings.

We also compare the performance of myopic and non-myopic planning in a *budget-constrained* environment. This specifically corresponds to an energy constrained environment, where cameras can be employed only a few times over the entire trajectory. This is augmented with resource constraints, so that the agent has to plan not only when to use the cameras, but also decide which camera to select. Specifically, the agent can only employ the multi-camera system a total of 15 times across all 50 timesteps and the agent can select which camera (out of the multi-camera system) to employ at each of the 15 instances. On the other timesteps, it must select an action that generates only a null observation. Figure 8 (bottom) shows that non-myopic planning is of critical importance in this setting. Whereas myopic planning greedily consumes the budget as quickly as possible, thus earning more reward in the beginning, non-myopic planning saves

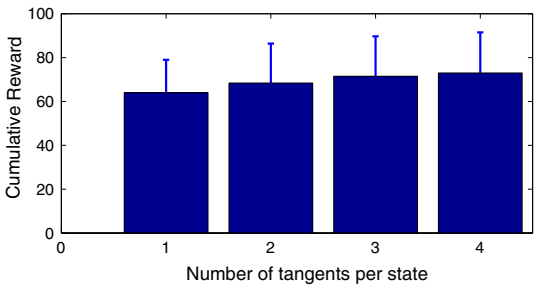

Fig. 7 Performance comparison as negative belief entropy is better approximated

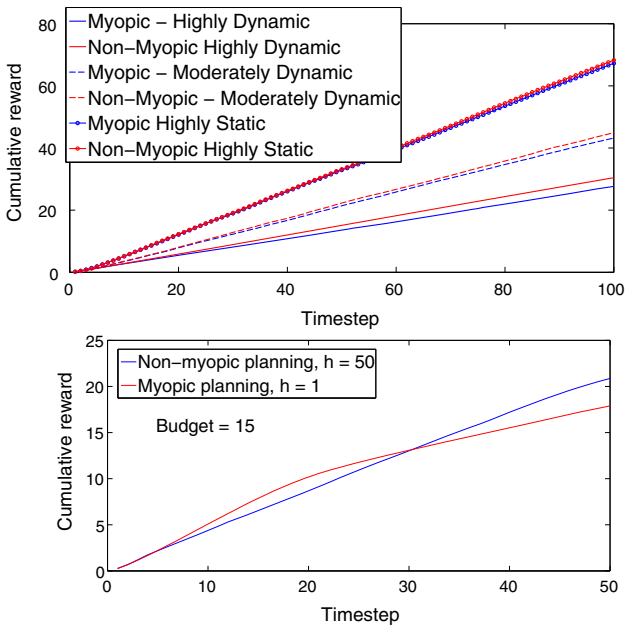

Fig. 8 (*Top*) Performance comparison for myopic versus non myopic policies; (*Bottom*) performance comparison for myopic versus non myopic policies in budget-based setting (Color figure online)

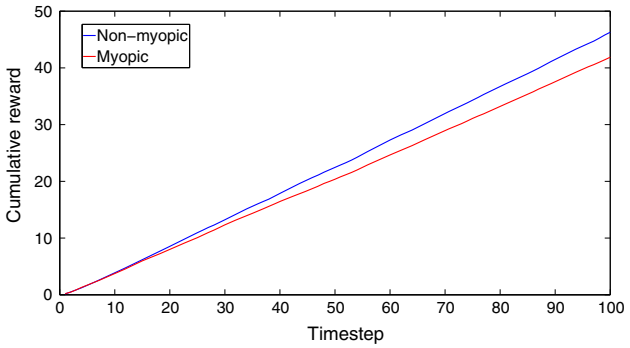

Fig. 9 Performance comparison for myopic versus non myopic policies when camera system is assisting a moving robot (Color figure online)

the budget for situations in which it is highly uncertain about the state.

Finally, we compare the performance of myopic and non-myopic planning when the multi-camera system can communicate with a mobile robot that also has sensors. This setting is typical of a networked robot system (Spaan et al. 2010) in which a robot coordinates with a multi-camera system to perform surveillance of a building, detect any emergency situations like fire, or help people navigate to their destination. Here, the task is to minimize uncertainty about the location of one person who is moving in the space monitored by the robot and the cameras. The robot's sensors are assumed to be more accurate than the stationary cameras. Specifically, the sensors attached to the robot can detect if a person is in the current cell with 90% accuracy compared to the stationary cameras, each of which has an accuracy of 75% of detecting a person in the cell it observes. The robot's sensor can observe the presence or absence of a person only for the cell that the robot occupies. In addition to using its sensors to generate observations about its current cell, the robot can also move forward or backward to an adjacent cell or choose to stay at the current cell. To model this task, the action vector introduced earlier is augmented with another action feature that indicates the direction of the robot's motion, which can take three values: forward, backward or stay.

Performance is quantified as the total number of times the correct location of the person is predicted by the system. Figure 9, which shows the performance of myopic and non-myopic policies for this task, demonstrates that when planning non-myopically the agent is able to utilize the accurate sensors more effectively as to compared to when planning myopically.

8.1.2 Multi-person tracking

To extend our analysis to a more challenging problem, we consider a simulated setting in which multiple people must be tracked simultaneously. Since $|S|$ grows exponentially in the number of people, the resulting POMDP quickly becomes intractable. Therefore, we compute instead a factored value function

$$V_t(b) = \sum_i V_t^i(b^i), \tag{29}$$

where $V_t^i(b^i)$ is the value of the agent's current belief b^i about the i-th person. Thus, $V_t^i(b^i)$ needs to be computed only once, by solving a POMDP of the same size as that in the single-person setting. During action selection, $V_t(b)$ is computed using the current b^i for each person. This kind of factorization corresponds to the assumption that each person's movement and observations is independent of that of other people. Although violated in practice, such an assumption can nonetheless yield good approximations.

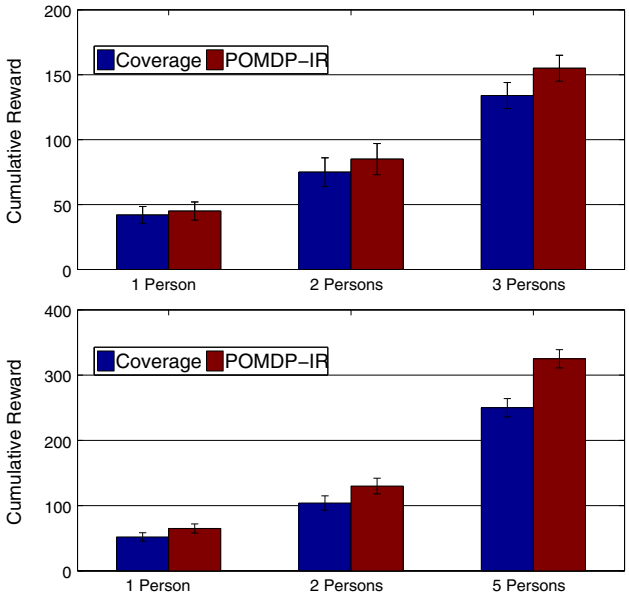

Fig. 10 (*Top*) Multi-person tracking performance for POMDP-IR and coverage policy; (*Bottom*) performance of POMDP-IR and coverage policy when only important cells must be tracked (Color figure online)

Figure 10 (top), which compares POMDP-IR to the coverage policy with one, two, and three people, shows that the advantage of POMDP-IR grows substantially as the number of people increases. Whereas POMDP-IR tries to maintain a good estimate of everyone's position, the coverage policy just tries to look at the cells where the maximum number of people might be present, ignoring other cells completely.

Finally, we compare POMDP-IR and the coverage policy in a setting in which the goal is only to reduce uncertainty about a set of "important cells" that are a subset of the whole state space. For POMDP-IR, we prune the set of prediction actions to allow predictions only about important cells. For the coverage policy, we reward the agent only for observ-

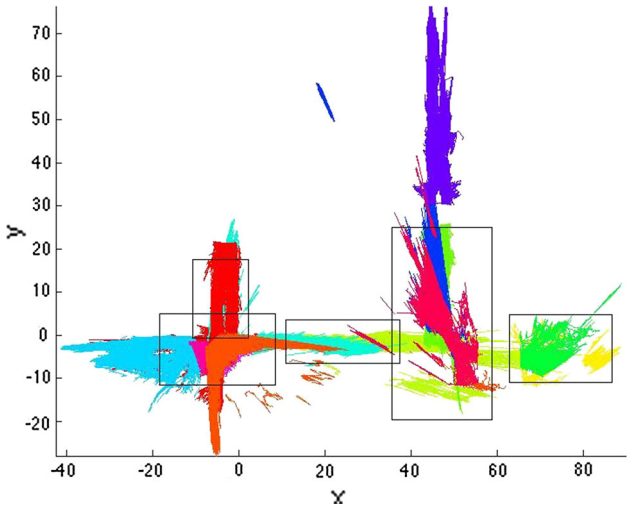

Fig. 11 Sample tracks for all the cameras. Each *color* represents all the tracks observed by a given camera. The *boxes* denote regions of high overlap between cameras (Color figure online)

ing people in important cells. The results, shown in Fig. 10 (bottom), demonstrate that the advantage of POMDP-IR over the coverage policy is even larger in this variant of the task. POMDP-IR makes use of information coming from cells that neighbor the important cells (which is of critical importance if the important cells do not have good observability), while the coverage policy does not. As before, the difference gets larger as the number of people increases.

8.2 Real data

Finally, we extended our analysis to a real-life dataset collected in a shopping mall. This dataset was gathered over 4 hours using 13 CCTV cameras located in a shopping mall (Bouma et al. 2013). Each camera uses a FPDW (Dollar et al. 2010) pedestrian detector to detect people in each camera image and in-camera tracking (Bouma et al. 2013) to generate tracks of the detected people's movements over time.

