Anytime-Constrained Reinforcement Learning

Jeremy McMahan **Xiaojin Zhu**
University of Wisconsin-Madison

Abstract

We introduce and study constrained Markov Decision Processes (cMDPs) with anytime constraints. An anytime constraint requires the agent to never violate its budget at any point in time, almost surely. Although Markovian policies are no longer sufficient, we show that there exist optimal deterministic policies augmented with cumulative costs. In fact, we present a fixed-parameter tractable reduction from anytime-constrained cMDPs to unconstrained MDPs. Our reduction yields planning and learning algorithms that are time and sample-efficient for tabular cMDPs so long as the precision of the costs is logarithmic in the size of the cMDP. However, we also show that computing non-trivial approximately optimal policies is NP-hard in general. To circumvent this bottleneck, we design provable approximation algorithms that efficiently compute or learn an arbitrarily accurate approximately feasible policy with optimal value so long as the maximum supported cost is bounded by a polynomial in the cMDP or the absolute budget. Given our hardness results, our approximation guarantees are the best possible under worst-case analysis.

1 INTRODUCTION

Suppose M is a constrained Markov Decision Process (cMDP). An *anytime constraint* requires that cost accumulated by the agent's policy π is within the budget at any time, almost surely: $\mathbb{P}^{\pi}_{M}\left[\forall k \in [H],\ \sum_{t=1}^{k} c_t \leq B\right] = 1$. If Π_M denotes the set of policies that respect the anytime constraints, then a solution to the anytime-constrained cMDP is a policy $\pi^* \in \arg\max_{\pi \in \Pi_M} V^{\pi}_{M}$. For example, consider planning a minimum-time route for an autonomous vehicle to travel from one city to another. Besides time, there are other important considerations including (Pan et al., 2017) (1) the route does not exhaust the vehicle's fuel, and (2) the route is safe. We can model (1) by defining the fuel consumed traveling a road to be its cost and the tank capacity to be the budget. Refueling stations are captured using negative costs. We can model many safety considerations (2) similarly.

Since nearly every modern system is constrained in some way, one key step to modeling more realistic domains with MDPs is allowing constraints. To address this, a rich literature of constrained reinforcement learning (CRL) has been developed, almost exclusively focusing on expectation constraints (Altman, 1999; Achiam et al., 2017; Borkar, 2005) or high-probability (chance) constraints (Charnes et al., 1958; Ono et al., 2015; Chow et al., 2018). However, in many applications, especially where safety is concerned or resources are consumed, anytime constraints are more natural. An agent would not be reassured by the overall expected or probable safety of a policy when it is faced with a reality or intermediate time where it is harmed. For instance, our car cannot run on expected or future gas, and must satisfy the B budget at any time along the route. Similarly, in goal-directed RL (Hou et al., 2014; Melo Moreira et al., 2021; Bertsekas and Tsitsiklis, 1991), the goal must be achieved; maximizing reward is only a secondary concern. These issues are especially crucial in medical applications (Coronato et al., 2020; Paragliola et al., 2018; Kolesar, 1970), disaster relief scenarios (Fan et al., 2021; Wu et al., 2019; Tsai et al., 2019), and resource management (Mao et al., 2016; Li et al., 2018; Peng and Shen, 2021; Bhatia et al., 2021).

Anytime constraints are natural, but introduce a plethora of new challenges. Traditional Markovian and history-dependent policies are rarely feasible and can be arbitrarily suboptimal; the cost history must also be considered. Naively using backward induction

Proceedings of the 27th International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238.

to compute an optimal cost-history-dependent policy is possible in principle for tabular cost distributions, but the time needed to compute the policy and the memory needed to store the policy would be super-exponential. Since the optimal solution value is a discontinuous function of the costs, using standard CRL approaches like linear programming is also impossible. In fact, not only is computing an optimal policy NP-hard but computing any policy whose value is approximately optimal is also NP-hard when at least two constraints are present.

Known works fail to solve anytime-constrained cMDPs. Expectation-constrained approaches (Altman, 1999; Paternain et al., 2019; Ding et al., 2020; Kalagarla et al., 2021; Efroni et al., 2020) and chance-constrained approaches (Ono et al., 2015; Xu and Mannor, 2011; Chow et al., 2018; Mowbray et al., 2022) yield policies that arbitrarily violate an anytime constraint. This observation extends to nearly every known setting: Knapsack constraints (Brantley et al., 2020; Cheung, 2019; Chen et al., 2021), risk constraints (Borkar and Jain, 2014; Chow et al., 2018), risk sensitivity (Yu et al., 1998; Hou et al., 2014; Steinmetz et al., 2016; Melo Moreira et al., 2021), quantile constraints (Jung et al., 2022; Yang et al., 2021), and instantaneous constraints (Li et al., 2021; Fisac et al., 2019; Gros et al., 2020). If we were instead to use these models with a smaller budget to ensure feasibility, the resultant policy could be arbitrarily suboptimal if any policy would be produced at all. Dangerous-state (Roderick et al., 2021; Thomas et al., 2021) and almost-sure (Castellano et al., 2022) constraints can be seen as a special case of our model with binary costs. However, their techniques do not generalize to our more complex setting. Moreover, absent expectation constraints, none of these approaches are known to admit polynomial time planning or learning algorithms.

Our Contributions. We present the first formal study of anytime-constrained cMDPs. Although traditional policies do not suffice, we show that deterministic augmented policies are always optimal. In fact, an optimal policy can be computed by solving an unconstrained, augmented MDP using any standard RL planning or learning algorithm. Using the intuition of safe exploration and an atypical forward induction, we derive an augmented state space rich enough to capture optimal policies without being prohibitively large. To understand the resultant augmented policies, we design new machinery requiring a combination of backward and forward induction to argue about optimality and feasibility. Overall, we show our reduction to standard RL is *fixed-parameter tractable* (FPT) (Downey and Fellows, 2012) in the cost precision when the cMDP and cost distribution are tabular. In particular, as long as the cost precision is logarithmic in the size of the cMDP, our planning (learning) algorithms are polynomial time (sample complexity), and the produced optimal policy can be stored with polynomial space.

Since we show computing any non-trivial approximately-optimal policy is NP-hard, we turn to approximate feasibility for the general case. For any $\epsilon > 0$, we consider additive and relative approximate policies that accumulate cost at most $B + \epsilon$ and $B(1+\epsilon)$ anytime, respectively. [1] Rather than consider every cumulative cost induced by safe exploration, our approximation scheme inductively accumulates and projects the costs onto a smaller space. Following the principle of optimism, the approximate cost is constructed to be an underestimate to guarantee optimal value. Our approach yields planning (learning) algorithms that produce optimal value, and approximately feasible policies in polynomial time (sample complexity) for any, possibly non-tabular, cost distribution whose maximum supported cost is bounded by a polynomial in the cMDP or by the absolute budget. Given our hardness results, this is the best possible approximation guarantee one could hope for under worst-case analysis. We also extend our methods to handle different budgets per time, general almost-sure constraints, and infinite discounting.

1.1 Related Work.

Knapsack Constraints. The knapsack-constrained frameworks (Brantley et al., 2020; Cheung, 2019; Chen et al., 2021) were developed to capture constraints on the learning process similar to bandits with knapsacks (Badanidiyuru et al., 2013). Brantley et al. (2020) and Cheung (2019) both constrain the total cost violation that can be produced during training. On the other hand, Chen et al. (2021) introduces the RLwK framework that constrains the total cost used per episode. In RLwK, each episode terminates when the agent violates the budget for that episode. In all of these models, the environment is given the ability to terminate the process early; the final policy produced after learning need not satisfy any kind of constraint. In fact, the agent still keeps the reward it accumulated before violation, the agent is incentivized to choose unsafe actions in order to maximize its reward. Thus, such methods produce infeasible policies for our anytime constraints regardless of the budget they are given.

[1] Critically, we can also ensure strict budget B feasibility but with a weaker value guarantee. See section 4.1.

Almost Sure Constraints. Performing RL while avoiding dangerous states (Roderick et al., 2021; Thomas et al., 2021; Gu et al., 2023) can be seen as a special case of both anytime and expectation constraints with binary costs and budget 0. However, these works require non-trivial assumptions, and being a special case of expectation constraints implies their techniques cannot solve our general setting. Similarly, Castellano et al. (2022) introduced almost sure constraints with binary costs, which can be seen as a special case of anytime constraints. However, they focus on computing minimal budgets, which need not lead to efficient solutions in general since the problem is NP-hard even with a budget of 1. Lastly, the infinite time-average case with almost sure constraints has been thoroughly studied (Ross and Varadarajan, 1989). Since we focus on finite-horizon and discounted settings, our policies would always have a time-average cost of 0 and so those methods cannot produce policies that are feasible for anytime constraints.

2 ANYTIME CONSTRAINTS

Constrained Markov Decision Processes. A (tabular, finite-horizon) Constrained Markov Decision Process is a tuple $M = (\mathcal{S}, \mathcal{A}, P, R, H, s_0, C, B)$, where (i) $\mathcal{S}$ is a finite set of states, (ii) $\mathcal{A}$ is a finite set of actions, (iii) $P_h(s,a) \in \Delta(S)$ is the transition distribution, (iv) $R_h(s,a) \in \Delta(\mathbb{R})$ is the reward distribution, (v) $H \in \mathbb{N}$ is the finite time horizon, (vi) $s_0 \in \mathcal{S}$ is the initial state, (vii) $C_h(s,a) \in \Delta(\mathbb{R}^d)$ is the cost distribution, and (viii) $B \in \mathbb{R}^d$ is the budget vector. Here, $d \in \mathbb{N}$ denotes the number of constraints. We overload notation by letting $C_h(s,a)$ denote both the cost distribution and its support. We also let $r_h(s,a) = \mathbb{E}[R_h(s,a)]$ denote the expected reward. Lastly, we let $S := |\mathcal{S}|$, $A := |\mathcal{A}|$, $[H] := \{1, \dots, H\}$, and $|M|$ be the description size of the cMDP.

Interaction Protocol. A complete history with costs takes the form $\tau = (s_1, a_1, c_1, \dots, s_H, a_H, c_H, s_{H+1})$, where $s_h \in \mathcal{S}$ denotes M's state at time h, $a_h \in \mathcal{A}$ denotes the agent's chosen action at time h, and $c_h \in C_h(s_h, a_h)$ denotes the cost incurred at time h. We let $\bar{c}_h := \sum_{t=1}^{h-1} c_t$ denote the cumulative cost up to (but not including) time h. Also, we denote by $\tau_h = (s_1, a_1, c_1, \dots, s_h)$ the partial history up to time h and denote by $\mathcal{H}_h$ the set of partial histories up to time h. The agent interacts with M using a policy $\pi = (\pi_h)_{h=1}^H$, where $\pi_h : \mathcal{H}_h \to \Delta(\mathcal{A})$ specifies how the agent chooses actions at time h given a partial history.

The agent starts at state s_0 with partial history $\tau_1 = (s_0)$. For any $h \in [H]$, the agent chooses an action $a_h \sim \pi_h(\tau_h)$. Afterward, the agent receives reward $r_h \sim R_h(s_h, a_h)$ and cost $c_h \sim C_h(s_h, a_h)$. Then, M transitions to state $s_{h+1} \sim P_h(s_h, a_h)$ and the history is updated to $\tau_{h+1} = (\tau_h, a_h, c_h, s_{h+1})$. This process is repeated for H steps total; the interaction ends once s_{H+1} is reached.

Objective. The agent's goal is to compute a π^* that is a solution to the following optimization problem:

$$\begin{aligned} &\max_{\pi} \mathbb{E}_M^\pi \left[\sum_{h=1}^{H} r_h(s_h, a_h) \right] \\ &\text{s.t. } \mathbb{P}_M^\pi \left[\forall t \in [H], \sum_{h=1}^{t} c_h \leq B \right] = 1. \end{aligned} \tag{ANY}$$

Here, $\mathbb{P}_M^\pi$ denotes the probability law over histories induced from the interaction of π with M, and $\mathbb{E}_M^\pi$ denotes the expectation with respect to this law. We let $V^\pi := \mathbb{E}_M^\pi \left[\sum_{t=1}^{H} r_t(s_t, a_t) \right]$ denote the value of a policy π, $\Pi_M := \left\{ \pi \mid \mathbb{P}_M^\pi \left[\forall k \in [H], \sum_{t=1}^{k} c_t \leq B \right] = 1 \right\}$ denote the set of feasible policies, and $V^* := \max_{\pi \in \Pi_M} V^\pi$ denote the optimal solution value. [2]. If there are no feasible policies, $V^* := -\infty$ by convention.

Optimal Solutions. It is well-known that expectation-constrained cMDPs always admit a randomized Markovian policy (Altman, 1999). However, under anytime constraints, feasible policies that do not remember the cumulative cost can be arbitrarily suboptimal. The intuition is that without knowing the cumulative cost, a policy must either play it too safe and suffer small value or risk an action that violates the constraint.

Proposition 1. *Any class of policies that excludes the full cost history is suboptimal for anytime-constrained cMDPs. In particular, Markovian policies can be arbitrarily suboptimal even for cMDPs with $S = 1$ and $A = H = 2$.*

Corollary 1. *(Approximately) optimal policies for cMDPs with expectation constraints, chance constraints, or their variants can arbitrarily violate an anytime constraint. Furthermore, (approximately) optimal policies for a cMDP defined by a smaller budget to achieve feasibility can be arbitrarily suboptimal.*

Although using past frameworks out of the box does not suffice, one might be tempted to use standard cMDP techniques, such as linear programming, to solve anytime-constrained problems. However, continuous optimization techniques fail since the optimal

[2] By using a negative budget, we capture the covering constraints that commonly appear in goal-directed problems. We consider the other variations of the problem in the Appendix.

anytime-constrained value is discontinuous in the costs and budgets. Even a slight change to the cost or budget can lead to a dramatically smaller solution value.

Proposition 2. *V^* is a continuous function of the rewards, but a discontinuous function of the costs and budgets.*

Intractability. In fact, solving anytime-constrained cMDPs is fundamentally harder than expectation-constrained cMDPs; solving (ANY) is NP-hard. The intuition is that anytime constraints capture the knapsack problem. With a single state, we can let the reward at time i be item i's value and the cost at time i be item i's weight. Any deterministic policy corresponds to choosing certain items to add to the knapsack, and a feasible policy ensures the set of items fit in the knapsack. Thus, an optimal deterministic policy for the cMDP corresponds to an optimal knapsack solution.

On the other hand, a randomized policy does not necessarily yield a solution to the knapsack problem. However, we can show that any randomized policy can be derandomized into a deterministic policy with the same cost and at least the same value. The derandomization can be performed by inductively choosing any supported action that leads to the largest value. This is a significant advantage over expectation and chance constraints which typically require stochastic policies.

Lemma 1 (Derandomization)**.** *For any randomized policy $\bar{\pi}$, there exists a deterministic policy π whose cumulative cost is at most $\bar{\pi}$'s anytime and whose value satisfies $V^{\pi} \geq V^{\bar{\pi}}$.*

Since the existence of a randomized solution implies the existence of a deterministic solution with the same value via Lemma 1, the existence of a high-value policy for an anytime-constrained cMDP corresponds to the existence of a high-value knapsack solution. Thus, anytime constraints can capture the knapsack problem. Our problem remains hard even if we restrict to the very specialized class of deterministic, non-adaptive (state-agnostic) policies, which are mappings from time steps to actions: $\pi : [H] \to \mathcal{A}$.

Theorem 1 (Hardness)**.** *Solving (ANY) is NP-complete even when $S = 1$, $A = 2$, and both the costs and rewards are deterministic, non-negative integers. This remains true even if restricted to the class of non-adaptive policies. Hardness also holds for stationary cMDPs so long as $S \geq H$.*

Given the hardness results in Theorem 1, it is natural to turn to approximation algorithms to find policies efficiently. The most natural approach would be to settle for a feasible, although, approximately-optimal policy. Unfortunately, even with only $d = 2$ constraints, it is intractable to compute a feasible policy with any non-trivial approximation factor. This also means designing an algorithm whose complexity is polynomial in d is likely impossible.

