Solving Markov Decision Processes with Partial State Abstractions

Samer B. Nashed[1]*, Justin Svegliato[1]*, Matteo Brucato[1], Connor Basich[1], Rod Grupen[1], Shlomo Zilberstein[1]

***Abstract*— Autonomous systems often use approximate planners that exploit state abstractions to solve large MDPs in real-time decision-making problems. However, these planners can eliminate details needed to produce effective behavior in autonomous systems. We therefore propose a novel model, a *partially abstract MDP*, with a set of abstract states that each *compress* a set of ground states to condense *irrelevant* details and a set of ground states that *expand* from a set of expanded abstract states to retain *relevant* details. This paper offers (1) a definition of a partially abstract MDP that (2) generalizes its ground MDP and its abstract MDP and exhibits bounded optimality depending on its abstract MDP along with (3) a lazy algorithm for planning and execution in autonomous systems. The result is a scalable approach that computes near-optimal solutions to large problems in minutes rather than hours.**

I. Introduction

Markov decision processes (MDP) are a general model for reasoning in fully observable, stochastic environments that have been used in many autonomous systems, such as search and rescue robots [1], [2], planetary rovers [3], [4], and self-driving cars [5], [6], [7], [8]. Typically, since autonomous systems operate in complex domains, there is a need to include many state factors in an MDP to produce effective operation. For example, a self-driving vehicle that uses an MDP to navigate roads and intersections includes each vehicle position, pedestrian location, and traffic light [9], [10], [11]. However, given the exponential growth of the state space in the number of state factors, autonomous systems must often solve an MDP approximately in real-time settings.

A common approach to solving MDPs approximately is to compute an optimal solution to some abstraction of an MDP. Recent work has focused on *abstract MDPs* that have abstract states that partition the ground states of a *ground MDP* according to certain criteria [12]. However, while abstract MDPs can be solved much faster than ground MDPs, they can eliminate details needed to produce effective behavior in autonomous systems. Ideally, for autonomous systems to produce effective behavior in an acceptable amount of time, an abstract MDP should not only condense unimportant details but also retain important details of the ground MDP.

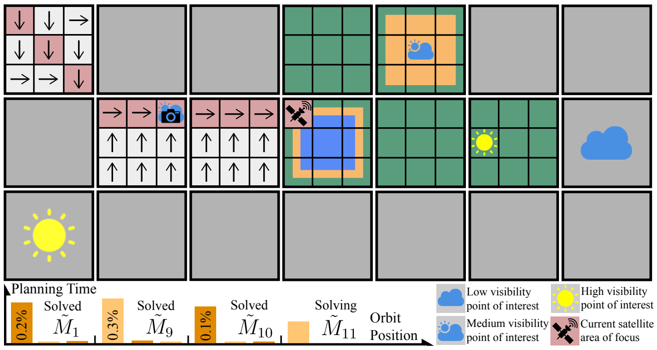

Fig. 1. A satellite orbiting Earth must take photos of 5 points of interests with 3 weather levels that change gradually. Initially, it sketches an abstract MDP $\bar{M}$ that compresses 45927 ground states of a ground MDP M into 672 abstract states (*gray*) of 3×3 abstract cells and 2 abstract weather levels. At each new abstract state along its path (*pink*), it refines the abstract MDP $\bar{M}$ into a partially abstract MDP $\tilde{M}$ that expands the new abstract state and other informative abstract states based on either a naive (*blue*), greedy (*orange*), or proactive (*green*) expansion strategy into ground states (*white*). The graph shows that each partially abstract MDP $\tilde{M}_i$ is solved in less than 1% of the *planning time* of the ground MDP M for each *orbit position*.

We therefore offer a novel model, called a *partially abstract MDP*, that generalizes ground MDPs and abstract MDPs. Similar to an abstract MDP, it has a set of abstract states that each *compress* a set of ground states to condense *irrelevant* details. However, similar to a ground MDP, it has a set of ground states that *expand* from a set of expanded abstract states to retain *relevant* details. Hence, a partially abstract MDP can be an adjustable balance of the *coarse-grained* abstract MDP and the *fine-grained* ground MDP.

Given an abstract MDP and a partially abstract MDP, we propose a lazy algorithm for planning and execution that allows an autonomous system to complete a task described by a ground MDP in Figure 1. Before starting the task, the system builds and solves an *abstract MDP* where each abstract state *compresses* a set of ground states. While completing the task, every time a new abstract state is encountered, the system builds and solves a *partially abstract MDP* where the new abstract state and other informative abstract states based on an expansion strategy *expand* into a set of ground states. By initially sketching an abstract MDP offline and later refining it into partially abstract MDPs online, the system produces effective behavior in an acceptable amount of time without solving the ground MDP. This algorithm is inspired by the *SketchRefine* method for solving large integer programs [13].

Our contributions are: (1) a definition of a partially abstract MDP, (2) an analysis that a partially abstract MDP generalizes its ground MDP and its abstract MDP and exhibits bounded optimality depending on its abstract MDP, and (3) a lazy algorithm for planning and execution in autonomous systems. Most importantly, we demonstrate that our approach is near-optimal and scalable in an Earth observation domain.

This work was supported in part by the NSF GRFP (DGE-1451512), the NSF (IIS-1724101 and IIS-1813490), and NASA (ESI-80NSSC19K0212).

*Both authors contributed equally.

[1]College of Information and Computer Sciences, University of Massachusetts Amherst, MA, USA. Emails: {snashed, jsvegliato, matteo, cbasich, grupen, shlomo}@cs.umass.edu

II. Related Work

The desire to solve large MDPs is not new, and techniques for doing so generally adopt one of three approaches. First, there are *approximate solvers* that use dynamic programming methods based on value or policy iteration [14], [15] and linear programming [16], [17], [18], [19]. Second, some methods compute *partial policies* on a subset of the ground states and re-plan if the agent encounters a state for which the partial policy is undefined [20], [21]. Third, optimal policies are computed on *abstractions* of the original problem, where there is a surjective mapping from the original ground states to the abstract states [12]. Our approach combines insights from both partial policies and abstractions but does not preclude the use of approximate solvers.