The dataset consists of 9915 tracks each specifying one person's x–y position over time. Figure 11 shows the sample tracks from all of the cameras.

To learn a POMDP model from the dataset, we divided the continuous space into 20 cells ($|S| = 21$: 20 cells plus an external state indicating the person has left the shopping mall). Using the data, we learned a maximum-likelihood tabular transition function. However, we did not have access to the ground truth of the observed tracks so we constructed them using the overlapping regions of the camera.

Because the cameras have many overlapping regions (see Fig. 11), we were able to manually match tracks of the same person recorded individually by each camera. The "ground truth" was then constructed by taking a weighted mean of the matched tracks. Finally, this ground truth was used to estimate noise parameters for each cell (assuming zero-mean Gaussian noise), which was used as the observation function. Figure 12 shows that, as before, POMDP-IR substantially outperforms the coverage policy for various numbers of cameras. In addition to the reasons mentioned before, the high overlap between the cameras contributes to POMDP-IR's superior performance. The coverage policy has difficulty ascertaining people's exact locations because it is rewarded only for observing them somewhere in a camera's large overlapping region, whereas POMDP-IR is rewarded for deducing their exact locations.

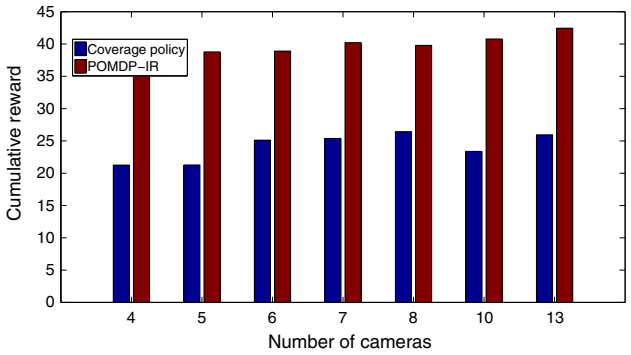

Fig. 12 Performance of POMDP-IR and the coverage policy on the shopping mall dataset (Color figure online)

8.3 Greedy PBVI

To empirically evaluate greedy PBVI, we tested it on the problem of tracking either one person or multiple people using a multi-camera system.

The reward function is described as a set of $|S|$ vectors, $\Gamma^{\rho} = \{\alpha_1 \ldots \alpha_{|S|}\}$, with $\alpha_i(s) = 1$ if $s = i$ and $\alpha_i(s) = 0$ otherwise. The initial belief is uniform across all states. We planned for horizon $h = 10$ with a discount factor $\gamma = 0.99$.

As baselines, we tested against regular PBVI and *myopic* versions of both greedy and regular PBVI that compute a policy assuming $h = 1$ and use it at each timestep. Figure 13 shows runtimes under different values of N and K. Since multi-person tracking uses the value function obtained by solving a single-person POMDP, single and multi-person tracking have the same runtimes. These results demonstrate that greedy PBVI requires only a fraction of the computational cost of regular PBVI. In addition, the difference in the runtime grows quickly as the action space gets larger: for $N = 5$ and $K = 2$ greedy PBVI is twice as fast, while for $N = 11$, $K = 3$ it is approximately nine times as fast. Thus, greedy PBVI enables much better scalability in the action space. Figure 14, which shows the cumulative reward under different values of N and K for single-person (top) and multi-person (bottom) tracking, verifies that greedy PBVI's

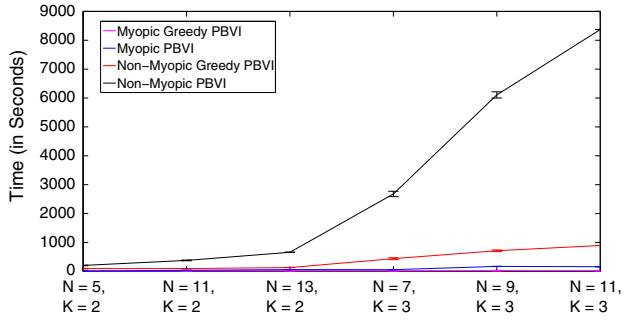

Fig. 13 Runtimes for the different methods (Color figure online)

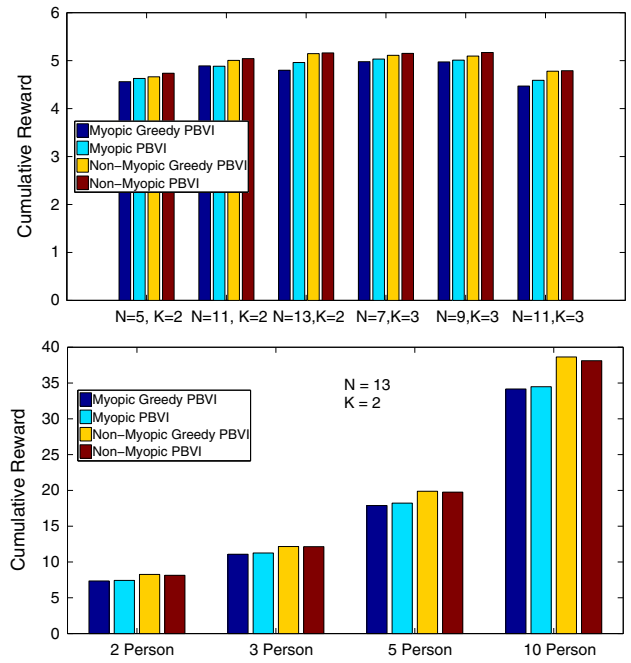

Fig. 14 Cumulative reward for single-person (*top*) and multi-person (*bottom*) tracking (Color figure online)

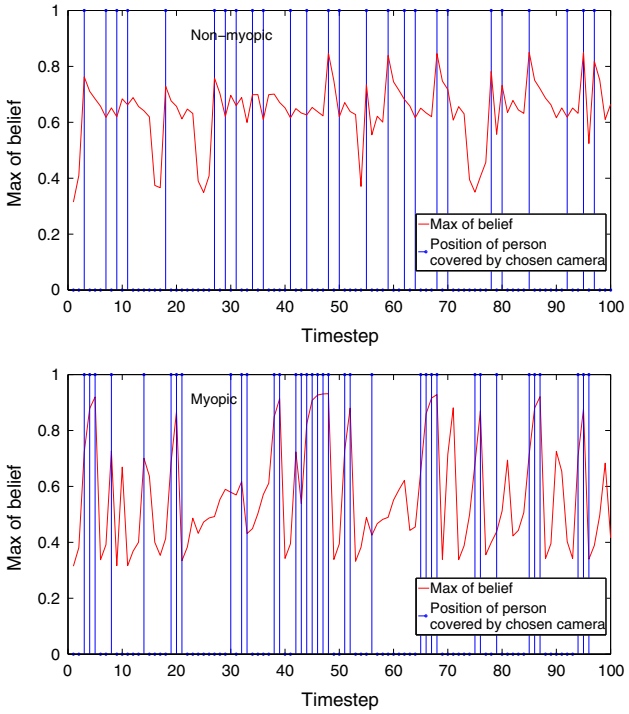

Fig. 15 Behavior of myopic versus non-myopic policy (Color figure online)

speedup does not come at the expense of performance, as greedy PBVI accumulates nearly as much reward as regular PBVI. They also show that both PBVI and greedy PBVI benefit from non-myopic planning. While the performance advantage of non-myopic planning is relatively modest, it increases with the number of cameras and people, which suggests that non-myopic planning is important to making active perception scalable.

Furthermore, an analysis of the resulting policies showed that myopic and non-myopic policies differ qualitatively. A myopic policy, in order to minimize uncertainty in the next step, tends to look where it believes the person to be. By contrast, a non-myopic policy tends to proactively look where the person might go next, so as to more quickly detect her new location when she moves. Consequently, non-myopic policies exhibit less fluctuation in belief and accumulate more reward, as illustrated in Fig. 15. The blue lines mark when the agent chooses the camera that can observe the cell occupied by the person. The red line plots the max of the agent's belief. The difference in fluctuation in belief is evident, as the max of the belief often drops below 0.5 for the myopic policy but rarely does so for the non-myopic policy.

9 Discussion and conclusions

In this article, we addressed the problem of active perception, in which an agent must take actions to reduce uncertainty about a hidden variable while reasoning about various constraints. Specifically, we modeled the task of surveillance with multi-camera tracking systems in large urban spaces as an active perception task. Since the state of the environment is dynamic, we model this task as a POMDP to compute closed-loop non-myopic policies that can reason about the long-term consequences of selecting a subset of sensors.

Formulating uncertainty reduction as an end in itself is a challenging task, as it breaks the PWLC property of the value function, which is imperative for solving POMDPs efficiently. ρPOMDP and POMDP-IR are two frameworks that allow formulating uncertainty reduction as an end in itself and do not break the PWLC property.

We showed that ρPOMDP and POMDP-IR are two equivalent frameworks for modeling active perception tasks. Thus,

results that apply to one framework are also applicable to the other one. While ρPOMDP does not restrict the definition of ρ to a PWLC function, in this work we restrict the definition of ρPOMDP to a case where ρ is approximated with a PWLC function, as it is not feasible to efficiently solve a ρPOMDP where ρ is not a PWLC function.

We model the action space of the active perception POMDP as selecting K out of N sensors, where K is the maximum number of sensors allowed by the resource constraints. Recent POMDP solvers enable scalability in the state space. However, for active perception, as the number of sensors grow, the action space grows exponentially. We proposed greedy PBVI, a POMDP planning method, that improves scalability in the action space of a POMDP. While we do not directly address the scaling in the observation space, we believe recent ideas on the factorization of the observation space (Veiga et al. 2014) can be combined with our approach to improve scalability in the state, action and observation spaces to solve active perception POMDPs.