Theorem 2 (Hardness of Approximation)**.** *For $d \geq 2$, computing a feasible solution to (ANY) is NP-hard. Furthermore, for any $\epsilon > 0$, it is NP-hard to compute a feasible policy π satisfying either $V^{\pi} \geq V^* - \epsilon$ or $V^{\pi} \geq V^*(1 - \epsilon)$.*

Remark 1. Note, Theorem 2 does not only rule out the existence of fully-polynomial-time approximation schemes (FPTAS). Since $\epsilon > 0$ is arbitrary, it rules out any non-trivial approximation similar to the (non-metric) Traveling Salesman Problem.

3 FPT REDUCTION

Despite our strong hardness results, Theorem 1 and Theorem 2, we show for a large class of cMDPs, (ANY) can be solved efficiently. The key is to augment the state space of the system to capture the constraint consideration. In this section, we assume the cost distributions have finite support; we generalize to broader classes of distributions in Section 4.

Assumption 1. $n := \sup_{h,s,a} |C_h(s,a)| < \infty$.

Proposition 1 illustrates that a key issue with standard policies is that they cannot adapt to the costs seen so far. This forces the policies to be overly conservative or to risk violating the budget. At the same time, cost-history-dependent policies are undesirable as they are computationally expensive to construct and store in memory.

Instead, we claim the agent can exploit a sufficient statistic of the cost sequence: the *cumulative cost*. By incorporating cumulative costs carefully, the agent can simulate an unconstrained MDP, $\overline{M}$, whose optimal policies are solutions to (ANY). The main challenge is defining the augmented states, $\overline{\mathcal{S}}_h$.

Augmented States. We could simply define $\overline{\mathcal{S}}_h$ to be $\mathcal{S} \times \mathbb{R}^d$, but this would result in an infinite state MDP with a discontinuous reward function, which cannot easily be solved. The ideal choice would be $\mathcal{F}_h := \{(s, \bar{c}) \in \mathcal{S} \times \mathbb{R}^d \mid \exists \pi \in \Pi_M,\ \mathbb{P}_M^{\pi}[s_h = s, \bar{c}_h = \bar{c}] > 0\}$, which is the minimum set containing all (state, cost)-pairs induced by feasible policies. However, $\mathcal{F}_h$ is difficult to characterize.

Instead, we consider a relaxation stemming from the idea of *safe exploration*. Namely, we look at the set of all (state, cost)-pairs that the agent could induce if it repeatedly interacted with M and only took actions that would not violate the constraint given the current history. This set can be constructed inductively. First,

Algorithm 1 Reduction to Unconstrained RL

Input: cMDP M
$\overline{M} \leftarrow$ Definition 2(M)
$\pi, \overline{V}^* \leftarrow \text{Solve}(\overline{M})$
if $\overline{V}^* = -\infty$ **then**
  return "Infeasible"
else
  return π

Algorithm 2 Augmented Interaction Protocol

Input: augmented policy π
$\overline{s}_1 = (s_0, 0)$ and $\overline{c}_1 = 0$.
for $h = 1$ to H **do**
  $a_h = \pi_h(\overline{s}_h)$.
  $c_h \sim C_h(s_h, a_h)$ and $s_{h+1} \sim P_h(s_h, a_h)$.
  $\overline{c}_{h+1} = \overline{c}_h + c_h$.
  $\overline{s}_{h+1} = (s_{h+1}, \overline{c}_{h+1})$.

the agent starts with $(s_0, 0)$ because it has yet to incur any costs. Then, if at time h, the agent has safely arrived at the pair $(s, \overline{c})$, the agent can now safely choose any action a for which $\Pr_{c\sim C_h(s,a)}[\overline{c} + c \le B] = 1$.

Definition 1 (Augmented States)**.** $\overline{\mathcal{S}}_1 := \{(s_0, 0)\}$, and for any $h \ge 1$,

$$\overline{\mathcal{S}}_{h+1} := \Big\{(s', \overline{c}') \mid \exists (s, \overline{c}) \in \overline{\mathcal{S}}_h, a \in \mathcal{A}, c' \in C_h(s,a), \overline{c}' = \overline{c} + c', \Pr_{c\sim C_h(s,a)}[\overline{c} + c \le B] = 1, P_h(s' \mid s, a) > 0\Big\}.$$

Unlike the backward induction approaches commonly used in MDP theory, observe that $\overline{\mathcal{S}}$ is constructed using *forward induction*. This feature is critical to computing a small, finite augmented-state space. We also point out that $\overline{\mathcal{S}}_h$ is a relaxation of $\mathcal{F}_h$ since actions chosen based on past costs without considering the future may not result in a fully feasible path. Nevertheless, the relaxation is not too weak; whenever 0 cost actions are always available, $\overline{\mathcal{S}}_h$ exactly matches $\mathcal{F}_h$.

Lemma 2. *$\forall h \in [H+1]$, $\overline{\mathcal{S}}_h \supseteq \mathcal{F}_h$ and $|\overline{\mathcal{S}}_h| < \infty$. Furthermore, equality holds if $\forall h, s, \exists a$ for which $C_h(s,a) = \{0\}$.*

If the agent records its cumulative costs and always takes safe actions, the interaction evolves according to the following unconstrained MDP.

Definition 2 (Augmented MDP)**.** The *augmented MPD* $\overline{M} := (\overline{\mathcal{S}}, \overline{\mathcal{A}}, \overline{P}, \overline{R}, H, \overline{s}_0)$ where,

- $\overline{\mathcal{S}}_h$ is defined in Definition 1.
- $\overline{\mathcal{A}}_h(s, \overline{c}) := \{a \in \mathcal{A} \mid \Pr_{c\sim C_h(s,a)}[\overline{c} + c \le B] = 1\}$.
- $\overline{P}_h((s', \overline{c}+c) \mid (s, \overline{c}), a) := P_h(s' \mid s, a) C_h(c \mid s, a)$.
- $\overline{R}_h((s, \overline{c}), a) := R_h(s, a)$.
- $\overline{s}_0 := (s_0, 0)$.

Theorem 3 (Optimality)**.** *Algorithm 1 solves* (ANY) *and can be implemented to run in finite time.*

We see from Theorem 3 that an anytime-constrained cMDP can be solved using Algorithm 1. If M is known, the agent can directly construct $\overline{M}$ using Definition 2 and then solve $\overline{M}$ using any RL planning algorithm. If M is unknown, the agent can still solve $\overline{M}$ by replacing the call to $\pi_h(s, \overline{c})$ in Algorithm 2 by a call to any RL learning algorithm.

Corollary 2 (Reduction)**.** *An optimal policy for an anytime-constrained cMDP can be computed from Algorithm 1 paired with any RL planning or learning algorithm. Thus, anytime-constrained RL reduces to standard RL.*

Augmented Policies. Observe that any Markovian policy π for $\overline{M}$ is a *augmented policy* that maps (state, cost)-pairs to actions. This policy can be translated into a full history policy or can be used directly through the new interaction protocol described in Algorithm 2. By recording the cumulative cost, the agent effectively simulates the $\pi - \overline{M}$ interaction through the $\pi - M$ interaction.

Analysis. To understand augmented policies, we need new machinery than typical MDP theory. Since traditional policies are insufficient for anytime constraints, we need to directly compare against cost-history-dependent policies. However, we cannot consider arbitrary histories, since an infeasible history could allow higher value. Rather, we focus on histories that are induced by safe exploration:

$$W_h(s, \overline{c}) := \Big\{\tau_h \in \mathcal{H}_h \mid \exists \pi,\ \mathbb{P}^\pi_{\tau_h}[s_h = s, \overline{c}_h = \overline{c}] = 1, \mathbb{P}^\pi_{\tau_k}[\overline{c}_{k+1} \le B] = 1 \quad \forall k \in [h-1]\Big\}.$$

Here, $\mathbb{P}^\pi_{\tau_h}[\cdot] := \mathbb{P}^\pi_M[\cdot \mid \tau_h]$ and $\mathbb{E}^\pi_{\tau_h}[\cdot] := \mathbb{E}^\pi_M[\cdot \mid \tau_h]$ denote the conditional probability and expectation given partial history τ_h. The condition $\mathbb{P}^\pi_{\tau_k}[\overline{c}_{k+1} \le B] = 1$ enforces that any action taken along the trajectory never could have violated the budget.

We must also restrict to policies that are feasible given such a history: $\Pi_M(\tau_h) := \{\pi \mid \mathbb{P}^\pi_{\tau_h}[\forall k \in [H], \overline{c}_{k+1} \le B] = 1\}$. Note that generally $\Pi_M(\tau_h) \supset \Pi_M$ so some $\pi \in \Pi_M(\tau_h)$ need not be feasible, but importantly $\Pi_M(s_0) = \Pi_M$ contains only

feasible policies. We define,

$$V_h^*(\tau_h) := \max_{\pi \in \Pi_M(\tau_h)} V_h^\pi(\tau_h) \; ; \overline{V}_h^*(s, \overline{c}) := \max_\pi \overline{V}_h^\pi(s, \overline{c}),$$

to be the optimal feasible value conditioned on τ_h, and the optimal value for $\overline{M}$ from time h onward starting from $(s, \overline{c})$, respectively. We show that optimal feasible solutions satisfy the *augmented bellman-optimality equations*.

Lemma 3. *For any $h \in [H+1]$, $(s, \overline{c}) \in \overline{\mathcal{S}}_h$, and $\tau_h \in W_h(s, \overline{c})$, $\overline{V}_h^*(s, \overline{c}) = V_h^*(\tau_h)$.*

The proof is more complex than the traditional bellman-optimality equations. It requires (1) backward induction to argue that the value is maximal under any safe partial history and (2) forward induction to argue the costs accrued respect the anytime constraints. It then follows that solutions to $\overline{M}$ are solutions to (ANY).

3.1 Complexity Analysis

To analyze the efficiency of our reduction, we define a combinatorial measure of a cMDP's complexity.

Definition 3 (Cost Diversity)**.** The *cost diversity*, D_M, is the total number of distinct cumulative costs the agent could face at any time:

$$D_M := \max_{h \in [H+1]} \left| \left\{ c \mid \exists s, (s, c) \in \overline{\mathcal{S}}_h \right\} \right|.$$

When clear from context, we refer to D_M as D.

We call D the diversity as it measures the largest cost population that exists in any generation h. The diversity naturally captures the complexity of an instance since the agent would likely encounter at least this many cumulative costs when computing or learning an optimal policy using any safe approach.

In particular, we can bound the complexity of the planning and learning algorithms produced from our reduction in terms of the diversity. For concreteness, we pair Algorithm 1 with backward induction (Altman, 1999) to produce a planning algorithm and with BPI-UCBVI (Menard et al., 2021) to produce a learning algorithm.

Proposition 3 (Complexity)**.** *Using Algorithm 1, an optimal policy for an anytime-constrained cMDP can be computed in $O\left(HS^2AnD\right)$ time and learned with $\tilde{O}\left(H^3SAD\log(\frac{1}{\delta})/\gamma^2\right)$ sample complexity. Furthermore, the amount of space needed to store the policy is $O\left(HSD\right)$.*

In the worst case, D could be exponentially large in the time horizon. However, for many cMDPs, D is small. One key factor in controlling the diversity is the precision needed to represent the supported costs in memory.

Lemma 4 (Precision)**.** *If the cost precision is at most k, then $D \leq H^d 2^{(k+1)d}$.*

We immediately see that when the costs have precision k, all of our algorithms have complexity polynomial in the size of M and exponential in k and d. By definition, this means our algorithms are fixed-parameter tractable in k so long as d is held constant. Moreover, we see as long as the costs can be represented with logarithmic precision, our algorithms have polynomial complexity.

Theorem 4 (Fixed-Parameter Tractability)**.** *For constant d, if $k = O(\log(|M|))$, planning (learning) for anytime-constrained cMDPs can be performed in polynomial time (sample complexity) using Algorithm 1, and the computed policy can be stored with polynomial space.*

Remark 2. Many cost functions can be represented with small precision. In practice, all modern computers use fixed precision numbers. So, in any real system, our algorithms have polynomial complexity. Although technically efficient, our methods can be prohibitively expensive when k or d is large. However, this complexity seems unavoidable since computing exact solutions to (ANY) is NP-hard in general.

4 APPROXIMATION ALGORITHMS

Since Theorem 2 rules out the possibility of traditional value-approximation algorithms due to the hardness of finding feasible policies, we relax the requirement of feasibility. We show that even with a slight relaxation of the constraint, solutions with optimal value can be found efficiently. Conversely, we can satisfy the constraint but with a weaker guarantee on value. Our approximate-feasibility methods can even handle infinite support distributions so long as they are bounded above.

Assumption 2. $c^{\max} := \sup_{h,s,a} \sup C_h(s, a) < \infty$.

If $Hc^{\max} \leq B$, then every policy is feasible, which just leads to a standard unconstrained problem. A similar phenomenon happens if $c^{\max} \leq 0$. Thus, we assume WLOG that $Hc^{\max} > B$ and $c^{\max} > 0$.

Definition 4 (Approximate Feasibility)**.** For any $\epsilon > 0$, a policy π is ϵ-additive feasible if,

$$\mathbb{P}_M^\pi \left[\forall t \in [H], \sum_{h=1}^{t} c_h \leq B + \epsilon \right] = 1, \qquad (1)$$

Algorithm 3 Approximate Reduction

Input: cMDP M and projection f

$\hat{M} \leftarrow$ Definition 5(M, f)
$\pi, \hat{V}^* \leftarrow \text{Solve}(\hat{M})$
if $\hat{V}^* = -\infty$ **then**
  return "Infeasible"
else
  return π

and ϵ-relative feasible if,

$$\mathbb{P}^{\pi}_{M}\left[\forall t \in [H], \sum_{h=1}^{t} c_h \leq B(1+\epsilon\sigma_B)\right] = 1, \quad (2)$$

where σ_B is the sign of B[3].

Approximation. The key to reducing the complexity of our reduction is lowering the cost diversity. Rather than consider every cost that can be accumulated from safe exploration, the agent can consider a smaller set of approximate cumulative costs. Specifically, for any cumulative cost $\overline{c}_h$ and cost c_h, the agent considers some $\hat{c}_{h+1} = f_h(\overline{c}_h, c_h)$ instead of $\overline{c}_{h+1} = \overline{c}_h + c_h$.

We view f as projecting a cumulative cost onto a smaller approximate cost space. Following the principle of optimism, we also ensure that $f(\overline{c}_h, c_h) \leq \overline{c}_h + c_h$. This guarantees that any optimal policy under the approximate costs achieves optimal value at the price of a slight violation in the budget.

If the agent records the approximate costs induced by the projection f, the interaction evolves according to the following unconstrained MDP.

Definition 5 (Approximate MDP)**.** The *approximate MPD* $\hat{M} := (\hat{\mathcal{S}}, \hat{\mathcal{A}}, \hat{P}, \hat{R}, H, \hat{s}_0)$ where,

$$\hat{S}_{h+1} := \Big\{(s', \hat{c}') \mid \exists (s, \hat{c}) \in \hat{S}_h, a \in \mathcal{A}, c' \in C_h(s,a), \hat{c}' = f_h(\hat{c}, c'), \Pr_{c \sim C_h(s,a)}[f_h(\hat{c}, c) \leq B] = 1, P_h(s' \mid s, a) > 0\Big\},$$

is defined using approximate costs produced by safe exploration with a projection step. The other objects are defined analogously to Definition 2.

Our approximation algorithms, Algorithm 3, equate to solving $\hat{M}$ for different choices of f. To use any Markovian π for $\hat{M}$, the agent just needs to apply f when updating its approximate costs. In effect, the agent *accumulates then projects* to create each approximate cost. The new interaction protocol is given by Algorithm 4.

[3]When the costs and budgets are negative, negating the constraint yields $\sum_{t=1}^{H} c_t \geq |B|(1-\ )$, which is the traditional notion of relative approximation for covering objectives.

Algorithm 4 Approximate Interaction Protocol

Input: policy π and projection f

$\hat{s}_1 = (s_0, 0)$ and $\hat{c}_1 = 0$.
for $h = 1$ to H **do**
  $a_h = \pi_h(\hat{s}_h)$
  $c_h \sim C_h(s_h, a_h)$ and $s_{h+1} \sim P_h(s_h, a_h)$
  $\hat{c}_{h+1} = f_h(\hat{c}_h, c_h)$
  $\hat{s}_{h+1} = (s_{h+1}, \hat{c}_{h+1})$

To derive our choice of f, we first observe that the cumulative cost can never surpass B. Furthermore, should the agent ever accumulate a cost of $B - Hc^{\max}$, it can no longer violate the budget along that trajectory. Thus, the agent's cumulative cost is always effectively within the interval $[B - Hc^{\max}, B]$.