Using abstractions to reduce the size of a problem is a natural and popular approach to solving large MDPs. The quality of these policies depends heavily on the abstraction scheme, and many abstraction methods have been proposed. Some strict definitions include bisimulation [22], statistical bisimulation [23], and bounded MDPs [24]. Abstractions based on homomorphisms [25], [26] and generic change of basis have also been proposed [27]. Abstractions also support hierarchical systems, such as MDP [28] and object hierarchies [29]. Abstractions for continuous variables [30] and across time [31] have been examined as well. Some work even uses temporal abstractions derived from analytically computed landmarks to summarize policies for stochastic shortest path (SSP) problems, a subclass of MDPs [32]. Another form of abstraction is determinization, or its more general form, reduced models, which forms abstractions over action outcomes [33], [34]. Our approach retains the potential for substantial state space reductions shared by abstraction-based approaches but remains more robust to abstraction schemes because it computes policies on ground states.

Computing partial policies is an approach with a history of success. FF-Replan [35], a remarkably simple yet effective algorithm for planning in MDPs, works by determinizing an MDP, constructing a plan to the determinization, and re-planning if the agent reaches an unexpected state. Recently, Soft-FLARES [36] achieved impressive results on large SSPs by probabilistically labeling ϵ-consistent state values within a horizon. There has also been work on designing meta-level controllers to reason about when to expand additional states within an MDP while solving for a partial policy [37]. Partial policies over *actions* have even been explored in an effort to bias Monte Carlo tree search over policies online [38]. Our approach benefits from many properties enjoyed by partial policy approaches, including work on selecting which states the partial policy should consider, and the eventual construction of a complete policy as the agent visits new states during deployment. One unique benefit of partially abstract MDPs is that the partial policy is globally influenced by relevant features of the ground state space via transitions between ground states and abstract states.

III. Background

A ground MDP is represented by a tuple $M = \langle S, A, T, R \rangle$ [39]. The space of states is S. The space of actions is A. The transition function $T : S \times A \times S \rightarrow [0, 1]$ represents the probability of reaching a state $s' \in S$ after performing an action $a \in A$ in a state $s \in S$. The reward function $R : S \times A \rightarrow \mathbb{R}$ represents the immediate reward of performing an action $a \in A$ in a state $s \in S$. A solution to an MDP is a policy $\pi : S \rightarrow A$ indicating that an action $\pi(s) \in A$ should be performed in a state $s \in S$. A policy π induces a value function $V^\pi : S \rightarrow \mathbb{R}$ representing the expected discounted cumulative reward $V^\pi(s) \in \mathbb{R}$ for each state $s \in S$ given a discount factor $0 \leq \gamma < 1$. An optimal policy π^* maximizes the expected discounted cumulative reward for each state $s \in S$ by meeting the Bellman optimality equation $V^*(s) = \max_{a \in A} \sum_{s' \in S} T(s, a, s')[R(s, a, s') + \gamma V^*(s')]$.

Specifying an abstract MDP $\bar{M}$ of a ground MDP M requires two functions [12]. First, an abstraction function $\phi : S \rightarrow \bar{S}$ maps a ground state $s \in S$ to an abstract state $\bar{s} \in \bar{S}$. Second, an inverse abstraction function $\phi^{-1} : \bar{S} \rightarrow \mathcal{P}(S)$ maps an abstract state $\bar{s} \in \bar{S}$ to a set of ground states $S \subseteq \mathcal{P}(S)$. The condition $\phi(s) = \bar{s} \Leftrightarrow s \in \phi^{-1}(\bar{s})$ must hold for each ground state $s \in S$ and abstract state $\bar{s} \in \bar{S}$.

An abstract MDP is represented by a tuple $\bar{M} = \langle \bar{S}, A, \bar{T}, \bar{R} \rangle$ [12]. The space of abstract states is $\bar{S} = \{\phi(s) \mid s \in S\}$ such that a set of ground states S is abstracted by an abstraction function ϕ. The space of ground actions is A. The abstract transition function is $\bar{T}(\bar{s}, a, \bar{s}') = \sum_{s \in \phi^{-1}(\bar{s})} \psi(s) \sum_{s' \in \phi^{-1}(\bar{s}')} T(s, a, s')$. The abstract reward function is $\bar{R}(\bar{s}, a) = \sum_{s \in \phi^{-1}(\bar{s})} \psi(s) R(s, a)$. Note that a weighting function $\psi : S \rightarrow [0, 1]$ represents the probability of being in a ground state $s \in S$ in an abstract state $\phi(s) \in \bar{S}$.

IV. Partial State Abstractions

We offer a novel model, a *partially abstract MDP*, that has two levels of abstraction: a set of abstract states that each compress a set of ground states and a set of ground states that expand from a set of expanded abstract states. We offer a description of a partially abstract MDP below.

Definition 1. *A* ***partially abstract MDP****, $\tilde{M} = \langle \tilde{S}, A, \tilde{T}, \tilde{R} \rangle$, is a partially abstract version of a ground MDP $M = \langle S, A, T, R \rangle$ and an abstract MDP $\bar{M} = \langle \bar{S}, A, \bar{T}, \bar{R} \rangle$, where*

- *$\tilde{S} = \alpha \cup \beta$ is a set of* ***partially abstract states*** *with a set of ground states $\alpha = \{\phi^{-1}(\bar{s}) \mid \bar{s} \in \Gamma\}$ and a set of abstract states $\beta = \{\bar{S} \setminus \Gamma\}$ such that a set of expanded abstract states $\Gamma \subseteq \bar{S}$ is expanded by an inverse abstraction function ϕ^{-1},*
- *A is a set of ground actions,*
- *$\tilde{T} : \tilde{S} \times A \times \tilde{S} \rightarrow [0, 1]$ is a* ***partially abstract transition function*** *composed of a ground transition function T and an abstract transition function $\bar{T}$, and*
- *$\tilde{R} : \tilde{S} \times A \rightarrow \mathbb{R}$ is a* ***partially abstract reward function*** *composed of a ground reward function R and an abstract reward function $\bar{R}$.*