By leveraging the theory of submodularity, we showed that the value function computed by greedy PBVI is guaranteed to have bounded error. Specifically, we extend Nemhauser's result on greedy maximization of submodular functions to long-term planning. To apply these results to the active perception task, we showed that under certain conditions the value function of an active perception POMDP is submodular. One such condition requires that the series of future of observations be independent of each other given the state. While this is a strong condition, it is only a sufficient condition and not may not be a necessary one. Thus, one line of future work is to attempt to relax this condition for proving the submodularity of the value function. Finally, we showed that, even with a PWLC approximation to the true value function, which is submodular, the error in the value function computed by greedy PBVI remains bounded, thus enabling us to compute efficiently value functions for active perception POMDPs.

Greedy PBVI is ideally suited for active perception POMDPs for which the value function is submodular. However, in real-life situations submodularity of value function might not always hold. For example, when there is occlusion in our setting, it is possible for combinations of sensors that when selected together yield higher utility than the sum of their utilities when selected individually. Similar cases can arise when a mobile robot is trying to sense the best point of view to observe a scene that is occluded. Thus in cases like these, greedy PBVI might not return the best solution.

Our empirical analysis established the critical factors involved in the performance of active perception tasks. We showed that a belief-based formulation of uncertainty reduction beats a corresponding state-based reward baseline as well as other simple policies. While the non-myopic policy beats the myopic one, in certain cases the gain is marginal. However, in cases involving mobile sensors and budgeted constraints, non-myopic policies become critically important. Finally, experiments on a real-world dataset showed that the performance of greedy PBVI is similar to the existing methods but requires only a fraction of the computational cost, leading to much better scalability for solving active perception tasks.

Acknowledgements We thank Henri Bouma and TNO for providing us with the dataset used in our experiments. We also thank the STW User Committee for its advice regarding active perception for multi-camera tracking systems. This research is supported by the Dutch Technology Foundation STW (project #12622), which is part of the Netherlands Organisation for Scientific Research (NWO), and which is partly funded by the Ministry of Economic Affairs. Frans Oliehoek is funded by NWO Innovational Research Incentives Scheme Veni #639.021.336.

Appendix

Results from Sect. 4

Theorem 1 *Let $\mathbf{M}_\rho$ be a ρPOMDP and π_ρ an arbitrary policy for $\mathbf{M}_\rho$. Furthermore let $\mathbf{M}_{IR}$ =* REDUCE- POMDP- ρ- IR($\mathbf{M}_\rho$) *and π_{IR} =* REDUCE- POLICY- ρ- IR(π_ρ). *Then, for all b,*

$$V_t^{IR}(b) = V_t^{\rho}(b), \tag{30}$$

where V_t^{IR} is the t-step value function for π_{IR} and V_t^{ρ} is the t-step value function for π_ρ.

Proof By induction on t. To prove the base case, we observe that, from the definition of $\rho(b)$,

$$V_0^{\rho}(b) = \rho(b) = \max_{\alpha_\rho^{a_p} \in \Gamma_\rho} \sum_s b(s)\alpha_\rho^{a_p}(s).$$

Since $\mathbf{M}_{IR}$ has a prediction action corresponding to each $\alpha_\rho^{a_p}$, thus the a_p corresponding to $\alpha = \text{argmax}_{\alpha_\rho^{a_p} \in \Gamma_\rho} \sum_s b(s)\alpha_\rho^{a_p}(s)$, must also maximize $\sum_s b(s)R(s, a_p)$. Then,

$$\begin{aligned} V_0^{\rho}(b) &= \max_{a_p} \sum_s b(s) R_{IR}(s, a_p) \\ &= V_0^{IR}(b). \end{aligned} \tag{31}$$

For the inductive step, we assume that $V_{t-1}^{IR}(b) = V_{t-1}^{\rho}(b)$ and must show that $V_t^{IR}(b) = V_t^{\rho}(b)$. Starting with $V_t^{IR}(b)$,

$$V_t^{IR}(b) = \max_{a_p} \sum_s b(s)R(s,a_p) + \sum_z \Pr\left(\mathbf{z}|b,\pi_{IR}^n(b)\right) V_{t-1}^{IR}(b^{\pi_{IR}^n(b),\mathbf{z}}), \tag{32}$$

where $\pi_{IR}^n(b)$ denotes the normal action of the tuple specified by $\pi_{IR}(b)$ and:

$$\begin{aligned}&\Pr\left(\mathbf{z}|b,\pi_{IR}^n(b)\right)\\ &= \sum_s \sum_{s''} O_{IR}\left(s'',\pi_{IR}^n(b),\mathbf{z}\right) T_{IR}\left(s,\pi_{IR}^n(b),s''\right) b(s).\end{aligned}$$

Using the reduction procedure, we can replace T_{IR} and O_{IR} and $\pi_{IR}^n(b)$ with their ρPOMDP counterparts on right hand side of the above equation:

$$\begin{aligned}&\Pr\left(\mathbf{z}|b,\pi_{IR}^n(b)\right)\\ &= \sum_s \sum_{s''} O_\rho(s'',\pi_\rho(b),z) T_\rho\left(s,\pi_\rho(b),s''\right) b(s)\\ &= \Pr(\mathbf{z}|b,\pi_\rho(b)).\end{aligned}$$

Similarly, for the belief update equation,

$$\begin{aligned}b^{\pi_{IR}^n(b),\mathbf{z}} &= \frac{O_{IR}\left(s',\pi_{IR}^n(b),\mathbf{z}\right)}{\Pr\left(\mathbf{z}|\pi_{IR}^n(b),b\right)} \sum_s b(s) T_{IR}\left(s,\pi_{IR}^n(b),s'\right)\\ &= \frac{O_\rho(s',\pi_\rho(b),\mathbf{z})}{\Pr(\mathbf{z}|\pi_\rho(b),b)} \sum_s b(s) T_\rho(s,\pi_\rho(b),s')\\ &= b^{\pi_\rho(b),\mathbf{z}}.\end{aligned}$$

Substituting the above result in (32) yields:

$$V_t^{IR}(b) = \max_{a_p} \sum_s b(s)R(s,a_p) + \sum_{\mathbf{z}} Pr(\mathbf{z}|b,\pi_\rho(b)) V_{t-1}^{IR}(b^{\pi_\rho(b),\mathbf{z}}). \tag{33}$$

Since the inductive assumption tells us that $V_{t-1}^{IR}(b) = V_{t-1}^{\rho}(b)$ and (31) shows that $\rho(b) = \max_{a_p} \sum_s b(s) R(s,a_p)$:

$$\begin{aligned}V_t^{IR}(b) &= \left[\rho(b) + \sum_{\mathbf{z}} Pr(\mathbf{z}|b,\pi_\rho(b)) V_{t-1}^{\rho}(b^{\pi_\rho(b),\mathbf{z}})\right]\\ &= V_t^{\rho}(b).\end{aligned} \tag{34}$$

□

Theorem 2 *Let* $\mathbf{M}_{IR}$ *be a POMDP-IR and* $\pi_{IR} = \langle \mathbf{a}_n, a_p \rangle$ *an policy for* $\mathbf{M}_{IR}$*, such that* $a_p = \max_{a_p'} b(s)R(s,a_p')$*. Furthermore let* $\mathbf{M}_\rho$ = REDUCE- POMDP- IR- $\rho(\mathbf{M}_{IR})$ *and* π_ρ = REDUCE- POLICY- IR- $\rho(\pi_{IR})$*. Then, for all b,*

$$V_t^{\rho}(b) = V_t^{IR}(b), \tag{35}$$

where V_t^{IR} *is the value of following* π_{IR} *in* $\mathbf{M}_{IR}$ *and* V_t^{ρ} *is the value of following* π_ρ *in* $\mathbf{M}_\rho$*.*

Proof By induction on t. To prove the base case, we observe that, from the definition of $\rho(b)$,

$$\begin{aligned}V_0^{IR}(b) &= \max_{a_p} \sum_s b(s)R(s,a_p)\\ &= \sum_s b(s)\alpha(s) \quad \Big\{\text{where } \alpha(s) \text{ is the } \alpha(s)\\ &\quad \text{corresponding to } a_p = \text{argmax}_{a_p'} \sum_s b(s)R(s,a_p').\Big\}\\ &= \rho(b)\\ &= V_0^{\rho}(b)\end{aligned} \tag{36}$$

For the inductive step, we assume that $V_{t-1}^{\rho}(b) = V_{t-1}^{IR}(b)$ and must show that $V_t^{\rho}(b) = V_t^{IR}(b)$. Starting with $V_t^{\rho}(b)$,

$$V_t^{\rho}(b) = \rho(b) + \sum_{\mathbf{z}} Pr(\mathbf{z}|b,\pi_\rho(b)) V_{t-1}^{\rho}(b^{\pi_\rho(b),\mathbf{z}}), \tag{37}$$

where $\pi_{IR}^n(b)$ denotes the normal action of the tuple specified by $\pi_{IR}(b)$ and:

$$Pr(\mathbf{z}|b,\pi_\rho(b)) = \sum_s \sum_{s''} O_\rho\left(s'',\pi_\rho(b),\mathbf{z}\right) T_\rho\left(s,\pi_\rho(b),s''\right) b(s). \tag{38}$$

From the reduction procedure, we can replace T_ρ and O_ρ and $\pi_\rho(b)$ with their POMDP-IR counterparts:

$$\begin{aligned}&Pr(\mathbf{z}|b,\pi_\rho(b))\\ &= \sum_s \sum_{s''} O_{IR}\left(s'',\pi_{IR}^n(b),\mathbf{z}\right) T_{IR}\left(s,\pi_{IR}^n(b),s''\right) b(s)\\ &= Pr\left(\mathbf{z}|b,\pi_{IR}(b)\right).\end{aligned} \tag{39}$$