Projection. Our approach is to evenly subdivide the interval $[B - Hc^{\max}, B]$ by length-ℓ intervals centered around 0. Then, the projection always maps a point in an interval to its left endpoint. Alternatively, we can think of ℓ as defining a new unit of measurement, and the projection maps each cumulative cost to its largest integer multiple of ℓ below the cumulative cost. Should the agent ever encounter an extremely negative cost, we safely truncate it to the projection of $B - (H-h)c^{\max}$.

In symbols, we define our projection by, $f_h(\hat{c}, c) :=$

$$\begin{cases} \hat{c} + \left\lfloor \frac{c}{\ell} \right\rfloor \ell & \text{if } \hat{c} + c \geq B - (H-h)c^{\max} \\ \left\lfloor \frac{B-(H-h)c^{\max}}{\ell} \right\rfloor \ell & \text{o.w.} \end{cases}$$

Critically, the projection is defined so that each approximate cost is an underestimate of the true cost, but no farther than ϵ away from the true cost (except when a cost smaller than $B - (H-h)c^{\max}$ is encountered).

Lemma 5. *For any feasible policy π for $\hat{M}$ and any $h \in [H+1]$, $\mathbb{P}^{\pi}_{M}[(\hat{c}_h \leq \overline{c}_h \leq \hat{c}_h + (h-1)\ell) \vee (\overline{c}_h, \hat{c}_h \leq B - (H-h+1)c^{\max})] = 1$. Also, $\left|\left\{\hat{c}_h \mid \exists s \in \mathcal{S}, (s, \hat{c}_h) \in \hat{\mathcal{S}}_h\right\}\right| \leq \left(\frac{H\|c^{\max}\|_1}{\ell} + 2\right)^d$.*

We see that solving $\hat{M}$ gives additive feasible solutions and $\hat{M}$ has far fewer states than $\overline{M}$.

Theorem 5 (Approximation)**.** *Algorithm 3 computes an $H\ell$-additive feasible policy whose value is at least the optimal value of* (ANY) *and that can be stored with $O\left(H^{d+1} S \|c^{\max}\|_{\infty}^{d} / \ell^{d}\right)$ space.*

Like with our original reduction, the interaction protocol in Algorithm 4 allows the agent to simulate $\hat{M}$

online through M. Thus, planning and learning in $\hat{M}$ can be done through M.

Remark 3. Note, the agent does not need to construct $\hat{S}$ using Definition 5; it suffices to consider the finite, stationary state space $\mathcal{S} \times \mathcal{C}$, where $\mathcal{C}$ is the ℓ-cover of $[B - Hc^{\max}, B]$ defined by f. Technically, for learning, the agent should already know or have learned a bound on $c^{\max}$ to know the approximate state space.

4.1 Approximation Guarantees

We can use Algorithm 3 with different choices of ℓ to achieve the traditional approximation guarantees defined in Definition 4.

Additive Approximation. Given any $\epsilon > 0$, we can compute an ϵ-additive feasible solution by choosing $\ell := \frac{\epsilon}{H}$. This approach is efficient so long as $c^{\max}$ is not too large, since $c^{\max}$ controls the number of discretized costs we need to consider.

Corollary 3 (Additive Reduction)**.** *For any $\epsilon > 0$, an optimal value, ϵ-additive feasible policy for an anytime-constrained cMDP can be computed in $O\left(H^{4d+1}S^2A\left\|c^{\max}\right\|_\infty^{2d}/\epsilon^{2d}\right)$ time and learned with $\tilde{O}\left(H^{2d+3}SA\left\|c^{\max}\right\|_\infty^{d}\log(\frac{1}{\delta})/\gamma^2\epsilon^{d}\right)$ sample complexity using Algorithm 3 with $\ell := \frac{\epsilon}{H}$. Furthermore, the amount of space needed to store the policy is $O\left(H^{2d+1}S\left\|c^{\max}\right\|_\infty^{d}/\epsilon^{d}\right)$. Thus, if d is constant and $c^{\max} \leq \text{poly}(|M|)$, our additive methods are polynomial time and sample complexity.*

Relative Approximation. Given any $\epsilon > 0$, we can compute an ϵ-relative feasible solution by choosing $\ell := \frac{\epsilon|B|}{H}$. This approach is efficient so long as $c^{\max}$ is not much larger than $|B|$. This allows us to capture cost ranges that are polynomial multiples of $|B|$, which could be exponentially large, unlike the additive approximation which requires that $c^{\max}$ to be polynomial.

Corollary 4 (Relative Reduction)**.** *For any $\epsilon > 0$, if $c^{\max} \leq x\,|B|$, an optimal value, ϵ-relative feasible policy for an anytime-constrained cMDP can be computed in $O\left(x^{2d}H^{4d+1}S^2A/\epsilon^{2d}\right)$ time and learned with $\tilde{O}\left(x^{d}H^{2d+3}SA/\epsilon^{d}\log(\frac{1}{\delta})/\gamma^2\right)$ sample complexity using Algorithm 3 with $\ell = \frac{\epsilon|B|}{H}$. Furthermore, the amount of space needed to store the policy is $O\left(x^{d}H^{2d+1}S/\epsilon^{d}\right)$. Thus, if d is constant and $x \leq \text{poly}(|M|)$, our methods are polynomial time and sample complexity.*

Corollary 5. *If all costs are positive, the H dependence in each guarantee of both the additive and relative approximation improves to H^{2d+1}, H^{d+3}, and H^{d+1}, respectively.*

Limitations. Using our additive approximation, we can efficiently handle any cMDP with $c^{\max} \leq \text{poly}(|M|)$. Using the relative approximation, we can even handle the case that $c^{\max}$ is exponentially large so long as $c^{\max} \leq \text{poly}(|M|)\,|B|$. Thus, we can efficiently compute approximately feasible solutions so long as $c^{\max} \leq \text{poly}(|M|)\max(1, |B|)$.

We point out that the condition $c^{\max} \leq \text{poly}(|M|)\,|B|$ is very natural. If the costs all have the same sign, any feasible policy induces costs with $c^{\max} \leq |B|$. In our driving example, the condition simply says the vehicle cannot use more fuel than the capacity of the tank. In fact, this bottleneck is not due to our approach; some bound on $c^{\max}$ is necessary for efficient computation as Proposition 4 shows.

Proposition 4. *For any fixed $\epsilon > 0$, computing an optimal-value, ϵ-additive or ϵ-relative feasible solution to the knapsack problem with negative weights is NP-hard. Hence, it is hard for anytime-constrained cMDPs.*

Feasibility Scheme. Let $OPT(B)$ denote the optimal value V^* of an anytime-constrained cMDP with budget B. Algorithm 3 provides an efficient approximation and guarantees at least $OPT(B)$ value but with the possibility of slightly going over budget, up to $B+\epsilon$ or $B(1+\epsilon)$. If it is important that the budget B is never violated, we can use the same approximate algorithm with one change: We instead give it $\hat{M}'$, which is $\hat{M}$ constructed from M under a smaller budget: (1) $B-\epsilon$ for the additive approximation and (2) $B/(1+\epsilon)$ for the relative approximation. Then, any over-budget by Algorithm 3 is compensated by the smaller budget, so that the cumulative cost is still under B. Thus we have both efficiency and (budget B) feasibility. The drawback is that the algorithm now only guarantees a value at least $OPT(B-\epsilon)$ or $OPT(B/(1+\epsilon))$, both can be much smaller than $OPT(B)$.

Proposition 5 (Feasible Solutions)**.** *If π is returned by Algorithm 3 using $\ell = \frac{\epsilon}{H}$ ($\ell = \frac{\epsilon|B|}{H}$) with budget $B' = B - \epsilon$ ($B' = \frac{B}{1+\epsilon}$), then π is feasible for (ANY).*

5 EXPERIMENTS

We test our methods on the family of NP-hard cMDP instances that we constructed in the proof of Theorem 1. Namely, cMDPs with one state, $\mathcal{S} = \{0\}$, two actions, $\mathcal{A} = \{0, 1\}$, and a single constraint, $d = 1$. The rewards and costs satisfy $r_h(0,0) = c_h(0,0) = 0$ for all h. For action 1, $r_h(0,1) = x_h$ and $c_h(0,1) = y_h$ where for all h, $x_h, y_h \sim \text{Unif}[0,1]$ are chosen and then fixed for each cMDP instance. Since the complexity blowup for anytime constraints is in the time horizon, we focus on varying the time horizon.

Figure 1: Value comparison of our relative approximation and feasibility scheme.

Figure 2: Running time of our relative approximation.

Already for a cMDP with $H = B = 15$, our exact reduction takes around a minute to run, which is unsurprising as the complexity grows like $30(2^{15})^2$. Instead of comparing to the reduction, we can instead compare our relative approximation (Corollary 4) to our feasibility scheme (Proposition 5). By definition, the relative approximation achieves at least the optimal value and the feasibility scheme is feasible. Thus, if both algorithms achieve value close to each other, we know that the approximation is not violating the budget much and the feasibility scheme is achieving near-optimal value.

We perform $N = 10$ trials for each $H \in \{10, 20, 30, 40, 50\}$. We consider two different budgets, $b \in \{.1, 10\}$, and $\epsilon = 0.1$. We report the value of the worst trial (the one where both methods are farthest apart) for each H in Figure 1. We see that both values are consistently close even in the worst case, which indicates the feasible solution is nearly optimal and the approximate solution is nearly feasible.

To test the scaling of our approximation algorithm, we ran $N = 10$ trials for each $H \in \{10, 20, \ldots, 100\}$. Here, we use a budget of $b = 100$ to maximize the cost diversity. This time, we tried both $\epsilon = 0.1$ and the even larger $\epsilon = 1$. We report the worst running time from each H under $\epsilon = 0.1$ in Figure 2. We see a cubic-like growth as expected from Corollary 5. Also, the approximation easily handled a large horizon of 100 in a mere 2 seconds, which drastically beats out the exact reduction. For $\epsilon = 1$ the results are even more striking with a maximum run time of .0008 seconds and the solutions are guaranteed to violate the budget by no more than a multiple of 2! We give additional details and results in the Appendix.

6 CONCLUSIONS

In this paper, we formalized and rigorously studied anytime-constrained cMDPs. Although traditional policies cannot solve anytime-constrained cMDPs, we showed that deterministic augmented policies suffice. We also presented a fixed-parameter tractable reduction based on cost augmentation and safe exploration that yields efficient planning and learning algorithms when the cost precision is $O(\log(|M|))$. In addition, we developed efficient planning and learning algorithms to find ϵ-approximately feasible policies with optimal value whenever the maximum supported cost is $O(\text{poly}(|M|)\max(1, |B|))$. Based on our hardness of approximation results, this is the best approximation guarantee we can hope for under worst-case analysis.

Although we have resolved many questions, there are still many more mysteries about anytime constraints. Primarily, we focus on worst-case analysis, which may be too pessimistic. Since anytime constraints are so sensitive to changes in cost, a smoothed or average case analysis could be promising. Going further, there may be useful classes of cMDPs for which (ANY) is efficiently solvable even in the worst case. Proving lower bounds on the exact complexity of computing solutions is also interesting. Lastly, learning a feasible policy without violation during the learning process is an important open question.

Acknowledgments

This project is supported in part by NSF grants 1836978, 2023239, 2202457, 2331669, ARO MURI W911NF2110317, AF CoE FA9550-18-1-0166, and DMS-2023239.

References

J. Achiam, D. Held, A. Tamar, and P. Abbeel. Constrained policy optimization. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 22–31. PMLR, 8 2017. URL https://proceedings.mlr.press/v70/achiam17a.html.

E. Altman. Constrained Markov Decision Processes. Chapman and Hall/CRC, 1999. doi: 10.1201/9781315140223.

A. Badanidiyuru, R. Kleinberg, and A. Slivkins. Bandits with knapsacks. In 2013 IEEE 54th Annual Symposium on Foundations of Computer Science, pages 207–216, 2013. doi: 10.1109/FOCS.2013.30.

D. P. Bertsekas and J. N. Tsitsiklis. An analysis of stochastic shortest path problems. Math. Oper. Res., 16(3):580–595, 8 1991. ISSN 0364-765X.

A. Bhatia, P. Varakantham, and A. Kumar. Resource constrained deep reinforcement learning. Proceedings of the International Conference on Automated Planning and Scheduling, 29(1):610–620, 5 2021. doi: 10.1609/icaps.v29i1.3528. URL https://ojs.aaai.org/index.php/ICAPS/article/view/3528.

V. Borkar. An actor-critic algorithm for constrained markov decision processes. Systems & Control Letters, 54(3):207–213, 2005. ISSN 0167-6911. doi: https://doi.org/10.1016/j.sysconle.2004.08.007. URL https://www.sciencedirect.com/science/article/pii/S0167691104001276.

V. Borkar and R. Jain. Risk-constrained markov decision processes. IEEE Transactions on Automatic Control, 59(9):2574–2579, 2014. doi: 10.1109/TAC.2014.2309262.

K. Brantley, M. Dudik, T. Lykouris, S. Miryoose, M. Simchowitz, A. Slivkins, and W. Sun. Constrained episodic reinforcement learning in concave-convex and knapsack settings. In NeurIPS, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/bc6d753857fe3dd4275dff707dedf329-Abstract.html.

A. Castellano, H. Min, E. Mallada, and J. A. Bazerque. Reinforcement learning with almost sure constraints. In R. Firoozi, N. Mehr, E. Yel, R. Antonova, J. Bohg, M. Schwager, and M. Kochenderfer, editors, Proceedings of The 4th Annual Learning for Dynamics and Control Conference, volume 168 of Proceedings of Machine Learning Research, pages 559–570. PMLR, 6 2022. URL https://proceedings.mlr.press/v168/castellano22a.html.

A. Charnes, W. W. Cooper, and G. H. Symonds. Cost horizons and certainty equivalents: An approach to stochastic programming of heating oil. Management Science, 4(3):235–263, 1958. doi: 10.1287/mnsc.4.3.235. URL https://doi.org/10.1287/mnsc.4.3.235.

X. Chen, J. Hu, L. Li, and L. Wang. Efficient reinforcement learning in factored mdps with application to constrained rl, 2021.

W. C. Cheung. Regret minimization for reinforcement learning with vectorial feedback and complex objectives. In Advances in Neural Information Processing Systems, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/file/a02ffd91ece5e7efeb46db8f10a74059-Paper.pdf.

Y. Chow, M. Ghavamzadeh, L. Janson, and M. Pavone. Risk-constrained reinforcement learning with percentile risk criteria. Journal of Machine Learning Research, 18(167):1–51, 2018. URL http://jmlr.org/papers/v18/15-636.html.

A. Coronato, M. Naeem, G. De Pietro, and G. Paragliola. Reinforcement learning for intelligent healthcare applications: A survey. Artificial Intelligence in Medicine, 109:101964, 2020. ISSN 0933-3657. doi: https://doi.org/10.1016/j.artmed.2020.101964. URL https://www.sciencedirect.com/science/article/pii/S093336572031229X.

D. Ding, X. Wei, Z. Yang, Z. Wang, and M. R. Jovanovic. Provably efficient safe exploration via primal-dual policy optimization, 2020.

R. Downey and M. Fellows. Parameterized Complexity. Monographs in Computer Science. Springer New York, 2012. ISBN 9781461205159. URL https://books.google.com/books?id=HyTjBwAAQBAJ.

Y. Efroni, S. Mannor, and M. Pirotta. Exploration-Exploitation in Constrained MDPs. arXiv e-prints, art. arXiv:2003.02189, Mar. 2020. doi: 10.48550/arXiv.2003.02189.

C. Fan, C. Zhang, A. Yahja, and A. Mostafavi. Disaster city digital twin: A vision for integrating artificial and human intelligence for disaster management. International Journal of Information Management, 56:102049, 2021. ISSN 0268-4012. doi: https://doi.org/10.1016/j.ijinfomgt.2019.102049. URL https://www.sciencedirect.com/science/article/pii/S0268401219302956.