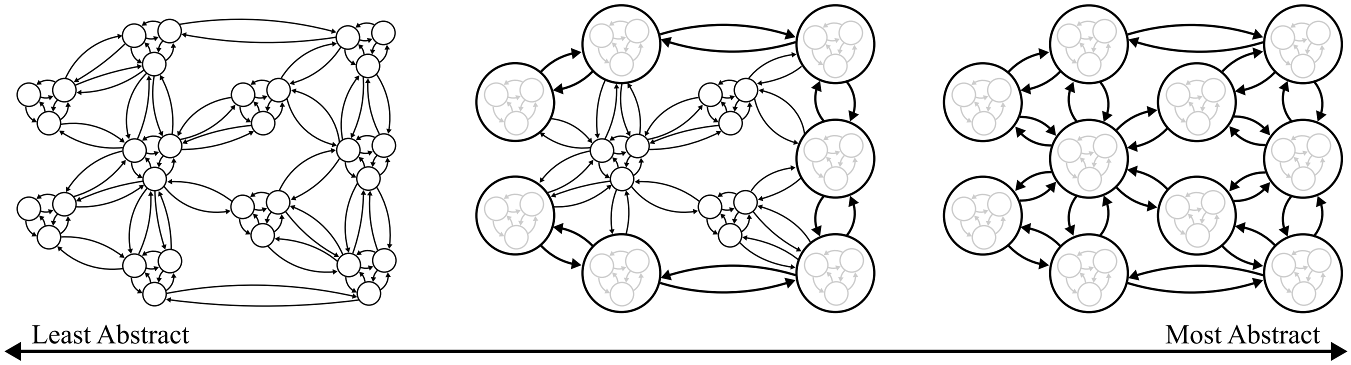

Fig. 2. An example range of state abstractions where a ground MDP is the least abstract and an abstract MDP is the most abstract. The ground MDP has ground states (*left*), the partially abstract MDP has ground states and abstract states (*center*), and the abstract MDP has abstract states (*right*). By adjusting the states of a partially abstract MDP, we can get a ground MDP if all abstract states are expanded or an abstract MDP if all ground states are compressed.

The partially abstract transition and reward functions of a partially abstract MDP are defined in the following way.

$$\tilde{T}(\tilde{s},a,\tilde{s}') = \begin{cases} T(\tilde{s},a,\tilde{s}') & \textit{if } \tilde{s}\in\alpha, \tilde{s}'\in\alpha \\ \sum_{s'\in\phi^{-1}(\tilde{s}')} T(\tilde{s},a,s') & \textit{if } \tilde{s}\in\alpha,\tilde{s}'\in\beta \\ \sum_{s\in\phi^{-1}(\tilde{s})}\psi(s)T(s,a,\tilde{s}') & \textit{if } \tilde{s}\in\beta,\tilde{s}'\in\alpha \\ \bar{T}(\tilde{s},a,\tilde{s}') & \textit{if } \tilde{s}\in\beta,\tilde{s}'\in\beta \end{cases}$$

$$\tilde{R}(\tilde{s},a) = \begin{cases} R(\tilde{s},a) & \textit{if } \tilde{s}\in\alpha \\ \bar{R}(\tilde{s},a) & \textit{if } \tilde{s}\in\beta \end{cases}$$

Figure 2 illustrates a spectrum of state abstractions that span a ground MDP, partially abstract MDPs, and an abstract MDP given an abstraction and inverse abstraction function.

A. Generality

A partially abstract MDP generalizes its ground MDP and its abstract MDP. In other words, a partially abstract MDP is a hybridization of a ground MDP and an abstract MDP. We prove that every ground MDP and every abstract MDP can naturally be represented as a partially abstract MDP below.

Remark 1. *Every ground MDP M and abstract MDP $\bar{M}$ is a partially abstract MDP $\tilde{M}$.*

Proof Sketch. Suppose a partially abstract MDP $\tilde{M}$ has a set of expanded abstract states Γ set to either $\bar{S}$ (for a ground MDP M) or $\emptyset$ (for an abstract MDP $\bar{M}$). The space of partially abstract states $\tilde{S}$ is then either a union of a set of ground states $\alpha = S$ and a set of abstract states $\beta = \emptyset$ (for a ground MDP M) or a union of a set of ground states $\alpha = \emptyset$ and a set of abstract states $\beta = \bar{S}$ (for an abstract MDP $\bar{M}$). Hence, the partially abstract transition and reward functions $\tilde{T}$ and $\tilde{R}$ reduce to either the ground functions T and R (for a ground MDP M) or the abstract functions $\bar{T}$ and $\bar{R}$ (for an abstract MDP $\bar{M}$). Every ground MDP M and abstract MDP $\bar{M}$ is therefore a partially abstract MDP $\tilde{M}$. □

B. Optimality

A partially abstract MDP can exhibit bounded optimality depending on the properties of its abstract MDP. At a high level, this section describes a well-known abstract MDP, outlines a standard solution method, and proves that a partially abstract MDP that uses this abstract MDP and solution method exhibits a lower bound on optimality.

We first consider an abstract MDP described as a *bounded parameter MDP* that is *ϵ-homogeneous* with respect to a ground MDP $M = \langle S, A, T, R\rangle$ [24]. A bounded parameter MDP is described by a tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}\rangle$. The space of states is $\mathcal{S} = \{\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_n\}$ that partitions the space of ground states $S = \mathcal{B}_1 \cup \mathcal{B}_2 \cup \cdots \cup \mathcal{B}_n$. The space of actions is $\mathcal{A} = A$. The transition function $\mathcal{T} : \mathcal{S}\times\mathcal{A}\times\mathcal{S} \to [0,1]\times[0,1]$ represents a closed form probability interval $[l_{\mathcal{T}(s,a,s')}, u_{\mathcal{T}(s,a,s')}]$ with $0 \leq l_{\mathcal{T}(s,a,s')} \leq u_{\mathcal{T}(s,a,s')} \leq 1$ of reaching a state $s' \in \mathcal{S}$ after performing an action $a \in \mathcal{A}$ in a state $s \in \mathcal{S}$. The reward function $\mathcal{R} : \mathcal{S}\times\mathcal{A} \to \mathbb{R}\times\mathbb{R}$ represents a closed form immediate reward interval $[l_{\mathcal{R}(s,a)}, u_{\mathcal{R}(s,a)}]$ with $-\infty \leq l_{\mathcal{R}(s,a)} \leq u_{\mathcal{R}(s,a)} \leq \infty$ of performing an action $a \in \mathcal{A}$ in a state $s \in \mathcal{S}$. A bounded parameter MDP $\mathcal{M}$ is ϵ-homogeneous if and only if the space of states $\mathcal{S} = \{\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_n\}$ partitions the space of ground states $S = \mathcal{B}_1 \cup \mathcal{B}_2 \cup \cdots \cup \mathcal{B}_n$ such that the conditions $|R(p) - R(q)| \leq \epsilon$ and $\sum_{r\in\mathcal{B}_j} T(p,a,r) - \sum_{r\in\mathcal{B}_j} T(q,a,r)| \leq \epsilon$ hold for each state $\mathcal{B}_i \in \mathcal{S}$, state $\mathcal{B}_j \in \mathcal{S}$, action $a \in \mathcal{A}$, ground state $p \in \mathcal{B}_i$, and ground state $q \in \mathcal{B}_i$ given a similarity threshold ϵ.