Similarly, for the belief update equation,

$$\begin{aligned}b^{\pi_\rho(b),\mathbf{z}} &= \frac{O_\rho(s',\pi_\rho(b),\mathbf{z})}{Pr(\mathbf{z}|\pi_\rho(b),b)} \sum_s b(s) T_\rho(s,\pi_\rho(b),s')\\ &= \frac{O_{IR}\left(s',\pi_{IR}^n(b),\mathbf{z}\right)}{Pr\left(\mathbf{z}|\pi_{IR}^n(b),b\right)} \sum_s b(s) T_{IR}\left(s,\pi_{IR}^n(b),s'\right)\\ &= b^{\pi_{IR}(b),\mathbf{z}}.\end{aligned} \tag{40}$$

Substituting the above result in (37) yields:

$$V_t^{\rho}(b) = \rho(b) + \sum_{\mathbf{z}} Pr(\mathbf{z}|b,\pi_{IR}(b)) V_{t-1}^{IR}(b^{\pi_{IR}(b),\mathbf{z}}). \tag{41}$$

Since the inductive assumption tells us that $V_{t-1}^{\rho}(b) = V_{t-1}^{IR}(b)$ and (36) shows that $\max_{a_p} \sum_s b(s)R(s,a_p) = \rho(b)$:

$$V_t^{\rho}(b) = \left[\max_{a_p} \sum_s b(s) R(s, a_p) + \sum_{\mathbf{z}} Pr(\mathbf{z}|b, \pi_{IR}(b)) V_{t-1}^{IR}(b^{\pi_{IR}(b),\mathbf{z}})\right] = V_t^{IR}(b).$$

□

Results from Sect. 7.1

The following Lemma proves that the error in the value function remains bounded after application of $\mathfrak{B}^G$.

Lemma 1 *If for all b,* $\rho(b) \geq 0$,

$$V_t^{\pi}(b) \geq (1-\epsilon) V_t^*(b), \tag{42}$$

and $Q_t^{\pi}(b, \mathfrak{a})$ *is non-negative, monotone, and submodular in* $\mathfrak{a}$*, then, for* $\epsilon \in [0, 1]$,

$$\left(\mathfrak{B}^G V_t^{\pi}\right)(b) \geq (1-e^{-1})(1-\epsilon)\left(\mathfrak{B}^G V_t^*\right)(b). \tag{43}$$

Proof Starting from (42) and, for a given $\mathfrak{a}$, on both sides multiplying $\gamma \geq 0$, taking the expectation over $\mathbf{z}$, and adding $\rho(b)$ (since $\rho(b) \geq 0$ and $\epsilon \leq 1$):

$$\rho(b) + \mathbb{E}_{\mathbf{z}|b,\mathfrak{a}}\left[V_t^{\pi}(b^{a,\mathbf{z}})\right] \geq (1-\epsilon)\left(\rho(b) + \mathbb{E}_{\mathbf{z}|b,\mathfrak{a}}\left[V_t^*(b^{a,\mathbf{z}})\right]\right).$$

From the definition of Q_t^{π} (3), we thus have:

$$Q_{t+1}^{\pi}(b, \mathfrak{a}) \geq (1-\epsilon) Q_{t+1}^*(b, \mathfrak{a}) \ \ \forall \mathfrak{a}. \tag{44}$$

From Theorem 3, we know

$$Q_{t+1}^{\pi}\left(b, \mathfrak{a}_{\pi}^G\right) \geq (1-e^{-1}) Q_{t+1}^{\pi}\left(b, \mathfrak{a}_{\pi}^*\right), \tag{45}$$

where $\mathfrak{a}_{\pi}^G = \texttt{greedy-argmax}(Q_{t+1}^{\pi}(b, \cdot), A^+, K)$ and $\mathfrak{a}_{\pi}^* = \text{argmax}_{\mathfrak{a}}\, Q_{t+1}^{\pi}(b, \mathfrak{a})$. Since $Q_{t+1}^{\pi}(b, \mathfrak{a}_{\pi}^*) \geq Q_{t+1}^{\pi}(b, \mathfrak{a})$ for any $\mathfrak{a}$,

$$Q_{t+1}^{\pi}\left(b, \mathfrak{a}_{\pi}^G\right) \geq (1-e^{-1}) Q_{t+1}^{\pi}\left(b, \mathfrak{a}_{*}^G\right), \tag{46}$$

where $\mathfrak{a}_{*}^G = \texttt{greedy-argmax}(Q_t^*(b, \cdot), A^+, K)$. Finally, (44) implies that $Q_{t+1}^{\pi}(b, \mathfrak{a}_*^G) \geq (1-\epsilon) Q_{t+1}^*(b, \mathfrak{a}_*^G)$, so:

$$Q_{t+1}^{\pi}\left(b, \mathfrak{a}_{\pi}^G\right) \geq (1-e^{-1})(1-\epsilon) Q_{t+1}^*\left(b, \mathfrak{a}_{*}^G\right)$$
$$\left(\mathfrak{B}^G V_t^{\pi}\right)(b) \geq (1-e^{-1})(1-\epsilon)\left(\mathfrak{B}^G V_t^*\right)(b). \tag{47}$$

□

Using Corollary 1 and Lemma 1, we can prove Theorem 4.

Theorem 4 *If for all policies* π, $Q_t^{\pi}(b, \mathfrak{a})$ *is non-negative, monotone and submodular in* $\mathfrak{a}$*, then for all b,*

$$V_t^G(b) \geq (1-e^{-1})^{2t} V_t^*(b). \tag{48}$$

Proof By induction on t. The base case, $t = 0$, holds because $V_0^G(b) = \rho(b) = V_0^*(b)$.

In the inductive step, for all b, we assume that

$$V_{t-1}^G(b) \geq (1-e^{-1})^{2t-2} V_{t-1}^*(b), \tag{49}$$

and must show that

$$V_t^G(b) \geq (1-e^{-1})^{2t} V_t^*(b). \tag{50}$$

Applying Lemma 1 with $V_t^{\pi} = V_{t-1}^G$ and $(1-\epsilon) = (1-e^{-1})^{2t-2}$ to (49):

$$\left(\mathfrak{B}^G V_{t-1}^G\right)(b) \geq (1-e^{-1})^{2t-2}(1-e^{-1})\left(\mathfrak{B}^G V_{t-1}^*\right)(b)$$
$$V_t^G(b) \geq (1-e^{-1})^{2t-1}\left(\mathfrak{B}^G V_{t-1}^*\right)(b).$$

Now applying Corollary 1 with $V_{t-1}^{\pi} = V_{t-1}^*$:

$$V_t^G(b) \geq (1-e^{-1})^{2t-1}(1-e^{-1})\left(\mathfrak{B}^* V_{t-1}^*\right)(b)$$
$$V_t^G(b) \geq (1-e^{-1})^{2t} V_t^*(b). \tag{51}$$

□

Results from Sect. 7.2

Proving that $Q_t^{\pi}(b, \mathfrak{a})$ is submodular in $\mathfrak{a}$ requires three steps. First, we show that $G_k^{\pi}(b^t, \mathfrak{a}^t)$ equals the *conditional entropy* of b^k over s^k given $\mathbf{z}^{t:k}$ and $\mathfrak{a}^t$. Second, we show that, under certain conditions, conditional entropy is a submodular set function. Third, we combine these two results to show that $Q_t^{\pi}(b, \mathfrak{a})$ is submodular.

Lemma 2 *If* $\rho(b) = -H_b(s)$*, then the expected reward at each time step equals the negative discounted conditional entropy of* b^k *over* s^k *given* $\mathbf{z}^{t:k}$*:*

$$G_k^{\pi}(b^t, \mathfrak{a}^t) = \left(H_{b^k}(s^k|\mathbf{z}^{t:k}, \mathfrak{a}^t)\right) = \left(H_{b^k}^{\mathfrak{a}^t}(s^k|\mathbf{z}^{t:k})\right) \ \forall\, \pi.$$

Proof To prove the above lemma, we take help of some additional notations and definitions, first we must elaborate on the definition of b^k:

$$b^k(s^k) \triangleq Pr\left(s^k|b^t, \mathfrak{a}^t, \pi, \mathbf{z}^{t:k}\right) = \frac{Pr(\mathbf{z}^{t:k}, s^k|b^t, \mathfrak{a}^t, \pi)}{Pr(\mathbf{z}^{t:k}|b^t, \mathfrak{a}^t, \pi)}. \tag{52}$$

For notational convenience, we also write this as:

$$b^k(s^k) \triangleq \frac{Pr^{\pi}_{b^t,\mathfrak{a}^t}(\mathbf{z}^{t:k}, s^k)}{Pr^{\pi}_{b^t,\mathfrak{a}^t}(\mathbf{z}^{t:k})}. \tag{53}$$

The entropy of b^k is thus:

$$H_{b^k}(s^k) = \sum_{s^k} \frac{Pr^{\pi}_{b^t,\mathfrak{a}^t}(\mathbf{z}^{t:k}, s^k)}{Pr^{\pi}_{b^t,\mathfrak{a}^t}(\mathbf{z}^{t:k})} \log\left(\frac{Pr^{\pi}_{b^t,\mathfrak{a}^t}(\mathbf{z}^{t:k}, s^k)}{Pr^{\pi}_{b^t,\mathfrak{a}^t}(\mathbf{z}^{t:k})}\right),$$

and the conditional entropy of b^k over s^k given $\mathbf{z}^{t:k}$ is:

$$H^{\mathfrak{a}^t}_{b^k}(s^k|\mathbf{z}^{t:k}) = \sum\nolimits_{s^k} \sum\nolimits_{\mathbf{z}^{t:k}} Pr^{\pi}_{b^t,\mathfrak{a}^t}(\mathbf{z}^{t:k}, s^k) \log\left(\frac{Pr^{\pi}_{b^t,\mathfrak{a}^t}(\mathbf{z}^{t:k}, s^k)}{Pr^{\pi}_{b^t,\mathfrak{a}^t}(\mathbf{z}^{t:k})}\right).$$