J. F. Fisac, N. F. Lugovoy, V. Rubies-Royo, S. Ghosh, and C. J. Tomlin. Bridging hamilton-jacobi safety analysis and reinforcement learning. In 2019 International Conference on Robotics and Automation (ICRA), page 8550–8556. IEEE Press, 2019. doi: 10.1109/ICRA.2019.8794107. URL https://doi.org/10.1109/ICRA.2019.8794107.

S. Gros, M. Zanon, and A. Bemporad. Safe reinforcement learning via projection on a safe set: How to achieve optimality? IFAC-PapersOnLine, 53(2):8076–8081, 2020. ISSN 2405-8963. doi: https://doi.org/10.1016/j.ifacol.2020.12.2276. URL https://www.sciencedirect.com/science/article/pii/S2405896320329360. 21st IFAC World Congress.

S. Gu, L. Yang, Y. Du, G. Chen, F. Walter, J. Wang, Y. Yang, and A. Knoll. A review of safe reinforcement learning: Methods, theory and applications, 2023.

P. Hou, W. Yeoh, and P. Varakantham. Revisiting risk-sensitive mdps: New algorithms and results. Proceedings of the International Conference on Automated Planning and Scheduling, 24 (1):136{144, 5 2014. doi: 10.1609/icaps.v24i1.13615. URL https://ojs.aaai.org/index.php/ICAPS/article/view/13615 .

W. Jung, M. Cho, J. Park, and Y. Sung. Quantile constrained reinforcement learning: A reinforcement learning framework constraining outage probability. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, Advances in Neural Information Processing Systems, volume 35, pages 6437{6449. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/2a07348a6a7b2c208ab5cb1ee0e78ab5-Paper-Conference.pdf .

K. C. Kalagarla, R. Jain, and P. Nuzzo. A sample-e cient algorithm for episodic nite-horizon mdp with constraints. Proceedings of the AAAI Conference on Arti cial Intelligence , 35 (9):8030{8037, 5 2021. doi: 10.1609/aaai.v35i9.16979. URL https://ojs.aaai.org/index.php/AAAI/article/view/16979 .

P. Kolesar. A markovian model for hospital admission scheduling. Management Science, 16(6):B384{B396, 1970. ISSN 00251909, 15265501. URL http://www.jstor.org/stable/2628725 .

J. Li, D. Fridovich-Keil, S. Sojoudi, and C. J. Tomlin. Augmented lagrangian method for instantaneously constrained reinforcement learning problems. In 2021 60th IEEE Conference on Decision and Control (CDC) , page 2982{2989. IEEE Press, 2021. doi: 10.1109/CDC45484.2021.9683088. URL https://doi.org/10.1109/CDC45484.2021.9683088 .

R. Li, Z. Zhao, Q. Sun, C.-L. I, C. Yang, X. Chen, M. Zhao, and H. Zhang. Deep reinforcement learning for resource management in network slicing. IEEE Access, 6:74429{74441, 2018. doi: 10.1109/ACCESS.2018.2881964.

H. Mao, M. Alizadeh, I. Menache, and S. Kandula. Resource management with deep reinforcement learning. In Proceedings of the 15th ACM Workshop on Hot Topics in Networks, HotNets '16, page 50{56, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450346610. doi: 10.1145/3005745.3005750. URL https://doi.org/10.1145/3005745.3005750 .

D. A. Melo Moreira, K. Valdivia Delgado, L. Nunes de Barros, and D. Deratani Mau a. E cient algorithms for risk-sensitive markov decision processes with limited budget. International Journal of Approximate Reasoning 139:143{165, 2021. ISSN 0888-613X. doi: https://doi.org/10.1016/j.ijar.2021.09.003. URL https://www.sciencedirect.com/science/article/pii/S0888613X21001389 .

P. Menard, O. D. Domingues, A. Jonsson, E. Kaufmann, E. Leurent, and M. Valko. Fast active learning for pure exploration in reinforcement learning. In M. Meila and T. Zhang, editors, Proceedings of the 38th International Conference on Machine Learning, volume 139 of Proceedings of Machine Learning Research, pages 7599{7608. PMLR, 7 2021. URL https://proceedings.mlr.press/v139/menard21a.html .

M. Mowbray, P. Petsagkourakis, E. del Rio-Chanona, and D. Zhang. Safe chance constrained reinforcement learning for batch process control. Computers & Chemical Engineering, 157:107630, 2022. ISSN 0098-1354. doi: https://doi.org/10.1016/j.compchemeng.2021.107630. URL https://www.sciencedirect.com/science/article/pii/S0098135421004087 .

M. Ono, M. Pavone, Y. Kuwata, and J. Balaram. Chance-constrained dynamic programming with application to risk-aware robotic space exploration. Auton. Robots, 39(4):555{571, 12 2015. ISSN 0929-5593. doi: 10.1007/s10514-015-9467-7. URL https://doi.org/10.1007/s10514-015-9467-7 .

X. Pan, Y. You, Z. Wang, and C. Lu. Virtual to real reinforcement learning for autonomous driving, 2017.

G. Paragliola, A. Coronato, M. Naeem, and G. De Pietro. A reinforcement learning-based approach for the risk management of e-health environments: A case study. In 2018 14th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS) pages 711{716, 2018. doi: 10.1109/SITIS.2018.00114.

S. Paternain, L. Chamon, M. Calvo-Fullana, and A. Ribeiro. Constrained reinforcement learning has zero duality gap. In Advances in Neural Information Processing Systems, volume 32, 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/c1aeb6517a1c7f33514f7ff69047e74e-Paper.pdf .

H. Peng and X. Shen. Multi-agent reinforcement learning based resource management in mec- and uav-assisted vehicular networks. IEEE Journal on Selected Areas in Communications, 39(1):131{141, 2021. doi: 10.1109/JSAC.2020.3036962.

M. L. Puterman. Markov Decision Processes: Discrete Stochastic Dynamic Programming John Wiley & Sons, Inc., USA, 1st edition, 1994. ISBN 0471619779.

M. Roderick, V. Nagarajan, and Z. Kolter. Provably safe pac-mdp exploration using analogies. In A. Banerjee and K. Fukumizu, editors, Proceedings of The 24th International Conference on Arti cial Intelligence and Statistics, volume 130 of Proceedings of Machine Learning Research, pages 1216{1224. PMLR, 4 2021. URL https://proceedings.mlr.press/v130/roderick21a.html .

K. W. Ross and R. Varadarajan. Markov decision processes with sample path constraints: The communicating case. Operations Research, 37(5):780{790, 1989. doi: 10.1287/opre.37.5.780. URL https://doi.org/10.1287/opre.37.5.780 .

M. Steinmetz, J. Ho mann, and O. Bu et. Revisiting goal probability analysis in probabilistic planning. Proceedings of the International Conference on Automated Planning and Scheduling, 26 (1):299{307, 3 2016. doi: 10.1609/icaps.v26i1.13740. URL https://ojs.aaai.org/index.php/ICAPS/article/view/13740 .

G. Thomas, Y. Luo, and T. Ma. Safe reinforcement learning by imagining the near future. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, Advances in Neural Information Processing Systems, volume 34, pages 13859{13869. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/73b277c11266681122132d024f53a75b-Paper.pdf.

Y. L. Tsai, A. Phatak, P. K. Kitanidis, and C. B. Field. Deep Reinforcement Learning for Disaster Response: Navigating the Dynamic Emergency Vehicle and Rescue Team Dispatch during a Flood. In AGU Fall Meeting Abstracts, volume 2019, pages NH33B{14, Dec. 2019.

C. Wu, B. Ju, Y. Wu, X. Lin, N. Xiong, G. Xu, H. Li, and X. Liang. Uav autonomous target search based on deep reinforcement learning in complex disaster scene. IEEE Access, 7:117227{117245, 2019. doi: 10.1109/ACCESS.2019.2933002.

H. Xu and S. Mannor. Probabilistic goal markov decision processes. In Proceedings of the Twenty-Second International Joint Conference on Arti cial Intelligence - Volume Volume Three, IJCAI'11, page 2046{2052. AAAI Press, 2011. ISBN 9781577355151.

Q. Yang, T. D. Sim-ao, S. H. Tindemans, and M. T. J. Spaan. Wcsac: Worst-case soft actor critic for safety-constrained reinforcement learning. Proceedings of the AAAI Conference on Arti cial Intelligence, 35(12):10639{10646, 5 2021. doi: 10.1609/aaai.v35i12.17272. URL https://ojs.aaai.org/index.php/AAAI/article/view/17272 .

S. X. Yu, Y. Lin, and P. Yan. Optimization models for the rst arrival target distribution function in discrete time. Journal of Mathematical Analysis and Applications, 225(1):193{223, 1998. ISSN 0022-247X. doi: https://doi.org/10.1006/jmaa.1998.6015. URL https://www.sciencedirect.com/science/article/pii/S0022247X98960152 .

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with speci cation of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all gures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear de nition of the speci c measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

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

(a) Citations of the creator If your work uses existing assets. [Not Applicable]

(b) The license information of the assets, if applicable. [Not Applicable]

(c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]

(d) Information about consent from data providers/curators. [Not Applicable]

(e) Discussion of sensible content if applicable, e.g., personally identi able information or offensive content. [Not Applicable]

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

(a) The full text of instructions given to participants and screenshots. [Not Applicable]

(b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]

(c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

A Proofs for Section 2

A.1 Proof of Proposition 1

Proof. Fix any $H \geq 2$. For any $h \in [H-1]$, let π be a cost-history-dependent policy that does not record the cost at time h. For any such π, we construct a cMDP instance for which π is arbitrarily suboptimal. This shows that any class of policies that does not consider the full cost history is insufficient to solve (ANY). In particular, the class of Markovian policies does not suffice.

Consider the simple cMDP M_h defined by a single state, $S = \{0\}$, two actions, $A = \{0, 1\}$, and horizon H. The initial state is trivially 0 and the transitions are trivially self-loops from 0 to 0. Importantly, M_h has non-stationary rewards and costs. The rewards are deterministic. For any $t \neq h+1$, $r_t(s,a) = 0$. For some large $x > 0$, $r_{h+1}(s,1) = x$ and $r_{h+1}(s,0) = 0$. The costs are deterministic except at time h. For any $t \notin \{h, h+1\}$, $c_t(s,a) = 0$. For $t = h+1$, $c_{h+1}(s,1) = B$ and $c_{h+1}(s,0) = 0$. For $t = h$, the costs are random:

$$C_h(s,a) := \begin{cases} B & \text{w.p. } \frac{1}{2} \\ 0 & \text{w.p. } \frac{1}{2} \end{cases}$$

The budget is any $B > 0$.

Clearly, an optimal cost-history-dependent policy can choose any action it likes other than at time $h+1$. At time $h+1$, an optimal policy chooses $a = 1$ if the cost incurred at time h was 0 and otherwise chooses action $a = 0$. The value of the optimal policy is $x/2$ since the agent receives total reward x whenever $c_h = 0$, which is half the time, and otherwise receives total reward 0. Thus, $V^{\pi^*}_{M_h} = x/2$.

On the other hand, consider π's performance. Since π does not record c_h, it cannot use c_h to make decisions. Hence, $p := P_{M_h}[a_{h+1} = 1]$ is independent of c_h's value. If $p > 0$ then with probability $1/2p > 0$, the agent accrues cost B at both time h and time $h+1$ so violates the constraint. Thus, if π is feasible, it must satisfy $p = 0$. Consequently, π can never choose $a = 1$ at time $h+1$ and so can never receive rewards other than 0. Thus, $V^{\pi}_{M_h} = 0 \ll x/2 = V^{\pi^*}_{M_h}$. By choosing x large enough, we see policies that do not consider the entire cost history can be arbitrarily suboptimal. By applying this argument to $H = 2$ and $h = 1$, we see that Markovian policies can be arbitrarily suboptimal and so do not suffice to solve anytime-constrained cMDPs. □

A.2 Proof of Corollary 1

Proof. Since optimal policies for expectation-constrained cMDPs are always Markovian, Proposition 1 immediately implies such policies are infeasible or arbitrarily suboptimal. In fact, we can see this using the same construction of M_h.

1. Under an expectation constraint, the optimal policy π^* can choose $p = 1/2$ and still maintain that $E_M[\sum_{t=1}^{H} c_t] = B/2 + pB = B$ B. Thus, such a policy violates the anytime constraint by accumulating cost $2B$ with probability $1/4$. In fact, if we generalize the construction of M_h to have $c_h = \frac{B}{2}$ with probability > 0 (where $= 1/2$ in the original construction), then the optimal expectation-constrained policy is the same but now accumulates cost $\frac{B}{2} + B$ with probability $/2 > 0$. Since $$ can be chosen to be arbitrarily small, the violation of the anytime constraint, which is $B/2$, can be arbitrarily large. Even if we relax the policy to just be $$-optimal, for any > 0 we can choose x large enough to where all $$-optimal policies still select action 1 with non-zero probability.

2. A similar construction immediately shows the arbitrarily infeasibility of optimal chance-constrained policies. Consider a chance constraint that requires $P_M[\sum_{t=1}^{H} c_t > B] \leq$ for some > 0. We can use the same construction as above but with an arbitrarily larger cost of $c_h = y\frac{B}{2}$ for some $y > 0$. Then, an optimal chance constrained policy can always let $p = 1$ since the cost only exceeds budget when $c_h > 0$ which happens with probability $$. Such a policy clearly violates the anytime constraint by $y\frac{B}{2}$, which is arbitrarily large by choosing y to be arbitrarily large. Also, observe this does not require us to consider Markovian policies since whether the budget was already violated at time h or not, the policy is still incentivized to choose action 1 at time $h+1$ as additional violation does not affect a chance-constraint. Again, considering an $$-optimal policy does not change the result.

3.

Suppose instead we computed an optimal policy using a smaller budget $B' < B$.

1. For expectation-constraints, to ensure the resultant policy is feasible for anytime constraints, we need that $p = 0$ as before. By inspection, it must be that $B' = B/2$ but then the value of the policy is 0 which is arbitrarily suboptimal as we saw before.
2. For chance-constraints, the situation is even worse. Consider the M_h but with $c_h = B$ w.p. ϵ. Then, no matter what $B' < B$ we choose, the resultant policy is not feasible. Specifically, an optimal cost-history-dependent policy under the event that $c_h = B/2$ will then choose $a_{h+1} = 1$ almost surely since the extent of the violation does not matter. But even ignoring this issue, under the event that $c_h = 0$ the policy would then have to choose $a_{h+1} = 0$ which is again arbitrarily suboptimal.

For the knapsack-constrained frameworks, the policy is allowed to violate the budget arbitrarily once per episode. Thus, no matter how we change the budget feasibility is never guaranteed: it will always choose $a_{h+1} = 1$ in any realization. The other frameworks also fail using slight modifications of the constructions above.

□

A.3 Proof of Proposition 2

Proof. For continuity with respect to rewards, notice that if a certain reward is not involved in the optimal solution, then any perturbation does not change V. On the other hand, if a reward is in an optimal solution, since V is defined by an expectation of the rewards, it is clear that V is continuous in that reward: a slight perturbation in the reward leads to the same or an even smaller perturbation in V due to the probability weighting.

On the other hand, V can be highly discontinuous in c and B. Suppose a cMDP has only one state and two actions with cost 0 and B, and reward 0 and $x \in \mathbb{R}_{>0}$ respectively. Then slightly increasing the cost or slightly decreasing the budget to create a new instance M moves a solution value of x all the way down to a solution value of 0. In particular, we see $V_M = x >> V_{\bar{M}}$ if we only perturb the budget slightly by some $\epsilon > 0$. □

A.4 Proof of Lemma 1

We first formally define the anytime cost of a policy π as,

$$C^\pi := \max_{h \in [H]} \max_{\tau_h \in \mathcal{H}_h : P^\pi[\tau_h] > 0} \bar{c}_h.$$

In words, C^π is the largest cost the agent ever accumulates at any time under any history.

Proof. Consider the deterministic policy π' defined by,

$$\pi'_h(\tau_h) := \max_{a \in \mathcal{A} : \pi_h(a|\tau_h) > 0} r_h(s,a) + \mathbb{E}_{c,s' \sim \mathbb{P}_h(s,a)} V_{h+1}(\tau_h, a, c, s'),$$

for every $h \in [H]$ and every $\tau_h \in \mathcal{H}_h$.