An ϵ-homogeneous bounded parameter MDP can then be solved using *interval value iteration* [40]. It first computes an interval value function $\mathcal{V} : S \to \mathbb{R}\times\mathbb{R}$ that represents a closed form value interval $[l_{\mathcal{V}(s)}, u_{\mathcal{V}(s)}]$ with $0 \leq l_{\mathcal{V}(s)} \leq u_{\mathcal{V}(s)} \leq 1$ for each ground state $s \in S$. It then computes a pessimistic policy $\pi_{pess} : S \to A$ that indicates that an action $\pi(s) \in A$ should be performed in a ground state $s \in S$ using the lower value bound $l_{\mathcal{V}}(s)$ of the interval value function $\mathcal{V}(s)$. Importantly, it has been shown that a pessimistic policy $\pi_{pess}(s)$ induces a ground value function $V(s)$ greater than or equal to the lower bound value $l_{\mathcal{V}(s)}$ of the interval value function $\mathcal{V}(s)$ for each ground state $s \in S$.

We now show that a partially abstract MDP that expands an abstract MDP based on an ϵ-homogeneous bounded-parameter MDP exhibits a lower bound on optimality below.

Proposition 1. *A partially abstract MDP $\tilde{M}$ that expands an abstract MDP $\bar{M}$ based on an ϵ-homogeneous bounded-parameter MDP $\mathcal{M}$ induces a ground value function $V(s)$ greater than or equal to the lower bound value $l_{\mathcal{V}(s)}$ of the interval value function $\mathcal{V}(s)$ for each ground state $s \in S$.*

Proof Sketch. Consider an abstract MDP $\bar{M}$ with a space of abstract states $\bar{S}$ based on an ϵ-homogeneous bounded-parameter MDP $\mathcal{M}$. Suppose the abstract MDP $\bar{M}$ expands a set of expanded abstract states $\Gamma \subseteq \bar{S}$ to get a partially abstract MDP $\tilde{M}$ with a set of partially abstract states $\tilde{S}$ that has a set of ground states $\alpha = \{\phi^{-1}(\bar{s}) \mid \bar{s} \in \Gamma\}$ and a set of abstract states $\beta = \{\bar{S} \setminus \Gamma\}$. By definition, the set of abstract states β is ϵ-homogeneous. Likewise, the set of ground states α is ϵ-homogeneous for any ϵ since each ground state can be treated as a unique abstract state. Thus, the partially abstract MDP $\tilde{M}$ induces a ground value function $V(s)$ greater than or equal to the lower bound value $l_{\mathcal{V}(s)}$ of the interval value function $\mathcal{V}(s)$ for each ground state $s \in S$. □

V. Lazy Planning and Execution

We propose a lazy algorithm that allows an autonomous system to use abstract MDPs and partially abstract MDPs for planning and execution in Algorithm 1. Initially, it builds and solves an *abstract MDP* where each abstract state *compresses* a set of ground states. Each time a new abstract state is encountered, it builds and solves a *partially abstract MDP* where the new abstract state and other informative abstract states based on an expansion strategy *expand* into a set of ground states. We discuss each phase of Algorithm 1 below.

1) Initialization: Algorithm 1 creates the ground policy by building and solving an abstract MDP offline (Lines 1-7). First, the abstract MDP is built from the ground MDP using the abstraction function and solved using the planner given the discount factor (Lines 1-2). Next, the ground policy is initialized to the abstract policy of the abstract state for each ground state of the ground MDP (Lines 3-6). Finally, the visited ground states are initialized (Line 7). The loop then starts at the first time step (Line 8).

2) Planning: Algorithm 1 updates the ground policy by building and solving a partially abstract MDP online (Lines 9-20). This phase only occurs if the current ground state is not in the visited ground states (Line 9). First, the expanded abstract states is a union of the current abstract state for the current ground state and informative abstract states generated by the ExpansionStrategy function given the ground MDP, the abstract MDP, and the current ground state (Line 10-12). Next, the ground states are computed from the expanded abstract states using the inverse abstraction function (Line 13) and the abstract states are computed from the abstract states of the abstract MDP and the expanded abstract states (Line 14). After, the partially abstract MDP is built from the ground MDP, the abstract MDP, the ground states, and the abstract states (Line 15) and solved using the planner given the discount factor (Line 16). Thereafter, if the planner does not exceed a planning time limit or encounter an error due to connectivity loss or node failure, the ground policy is updated to the partially abstract policy for each ground state of the current abstract state (Lines 17-19). Finally, the visited ground states are updated with the ground states of the current abstract state (Line 20).