Then, by definition of $G^{\pi}_k(b^t, \mathfrak{a}^t)$,

$$G^{\pi}_k(b^t, \mathfrak{a}^t) = \left(-\sum_{\mathbf{z}^{t:k}} Pr^{\pi}_{b^t,\mathfrak{a}^t}(\mathbf{z}^{t:k}) H_{b^k}(s^k)\right)$$

By definition of entropy,

$$= \sum\nolimits_{\mathbf{z}^{t:k}} Pr^{\pi}_{b^t,\mathfrak{a}^t}(\mathbf{z}^{t:k}) \left[\sum\nolimits_{s^k} \frac{Pr^{\pi}_{b^t,\mathfrak{a}^t}(\mathbf{z}^{t:k}, s^k)}{Pr^{\pi}_{b^t,\mathfrak{a}^t}(\mathbf{z}^{t:k})} \log\left(\frac{Pr^{\pi}_{b^t,\mathfrak{a}^t}(\mathbf{z}^{t:k}, s^k)}{Pr^{\pi}_{b^t,\mathfrak{a}^t}(\mathbf{z}^{t:k})}\right)\right]$$

$$= \sum_{\mathbf{z}^{t:k}} \left[\sum_{s^k} Pr^{\pi}_{b^t,\mathfrak{a}^t}(\mathbf{z}^{t:k}, s^k) \log\left(\frac{Pr^{\pi}_{b^t,\mathfrak{a}^t}(\mathbf{z}^{t:k}, s^k)}{Pr^{\pi}_{b^t,\mathfrak{a}^t}(\mathbf{z}^{t:k})}\right)\right]$$

By definition of conditional entropy,

$$= \left(-H^{\mathfrak{a}^t}_{b^k}(s^k|\mathbf{z}^{t:k})\right).$$ □

Lemma 3 *If $\mathfrak{z}$ is conditionally independent given s then $-H(s|\mathfrak{z})$ is submodular in $\mathfrak{z}$, i.e., for any two observations $\mathfrak{z}_M$ and $\mathfrak{z}_N$,*

$$H(s|\mathfrak{z}_M \cup \mathfrak{z}_N) + H(s|\mathfrak{z}_M \cap \mathfrak{z}_N) \geq H(s|\mathfrak{z}_M) + H(s|\mathfrak{z}_N). \tag{54}$$

Proof By Bayes' rule for conditional entropy (Cover and Thomas 1991):

$$H(s|\mathfrak{z}_M \cup \mathfrak{z}_N) = H(\mathfrak{z}_M \cup \mathfrak{z}_N|s) + H(s) - H(\mathfrak{z}_M \cup \mathfrak{z}_N). \tag{55}$$

Using conditional independence, we know $H(\mathfrak{z}_M \cup \mathfrak{z}_N|s) = H(\mathfrak{z}_M|s) + H(\mathfrak{z}_N|s)$. Substituting this in (55), we get:

$$H(s|\mathfrak{z}_M \cup \mathfrak{z}_N) = H(\mathfrak{z}_M|s) + H(\mathfrak{z}_N|s) + H(s) - H(\mathfrak{z}_M \cup \mathfrak{z}_N). \tag{56}$$

By Bayes' rule for conditional entropy:

$$H(s|\mathfrak{z}_M \cap \mathfrak{z}_N) = H(\mathfrak{z}_M \cap \mathfrak{z}_N|s) + H(s) - H(\mathfrak{z}_M \cap \mathfrak{z}_N). \tag{57}$$

Adding (55) and (57):

$$\begin{aligned} &H(s|\mathfrak{z}_M \cap \mathfrak{z}_N) + H(s|\mathfrak{z}_M \cup \mathfrak{z}_N) \\ &\quad = H(\mathfrak{z}_M|s) + H(\mathfrak{z}_N|s) \\ &\qquad + H(\mathfrak{z}_M \cap \mathfrak{z}_N|s) + 2H(s) \\ &\qquad - H(\mathfrak{z}_M \cup \mathfrak{z}_N) - H(\mathfrak{z}_M \cap \mathfrak{z}_N). \end{aligned} \tag{58}$$

By Bayes' rule for conditional entropy:

$$\begin{aligned} &H(\mathfrak{z}_M|s) = H(s|\mathfrak{z}_M) + H(\mathfrak{z}_M) - H(s), \text{ and} \\ &H(\mathfrak{z}_N|s) = H(s|\mathfrak{z}_N) + H(\mathfrak{z}_N) - H(s) \end{aligned} \tag{59}$$

Substituting $H(\mathfrak{z}_M|s)$ and $H(\mathfrak{z}_N|s)$ in (58):

$$\begin{aligned} &H(s|\mathfrak{z}_M \cap \mathfrak{z}_N) + H(s|\mathfrak{z}_M \cup \mathfrak{z}_N) \\ &\quad = H(s|\mathfrak{z}_M) + H(s|\mathfrak{z}_N) \\ &\qquad + H(\mathfrak{z}_M \cap \mathfrak{z}_N|s) + [H(\mathfrak{z}_M) \\ &\qquad + H(\mathfrak{z}_N) - H(\mathfrak{z}_M \cup \mathfrak{z}_N) - H(\mathfrak{z}_M \cap \mathfrak{z}_N)]. \end{aligned}$$

Since entropy is submodular $[H(\mathfrak{z}_M) + H(\mathfrak{z}_N) - H(\mathfrak{z}_M \cup \mathfrak{z}_N) - H(\mathfrak{z}_M \cap \mathfrak{z}_N)]$ is positive and since entropy is positive, $H(\mathfrak{z}_M \cap \mathfrak{z}_N|s)$ is positive. Thus,

$$\begin{aligned} H(s|\mathfrak{z}_M \cap \mathfrak{z}_N) + H(s|\mathfrak{z}_M \cup \mathfrak{z}_N) &= H(s|\mathfrak{z}_M) + H(s|\mathfrak{z}_N) \\ &\quad + \text{a positive term.} \end{aligned}$$

This implies $H(s|\mathfrak{z}_M \cup \mathfrak{z}_N) + H(s|\mathfrak{z}_M \cap \mathfrak{z}_N) \geq H(s|\mathfrak{z}_M) + H(s|\mathfrak{z}_N)$. □

Lemma 4 *If $\mathfrak{z}^{t:k}$ is conditionally independent given s^k and $\rho(b) = -H_b(s)$, then $G^{\pi}_k(b^t, \mathfrak{a}^t)$ is submodular in $\mathfrak{a}^t \; \forall \; \pi$.*

Proof Let $\mathfrak{a}^t_M$ and $\mathfrak{a}^t_N$ be two actions and $\mathfrak{z}^{t:k}_M$ and $\mathfrak{z}^{t:k}_N$ the observations they induce. Then, from Lemma 2,

$$G^{\pi}_k\left(b^t, \mathfrak{a}^t_M\right) = \left(-H^{\mathfrak{a}^t}_{b^k}\left(s^k|\mathfrak{z}^{t:k}_M\right)\right). \tag{60}$$

From Lemma 3,

$$\begin{aligned} &H^{\mathfrak{a}^t}_{b^k}\left(s^k|\mathfrak{z}^{t:k}_M \cup \mathfrak{z}^{t:k}_N\right) + H^{\mathfrak{a}^t}_{b^k}\left(s^k|\mathfrak{z}^{t:k}_M \cap \mathfrak{z}^{t:k}_N\right) \\ &\quad \geq H^{\mathfrak{a}^t}_{b^k}\left(s^k|\mathfrak{z}^{t:k}_M\right) + H^{\mathfrak{a}^t}_{b^k}\left(s^k|\mathfrak{z}^{t:k}_N\right) \end{aligned}$$

Using definition of G

$$\begin{aligned} &G^{\pi}_k\left(b^t, \mathfrak{a}^t_M \cup \mathfrak{a}^t_N\right) + G^{\pi}_k\left(b^t, \mathfrak{a}^t_N \cap \mathfrak{a}^t_M\right) \\ &\quad \leq G^{\pi}_k\left(b^t, \mathfrak{a}^t_M\right) + G^{\pi}_k\left(b^t, \mathfrak{a}^t_N\right). \end{aligned}$$ □

Theorem 5 *If $\mathfrak{z}^{t:k}$ is conditionally independent given s^k and $\rho(b) = -H_b(s)$, then $Q^{\pi}_t(b, \mathfrak{a})$ is submodular in $\mathfrak{a}$, for all π.*

Proof $\rho(b)$ is trivially submodular in $\mathfrak{a}$ because it is independent of $\mathfrak{a}$. Furthermore, Lemma 4 shows that $G_k^\pi(b^t, \mathfrak{a}^t)$ is submodular in $\mathfrak{a}^t$. Since a positively weighted sum of submodular functions is also submodular (Krause and Golovin 2014), this implies that $\sum_{k=1}^{t-1} G_k^\pi(b^t, \mathfrak{a}^t)$ and thus $Q_t^\pi(b, \mathfrak{a})$ are also submodular in $\mathfrak{a}$. □

Lemma 5 *If V_t^π is convex over the belief space for all t, then $Q_t^\pi(b, \mathfrak{a})$ is monotone in $\mathfrak{a}$, i.e., for all b and $\mathfrak{a}_M \subseteq \mathfrak{a}_N$, $Q_t^\pi(b, \mathfrak{a}_M) \leq Q_t^\pi(b, \mathfrak{a}_N)$.*

Proof By definition of $Q_t^\pi(b, \mathfrak{a})$,

$$Q_t^\pi(b, \mathfrak{a}_M) = \left[\rho(b) + \mathbb{E}_{\mathfrak{z}_M}\left[V_{t-1}^\pi(b^{\mathfrak{a}_M, \mathfrak{z}_M})|b, \mathfrak{a}_M\right]\right]. \quad (61)$$