We first show that for any $\tau_h \in \mathcal{H}_h$, if $P^{\pi'}[\tau_h] > 0$ then $P^\pi[\tau_h] > 0$. This means that the set of partial histories induced by π' with non-zero probability are a subset of those induced by π. Hence,

$$C^{\pi'} = \max_{h \in [H]} \max_{\tau_h \in \mathcal{H}_h : P^{\pi'}[\tau_h] > 0} \bar{c}_h \le \max_{h \in [H]} \max_{\tau_h \in \mathcal{H}_h : P^{\pi}[\tau_h] > 0} \bar{c}_h = C^\pi.$$

We show the claim using induction on h. For the base case, we consider $h = 1$. By definition, we know that for both policies, $P^{\pi'}[s_0] = P^\pi[s_0] = 1$. For the inductive step, consider any $h \ge 1$ and suppose that $P^{\pi'}[\tau_{h+1}] > 0$. Decompose τ_{h+1} into $\tau_{h+1} = (\tau_h, a, c, s')$ and let $s = s_h$. As we have shown many times before,

$$0 < P^{\pi'}[\tau_{h+1}] = \pi'_h(a \mid \tau_h) C_h(c \mid s,a) P_h(s' \mid s,a) P^{\pi'}[\tau_h]$$

Thus, it must be the case that $\pi'_h(\tau_h) = a$ (since π' is deterministic), $C_h(c \mid s,a) > 0$, $P_h(s' \mid s,a) > 0$, and $P^{\pi'}[\tau_h] > 0$. By the induction hypothesis, we then know that $P[\tau_h] > 0$. Since by definition $\pi'_h(\tau_h) = a \in \{a' \in A \mid \pi_h(a' \mid \tau_h) > 0\}$, we then see that,

$$P[\tau_{h+1}] = \pi_h(a \mid \tau_h) C_h(c \mid s,a) P_h(s' \mid s,a) P[\tau_h] > 0$$

This completes the induction.

Next, we show that for any $h \in [H]$ and $\tau_h \in \mathcal{H}_h$, $V_h^{\pi'}(\tau_h) \geq V_h(\tau_h)$. This implies that $V_M^{\pi'} = V_1^{\pi'}(s_0) \geq V_1(s_0) = V_M$ which proves the second claim. We proceed by backward induction on h. For the base case, we consider $h = H+1$. By definition, both policies achieve value $V_{H+1}^{\pi'}(\langle\rangle) = 0 = V_{H+1}(\langle\rangle)$. For the inductive step, consider $h \leq H$. By (PE),

$$\begin{aligned} V_h^{\pi'}(\tau_h) &= r_h(s, \pi'(\tau_h)) + E_{c,s'}[V_{h+1}^{\pi'}(\tau_{h+1})] \\ &\geq r_h(s, \pi(\tau_h)) + E_{c,s'}[V_{h+1}(\tau_{h+1})] \\ &\geq E_a[r_h(s,a) + E_{c,s'}[V_{h+1}(\tau_{h+1})]] \\ &= V_h(\tau_h). \end{aligned}$$

The second line used the induction hypotheses. The third lines used the fact that the maximum value is at least any weighted average. This completes the induction.

Thus, we see that π' satisfies $C_M^{\pi'} \leq C_M$ and $V_M^{\pi'} \geq V_M$ as was to be shown. Furthermore, we see that π' can be computed from π in linear time in the size of π by just computing $V_h(\tau_h)$ by backward induction and then directly computing a solution for each partial history.

□

A.5 Proof of Theorem 1

Proof. We present a poly-time reduction from the knapsack problem. Suppose we are given n items each with a non-negative integer value v_i and weight w_i. Let B denote the budget. We construct an MDP M with $S = \{0\}$, $A = \{0,1\}$, and $H = n$. Naturally, having a single state implies the initial state is $s_0 = 0$, and the transition is just a self-loop: $P(0 \mid 0,a) = 1$ for any $a \in A$. The rewards of M correspond to the knapsack values $r_i(s,1) = v_i$. The costs of M correspond to the knapsack weights: $c_i(s,1) = w_i$. The budget remains B.

Clearly, any (possibly non-stationary) deterministic policy corresponds to a choice of items for the knapsack. By definition of the rewards and costs, $\pi_h(\langle\rangle) = 1$ if and only if the agent gets reward v_h and accrues cost c_h. Thus, there exists a deterministic π with $V_M \geq V$ and $C_M \leq B$ if and only if $\exists I \subseteq [n]$ with $\sum_{i \in I} v_i \geq V$ and $\sum_{i \in I} w_i \leq B$. From Lemma 1 if there exists a stochastic optimal policy for the cMDP with value at least V and anytime cost at most B, then there exists a deterministic policy with value at least V and anytime cost at most B. As M can be constructed in linear time from the knapsack instance, the reduction is complete. □

A.6 Proof of Theorem 2

Proof. We show that computing a feasible policies for anytime constrained cMDPs with only $d = 2$ constraints is NP-hard via a reduction from Partition. Suppose $X = \{x_1, \ldots, x_n\}$ is a set of non-negative integers. Let $\text{Sum}(X) := \sum_{i=1}^{n} x_i$. We define a simple cMDP similar to the one in the proof of Theorem 1. Again, we define $S = \{0\}$, $A = \{0,1\}$, and $H = n$. The cost function is deterministic, defined by $c_{i,h}(s,i) = x_h$ and $c_{i,h}(s,1-i) = 0$. The budgets are $B_0 = B_1 = \text{Sum}(X)/2$.

Intuitively, at time h, choosing action $a_h = 0$ corresponds to placing x_h in the left side of the partition and $a_h = 1$ corresponds to placing x_h in the right side. The total cumulative cost for each constraint corresponds to the sum of elements in each side of the partition. If both sides sum to at most $\text{Sum}(X)/2$ then it must be the case that both are exactly $\text{Sum}(X)/2$ and so we have found a solution to the Partition problem.

Formally, we show that $\exists \pi \in \Pi_M$ if and only if $\exists Y \subseteq [n]$ with $\text{Sum}(Y) = \text{Sum}(Z) = \text{Sum}(X)/2$ where $Z = X \setminus Y$.

- ($\Rightarrow$) Suppose π is a feasible deterministic policy for M (We can assume deterministic again by Lemma 1). Define $Y := \{i \mid \pi_h(s) = 0\}$ and $Z := \{i \mid \pi_h(s) = 1\}$. Since π is deterministic we know that each item is assigned to one set or the other and so $Y \cup Z = X$.

Algorithm 5 Compute $\{\bar{S}_h\}_h$

Input: (M, C, B)

$\bar{S}_1 = \{(s_0, 0)\}$
for $h = 1$ **to** $H - 1$ **do**
  $\bar{S}_{h+1} = \emptyset$
  for $(s, \bar{c}) \in \bar{S}_h$ **do**
    for $s' \in S$ **do**
      for $a \in A$ **do**
        if $P_h(s' \mid s, a) > 0$ and $\Pr_{c \sim C_h(s,a)}[\bar{c} + c \leq B] = 1$ **then**
          for $c \in C_h(s, a)$ **do**
            $\bar{S}_{h+1} \leftarrow \bar{S}_{h+1} \cup \{(s', \bar{c} + c)\}$
return $\{\bar{S}_h\}_h$.

By definition of the constraints, we have that $P_M[\sum_{h=1}^{H} c_{i,h} \leq B_i] = 1$. Since all quantities are deterministic this means that $\sum_{h=1}^{H} c_{i,h} \leq B_i$. By definition of Y and Z we further see that $\text{Sum}(Y) = \sum_{h=1}^{H} c_{0,h} \leq \text{Sum}(X)/2$ and $\text{Sum}(Z) = \sum_{h=1}^{H} c_{1,h} \leq \text{Sum}(X)/2$. Since,

$$\text{Sum}(X) = \text{Sum}(Y \cup Z) = \text{Sum}(Y) + \text{Sum}(Z) \leq \text{Sum}(X)/2 + \text{Sum}(X)/2 = \text{Sum}(X),$$

the inequality must be an equality. Using $\text{Sum}(Y) = \text{Sum}(X) - \text{Sum}(Z)$, then implies that $\text{Sum}(Y) = \text{Sum}(Z) = \text{Sum}(X)/2$ and so (Y, Z) is a solution to the partition problem.

$\hat{}$ ($\Leftarrow$) On the other hand, suppose that (Y, Z) is a solution to the partition problem. We can define $\pi_h(s) = 0$ if $h \in Y$ and $\pi_h(s) = 1$ if $h \in Z$. By definition, we see that $\text{Sum}(Y) = \sum_{h=1}^{H} c_{0,h} = \text{Sum}(X)/2 = B_0$ and $\text{Sum}(Z) = \sum_{h=1}^{H} c_{1,h} = \text{Sum}(X)/2 = B_1$. Thus, π is feasible for M.

As the construction of M can clearly be done in linear time by copying the costs and computing $\text{Sum}(X)/2$, the reduction is polynomial time. Thus, it is NP-hard to compute a feasible policy for an anytime-constrained cMDP.

Since the feasibility problem is NP-hard, it is easy to see approximating the problem is NP-hard by simply defining a reward of 1 at the last time step. Then, if an approximation algorithm yields any finite-value policy, we know it must be feasible since infeasible policies yield -1 value. Thus, any non-trivial approximately-optimal policy to an anytime-constrained cMDP is NP-hard to compute. □

B Proofs for Section 3

The complete forward induction algorithm that computes $\bar{S}$ as defined in Definition 2 is given in Algorithm 5.

Suppose $x_{h+1} \in H_{h+1}$ is any partial history satisfying $P_\pi[x_{h+1}] > 0$. By the Markov property (Equation (2.1.11) from Puterman (1994)), we have that

$$P_\pi[x_{h+1}] = \pi_h(a \mid x_h) C_h(c \mid s, a) P_h(s' \mid s, a). \tag{MP}$$

Thus, it must be the case that $x_{h+1} = (x_h, a, c, s')$ where $\pi_h(a \mid x_h) > 0$, $C_h(c \mid s, a) > 0$, and $P_h(s' \mid s, a) > 0$.

B.1 Proof of Lemma 2

We show an alternative characterization of the safe exploration state set:

$$SE_h := \left\{ (s, \bar{c}) \mid \exists \pi \; \exists x_h \in H_h : P_\pi[s_h = s, \bar{c}_h = \bar{c}] = 1 \text{ and } \forall k \in [h-1] \; P_\pi[\bar{c}_{k+1} \leq B] = 1 \right\}. \tag{3}$$

Observe that $P_\pi[s_h = s, \bar{c}_h = \bar{c}] = 1$ is equivalent to requiring that for x_h, $s_h = s$, $\bar{c}_h = \bar{c}$, and $P[x_h] > 0$.

Lemma 6. For all $h \in [H+1]$, $\bar{S}_h = SE_h$.

We break the proof into two claims.

Claim 1. For all $h \in [H+1]$, $\bar{S}_h \subseteq SE_h$.

Proof. We proceed by induction on h. For the base case, we consider $h = 1$. By definition, $\bar{S}_1 = \{(s_0, 0)\}$. For $\tau_1 = s_0$, we have $\bar{c}_1 = 0$ is an empty sum. Thus, for any $\tau \in \mathcal{T}$, $P[s_1 = s_0, \bar{c}_1 = 0 \mid \tau_1] = 1$. Also, $\mathcal{B}[h-1] = \mathcal{B}[0] = \emptyset$ and so the second condition vacuously holds. Hence, $(s_0, 0) \in SE_1$ implying that $\bar{S}_1 \subseteq SE_1$.

For the inductive step, we consider any $h > 1$. Let $(s', \bar{c}') \in \bar{S}_{h+1}$. By definition, we know that there exists some $(s, \bar{c}) \in \bar{S}_h$, $a \in A$, and $c \in \mathcal{R}$ satisfying,

$$C_h(c \mid s, a) > 0, \quad \Pr_{c \sim C_h(s,a)}[\bar{c} + c \in \mathcal{B}] = 1, \quad \bar{c}' = \bar{c} + c, \text{ and } P_h(s' \mid s, a) > 0.$$

By the induction hypothesis, $(s, \bar{c}) \in SE_h$ and so there also exists some $\tau \in \mathcal{T}$ and some $\pi_h \in \Pi_h$ satisfying,

$$P_{\pi_h}[s_h = s, \bar{c}_h = \bar{c}] = 1 \text{ and } \forall k \in [h-1], \; P_{\pi_k}[\bar{c}_{k+1} \in \mathcal{B}] = 1.$$

Overwrite $\pi_h(\tau_h) = a$ and define $\tau_{h+1} = (\tau_h, a, c, s')$. Then, by definition of the interaction with M, $P[\tau_{h+1}] = P[\tau_h]\,\pi_h(a \mid \tau_h) C_h(c \mid s, a) P_h(s' \mid s, a) > 0$. Here we used the fact that if $P_{\pi_h}[s_h = s, \bar{c}_h = \bar{c}] = 1$ then $P[\tau_h] > 0$ by the definition of conditional probability. Thus, $P_{\pi_{h+1}}[s_{h+1} = s', \bar{c}_{h+1} = \bar{c}'] = 1$. By assumption, $P_{\pi_k}[\bar{c}_{k+1} \in \mathcal{B}]$ holds for all $k \in [h-1]$. For $k = h$, we have

$$\begin{aligned} P_{\pi_h}[\bar{c}_{h+1} \in \mathcal{B}] &= P_{\pi_h}[\bar{c}_h + c_h \in \mathcal{B} \mid s_h = s, \bar{c}_h = \bar{c}] \\ &= \sum_{a'} \pi_h(a' \mid \tau_h) \Pr_{C_h(s,a')}[\bar{c} + c \in \mathcal{B}] \\ &= \Pr_{C_h(s,a)}[\bar{c} + c \in \mathcal{B}] \\ &= 1. \end{aligned}$$

The first line used the law of total probability, the fact that $P_{\pi_h}[s_h = s, \bar{c}_h = \bar{c}] = 1$, and the recursive decomposition of cumulative costs. The second line uses law of total probability on c_h. The third line follows since $\pi_h(a \mid \tau_h) = 1$ since $\pi_h(\tau_h) = a$ deterministically. The last line used the fact that $\Pr_{c \sim C_h(s,a)}[\bar{c} + c \in \mathcal{B}] = 1$. Thus, we see that $(s', \bar{c}') \in SE_{h+1}$. □

Claim 2. For all $h \in [H+1]$, $\bar{S}_h \supseteq SE_h$.

Proof. We proceed by induction on h. For the base case, we consider $h = 1$. Observe that for any $\tau \in \mathcal{T}$, the only τ_1 that has non-zero probability is $\tau_1 = s_0$ since M starts at time 1 in state s_0. Also, $\bar{c}_1 = 0$ since no cost has been accrued by time 1. Thus, $SE_1 = \{(s_0, 0)\} = \bar{S}_1$.

For the inductive step, we consider any $h > 1$. Let $(s', \bar{c}') \in SE_{h+1}$. By definition, there exists some $\tau \in \mathcal{T}$ and some $\pi \in \Pi$ satisfying,

$$P_{\pi_{h+1}}[s_{h+1} = s', \bar{c}_{h+1} = \bar{c}'] = 1 \text{ and } \forall k \in [h], \; P_{\pi_k}[\bar{c}_{k+1} \in \mathcal{B}] = 1.$$

Decompose $\tau_{h+1} = (\tau_h, a, c, s')$ where $s_h = s$ and $\bar{c}_h = \bar{c}$. Since $P_{\pi_{h+1}}[s_{h+1} = s', \bar{c}_{h+1} = \bar{c}'] = 1$, we observe that

$$0 < P[\tau_{h+1}] = P[\tau_h]\,\pi_h(a \mid \tau_h) C_h(c \mid s, a) P_h(s' \mid s, a).$$

Thus, $P_{\pi_h}[s_h = s, \bar{c}_h = \bar{c}] = 1$. Also, we immediately know that $P_{\pi_k}[\bar{c}_{k+1} \in \mathcal{B}]$ $\forall k \in [h-1]$ since any sub-history of τ_h is also a sub-history of τ_{h+1}. Hence, $(s, \bar{c}) \in SE_h$ and so the induction hypothesis implies that $(s, \bar{c}) \in \bar{S}_h$. We have already seen that $\bar{c}' = \bar{c} + c$, $C_h(c \mid s, a) > 0$ and $P_h(s' \mid s, a)$. To show that $(s', \bar{c}') \in \bar{S}_{h+1}$, it then suffices to argue that $\Pr_{c \sim C_h(s,a)}[\bar{c} + c \in \mathcal{B}] = 1$. To this end, observe as in Claim 1 that,

$$1 = P_{\pi_h}[\bar{c}_{h+1} \in \mathcal{B}] = \sum_{a'} \pi_h(a' \mid \tau_h) \Pr_{C_h(s,a')}[\bar{c} + c \in \mathcal{B}].$$

This implies that for all a', $\Pr_{C_h(s,a')}[\bar{c} + c \in \mathcal{B}] = 1$, otherwise we would have,

$$\sum_{a'} \pi_h(a' \mid \tau_h) \Pr_{C_h(s,a')}[\bar{c} + c \in \mathcal{B}] < \sum_{a'} \pi_h(a' \mid \tau_h) = 1;$$

which is a contradiction. Thus, $c + C_h(s, a) \subseteq \mathcal{B}$ and so $(s', \bar{c}') \in \bar{S}_{h+1}$. □

Proof of Lemma 2.