Algorithm 1: A lazy algorithm for planning and execution in autonomous systems.

```
Input: A ground MDP $M = \langle S, A, T, R \rangle$, a horizon $T$, a
       discount factor $\gamma$, an abstraction function $\phi$, an
       inverse abstraction function $\phi^{-1}$, an initial ground
       state $s \in S$, and a planner $\Xi$
1  $\bar{M} \leftarrow$ BuildAMDP($M$, $\phi$)
2  $\bar{\pi} \leftarrow \Xi$.Solve($\bar{M}$, $\gamma$)
3  $\pi \leftarrow \varnothing$
4  for $s'$ in $S$ do
5      $\bar{s}' \leftarrow \phi(s')$
6      $\pi(s') \leftarrow \bar{\pi}(\bar{s}')$
7  $\Lambda \leftarrow \emptyset$
8  while $t \to T$ do
9      if $s$ not in $\Lambda$ then
10         $\bar{s} \leftarrow \phi(s)$
11         $\rho \leftarrow$ ExpansionStrategy($M$, $\bar{M}$, $s$)
12         $\Gamma \leftarrow \rho \cup \{\bar{s}\}$
13         $\alpha \leftarrow \{\phi^{-1}(\bar{s}) \mid \bar{s} \in \Gamma\}$
14         $\beta \leftarrow \bar{S} \setminus \Gamma$
15         $\tilde{M} \leftarrow$ BuildPAMDP($M$, $\bar{M}$, $\alpha$, $\beta$)
16         $\tilde{\pi} \leftarrow \Xi$.Solve($\tilde{M}$, $\gamma$)
17         if $\tilde{\pi}$ is not $\emptyset$ then
18             for $s'$ in $\phi^{-1}(\bar{s})$ do
19                 $\pi(s') \leftarrow \tilde{\pi}(s')$
20             $\Lambda \leftarrow \Lambda \cup \phi^{-1}(\bar{s})$
21     $a \leftarrow \pi(s)$
22     $s \leftarrow$ Perform($a$)
```

3) Execution: Algorithm 1 performs an action that follows the ground policy (Lines 21-22). First, the current ground action is calculated from the ground policy given the current ground state (Line 21). Next, the current state is generated by performing the current action (Line 22). The loop then continues to the next time step (Line 8).

At each time step, the algorithm must determine the expanded abstract states for the partially abstract MDP. Minimally, the algorithm selects the abstract state of the current ground state for expansion on Line 10. However, the algorithm also selects informative abstract states for expansion using the ExpansionStrategy function on Line 11. Ideally, the expanded abstract states, namely the current abstract state and the informative abstract states, must be set in a way that produces effective behavior in real time.

Any planner can be used by the algorithm. As the main requirement, the planner must have an MDP as input and a policy as output. In fact, it can use any exact planner based on dynamic programming, such as value or policy iteration [39], heuristic search, such as LAO* [41], or linear programming [42]. It can even use approximate planners based on real-time dynamic programming [43] or determinization [34].

The algorithm offers three desirable properties for autonomous systems. First, it is *lazy* as it only computes the ground policy for the ground states of visited abstract states. Second, it is *attentional* since it only solves partially abstract MDPs that expand the new abstract state and informative abstract states instead of the ground MDP. This retains small relevant regions but ignores large irrelevant regions of the ground MDP, which increases the accuracy while decreasing the complexity of planning. Third, it is *anytime* because it uses a default ground policy from the abstract MDP if the planner either exceeds some planning time limit or encounters an error due to connectivity loss or node failure.

VI. Earth Observation

We now turn to an application of our approach to an Earth observation domain [44]. A satellite orbiting Earth must take photos of various points of interests P with different weather levels W that change stochastically. The satellite starts at longitude $x \in X$ with its camera focused at latitude $y \in Y$. The satellite can then either do *no operation*, shift its camera *north* to the northern latitude $y' \in Y$, shift its camera *south* to the southern latitude $y'' \in Y$, or take a *photo* of the rectangular region of Earth at latitude and longitude $y \in Y$ and $x \in X$ with a photo quality based on the weather level $w \in W$. Concurrently, the satellite moves east to the eastern longitude $x' \in X$. This repeats indefinitely. We define the ground, abstract, and partially abstract MDPs below.

1) Ground MDP: We use a ground MDP $M = \langle S, A, T, R \rangle$ that represents the Earth observation problem. The set of states $S = X \times Y \times W^{|P|}$ is a cross product of a set of longitudes $X = \{x_1, x_2, \ldots, x_{\ell_X}\}$ that represents the position of the satellite, a set of latitudes $Y = \{y_1, y_2, \ldots, y_{\ell_Y}\}$ that represents the focus of its camera, a set of weather levels $W = \{w_1, w_2, \ldots, w_{\ell_W}\}$, and a set of points of interests $P = \{p_1, p_2, \ldots, p_{\ell_P}\}$ for the state factor sizes ℓ_X, ℓ_Y, ℓ_W, and ℓ_P. The set of actions $A = \{\otimes, \Uparrow, \Downarrow, \odot\}$ has a *no operation* action $\otimes$, a *north* action $\Uparrow$, a *south* action $\Downarrow$, and a *photo* action $\odot$. The transition function $T : S \times A \times S \rightarrow [0, 1]$ reflects the probability that the weather level $w \in W$ of each point of interest $p \in P$ changes as each action moves the satellite to the next eastern longitude $x' \in X$, while the *north* and *south* actions $\Uparrow$ and $\Downarrow$ change the focus of the camera north and south to the northern and southern latitudes $y' \in Y$ and $y'' \in Y$ and the *no operation* and *photo* actions $\otimes$ and $\odot$ do not change latitude $y \in Y$. The reward function $R : S \times A \rightarrow \mathbb{R}$ reflects the reward gained after performing the *photo* action $\odot$ at any latitude $y \in Y$ and longitude $x \in X$ at a point of interest $p \in P$ with a photo quality based on the weather level $w \in W$ and a nil reward for any other latitude $y' \in Y$ and longitude $x' \in X$.

2) Abstract MDP: We use an abstract MDP $\bar{M} = \langle \bar{S}, A, \bar{T}, \bar{R} \rangle$ that partitions both the latitudes and longitudes rectangularly and the weather levels contiguously into different abstract states. The set of abstract states $\bar{S} = \bar{X} \times \bar{Y} \times \bar{W}^{|P|}$ is a cross product of a set of abstract longitudes $\bar{X} = \{\bar{x}_{\lfloor 1/\bar{\ell}_X \rfloor}, \bar{x}_{\lfloor 2/\bar{\ell}_X \rfloor}, \ldots, \bar{x}_{\lfloor \ell_X/\bar{\ell}_X \rfloor}\}$ that represents the abstract position of the satellite, a set of abstract latitudes $\bar{Y} = \{\bar{Y}_{\lfloor 1/\bar{\ell}_Y \rfloor}, \bar{Y}_{\lfloor 2/\bar{\ell}_Y \rfloor}, \ldots, \bar{Y}_{\lfloor \ell_Y/\bar{\ell}_Y \rfloor}\}$ that represents the abstract focus of its camera, a set of abstract weather levels $\bar{W} = \{\bar{w}_{\lfloor 1/\bar{\ell}_W \rfloor}, \bar{w}_{\lfloor 2/\bar{\ell}_W \rfloor}, \ldots, \bar{w}_{\lfloor \ell_W/\bar{\ell}_W \rfloor}\}\}$, and a set of points of interests P for the abstract partition sizes $\bar{\ell}_X$, $\bar{\ell}_Y$, and $\bar{\ell}_W$ given an abstraction function ϕ. The attributes A, $\tilde{T}$, and $\tilde{R}$ follow directly from the abstract MDP definition.