Since $\rho(b)$ is independent of $\mathfrak{a}_M$, we need only show that the second term is monotone in $\mathfrak{a}$. Let $\mathfrak{a}_P = \mathfrak{a}_N \setminus \mathfrak{a}_M$ and

$$F_b^\pi(\mathfrak{a}_N) = \mathbb{E}_{\mathfrak{z}_N}\left[V_{t-1}^\pi(b^{\mathfrak{a}_N, \mathfrak{z}_N})||b, \mathfrak{a}_N\right]. \quad (62)$$

Since $\mathfrak{a}_N = \{\mathfrak{a}_M \cup \mathfrak{a}_P\}$,

$$F_b^\pi(\mathfrak{a}_N) = \mathbb{E}_{\{\mathfrak{z}_M, \mathfrak{z}_P\}}\left[V_{t-1}^\pi(b^{\{\mathfrak{a}_M, \mathfrak{a}_P\}, \{\mathfrak{z}_M, \mathfrak{z}_P\}})|b, \{\mathfrak{a}_M, \mathfrak{a}_P\}\right].$$

Separating expectations,

$$F_b^\pi(\mathfrak{a}_N) = \mathbb{E}_{\mathfrak{z}_M}\left[\mathbb{E}_{\mathfrak{z}_P}\left[V_{t-1}^\pi(b^{\{\mathfrak{a}_M, \mathfrak{a}_P\}, \{\mathfrak{z}_M, \mathfrak{z}_P\}})|b, \mathfrak{a}_P\right]|b, \mathfrak{a}_M\right]$$

Applying Jensen's inequality, since V_{t-1}^π is convex,

$$F_b^\pi(\mathfrak{a}_N) \geq \mathbb{E}_{\mathfrak{z}_M}\left[V_{t-1}^\pi\left(\mathbb{E}_{\mathfrak{z}_P}\left[b^{\mathfrak{a}_M, \mathfrak{a}_P, \mathfrak{z}_M, \mathfrak{z}_P}|b, \mathfrak{a}_P\right]\right)||b, \mathfrak{a}_M\right]$$

Since the expectation of the posterior is the prior,

$$\begin{aligned} F_b^\pi(\mathfrak{a}_N) &\geq \mathbb{E}_{\mathfrak{z}_M}\left[V_{t-1}^\pi\left(b^{\mathfrak{a}_M, \mathfrak{z}_M}\right)|b, \mathfrak{a}_M\right] \\ F_b^\pi(\mathfrak{a}_N) &\geq F_b^\pi(\mathfrak{a}_M). \end{aligned} \quad (63)$$

Consequently, we have:

$$\begin{aligned} \rho(b) + F_b^\pi(\mathfrak{a}_N) &\geq \rho(b) + F_b^\pi(\mathfrak{a}_M) \\ Q_t^\pi(b, \mathfrak{a}_N) &\geq Q_t^\pi(b, \mathfrak{a}_M). \end{aligned} \quad (64)$$

□

Lemma 5 requires that V_t^π be convex in belief space. To establish this for V_t^G, we must first show that $\mathfrak{B}^G$ preserves the convexity of the value function:

Lemma 6 *If ρ and V_{t-1}^π are convex over the belief simplex, then $\mathfrak{B}^G V_{t-1}^\pi$ is also convex.*

Proof

$$\begin{aligned} \mathfrak{B}^G V_{t-1}^\pi(b) &= \max_{\mathfrak{a}}^G\left[\rho(b) + \sum_{\mathbf{z}} \Pr(\mathbf{z}|b, \mathfrak{a}) V_{t-1}^\pi(b^{\mathfrak{a}, \mathbf{z}})\right] \\ &= \rho(b) + \sum_{\mathbf{z}} \Pr(\mathbf{z}|b, \mathfrak{a}^G) V_{t-1}^\pi(b^{\mathfrak{a}^G, \mathbf{z}}). \end{aligned}$$

The updated belief is $b^{\mathfrak{a}^G, \mathbf{z}}(\mathbf{s}) = \frac{\Pr(\mathbf{z}, s|\mathfrak{a}^G), b}{\Pr(\mathbf{z}|\mathfrak{a}^G, b)}$, which is the same as ω in Lemma A.1 in Araya-López et al. (2010). Thus by direct application of Lemma A.1 in Araya-López et al. (2010), $\mathfrak{B}^G V_{t-1}^\pi(b)$ is convex. □

Theorem 6 *If $\mathfrak{z}^{t:k}$ is conditionally independent given s^k and $\rho(b) = -H_b(s) + log(\frac{1}{|S|})$, then for all b,*

$$V_t^G(b) \geq (1 - e^{-1})^{2t} V_t^*(b). \quad (65)$$

Proof Follows from Theorem 4, given $Q_t^G(b, \mathfrak{a})$ is non-negative, monotone and submodular. For $\rho(b) = -H_b(s) + \log(\frac{1}{|S|})$, it is easy to see that $Q_t^G(b, \mathfrak{a})$ is non-negative, as entropy is always positive (Cover and Thomas 1991) and is maximum when $b(s) = \frac{1}{|S|}$ for all s (Cover and Thomas 1991). Theorem 5 showed that $Q_t^G(b, \mathfrak{a})$ is submodular if $\rho(b) = -H_b(s)$. The monotonicity of Q_t^G follows the fact that $-H_b(\mathbf{s})$ is convex (Cover and Thomas 1991): since Lemma 6 shows that $\mathfrak{B}^G$ preserves convexity, V_t^G is convex if $\rho(b) = -H_b(s)$; Lemma 5 then shows that $Q_t^G(b, \mathfrak{a})$ is monotone in $\mathfrak{a}$. □

Results from Sect. 7.3

Lemma 7 *For all beliefs b, the error between $V_t^G(b)$ and $\tilde{V}_t^G(b)$ is bounded by $C\delta^\alpha$. That is, $||V_t^G - \tilde{V}_t^G||_\infty \leq C\delta^\alpha$.*

Proof Follows exactly the strategy by Araya-López et al. (2010) used to prove (27), which places no conditions on π and thus holds as long as $\mathfrak{B}^G$ is a contraction mapping. Since for any policy the Bellman operator $\mathfrak{B}^\pi$ defined as:

$$\left(\mathfrak{B}^\pi V_{t-1}\right)(b) = \left[\rho(b, \mathfrak{a}_\pi) + \sum_{\mathbf{z}\in\Omega} \Pr(\mathbf{z}|\mathfrak{a}_\pi, b) V_{t-1}(b^{\mathfrak{a}_\pi, \mathbf{z}})\right],$$

is a contraction mapping (Bertsekas 2007), the bound holds for $\tilde{V}_t^G$. □

Let $\eta = C\delta^\alpha$ and $\tilde{Q}_t^*(b, \mathfrak{a}) = \tilde{\rho}(b) + \sum_{\mathbf{z}} \Pr(\mathbf{z}|b, \mathfrak{a}) \tilde{V}_{t-1}^*(b^{\mathfrak{a}, \mathbf{z}})$ denote the value of taking action $\mathfrak{a}$ in belief b under an optimal policy. Let $\tilde{Q}_t^G(b, \mathfrak{a}) = \tilde{\rho}(b) + \sum_{\mathbf{z}} \Pr(\mathbf{z}|b, \mathfrak{a}) \tilde{V}_{t-1}^G(b^{\mathfrak{a}, \mathbf{z}})$ be the action-value function computed by greedy PBVI with immediate reward being $\tilde{\rho}(b)$. Also, let

$$\tilde{Q}_t^{\pi}(b,\mathfrak{a}) = \tilde{\rho}(b) + \sum_{\mathbf{z}} \Pr(\mathbf{z}|b,\mathfrak{a})\tilde{V}_{t-1}^{\pi}(b^{\mathfrak{a},\mathbf{z}}),$$
$$\tilde{V}_t^{\pi}(b) = \tilde{\rho}(b) + \sum_{\mathbf{z}} \Pr(\mathbf{z}|b,\mathfrak{a}_{\pi})\tilde{V}_{t-1}^{\pi}(b^{\mathfrak{a}_{\pi},\mathbf{z}}), \quad (66)$$

denote the value function for a given policy π, when the belief based reward is $\tilde{\rho}(b)$. As mentioned before, it is not guaranteed that $\tilde{Q}_t^G(b,\mathfrak{a})$ is submodular. Instead, we show that it is ϵ-*submodular*:

Definition 6 The set function $f(\mathfrak{a})$ is ϵ-submodular in $\mathfrak{a}$, if for every $\mathfrak{a}_M \subseteq \mathfrak{a}_N \subseteq A^+$, $a_e \in A^+ \setminus \mathfrak{a}_N$ and $\epsilon \geq 0$,

$$f(a_e \cup \mathfrak{a}_M) - f(\mathfrak{a}_M) \geq f(a_e \cup \mathfrak{a}_N) - f(\mathfrak{a}_N) - \epsilon.$$

Lemma 8 *If* $||V_{t-1}^{\pi} - \tilde{V}_{t-1}^{\pi}||_{\infty} \leq \eta$, *and* $Q_t^{\pi}(b,\mathfrak{a})$ *is submodular in* $\mathfrak{a}$, *then* $\tilde{Q}_t^{\pi}(b,\mathfrak{a})$ *is* ϵ'*-submodular in* $\mathfrak{a}$ *for all* b, *where* $\epsilon' = 8\eta$.