Proof.

First Claim. $_h$ Fix any $(s;\bar{c})$ and suppose that $P_M[s_h = s;\bar{c}_h = \bar{c}] > 0$ where 2 $_M$. Since 2 $_M$, we know that P_M $8k \in [H]$ $\sum_{t=1}^{k} c_t$ B $= 1$. Since the history distribution has nite support whenever the cost distributions do, we see for any history $_{h+1}$ 2 H_{h+1} with $P_M[\ _{h+1}] > 0$ it must be the case that $\bar{c}_{h+1}$ B. In fact, it must also be the case that $Pr_{c\ C_h(s_h;a_h)}[c + \bar{c}_h$ $B] = 1$ otherwise there exists a realization of$c$ and $\bar{c}_h$ under for which the anytime constraint would be violated.

Moreover, this must hold for any subhistory of $_{h+1}$ since those are also realized with non-zero probability under . In symbols, we see that$P_{\ k}[\bar{c}_{k+1}$ $B] = 1$ for all $k \in [h]$. Thus, $(s;\bar{c}) \in SE_h$. Since $(s;\bar{c})$ was arbitrary, we conclude by Lemma 6 that $\bar{S}_h = SE_h$ $F_{\ h}$.

observe that $|\bar{S}_1| = 1$. By the inductive de nition of $\bar{S}_{h+1}$, we see that for any $(s;\bar{c}) \in \bar{S}_h$, $(s;\bar{c})$ is responsible for adding at most $\sum_{a\in A} |C_h(s;a)|$ SAn pairs $(s';\bar{c}')$ into $\bar{S}_{h+1}$. Speci cally, each next state s', current action a, and current cost $c \in C_h(s;a)$ yields at most one new element of$\bar{S}_{h+1}$. Thus, $|\bar{S}_{h+1}|$ $SA|\bar{S}_h|$. Since $S;A;n < 1$, we see inductively that $|\bar{S}_h| < 1$ for all $h \in [H+1]$.

Second Claim. Suppose that for each $(h;s)$ there is some$a$ with $C_h(s;a) = \{0\}$. For any $(s;\bar{c}) \in SE_{h+1}$, we know that there exists some and $_{h+1}$ for which $P_{\ k}[\bar{c}_{k+1}$ $B] = 1$ for all $k \in [h]$. Now de ne the deterministic policy ^{0} by $^0_k(\ _k) = \ _k(\ _k)$ for all subhistories $_k$ of $_{h+1}$, and $^0_k(\ _k) = a$ for any a with $C_k(s_k;a) = \{0\}$ otherwise. Clearly, ^{0} never accumulates more cost than a subhistory of$_{h+1}$ since it always takes 0 cost actions after inducing a di erent history than one contained in $_{h+1}$.

Since under any subhistory of $_{h+1}$, ^{0} satis es the constraints by de nition of SE_{h+1}, we know that $^0 \in$ $_M$. We also see that$P^{^0}_M[s_{h+1} = s;\bar{c}_{h+1} = \bar{c}] > 0$ and so $(s;\bar{c}) \in F_{h+1}$. Since $(s;\bar{c})$ was arbitrary we have that $SE_{h+1} = F_{h+1}$. As h was arbitrary the claim holds. □

Observation 1. For all $h > 1$, if $(s;\bar{c}) \in \bar{S}_h$, then $\bar{c}$ B.

Proof. For any h 1, any $(s;\bar{c}) \in \bar{S}_{h+1}$ satis es $\bar{c}' = \bar{c} + c$ where $c \in C_h(s;a)$ and $Pr_{c\ C_h(s;a)}[\bar{c} + c$ $B] = 1$. Since$C_h(s;a)$ has nite support, this means that for any such $c \in C_h(s;a)$, we have that $\bar{c} + c$ B. In particular, $\bar{c}' = \bar{c} + c$ B. □

B.2 Proof of Lemma 3

The tabular policy evaluation equations(Equation 4.2.6 Puterman (1994)) naturally translate to the cost setting as follows:

$$V_h(\ _h) = \sum_{a\in A} {}_h(a \mid \ _h)\ \ r_h(s;a) + \sum_{c}\sum_{s'} C_h(c \mid s;a)P_h(s' \mid s;a)V_{h+1}(\ _h;a;c;s')\ \ :$$

We can write this more generally in the compact form:

$$V_h(\ _h) = E_{\ _h}\ \ r_h(s;a) + E_{\ _{h+1}}\ \ V_{h+1}(\ _{h+1})\ \ : \tag{PE}$$

The classicBellman Optimality Equations (Equation 4.3.2 Puterman (1994)) are,

$$\bar{V}_h(s) = \max_{a\in A\ (s)}\ r_h(s;a) + E_{s'}\ \ \bar{V}_{h+1}(s')$$

Observe that the optimality equations for $\bar{M}$ are,

$$\bar{V}_h(\bar{s}) = \max_{\bar{a}\in \bar{A}_h(\bar{s})}\ r_h(\bar{s};\bar{a}) + E_{\bar{s}'}\ \ \bar{V}_{h+1}(s')\ \ ;$$

which reduce to

$$\bar{V}_h(s;\bar{c}) = \max_{a:Pr_{c\ C_h(s;a)}[\bar{c}+c\ \ B]=1}\ r_h(s;a) + E_{c;s'}\ \ \bar{V}_{h+1}(s';\bar{c}+c)\ \ ; \tag{BE}$$

where $\overline{V}_{H+1}(s;\bar{c}) = 0$. We then define $\overline{V}_h(s;\bar{c}) = \sup V_h(s;\bar{c})$. Note, if chooses any action a for which $a \notin \overline{A}(s;\bar{c})$, then $V_h(s;\bar{c}) := -1$ and we call infeasible for $\overline{M}$.

Observation 2. For any $_h \in W_h(s;\bar{c})$, if $a \in A$ satisfies $\Pr_{c \sim C_h(s;a)}[\bar{c}+c \leq B] = 1$, $s' \in S$ satisfies $P_h(s' \mid s;a) > 0$, and $c \in C_h(s;a)$, then for $_{h+1} := (_h; a; c; s')$, $_{h+1} \in W_{h+1}(s', \bar{c}+c)$.

Proof. If $_h \in W_h(s;\bar{c})$, then there exists some $$ with,

$$P_{_h}[s_h = s; \bar{c}_h = \bar{c}] = 1 \text{ and } \forall k \in [h-1], P_{_k}[\bar{c}_{k+1} \leq B] = 1:$$

Define $_h(_h) = a$. Immediately, we see,

$$P[_{h+1}] = P[_h]\,_h(a \mid _h) C_h(c \mid s;a) P_h(s' \mid s;a) > 0;$$

so $P_{_{h+1}}[s_{h+1} = s', \bar{c}_{h+1} = \bar{c}+c] = 1$. Also, $P_{_h}[\bar{c}_{h+1} \leq B] = \Pr_{c \sim C_h(s;a)}[\bar{c}+c \leq B] = 1$ using the same argument as in Claim 1. Thus, $_{h+1} \in W_{h+1}(s', \bar{c}+c)$. □

Now, we split the proof of Lemma 3 into two claims.

Claim 3. For any $h \in [H+1]$, if $(s;\bar{c}) \in \overline{S}_h$ and $_h \in W_h(s;\bar{c})$, then $\overline{V}_h(s;\bar{c}) \geq V(_h)$.

Proof. We proceed by induction on h. For the base case, we consider $h = H+1$. Let $(s;\bar{c}) \in \overline{S}_{H+1}$ and $_{H+1} \in W_{H+1}(s;\bar{c})$. By definition, $\overline{V}_{H+1}(s;\bar{c}) = 0$. If $_M(_{H+1}) = ?$, then $V(_{H+1}) = -1 < 0 = \overline{V}_{H+1}(s;\bar{c})$. Otherwise, for any $\in _M(_{H+1})$, $V_{H+1}(_{H+1}) = 0$ by definition. Since was arbitrary we see that $V(_{H+1}) = 0 \geq 0 = \overline{V}_{H+1}(s;\bar{c})$.

For the inductive step, we consider $h \leq H$. Fix any $(s;\bar{c}) \in \overline{S}_h$ and let $_h \in W_h(s;\bar{c})$. If $_M(_h) = ?$, then $V(_h) = -1 \leq \overline{V}_h(s;\bar{c})$. Otherwise, fix any $\in _M(_h)$. Suppose $_{h+1} = (_h; a; c; s')$ where $_h(a \mid _h) > 0$, $C_h(c \mid s;a) > 0$, and $P_h(s' \mid s;a) > 0$. For any full history $\in H$ satisfying $P_{_{h+1}}[\,] > 0$, we have $P_{_h}[\,] = P_{_{h+1}}[\,]P_{_h}[_{h+1}] > 0$. Since $\in _M(_h)$, we know that for all complete histories $\in H$ with $P_{_h}[\,] > 0$ that $\bar{c}_{k+1} \leq B$ for all $k \in [H]$. Consequently, for any $\in H$ satisfying $P_{_{h+1}}[\,] > 0$, $\bar{c}_{k+1} \leq B$ for all $k \in [H]$. This means that $P_{_{h+1}}[\bar{c}_{k+1} \leq B] = 1$ for all $k \in [H]$ and so $\in _M(_{h+1})$.

By (BE),

$$\begin{aligned}
\overline{V}_h(s;\bar{c}) &= \max_{a:\Pr_{c \sim C_h(s;a)}[\bar{c}+c \leq B]=1} r_h(s;a) + E_{c;s'}\left[\overline{V}_{h+1}(s', \bar{c}+c)\right] \\
&\geq \max_{a:\Pr_{c \sim C_h(s;a)}[\bar{c}+c \leq B]=1} r_h(s;a) + E_{c;s'}\left[V_{h+1}(_{h+1})\right] \\
&\geq \sum_{a:\Pr_{c \sim C_h(s;a)}[\bar{c}+c \leq B]=1} {}_h(a \mid _h)\left[r_h(s;a) + E_{_{h+1}}\left[V_{h+1}(_{h+1})\right]\right] \\
&\geq \sum_{a:\Pr_{c \sim C_h(s;a)}[\bar{c}+c \leq B]=1} {}_h(a \mid _h)\left[r_h(s;a) + E_{_{h+1}}\left[V_{h+1}(_{h+1})\right]\right] \\
&= \sum_{a \in A} {}_h(a \mid _h)\left[r_h(s;a) + E_{_{h+1}}\left[V_{h+1}(_{h+1})\right]\right] \\
&= V_h(_h):
\end{aligned}$$

The second line follows from the induction hypothesis, where $_{h+1} = (_h; a; c; s') \in W_{h+1}(s', \bar{c}+c)$ by Observation 2. The third line follows since the finite maximum is larger than the finite average (see Lemma 4.3.1 of Puterman (1994)). The fourth line follows since $\in _M(_{h+1})$ implies $V(_{h+1}) \geq V_{h+1}(_h)$. $\Pr_{c \sim C_h(s;a)}[\bar{c}+c \leq B] = 1$. The fifth line follows since cannot place non-zero weight on any action a satisfying $\bar{c}+C_h(s;a) > B$. Otherwise, we would have,

$$P_{_h}[\bar{c}_{h+1} > B]\ _h(a \mid _h) \Pr_{c \sim C_h(s;a)}[\bar{c}+c > B] > 0;$$

contradicting that $\in _M(_h)$. The final line uses (PE).

Since was arbitrary, we see that $\overline{V}_h(s;\bar{c}) \geq V(_h)$.

□

Claim 4. For any $h \in [H+1]$, if $(s,\bar{c}) \in \bar{S}_h$ and $\pi_h \in W_h(s,\bar{c})$, then $\bar{V}_h(s,\bar{c}) \geq V^{\pi}(\pi_h)$.

Proof. We proceed by induction on h. For the base case, we consider $h = H+1$. If $(s,\bar{c}) \in \bar{S}_{H+1}$ and $\pi_{H+1} \in W_{H+1}(s,\bar{c})$, then by definition there exists some $\pi \in \Pi$ for which $P^{\pi}[\pi_{H+1}] > 0$ and $P^{\pi}_k[\bar{c}_{k+1} \leq B] = 1$ for all $k \in [H]$. We saw in the proof of Claim 2 that for any $k \leq H$, $(s_{k+1},\bar{c}_{k+1}) \in \bar{S}_{k+1}$. Thus, by Observation 1, we have $\bar{c}_{k+1} \leq B$ for all $k \in [H]$ which implies that $P^{\pi}_{H+1}[\bar{c}_{k+1} \leq B] = 1$ for all $k \in [H]$. Hence, $\pi \in \Pi_M(\pi_{H+1})$ is a feasible solution to the optimization defining $V^{\pi}(\pi_{H+1})$ implying that $V^{\pi}(\pi_{H+1}) = 0$. By definition, we also have $\bar{V}_{H+1}(s,\bar{c}) = 0 \geq V^{\pi}(\pi_{H+1})$.

For the inductive step, we consider $h \leq H$. Let $(s,\bar{c}) \in \bar{S}_h$ and $\pi_h \in W_h(s,\bar{c})$. Consider the deterministic optimal partial policy π for $\bar{M}$ defined by solutions to (BE). Formally, for all $t \geq h$,

$$\pi_t(s,\bar{c}) \in \operatorname*{arg\,max}_{a:\Pr_{C_t}(s,a)[\bar{c}+c \leq B]=1} \left[r_t(s,a) + \mathbb{E}_{c,s'}\left[\bar{V}^{\pi}_{t+1}(s',\bar{c}+c) \right] \right].$$

If there is no feasible action for any of these equations of form $\pi_t(s',\bar{c}')$ where $t \geq h$ and $(s',\bar{c}') \in \bar{S}_t$ are reachable from $(h,s,\bar{c})$ with non-zero probability, then $\bar{V}^{\pi}_h(s,\bar{c}) = -\infty$. In this case, clearly $\bar{V}^{\pi}_h(s,\bar{c}) \leq V^{\pi}(\pi_h)$. Otherwise, suppose solutions to (BE) exist so that π is well-defined from $(s,\bar{c})$ at time h onward. It is well known (see Theorem 4.3.3 from Puterman (1994)) that $\bar{V}^{\pi}_t(s',\bar{c}') = \bar{V}^{*}_t(s',\bar{c}')$ for all $t \geq h$ and all $(s',\bar{c}') \in \bar{S}_t$. We unpack π into a partial policy π for M defined by,

$$\pi_t(\pi_t) = \begin{cases} \pi_t(s_t,\bar{c}_t) & \text{if } (s_t,\bar{c}_t) \in \bar{S}_t \\ a_1 & \text{o.w.} \end{cases}$$

Here, a_1 is an arbitrary element of A. To make π a full policy, we can define π_t arbitrarily for any $t < h$.

We first show that for all $t \geq h$, $P^{\pi}_h[(s_t,\bar{c}_t) \in \bar{S}_t] = 1$. We proceed by induction on t. For the base case, we consider $t = h$. By assumption, $\pi_h \in W_h(s,\bar{c})$ so $(s_h,\bar{c}_h) = (s,\bar{c}) \in \bar{S}_h$. Thus, $P^{\pi}_h[(s_h,\bar{c}_h) \in \bar{S}_h] = P^{\pi}_h[(s,\bar{c}) \in \bar{S}_h] = 1$.