TABLE I

The Earth observation problem simulation parameters.

ID	ℓ_X	ℓ_Y	ℓ_W	ℓ_P	$\lvert S \rvert$	$\bar{\ell}_X$	$\bar{\ell}_Y$	$\bar{\ell}_W$	$\lvert \bar{S} \rvert$	${\lvert \bar{S} \rvert}/{\lvert S \rvert}$	RAND$_P$	RAND$_W$	τ
A	6	3	4	2	288	3	3	2	8	0.028	20	5	100
B	6	3	4	3	1152	3	3	2	16	0.014	20	5	100
C	6	3	4	4	4608	3	3	2	32	0.007	20	5	100
D	12	6	4	2	1152	3	3	2	32	0.028	20	5	100
E	12	6	4	3	4608	3	3	2	64	0.014	20	5	100
F	12	6	4	4	18432	3	3	2	128	0.007	5	5	25
G	24	12	4	2	4608	3	3	2	128	0.028	20	5	100
H	24	12	4	3	18432	3	3	2	256	0.014	5	5	25
I	24	12	4	4	73728	3	3	2	512	0.007	5	2	10
J	24	18	4	2	6912	3	3	2	192	0.028	10	5	50
K	24	18	4	3	27648	3	3	2	384	0.014	5	5	25
L	24	18	4	4	110592	3	3	2	768	0.007	5	2	10

3) Partially Abstract MDP: We use a partially abstract MDP $\langle \tilde{S}, A, \tilde{T}, \tilde{R} \rangle$ that expands the current abstract state and all abstract states generated by a given expansion strategy. The set of partially abstract states $\tilde{S} = \alpha \cup \beta$ is a union of a set of ground states $\alpha = \{\phi^{-1}(\bar{s}) \mid \bar{s} \in \Gamma\}$ and a set of abstract states $\beta = \{\bar{S} \setminus \Gamma\}$ such that the set of expanded abstract states $\Gamma \subseteq \bar{S}$ contains the current abstract state $\bar{s} \in \bar{S}$ and other informative abstract states generated by an ExpansionStrategy function given an inverse abstraction function ϕ^{-1}. The attributes A, $\tilde{T}$, and $\tilde{R}$ follow directly from the definition of a partially abstract MDP.

VII. Experiments

We show that our approach is near-optimal and scalable by comparing it to a standard approach that solves a ground MDP directly across a set of Earth observation problems.

Table I summarizes the parameters of the Earth observation problems. For each problem, we perform τ random trials of 5000 simulation steps that are initialized with random points of interest RAND$_P$ and a random weather level process RAND$_W$. These initializations are held constant across each approach. In the first time step of each trial, the satellite starts at a state with longitude 0 and latitude 0. During each step, the satellite performs an action, gains a reward, and transitions to a successor state. All trials were run on a 3.7 GHz quad-core CPU with 32 GB 1333 MHz DDR3 RAM.

A ground MDP and an abstract MDP are specified for each problem. Each ground MDP uses parameters ℓ_X, ℓ_Y, ℓ_W, and ℓ_P that correspond to the state factor size of the

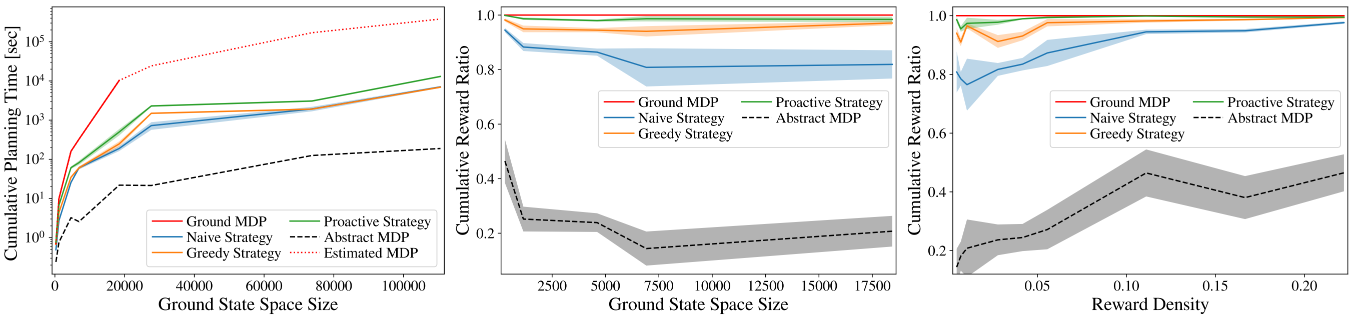

Fig. 3. The result plots (a), (b), and (c) for the Earth observation domain. In (a), a *log scale* is used, the *dotted line* is a projection for any ground MDP that is infeasible to solve, and the *dashed line* is the cumulative planning time for each abstract MDP. *Shaded* regions denote a confidence interval of 95%.

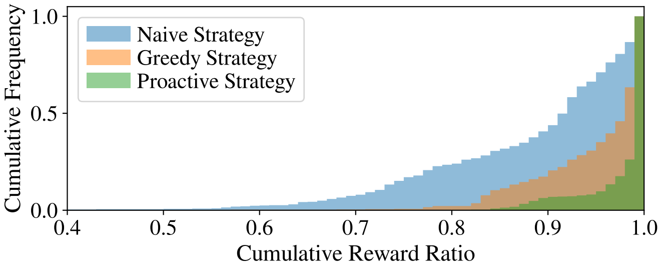

Fig. 4. A cumulative histogram over the cumulative reward ratios.

longitudes, latitudes, weather levels, and points of interest. The state space size is $|S| = \ell_X \times \ell_Y \times \ell_W^{\ell_P}$. Each abstract MDP uses parameters $\bar{\ell}_X$, $\bar{\ell}_Y$, and $\bar{\ell}_W$ that correspond to the abstract partition size for the abstract longitudes, abstract latitudes, and abstract weather levels. The abstract state space size is $|\bar{S}| = \bar{\ell}_X \times \bar{\ell}_Y \times \bar{\ell}_W^{\ell_P}$. The compression ratio is ${}^{|\bar{S}|}/_{|S|}$.