Proof Since, $||V_{t-1}^{\pi} - \tilde{V}_{t-1}^{\pi}||_{\infty} \leq \eta$, then for all beliefs b,

$$V_{t-1}^{\pi}(b) - \tilde{V}_{t-1}^{\pi}(b) \leq \eta, \quad (67)$$

For a given $\mathfrak{a}$, on both sides, take the expectation over $\mathbf{z}$ and since $\rho(b) - \tilde{\rho}(b) \leq \eta$, ,

$$\rho(b) - \tilde{\rho}(b) + \mathbb{E}_{\mathbf{z}|b,\mathfrak{a}} V_{t-1}^{\pi}(b) - \mathbb{E}_{\mathbf{z}|b,\mathfrak{a}} \tilde{V}_{t-1}^{\pi}(b) \leq 2\eta$$

Therefore for all b, $\mathfrak{a}$,

$$Q_t^{\pi}(b,\mathfrak{a}) - \tilde{Q}_t^{\pi}(b,\mathfrak{a}) \leq 2\eta \quad (68)$$

Now since $Q_t^{\pi}(b,\mathfrak{a})$ is submodular, it satisfies the following equation,

$$Q_t^{\pi}(b, a_e \cup \mathfrak{a}_M) - Q_t^{\pi}(b,\mathfrak{a}_M) \geq Q_t^{\pi}(b, a_e \cup \mathfrak{a}_N) - Q_t^{\pi}(b,\mathfrak{a}_N), \quad (69)$$

for every $\mathfrak{a}_M \subseteq \mathfrak{a}_N \subseteq A^+$, $a_e \in A^+ \setminus \mathfrak{a}_N$ For each action that appear in (69), that is, $\{a_e \cup \mathfrak{a}_M\}$, $\mathfrak{a}_M$, $\{a_e \cup \mathfrak{a}_N\}$ and $\mathfrak{a}_N$, the value computed by $\tilde{Q}_t^{\pi}$ for belief b will be an approximation to Q_t^{π}. Thus the inequality in (69) that holds for Q_t^{π}, may not hold for $\tilde{Q}_t^{\pi}$. The worst case possible is, for some combination of b, $\{a_e \cup \mathfrak{a}_M\}$, $\mathfrak{a}_M$, $\{a_e \cup \mathfrak{a}_N\}$, $\tilde{Q}_t^{\pi}(b, a_e \cup \mathfrak{a}_M)$ and $Q_t^{\pi}(b,\mathfrak{a}_N)$ underestimates the true value of $Q_t^{\pi}(b, a_e \cup \mathfrak{a}_M)$ and $\tilde{Q}_t^{\pi}(b,\mathfrak{a}_N)$ by 2η each and $\tilde{Q}_t^{\pi}(b,\mathfrak{a}_M)$ and $\tilde{Q}_t^{\pi}(b, a_e \cup \mathfrak{a}_N)$ overestimates the value of $Q_t^{\pi}(b,\mathfrak{a}_M)$ and $Q_t^{\pi}(b, a_e \cup \mathfrak{a}_N)$ by 2η each. This can be written formally as: $\tilde{Q}_t^{\pi}(b, a_e \cup \mathfrak{a}_M) - \tilde{Q}_t^{\pi}(b,\mathfrak{a}_M) \geq \tilde{Q}_t^{\pi}(b, a_e \cup \mathfrak{a}_N) - \tilde{Q}_t^{\pi}(b,\mathfrak{a}_N) - 8\eta$. □

Lemma 9 *If* $\tilde{Q}_t^{\pi}(b,\mathfrak{a})$ *is non-negative, monotone and* ϵ*-submodular in* $\mathfrak{a}$, *then*

$$\tilde{Q}_t^{\pi}(b,\mathfrak{a}^G) \geq (1 - e^{-1})\tilde{Q}_t^{\pi}(b,\mathfrak{a}^*) - 4\chi_K\epsilon, \quad (70)$$

where $\chi_K = \sum_{p=0}^{K-1}(1 - K^{-1})^p$.

Proof Let $\mathfrak{a}^*$ be the optimal set of action features of size K, $\mathfrak{a}^* = \text{argmax}_{\mathfrak{a}}\, \tilde{Q}_t^{\pi}(b,\mathfrak{a})$ and let $\mathfrak{a}^l$ be the greedily selected set of size l, that is, $\mathfrak{a}^l = \texttt{greedy-argmax}(\tilde{Q}_t^{\pi}(b,\cdot), A^+, l)$ Also, let $\mathfrak{a}^* = \{a_1^* \ldots a_K^*\}$ be the elements of set $\mathfrak{a}^*$. Then,

By monotonicity of $\tilde{Q}_t^{\pi}(b,\mathfrak{a})$

$$\tilde{Q}_t^{\pi}(b,\mathfrak{a}^*) \leq \tilde{Q}_t^{\pi}(b,\mathfrak{a}^* \cup \mathfrak{a}^l)$$

Re-writing as a telescoping sum

$$= \tilde{Q}_t^{\pi}(b,\mathfrak{a}^l) + \sum_{j=1}^{K} \Delta_{\tilde{Q}_b}\left(a_j^* | \mathfrak{a}^l \cup \{a_1^* \ldots a_{j-1}^*\}\right)$$

Using Lemma 8, since Q is ϵ' − submodular

$$\leq \tilde{Q}_t^{\pi}(b,\mathfrak{a}^l) + \sum_{j=1}^{K} \Delta_{\tilde{Q}_b}(a_j^* | \mathfrak{a}^l) + 4K\epsilon$$

As $\mathfrak{a}^{l+1}$ is built greedily from $\mathfrak{a}^l$ in order to maximize $\Delta_{\tilde{Q}_b}$

$$\leq \tilde{Q}_t^{\pi}(b,\mathfrak{a}^l) + \sum_{j=1}^{K}\left(\tilde{Q}_t^{\pi}(b,\mathfrak{a}^{l+1}) - \tilde{Q}_t^{\pi}(b,\mathfrak{a}^l)\right) + 4K\epsilon$$

As $|\mathfrak{a}^*| = K$

$$= \tilde{Q}_t^{\pi}(b,\mathfrak{a}^l) + K\left(\tilde{Q}_t^{\pi}(b,\mathfrak{a}^{l+1}) - \tilde{Q}_t^{\pi}(b,\mathfrak{a}^l)\right) + 4K\epsilon$$

Let $\delta_l := \tilde{Q}_t^{\pi}(b,\mathfrak{a}^*) - \tilde{Q}_t^{\pi}(b,\mathfrak{a}^l)$, which allows us to rewrite above equation as: $\delta_l \leq K(\delta_l - \delta_{l+1}) + 4K\epsilon$. Hence, $\delta_{l+1} \leq (1 - \frac{1}{K})\delta_l + 4\epsilon$. Using this relation recursively, we can write, $\delta_K \leq (1 - \frac{1}{K})^K\delta_0 + 4\sum_{p=0}^{K-1}(1 - \frac{1}{K})^p\epsilon$. Also, $\delta_0 = \tilde{Q}_t^{\pi}(b,\mathfrak{a}^*) - \tilde{Q}_t^{\pi}(b,\mathfrak{a}^0)$ and using the inequality $1 - x \leq e^{-x}$, we can write $\delta_K \leq e^{-\frac{K}{K}}\tilde{Q}_t^{\pi}(b,\mathfrak{a}^*) + 4\sum_{p=0}^{K-1}(1 - K^{-1})\epsilon$. Substituting δ_K and rearranging terms (Also $\chi_K = \sum_{p=0}^{K-1}(1 - \frac{1}{K})^p$): $\tilde{Q}_t^{\pi}(b,\mathfrak{a}^G) \geq (1 - e^{-1})\tilde{Q}_t^{\pi}(b,\mathfrak{a}^*) - 4\chi_K\epsilon$. □

Theorem 7 *For all beliefs, the error between* $\tilde{V}_t^G(b)$ *and* $\tilde{V}_t^*(b)$ *is bounded, if* $\rho(b) = -H_b(s)$, *and* $\mathfrak{z}^{t:k}$ *is conditionally independent given* s^k.

Proof Theorem 6 shows that, if $\rho(b) = -H_b(s)$, and $\mathfrak{z}^{t:k}$ is conditionally independent given s^k, then $Q_t^G(b,\mathfrak{a})$ is submodular. Using Lemma 8, for $V_t^{\pi} = V_t^G$, $\tilde{V}_t^{\pi} = \tilde{V}_t^G$, $Q_t^{\pi}(b,\mathfrak{a}) = Q_t^G(b,\mathfrak{a})$ and $\tilde{Q}_t^{\pi}(b,\mathfrak{a}) = \tilde{Q}_t^G(b,\mathfrak{a})$, it is easy to see that $\tilde{Q}_t^G(b,\mathfrak{a})$ is ϵ-submodular. This satisfies one condition of Lemma 9. The convexity of $\tilde{V}_t^G(b)$ follows from Lemma 6 and that $\tilde{\rho}(b)$ is convex. Given that $\tilde{V}_t^G(b)$ is convex, the monotonicity of $\tilde{Q}_t^G(b,\mathfrak{a})$ follows from Lemma 5. Since $\tilde{\rho}(b)$ is non-negative, $\tilde{Q}_t^G(b,\mathfrak{a})$ is non-negative too. Now we can apply Lemma 9 to prove that the error generated by a one-time application of the greedy Bellman operator to $\tilde{V}_t^G(b)$, instead of the Bellman optimality operator, is bounded. It is thus easy to see that the error between $\tilde{V}_t^G(b)$, produced by multiple applications of the greedy Bellman operator, and $\tilde{V}_t^*(b)$ is bounded for all beliefs. □

References

Araya-López, M., Thomas, V., Buffet, O., & Charpillet, F. (2010). A POMDP extension with belief-dependent rewards. In *Advances in neural information processing systems* (pp. 64–72). MIT Press.

Aström, K. J. (1965). Optimal control of Markov decision processes with incomplete state estimation. *Journal of Mathematical Analysis and Applications*, *10*, 174–205.

Bajcsy, R. (1988). Active perception. *Proceedings of the IEEE*, *76*(8), 966–1005.

Bertsekas, D. P. (2007). *Dynamic programming and optimal control* (3rd ed., Vol. II). Belmont: Athena Scientific.