For the inductive step, we consider $t > h$. By the induction hypothesis, we know that $P^{\pi}_h[(s_t,\bar{c}_t) \in \bar{S}_t] = 1$. By the law of total probability, it is then clear that,

$$\begin{aligned} P^{\pi}_h[(s_{t+1},\bar{c}_{t+1}) \in \bar{S}_{t+1}] &= P^{\pi}_h[(s_{t+1},\bar{c}_{t+1}) \in \bar{S}_{t+1} \mid (s_t,\bar{c}_t) \in \bar{S}_t] \\ &= \sum_{(s',\bar{c}') \in \bar{S}_t} P^{\pi}_h[(s_{t+1},\bar{c}_{t+1}) \in \bar{S}_{t+1} \mid s_t = s', \bar{c}_t = \bar{c}'] P^{\pi}_h[s_t = s', \bar{c}_t = \bar{c}']. \end{aligned}$$

Above we have used the fact that for any $(s',\bar{c}') \in \bar{S}_t$, the event that $\{s_t = s', \bar{c}_t = \bar{c}', (s_t,\bar{c}_t) \in \bar{S}_t\} = \{s_t = s', \bar{c}_t = \bar{c}'\}$.

For any π_t with $(s_t,\bar{c}_t) = (s',\bar{c}') \in \bar{S}_t$, by definition, $\pi_t(\pi_t) = \pi_t(s',\bar{c}') = a' \in \{a \in A \mid \bar{c}' + C_t(s',a) \leq B\}$. By the inductive definition of $\bar{S}_{t+1}$, we then see that $(s'', \bar{c}' + c') \in \bar{S}_{t+1}$ for any $s'' \in P_t(s',a')$ and $c' \in C_t(s',a')$. Hence, $P^{\pi}_h[(s_{t+1},\bar{c}_{t+1}) \in \bar{S}_{t+1} \mid s_t = s', \bar{c}_t = \bar{c}'] = 1$. We then see that,

$$\begin{aligned} P^{\pi}_h[(s_{t+1},\bar{c}_{t+1}) \in \bar{S}_{t+1}] &= \sum_{(s',\bar{c}') \in \bar{S}_t} P^{\pi}_h[s_t = s', \bar{c}_t = \bar{c}'] \\ &= P^{\pi}_h[(s_t,\bar{c}_t) \in \bar{S}_t] \\ &= 1 \end{aligned}$$

This completes the induction.

Since under π_h, π induces only histories whose state and cumulative cost are in $\bar{S}$, we see that π's behavior is identical to π's almost surely. In particular, it is easy to verify by induction using (PE) and Observation 2 that,

$$\begin{aligned} V^{\pi}_h(\pi_h) &= \mathbb{E}^{\pi}_{h}\left[r_h(s,a) + \mathbb{E}_{h+1}\left[V^{\pi}_{h+1}(\pi_{h+1}) \right] \right] \\ &= \mathbb{E}_{(s,\bar{c})}\left[r_h((s,\bar{c}),a) + \mathbb{E}_{(s',\bar{c}')}\left[\bar{V}^{\pi}_{h+1}(s',\bar{c}') \right] \right] \\ &= \bar{V}^{\pi}_h(s,\bar{c}) \\ &= \bar{V}^{*}_h(s,\bar{c}). \end{aligned}$$

By Observation 1, we see if $(s_{k+1}, \bar{c}_{k+1}) \in \bar{S}_{k+1}$ then $\bar{c}_{k+1} \in B$. It is then clear by monotonicity of probability that $P_h[c_{k+1} \in B] \geq P_h[(s_{k+1}, c_{k+1}) \in \bar{S}_{k+1}] = 1$ for all $k \in [H]$. Hence, $\pi \in \Pi_M(\pi_h)$ and so $\bar{V}_h(s, \bar{c}) = V_h(\pi_h) \geq V(\pi_h)$.

□

Observation 3 (Cost-Augmented Probability Measures). We note we can treat π defined in the proof of Claim 4 as a history dependent policy in the same way we defined. Doing this induces a probability measure over histories. We observe that measure is identical as the one induced by the true history-dependent policy π. Thus, we can directly use augmented policies with $\bar{M}$ and reason about their values and costs with respect to M.

B.3 Proof of Theorem 3

Proof. From Lemma 3, we see that $V^* = V(s_0) = \bar{V}_1(s_0, 0) = \bar{V}^*$. Furthermore, in Claim 4, we saw the policy defined by the optimality equations (BE) achieves the optimal value, $\bar{V}^{\pi} = \bar{V}^* = V^*$. Furthermore, π behaves identically to a feasible history-dependent policy almost surely. In particular, as argued in Claim 4 both policies only induce cumulative costs appearing in $\bar{S}_h$ at any time h and so by Observation 1 we know that both policies' cumulative costs are at most B anytime. □

B.4 Proof of Corollary 2

The theorem follows immediately from Theorem 3 and the argument from the main text.

B.5 Proof of Proposition 3

Proof. By definition of $\bar{S}$, it is clear that $|\bar{S}_h| \leq |S| D$, and by inspection we see that $|\bar{A}| \leq |A|$. The agent can construct $\bar{S}$ using our forward induction procedure, Algorithm 5, in $O(\sum_{h=1}^{H-1} |\bar{S}_h| |S| A n) = O(HS^2AnD)$ time. Also, the agent can compute $\bar{P}$ by forward induction in the same amount of time so long as the agent only records the non-zero transitions. Thus, $\bar{M}$ can be computed in $O(HS^2AnD)$ time.

1. By directly using backward induction on $\bar{M}$ Puterman (1994), we see that an optimal policy can be computed in $O(H|\bar{S}|^2|\bar{A}|) = O(HS^2AD^2)$ time. However, this analysis can be refined: for any sub-problem of the backward induction $(h, (s, \bar{c}))$ and any action a, there are at most nS state-cost pairs that can be reached in the next period (namely, those of the form $(s', \bar{c} + c)$) rather than SD. Thus, backward induction runs in $O(HS^2AnD)$ time, and so planning in total can be performed in $O(HS^2AnD)$ time.
2. Similarly, PAC (probably-approximately correct) learning can be done with sample complexity $\tilde{O}(H^3|\bar{S}||\bar{A}| \log(\frac{1}{\delta})/\varepsilon^2) = \tilde{O}(H^3SAD \log(\frac{1}{\delta})/\varepsilon^2)$ Menard et al. (2021), where δ is the confidence and ε is the accuracy. Note, we are translating the guarantee to the non-stationary state set setting which is why the $|\bar{S}|$ term becomes SD instead of HSD.

□

B.6 Proof of Lemma 4

Proof. Suppose each cost is represented with k bits of precision. For simplicity, we assume that k includes a possible sign bit. By ignoring insignificant digits, we can write each number in the form $2^i b_{-i} + \cdots + 2^{-1} b_{-1} + 2^0 b_0 + \cdots + 2^{k-i-1} b_{k-i}$ for some i. By dividing by 2^{-i}, each number is of the form $2^0 b_0 + \cdots + 2^{k-1} b_{k-1}$. Notice, the largest possible number that can be represented in this form is $\sum_{i=0}^{k-1} 2^i = 2^k - 1$. Since at each time h, we potentially add the maximum cost, the largest cumulative cost ever achieved is at most $2^k H - 1$. Since that is the largest cost achievable, no more than $2^k H$ can ever be achieved through all H times. Similarly, no cost can be achieved smaller than $-2^k H$.

Thus each cumulative cost is in the range $[-2^k H + 1, 2^k H - 1]$ and so at most $2^{k+1} H$ cumulative costs can ever be created. By multiplying back the 2^{-i} term, we see at most $2^{k+1} H$ costs are ever generated by numbers with k bits of precision. Since this argument holds for each constraint independently, the total number of cumulative cost vectors that could ever be achieved is $(H 2^{k+1})^d$. Hence, $D \leq H^d 2^{(k+1)d}$. □

B.7 Proof of Theorem 4

Theorem 4 follows immediately from Proposition 3, Lemma 4, and the definition of fixed-parameter tractability Downey and Fellows (2012).

C Proofs for Section 4

For any h we let $\hat{c}_{h+1} := f(\tau_{h+1})$ be a random variable of the history defined inductively by $\hat{c}_1 = 0$ and $\hat{c}_{k+1} = f_k(\hat{c}_k, c_k)$ for all $k \leq h$. Notice that since f is a deterministic function, $\hat{c}_k$ can be computed from τ_{h+1} for all $k \in [h+1]$. Then, a probability distribution over $\hat{c}$ is induced by the one over histories. As such, approximate-cost augmented policies can also be viewed as history-dependent policies for $\hat{M}$ as in Observation 3.

C.1 Proof of Lemma 5

Proof. We proceed by induction on h. Fix any feasible policy π for $\hat{M}$. For the base case, we consider $h = 1$. By definition, $\bar{c}_1 = 0 = \hat{c}_1$ and so the claim trivially holds. For the inductive step, we consider any $h \geq 1$. By the induciton hypothesis, we know that $\hat{c}_h \leq \bar{c}_h \leq \hat{c}_h + (h-1)\ell$ or $\hat{c}_h, \bar{c}_h \geq B - (H - h + 1)c^{\max}$ almost surely. We split the proof into cases.

1. First, suppose that $\hat{c}_h \leq \bar{c}_h \leq \hat{c}_h + (h-1)\ell$.
   (a) Furthermore, suppose that $\hat{c}_h + c_h \leq B - (H-h)c^{\max}$ so that $\hat{c}_{h+1} = f_h(\hat{c}_h, c_h) = \hat{c}_h + \frac{c_h}{\ell}\ell$. By definition of the floor function, $\frac{c_h}{\ell}\ell \leq c_h$. Thus,
   $$\hat{c}_{h+1} \leq \hat{c}_h + \frac{c_h}{\ell}\ell = \hat{c}_h + c_h \leq \bar{c}_h + c_h = \bar{c}_{h+1},$$
   holds almost surely, where we used the inductive hypothesis with our case assumption to infer that $\hat{c}_h \leq \bar{c}_h$ almost surely in the second inequality. Also, by definition of the floor function, $\frac{c_h}{\ell}\ell > \frac{c_h}{\ell}\ell - 1$. We then see that,
   $$\bar{c}_{h+1} = \bar{c}_h + \frac{c_h}{\ell}\ell \leq \bar{c}_h + \left(\left\lfloor \frac{c_h}{\ell} \right\rfloor + 1\right)\ell \leq \hat{c}_h + (h-1)\ell + \left\lfloor \frac{c_h}{\ell} \right\rfloor \ell + \ell = \hat{c}_{h+1} + h\ell.$$
   The first inequality used the induction hypothesis with our case assumption and the second used the property of floors.
   (b) Now, suppose that $\hat{c}_h + c_h < B - (H-h)c^{\max}$ so that $\hat{c}_{h+1} = f_h(\hat{c}_h, c_h) = \left\lfloor \frac{B - (H-h)c^{\max}}{\ell} \right\rfloor \ell$.
      i. If $\bar{c}_{h+1} \leq \hat{c}_{h+1}$, then by definition we have,
      $$\bar{c}_{h+1}, \hat{c}_{h+1} \leq \frac{B - (H-h)c^{\max}}{\ell}\ell \leq B - (H-h)c^{\max},$$
      and we are done.
      ii. Otherwise, if $\hat{c}_{h+1} < \bar{c}_{h+1}$, then we see that,
      $$\bar{c}_{h+1} = \bar{c}_h + c_h \leq \hat{c}_h + (h-1)\ell + c_h < B - (H-h)c^{\max} + (h-1)\ell$$
      $$\left(\frac{B - (H-h)c^{\max}}{\ell} + 1\right)\ell + (h-1)\ell = \hat{c}_{h+1} + h\ell;$$
      where the first inequality used the induction hypothesis with our case assumption.
2. Lastly, suppose that $\bar{c}_h, \hat{c}_h \geq B - (H-h+1)c^{\max}$. Then, it is clear that,
   $$\bar{c}_{h+1} = \bar{c}_h + c_h \geq \bar{c}_h + c^{\max} \geq B - (H-h+1)c^{\max} + c^{\max} = B - (H-h)c^{\max}.$$
   Similarly, we see that either,
   $$\hat{c}_{h+1} = \hat{c}_h + \left\lfloor \frac{c_h}{\ell} \right\rfloor \ell \geq \hat{c}_h + c_h \geq \hat{c}_h + c^{\max} \geq B - (H-h+1)c^{\max} + c^{\max} = B - (H-h)c^{\max},$$
   or,
   $$\hat{c}_{h+1} = \frac{B - (H-h)c^{\max}}{\ell}\ell \geq B - (H-h)c^{\max}.$$

This completes the induction.

We next show the second claim. By definition, any approximate cost is an integer multiple of ℓ where the integer is in the range $\{\lfloor \frac{B-Hc^{\max}}{\ell} \rfloor, \ldots, \lfloor \frac{B}{\ell} \rfloor\}$. The number of elements in this set is exactly,

$$\left\lfloor \frac{B}{\ell} \right\rfloor - \left\lfloor \frac{B - Hc^{\max}}{\ell} \right\rfloor + 1 \leq \frac{B}{\ell} - (\frac{B - Hc^{\max}}{\ell} - 1) + 1 = \frac{Hc^{\max}}{\ell} + 2.$$

When there are d constraints, this analysis applies to each separately since we do vector operations component-wise. Thus, the total number of approximate costs is $(\frac{H\|c^{\max}\|_1}{\ell} + 2)^d$.

□

C.2 Proof of Theorem 5

Proof. We first note that the same argument used to prove Theorem 3 immediately extends to the approximate MDP and implies that any feasible π for $\hat{M}$ satisfies $\mathbb{P}^{\pi}_{\hat{M}}[\forall t \in [H],\ \hat{c}_{t+1} \leq B] = 1$. Also, we note since $\hat{c}$ is a deterministic function of the history, we can view any policy π for $\hat{M}$ as a cost-history-dependent policy for M similar to in the proof of Observation 3. Thus, Lemma 5 implies that for any feasible π for $\hat{M}$ and any $h \in [H+1]$, $\mathbb{P}^{\pi}_{M}[\hat{c}_h \leq \bar{c}_h \leq \hat{c}_h + (h-1)\ell \vee \bar{c}_h, \hat{c}_h \leq B - (H-h+1)c^{\max}] = 1$. Since $\hat{c}_{h+1} \leq B$ a.s., we immediately see that $\mathbb{P}^{\pi}_{M}[\bar{c}_{h+1} \leq B + h\ell] = 1$ for all $h \in [H]$.

Furthermore, we observe that any feasible policy π for the anytime constraint is also feasible for $\hat{M}$ since $\mathbb{P}^{\pi}_{M}[\hat{c}_h \leq \bar{c}_h \vee \bar{c}_h, \hat{c}_h \leq B - (H-h+1)c^{\max}] = 1$ implies that $\mathbb{P}^{\pi}_{M}[\hat{c}_{h+1} \leq B] = 1$ since $\bar{c}_{h+1} \leq B$ almost surely. Since the rewards of $\hat{M}$ only depends on the state and action, we see π achieves the same value in both MDPs. Thus, $\hat{V}^* \geq V^*$.

Lastly, Lemma 5 implies that $D_{\hat{M}} \leq (\frac{H\|c^{\max}\|_1}{\ell} + 2)^d$ which with Proposition 3 gives the storage complexity. □

C.3 Proof of Corollary 3

Proof. The proof is immediate from Theorem 5 and Proposition 3. □

C.4 Proof of Corollary 4

Proof. The proof is immediate from Theorem 5 and Proposition 3. □

C.5 Proof of Corollary 5

First observe that if $B < 0$ then the instance is trivially infeasible which can be determined in linear time. Otherwise, the immediate cost (in addition to the cumulative cost) induced by any feasible π is always in the range $[0, B]$. Specifically, the larger costs xB can never be accrued since there are no negative costs now to offset them, so we can effectively assume that $c^{\max} \leq B$. Since the floor of any non-negative number is non-negative, the integer multiples of ℓ needed are now in the range $[0, \lfloor c^{\max}/\ell \rfloor] \subseteq [0, \lfloor B/\ell \rfloor]$. Thus, we have $O(\frac{Hc^{\max}}{\epsilon})$ approximate costs for the additive approximation since $\ell = \epsilon/H$, and $O(\frac{H}{\epsilon})$ approximate costs for the relative approximation since $\ell = \epsilon B/H$. The complexities are reduced accordingly.