We evaluate both approaches on each Earth observation problem along two metrics. First, we measure the *cumulative planning time*: the total planning time required by an agent to solve all partially abstract MDPs for our approach and the ground MDP for the standard approach. The planning time required to solve each abstract MDP is measured separately. Second, we measure the *cumulative reward ratio*: an approximation ratio between 0 and 1 computed as the total reward gained by the agent using an approach divided by the reward gained by an agent that behaves optimally. The value of the policies generated by each approach cannot be measured directly using policy evaluation because state abstractions can change the scale of the values.

Our approach expands the current abstract state and uses the three EXPANSIONSTRATEGY functions shown in Figure 1. NAIVESTRATEGY expands no informative abstract states. GREEDYSTRATEGY expands any informative abstract state with the current abstract weather level that contains a point of interest within 1 abstract state of the current abstract state. PROACTIVESTRATEGY expands any informative abstract state with the current abstract weather level contained by the overlapping rectangles that are formed by the current abstract state and any abstract state that contains a point of interest within 2 abstract states of the current abstract state.

Our results are shown in four figures. Figure 3(a) shows how the *mean* cumulative planning time over all trials increases with the ground state space size. Our approach incurs initial overhead but yields speedups of multiple orders of magnitude for large ground state spaces. Figure 3(b) shows how the *mean* cumulative reward ratio over all trials varies with the ground state space size while Figure 3(c) shows how the *mean* cumulative reward ratio over all trials increases with the reward density (the fraction of ground states that generate positive reward). Both figures indicate near-optimal performance on average. We suspect the variation among trials is caused by the suitability of expansion strategies to different geometries of points of interest. Figure 4 shows the cumulative frequency for each *mean* cumulative reward ratio over all trials. The short tails for greedy and proactive strategies suggest high performance and low variance.

There are three key takeaways. First, our approach with the greedy and proactive expansion strategies performs nearly as well as the standard approach in a fraction of the time. Next, our approach offers a trade-off between solution quality and computation time based on the choice of expansion strategy and abstraction strategy. In fact, *less* aggressive expansion strategies that expand *fewer* abstract states or *more* aggressive abstraction functions with *larger* abstract partitions can be used if planning time constraints are severe. This is consistent with the idea that partially abstract MDPs span a continuum of abstractions. Finally, our approach enables an agent to only wait until the abstract MDP and the first partially abstract MDP is built and solved while the standard approach must wait until the ground MDP is built and solved. This means that the planning time of our approach can not only be amortized over the deployment of the agent but also minimized through concurrent planning and execution.

VIII. CONCLUSION

We offer a novel model, a partially abstract MDP, that reduces the complexity of a problem while maintaining high quality solutions by simultaneously using different levels of abstraction. We also provide theoretical results on the generality of partially abstract MDPs and their bounded optimality under certain conditions and propose a lazy algorithm that enables autonomous systems to leverage partially abstract MDPs during operation. Finally, we demonstrate the efficiency and accuracy of our approach on an Earth observation domain. Future work will explore sophisticated strategies for state abstraction and state expansion.

References

[1] M. A. Goodrich, B. S. Morse, D. Gerhardt, J. L. Cooper, M. Quigley, J. A. Adams, and C. Humphrey, "Supporting wilderness search and rescue using a camera-equipped mini UAV," *Journal of Field Robotics*, 2008.

[2] L. Pineda, T. Takahashi, H.-T. Jung, S. Zilberstein, and R. Grupen, "Continual planning for search and rescue robots," in *IEEE/RAS International Conference on Humanoid Robots*. IEEE, 2015, pp. 243–248.

[3] J. F. Mustard, D. Beaty, and D. Bass, "Mars 2020 science rover: Science goals and mission concept," in *AAS/Division for Planetary Sciences Meeting Abstracts*, vol. 45, 2013.

[4] Y. Gao and S. Chien, "Review on space robotics: Toward top-level science through space exploration," *Science Robotics*, vol. 2, no. 7, 2017.

[5] J. Svegliato, K. H. Wray, S. J. Witwicki, J. Biswas, and S. Zilberstein, "Belief space metareasoning for exception recovery," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2019.

[6] C. Basich, J. Svegliato, K. H. Wray, S. Witwicki, J. Biswas, and S. Zilberstein, "Learning to optimize autonomy in competence-aware systems," in *19th International Conference on Autonomous Agents and Multiagent Systems*, 2020.

[7] C. Basich, J. Svegliato, S. Zilberstein, K. H. Wray, and S. J. Witwicki, "Improving competence for reliable autonomy," in *ECAI Workshop on Agents and Robots for Reliable Engineered Autonomy*, 2021.

[8] J. Svegliato, S. B. Nashed, and S. Zilberstein, "Ethically compliant sequential decision making," in *35th AAAI Conference on Artificial Intelligence*, 2021.

[9] K. H. Wray, S. J. Witwicki, and S. Zilberstein, "Online decision-making for scalable autonomous systems," in *International Joint Conference on Artificial Intelligence*, 2017.

[10] S. B. Nashed, D. M. Ilstrup, and J. Biswas, "Localization under topological uncertainty for lane identification of autonomous vehicles," in *IEEE International Conference on Robotics and Automation*, 2018.

[11] S. Parr, I. Khatri, J. Svegliato, and S. Zilberstein, "Agent-aware state estimation: Effective traffic light classification for autonomous vehicles," in *lo 2020 Workshop on Sensing, Estimating and Understanding the Dynamic World*, 2020.

[12] L. Li, T. J. Walsh, and M. L. Littman, "Towards a unified theory of state abstraction for MDPs," in *International Symposium on Artificial Intelligence and Mathematics*, 2006.