Bonet, B., & Geffner, H. (2009). Solving pomdps: Rtdp-bel vs. point-based algorithms. In *Proceedings of the Twenty-First International Jont Conference on Artifical Intelligence, IJCAI'09* (pp. 1641–1646).

Bouma, H., Baan, J., Landsmeer, S., Kruszynski, C., van Antwerpen, G., & Dijk, J. (2013). Real-time tracking and fast retrieval of persons in multiple surveillance cameras of a shopping mall. In *Proceedings of SPIE* (Vol. 8756, pp. 87560A–1).

Burgard, W., Fox, D., & Thrun, S. (1997). Active mobile robot localization by entropy minimization. In *Proceedings of the Second EUROMICRO Workshop on Advanced Mobile Robots 1997* (pp. 155–162). IEEE.

Chen, Y., Javdani, S., Karbasi, A., Bagnell, J. A., Srinivasa, S., & Krause, A. (2015). Submodular surrogates for value of information. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence* (pp. 3511–3518).

Cheng, H. T. (1988). *Algorithms for partially observable Markov decision processes*. Ph.D. thesis, University of British Columbia.

Cover, T. M., & Thomas, J. A. (1991). Entropy, relative entropy and mutual information. In *Elements of information theory* (pp. 12–49). Wiley.

Dollar, P., Belongie, S., & Perona, P. (2010). The fastest pedestrian detector in the west. In *Proceedings of the British Machine Vision Conference, BMVA Press* (pp. 68.1–68.11).

Eck, A., & Soh, L. K. (2012). Evaluating POMDP rewards for active perception. In *Proceedings of the Eleventh International Conference on Autonomous Agents and Multiagent Systems* (pp. 1221–1222).

Fisher, M. L., Nemhauser, G. L., & Wolsey, L. A. (1978). *An analysis of approximations for maximizing submodular set functions—II*. Berlin: Springer.

Gilbarg, D., & Trudinger, N. (2001). *Elliptic partial differential equations of second order*. Washington: U.S. Government Printing Office.

Golovin, D., & Krause, A. (2011). Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *Journal of Artificial Intelligence Research (JAIR)*, *42*, 427–486.

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

Ji, S., Parr, R., & Carin, L. (2007). Nonmyopic multiaspect sensing with partially observable Markov decision processes. *IEEE Transactions on Signal Processing*, *55*, 2720–2730.

Joshi, S., & Boyd, S. (2009). Sensor selection via convex optimization. *IEEE Transactions on Signal Processing*, *57*, 451–462.

Kaelbling, L. P., Littman, M. L., & Cassandra, A. R. (1998). Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, *101*, 99–134.

Kochenderfer, M. J. (2015). *Decision making under uncertainty: Theory and application*. Cambridge: MIT Press.

Krause, A., & Golovin, D. (2014). Submodular function maximization. In L. Bordeaux, Y. Hamadi, & P. Kohli (Eds.), *Tractability: Practical approaches to hard problems*. Cambridge: Cambridge University Press.

Krause, A., & Guestrin, C. (2005). Optimal nonmyopic value of information in graphical models—efficient algorithms and theoretical limits. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence* (pp. 1339–1345).

Krause, A., & Guestrin, C. (2007). Near-optimal observation selection using submodular functions. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence* (pp. 481–492).

Krause, A., & Guestrin, C. (2009). Optimal value of information in graphical models. *Journal of Artificial Intelligence Research*, *35*, 557–591.

Kreucher, C., Kastella, K., & Hero, A. O., III. (2005). Sensor management using an active sensing approach. *Signal Processing*, *85*, 607–624.

Krishnamurthy, V., & Djonin, D. V. (2007). Structured threshold policies for dynamic sensor scheduling—a partially observed Markov decision process approach. *IEEE Transactions on Signal Processing*, *55*(10), 4938–4957.

Kumar, A., & Zilberstein, S. (2009). Event-detecting multi-agent MDPs: Complexity and constant-factor approximation. In *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence* (pp. 201–207).

Kurniawati, H., Hsu, D., & Lee, W. S. (2008). Sarsop: Efficient point-based POMDP planning by approximating optimally reachable belief spaces. In *Proceedings robotics: Science and systems*.

Kurniawati, H., Du, Y., Hsu, D., & Lee, W. S. (2011). Motion planning under uncertainty for robotic tasks with long time horizons. *The International Journal of Robotics Research*, *30*(3), 308–323.

Littman, M. L. (1996). *Algorithms for sequential decision making*. Ph.D. thesis, Brown University.

Lovejoy, W. S. (1991). Computationally feasible bounds for partially observed Markov decision processes. *Operations Research*, *39*, 162–175.

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

Natarajan, P., Hoang, T. N., Low, K. H., Kankanhalli, M. (2012). Decision-theoretic approach to maximizing observation of multiple targets in multi-camera surveillance. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems* (pp. 155–162).

Nemhauser, G., Wolsey, L., & Fisher, M. (1978). An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, *14*, 265–294.

Oliehoek, F. A., Whiteson, S., & Spaan, M. T. J. (2013). Approximate solutions for factored Dec-POMDPs with many agents. In *Proceedings of the Twelfth International Joint Conference on Autonomous Agents and Multiagent Systems* (pp. 563–570).

Pineau, J., & Gordon, G. J. (2007). POMDP planning for robust robot control. In S. Thrun, R. Brooks, & H. Durrant-Whyte (Eds.), *Robotics Research* (pp. 69–82). Springer.

Pineau, J., Gordon, G. J., & Thrun, S. (2006). Anytime point-based approximations for large POMDPs. *Journal of Artificial Intelligence Research*, *27*, 335–380.

Poupart, P. (2005). *Exploiting structure to efficiently solve large scale partially observable Markov decision processes*. Ph.D. thesis, University of Toronto.

Raphael, C., & Shani, G. (2012). The skyline algorithm for POMDP value function pruning. *Annals of Mathematics and Artificial Intelligence*, *65*(1), 61–77.

Ross, S., Pineau, J., Paquet, S., & Chaib-Draa, B. (2008). Online planning algorithms for POMDPs. *Journal of Artificial Intelligence Research*, *32*, 663–704.

Satsangi, Y., Whiteson, S., & Oliehoek, F. (2015). Exploiting submodular value functions for faster dynamic sensor selection. In *AAAI 2015: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence* (pp. 3356–3363).

Satsangi, Y., Whiteson, S., & Oliehoek, F. A. (2016). PAC greedy maximization with efficient bounds on information gain for sensor selection. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI'16* (pp. 3220–3227). AAAI Press.

Shani, G., Pineau, J., & Kaplow, R. (2013). A survey of point-based POMDP solvers. *Autonomous Agents and Multi-Agent Systems*, *27*(1), 1–51.

Silver, D., Veness, J. (2010). Monte-carlo planning in large POMDPs. In *Advances in neural information processing systems* (pp. 2164–2172).

Smallwood, R. D., & Sondik, E. J. (1973). The optimal control of partially observable Markov processes over a finite horizon. *Operations Research*, *21*, 1071–1088.

Sondik, E. J. (1971). The optimal control of partially observable Markov processes. Ph.D. thesis, Stanford University, California, United States.

Spaan, M. T. J. (2008). Cooperative active perception using POMDPs. In *AAAI Conference on Artificial Intelligence 2008: Workshop on Advancements in POMDP Solvers*.

Spaan, M. T. J. (2012). Partially observable Markov decision processes. In M. Wiering & M. van Otterlo (Eds.), *Reinforcement learning: State of the art* (pp. 387–414). Berlin: Springer.

Spaan, M. T. J., & Lima, P. U. (2009). A decision-theoretic approach to dynamic sensor selection in camera networks. In *International Conference on Automated Planning and Scheduling* (pp. 279–304).

Spaan, M. T. J., & Vlassis, N. (2005). Perseus: Randomized point-based value iteration for POMDPs. *Journal of Artificial Intelligence Research*, *24*, 195–220.

Spaan, M. T. J., Veiga, T. S., & Lima, P. U. (2010). Active cooperative perception in network robot systems using POMDPs. In *International Conference on Intelligent Robots and Systems* (pp. 4800–4805).

Spaan, M. T. J., Veiga, T. S., & Lima, P. U. (2015). Decision-theoretic planning under uncertainty with information rewards for active cooperative perception. *Autonomous Agents and Multi-Agent Systems*, *29*, 1157–1185.

Veiga, T., Spaan, M. T. J., & Lima, P. U. (2014). Point-based POMDP solving with factored value function approximation. In *AAAI 2014: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*.

White, C. C. (1991). A survey of solution techniques for the partially observed Markov decision process. *Annals of Operations Research*, *32*(1), 215–230.

Williams, J., Fisher, J., & Willsky, A. (2007). Approximate dynamic programming for communication-constrained sensor network management. *IEEE Transactions on Signal Processing*, *55*, 4300–4311.

Yash Satsangi is a Ph.D. student in the Informatics Department at University of Amsterdam. Before joining University of Amsterdam as a Ph.D. student he studied Electrical Engineering at Columbia University, New York.

Shimon Whiteson studied English and Computer Science at Rice University before completing his doctorate in Computer Science at the University of Texas at Austin in 2007. He then spent eight years as an Assistant and then an Associate Professor at the University of Amsterdam before joining Oxford as an Associate Professor and a Fellow in Computer Science at St. Catz in 2015.

Frans A. Oliehoek currently is a Lecturer at the Department of Computer Science of the University of Liverpool. He received his Ph.D. in Computer Science (2010) from the University of Amsterdam. Frans' research focuses on decision making under uncertainty, with emphasis on multiagent systems.

Matthijs T. J. Spaan received his doctorate in Computer Science (2006) from the University of Amsterdam. Currently, he is an Associate Professor at Delft University of Technology.