C.6 Proof of Proposition 4

Proof. Note that computing an optimal-value, ϵ-additive solution for the knapsack problem is equivalent to just solving the knapsack problem when $\epsilon < 1$. In particular, since each weight is integer-valued, if the sum of the weights is at most $B + \epsilon < B + 1$ then it is also at most B. By scaling the weights and budget by $\lceil 2\epsilon \rceil$, the same argument implies hardness for $\epsilon \geq 1$.

For relative approximations, we present a reduction from Partition to the problem of finding an optimal-value, ϵ-relative feasible solution to the knapsack problem with negative weights. Again, we focus on the $\epsilon < 1$ regime but note the proof extends using scaling. Let $X = \{x_1, \ldots, x_n\}$ be the set of positive integers input to the partition problem and $Sum(X) := \sum_{i=1}^{n} x_i$. Observe that $Sum(X)/2$ must be an integer else the instance is

trivially a "No" instance. Define $v_i = 2x_i$ and $w_i = 2x_i$ for each $i \in [n]$. Also, we define a special item 0 with $v_0 = -Sum(X)$ and $w_0 = -Sum(X)$. We define the budget to be $B = 1$. We claim that there exists some $Y \subseteq [n]$ with $Sum(Y) = Sum(\overline{Y}) = Sum(X)/2$ if and only if there exists an $I \subseteq [n] \cup \{0\}$ with $\sum_{i \in I} v_i \geq 0$ and $\sum_{i \in I} w_i \leq B(1+\epsilon)$.

- $[\implies]$ if Y is a solution to Partition, then we define $I = Y \cup 0$. We observe that,

$$\sum_{i \in I} v_i = -Sum(X) + 2\sum_{i \in S} x_i = -Sum(X) + 2Sum(X)/2 = 0.$$

  Similarly, $\sum_{i \in I} w_i = 0 < 1 \leq B(1+\epsilon)$. Thus, I satisfies the conditions.

- $[\impliedby]$ if I is an ϵ-relative feasible solution to Knapsack, observe that I must contain 0. In particular, each $w_i = 2x_i \geq 2 > (1+\epsilon) = B(1+\epsilon)$ and so for approximate feasibility to hold it must be the case that a negative weight was included. Let $Y = I \setminus 0$. Then, we see that,

$$0 \leq \sum_{i \in I} v_i = -Sum(X) + 2\sum_{i \in Y} x_i = -Sum(X) + 2Sum(Y).$$

  Thus, $Sum(Y) \geq Sum(X)/2$. Similarly,

$$1 + \epsilon \geq \sum_{i \in I} w_i = -Sum(X) + 2Sum(Y).$$

  Thus, $Sum(Y) \leq Sum(X)/2 + (1+\epsilon)/2 < Sum(X)/2 + 1$ since $\epsilon < 1$. Because $Sum(Y)$ is a sum of positive integers, and $Sum(X)/2$ is a positive integer, it must be the case that $Sum(Y) \leq Sum(X)/2$. Pairing this with $Sum(Y) \geq Sum(X)/2$ implies equality holds. Thus, Y is a solution to Partition.

Since the transformation can be made in linear time, we see that the reduction is polynomial time. Since Partition is NP-hard, we then see finding an optimal-value, ϵ-relative feasible solution to the knapsack problem with negative weights is NP-hard.

□

C.7 Proof of Proposition 5

Proof. The proof is immediate from Corollary 3 and Corollary 4. □

D Extensions

D.1 Generalized Anytime Constraints

Consider the constraints of the form,

$$\mathbb{P}^{\pi}_{M}\left[\forall k \in [H], \sum_{t=1}^{k} c_t \in [L_k, U_k]\right] = 1. \tag{4}$$

All of our exact methods carry over to this more general setting by simply tweaking the safe exploration set. In particular, we define,

$$\overline{\mathcal{S}}_{h+1} := \Big\{(s', \bar{c}') \in \mathcal{S} \times \mathbb{R}^d \mid \exists (s, \bar{c}) \in \overline{\mathcal{S}}_h, \exists a \in \mathcal{A}, \exists c \in C_h(s,a), \\ \bar{c}' = \bar{c} + c, \underset{c \sim C_h(s,a)}{\Pr}[c + \bar{c} \in [L_h, U_h]] = 1,\ P_h(s' \mid s, a) > 0\Big\}. \tag{5}$$

Similarly, each quantity in the analysis changes to consider the different intervals per time step. The proof is otherwise identical.

For the approximate methods, the additive results imply the costs are at most $U_k + \epsilon$ anytime, and since the costs are independent of the new restrictions, the complexity guarantees are the same. We could similarly give an approximation concerning the lower bound by using pessimistic costs. For the relative approximation, we now define ℓ with respect to $|U^{min}| = \min_k |U_k|$ and all costs should lie below $|U^{min}|$. The guarantees then translate over with $|U^{min}|$ taking the role of $|B|$.

D.2 General Almost-Sure Constraints

General almost-sure constraints require that,

$$\mathbb{P}_M^\pi \left[\sum_{t=1}^{H} c_t \leq B \right] = 1. \tag{6}$$

This can be easily captured by the generalized anytime constraints by making L_k smaller than $kc^{\min}$ and U_k larger than $kc^{\max}$ for any $k < H$ so that the process is effectively unconstrained until the last time step where $U_H = B$.

Observe then when applying our relative approximation, $U^{min} = U_H = B$ and so the guarantees translate similarly as to the original anytime constraints. In particular, although $c^{\max} \leq |B|$, the cumulative cost could be up to $H|B|$. This means the multiples of ℓ that need to be considered are in the set $\left\{ \lfloor -xH^2/\epsilon \rfloor, \ldots, \lfloor xH^2/\epsilon \rfloor \right\}^d$. This changes the exact constants considered, but the asymptotic guarantees are the same. We do note however that the improvements in Corollary 5 do not extend to the general almost-sure case.

On the other hand, the additive approximation results now have $\|2Hc^{\max} - B\|_\infty$ terms instead of $\|c^{\max}\|_\infty$ terms. The asymptotic bounds then have $\|c^{\max} - B/H\|_\infty$ terms.

D.3 Infinite Discounting

If the rewards and costs are discounted, it is easy to see that Theorem 3 still holds but the resultant MDP has infinite states and discontinuous reward function. However, our approximate methods work well. By simply using the horizon H to be the earliest time in which $\sum_{t=H+1}^{\infty} \gamma^t c_t \leq \epsilon$ almost surely, we can use our reduction to get an ϵ-additive feasible policy. Pairing this with our approximation algorithms gives a computationally efficient solution. To get a desired accuracy the effective horizon H may need to be increased before using the approximation algorithms.

E Additional Experiments

For all of our experiments, we generate cMDPs built from the same structure as those in the proof of Theorem 1, but with deterministic costs and rewards chosen uniformly at random over $[0, 1]$. Formally, we consider cMDPs with a single state ($\mathcal{S} = \{0\}$), two actions ($\mathcal{A} = \{0, 1\}$), and non-stationary cost and reward functions. For all h, $r_h(s, 0) = c_h(s, 0) = 0$, and $r_h(s, 1), c_h(s, 1) \in U[0, 1]$. Despite these being worst-case hard instances, capturing the knapsack problem, we conjecture that under well-concentrated costs, such as uniform costs, that our relative approximate method Corollary 4 and no-violation methods Proposition 5 would perform well. We test this and the performance of all of our methods on these hard cMDPs. Since the complexity blowup for cMDPs is in the time horizon, we focus on testing our methods on hard instances with varying time horizons.

E.1 Approximation Scheme Performance

We first test how well our approximation scheme Algorithm 4 does compared to our exact, FPT-reduction, method Corollary 2. We see even for the fairly large choice of precision parameter, $\epsilon = .1$, that our approximation performs very well. We tested both methods on $N = 5$ hard cMDPs with fixed time horizons in the range of $1, \ldots, H_{\max} = 16$. Note, already for a single instance with $H = 15$, the exact method takes over a minute to run, which is unsurprising as the complexity grows like 2^{15}. However, we will show our approximate methods easily handle much larger instances in Appendix E.4.

Here, we consider two extreme choices of budget, $Bs = [.1, 10]$. The first is chosen so that few costs are induced before hitting the budget, so we expect even the exact method to be fast. The second is chosen to be close to the

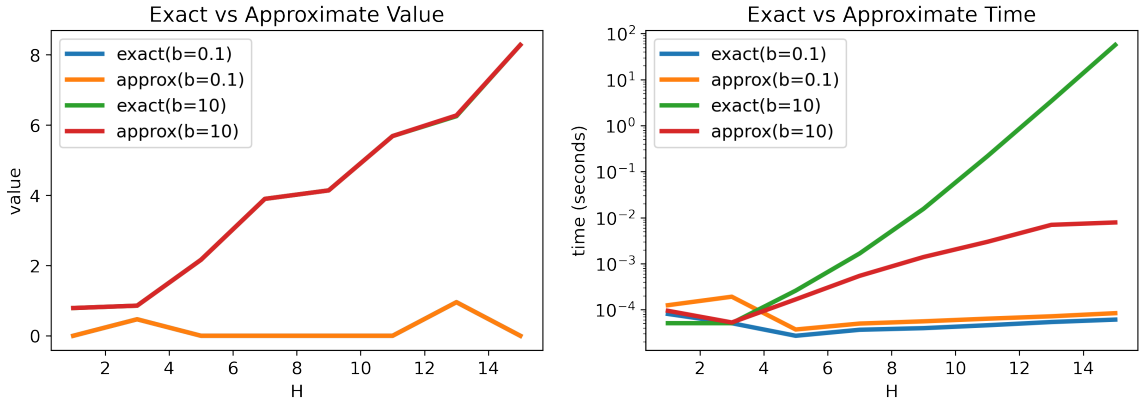

Figure 3: Our Approximation Scheme vs Our Exact Method on hard cMDP instances.

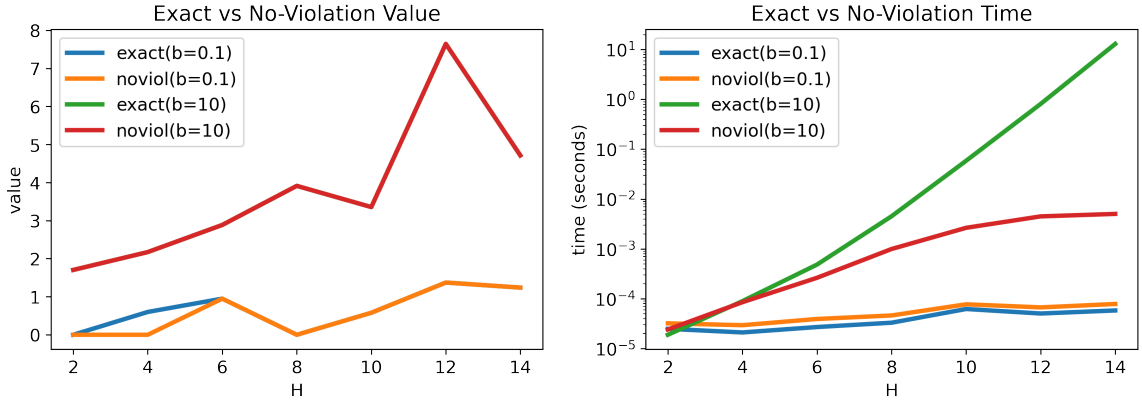

Figure 4: Our No-Violation Scheme vs Our Exact Method on hard cMDP instances.

maximum time horizon, which is an upper bound on the value ever achievable, so that many cumulative costs could be induced. We expect the exact method would run quite slowly on a large budget.

In all of our experiments, we present the worst-case for our methods. In particular, out of the N trials for the cMPDs we run on a fixed H, we report the run-time that was worst for our approximation method, and the approximate value that was closest to the exact method (the approximate value, in general, could be much larger than the exact method due to the optimistic costs Corollary 4).

The results are given in Figure 3. We see that for small budgets, the approximate method does not have much benefit and can even be slightly faster. However, for larger budgets, the approximation scheme is leagues faster, completing in one-hundredth of a second instead of over a minute and a half like the exact method. We also see in these instances the approximation method exactly matched the exact method value, which is an indicator that the approximate policies might not go over budget. We see further evidence of this in the no-violation results we see next.

E.2 No-Violation Scheme Performance

Next, we use the same setup (except $H_{\max} = 14$ now), but compare the no-violation scheme to the exact method. Here, we report the values for trials when the no-violation scheme was the furthest below the exact method (recall that, unlike the approximation scheme, the no-violation scheme is generally suboptimal). The results are summarized in Figure 4. We see in fact that the no-violation scheme gets optimal value in nearly every trial we ran! Also, the scheme is much faster than the exact method as expected.

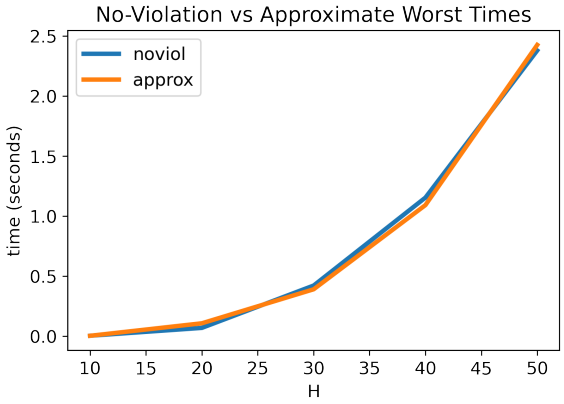

Figure 5: Our No-Violation Scheme vs Our Approximation Scheme on hard cMDP instances.

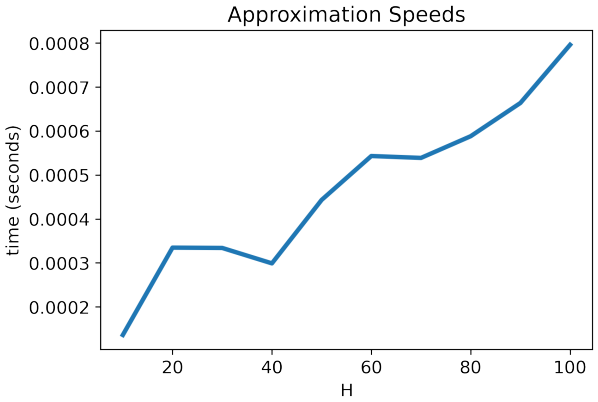

Figure 6: Our Approximation Scheme on large, hard cMDP instances.

E.3 Approximation vs No-Violation Scheme

Since the exact method is too slow to handle instances with $H = 20$, we need another way to test the efficacy of the approximation and no-violation schemes. In fact, we can just compare them to each other. Since the approximation scheme is always at least the optimal value, and the no-violation scheme is at most the optimal value when they coincide both schemes are optimal. Furthermore, since the no-violation scheme does not violate the budget, if they coincide it gives evidence that the approximate method is producing policies that are within budget anytime.

Now, we let $H_{\max} = 50$ and perform $N = 10$ trials for each $H \in \{10, 20, 30, 40, 50\}$. The results are given in Figure 5. We see the no-violation scheme consistently was close to the approximate value, which indicates it is achieving nearly optimal value. Furthermore, we see both methods scale as expected: roughly quadratically in H.

E.4 Approximation Scale Test

Lastly, we wanted to see how large an instance the approximation scheme (and equivalently the no-violation scheme), could handle. We tested the scheme with $H_{\max} = 100$. Specifically, we did $N = 10$ trials for each $H \in \{10, 20, \ldots, 100\}$. This time, we tried both $\epsilon = .1$ and the even larger $\epsilon = 1$. The results are summarized in Figure 6. We see for $\epsilon = .01$ the quadratic-like growth is still present, and yet the scheme handled a huge horizon of 100 in a mere 2 seconds. For $\epsilon = 1$ the results are even more striking with a maximum run time of .0008 seconds and the solutions are guaranteed to violate the budget by no more than a multiple of 2! The method handles an instance 10 times larger than the exact method could and yet runs in a fraction of the time.

E.5 Code Details

We conducted our experiments using standard python3 libraries. We provide our code in a jupyter notebook with an associated database file so that our experiments can be easily reproduced. The notebook already reads in the database by default so no file management is needed. Simply ensure the notebook is in the same directory as the database folder. [4]

[4]The code can be found at https://github.com/jermcmahan/anytime-constraints.git