[13] M. Brucato, J. F. Beltran, A. Abouzied, and A. Meliou, "Scalable package queries in relational database systems," *International Conference on Very Large Data Bases*, vol. 9, no. 7, pp. 576–587, 2016.

[14] D. P. Bertsekas, "Approximate policy iteration: A survey and some new methods," *Journal of Control Theory and Applications*, vol. 9, no. 3, pp. 310–335, 2011.

[15] W. B. Powell, "Perspectives of approximate dynamic programming," *Annals of Operations Research*, vol. 241, no. 1-2, pp. 319–356, 2016.

[16] C. Guestrin, D. Koller, R. Parr, and S. Venkataraman, "Efficient solution algorithms for factored MDPs," *Journal of Artificial Intelligence Research*, vol. 19, pp. 399–468, 2003.

[17] M. Petrik and S. Zilberstein, "Constraint relaxation in approximate linear programs," in *Annual International Conference on Machine Learning*, 2009, pp. 809–816.

[18] P. Poupart, A. Malhotra, P. Pei, K.-E. Kim, B. Goh, and M. Bowling, "Approximate linear programming for constrained partially observable Markov decision processes." in *AAAI Conference on Artificial Intelligence*, vol. 1, 2015, pp. 3342–3348.

[19] A. Malek, Y. Abbasi-Yadkori, and P. Bartlett, "Linear programming for large-scale Markov decision problems," in *International Conference on Machine Learning*, 2014, pp. 496–504.

[20] T. Smith and R. Simmons, "Focused real-time dynamic programming for MDPs: Squeezing more out of a heuristic," in *AAAI Conference on Artificial Intelligence*, 2006, pp. 1227–1232.

[22] R. Givan, T. Dean, and M. Greig, "Equivalence notions and model minimization in Markov decision processes," *Artificial Intelligence*, vol. 147, no. 1-2, pp. 163–223, 2003.

[21] L. E. Pineda, K. H. Wray, and S. Zilberstein, "Fast SSP solvers using short-sighted labeling," in *AAAI Conference on Artificial Intelligence*, 2017.

[23] N. Ferns, P. Panangaden, and D. Precup, "Metrics for finite Markov decision processes." in *Conference on Uncertainty in Artificial Intelligence*, vol. 4, 2004, pp. 162–169.

[24] T. L. Dean, R. Givan, and S. Leach, "Model reduction techniques for computing approximately optimal solutions for Markov decision processes," *arXiv preprint arXiv:1302.1533*, 1997.

[25] B. Ravindran and A. G. Barto, "Model minimization in hierarchical reinforcement learning," in *International Symposium on Abstraction, Reformulation, and Approximation*. Springer, 2002, pp. 196–211.

[26] O. Biza and R. Platt, "Online abstraction with MDP homomorphisms for deep learning," *arXiv preprint arXiv:1811.12929*, 2018.

[27] H. Yu and D. P. Bertsekas, "Basis function adaptation methods for cost approximation in MDP," in *IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning*. IEEE, 2009, pp. 74–81.

[28] T. G. Dietterich, "Hierarchical reinforcement learning with the MAXQ value function decomposition," *Journal of Artificial Intelligence Research*, vol. 13, pp. 227–303, 2000.

[29] D. Ruiken, T. Q. Liu, T. Takahashi, and R. A. Grupen, "Reconfigurable tasks in belief-space planning," in *IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2016, pp. 1257–1263.

[30] L. Li and M. L. Littman, "Lazy approximation for solving continuous finite-horizon MDPs," in *AAAI Conference on Artificial Intelligence*, vol. 5, 2005, pp. 1175–1180.

[31] M. Hauskrecht, N. Meuleau, L. P. Kaelbling, T. L. Dean, and C. Boutilier, "Hierarchical solution of Markov decision processes using macro-actions," *arXiv preprint arXiv:1301.7381*, 2013.

[32] S. Sreedharan, S. Srivastava, and S. Kambhampati, "TLdR: Policy summarization for factored SSP problems using temporal abstractions," in *International Conference on Automated Planning and Scheduling*, vol. 30, 2020, pp. 272–280.

[33] L. E. Pineda and S. Zilberstein, "Planning under uncertainty using reduced models: Revisiting determinization," in *24th International Conference on Automated Planning and Scheduling*. Citeseer, 2014.

[34] S. Saisubramanian and S. Zilberstein, "Adaptive outcome selection for planning with reduced models," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2019, pp. 1655–1660.

[35] S. W. Yoon, A. Fern, and R. Givan, "FF-Replan: A baseline for probabilistic planning." in *International Conference on Automated Planning and Scheduling*, vol. 7, 2007, pp. 352–359.

[36] L. E. Pineda and S. Zilberstein, "Soft labeling in stochastic shortest path problems." in *International Conference on Autonomous Agents and Multiagent Systems*, 2019, pp. 467–475.

[37] G. Alexander, A. Raja, and D. J. Musliner, "Controlling deliberation in a Markov decision process-based agent," in *International Joint Conference on Autonomous Agents and Multiagent Systems*. Citeseer, 2008, pp. 461–468.

[38] J. Pinto and A. Fern, "Learning partial policies to speedup MDP tree search." in *Conference on Uncertainty in Artificial Intelligence*. Citeseer, 2014, pp. 672–681.

[39] R. Bellman, "Dynamic programming," *Science*, 1966.

[40] R. Givan, S. Leach, and T. Dean, "Bounded-parameter Markov decision processes," *Artificial Intelligence*, vol. 122, no. 1-2, pp. 71–109, 2000.

[41] E. A. Hansen and S. Zilberstein, "LAO*: A heuristic search algorithm that finds solutions with loops," *Artificial Intelligence*, vol. 129, no. 1-2, pp. 35–62, 2001.

[42] A. S. Manne, "Linear programming and sequential decisions," *Management Science*, 1960.

[43] A. G. Barto, S. J. Bradtke, and S. P. Singh, "Learning to act using real-time dynamic programming," *Artificial intelligence*, vol. 72, no. 1-2, pp. 81–138, 1995.

[44] A. Hertle, C. Dornhege, T. Keller, R. Mattmüller, M. Ortlieb, and B. Nebel, "An experimental comparison of classical, fond and probabilistic planning," in *Joint German/Austrian Conference on Artificial Intelligence*. Springer, 2014, pp. 297–308.