Proceedings of Machine Learning Research vol xxx:1–22, 2021

Reinforcement Learning with Almost Sure Constraints

Agustin Castellano ACASTE11@JHU.EDU
Hancheng Min HANCHMIN@JHU.EDU
Johns Hopkins University, Baltimore, MD, USA

Juan Bazerque JBAZERQUE@FING.EDU.UY
Universidad de la República, Montevideo, Uruguay

Enrique Mallada MALLADA@JHU.EDU
Johns Hopkins University, Baltimore, MD, USA

Abstract

In this work we address the problem of finding feasible policies for Constrained Markov Decision Processes under probability one constraints. We argue that stationary policies are not sufficient for solving this problem, and that a rich class of policies can be found by endowing the controller with a scalar quantity, so called budget, that tracks how close the agent is to violating the constraint. We show that the minimal budget required to act safely can be obtained as the smallest fixed point of a Bellman-like operator, for which we analyze its convergence properties. We also show how to learn this quantity when the true kernel of the Markov decision process is not known, while providing sample-complexity bounds. The utility of knowing this minimal budget relies in that it can aid in the search of optimal or near-optimal policies by shrinking down the region of the state space the agent must navigate. Simulations illustrate the different nature of probability one constraints against the typically used constraints in expectation.

Keywords: Constrained MDPs, Safe RL, RL for physical systems, sample-efficient learning.

1. Introduction

With the huge availability of data made possible by cheap sensors and widespread telecommunications, the control paradigm has shifted: the previous-century approach, which relied heavily on system modeling followed by careful control design is now moving towards an *improve-as-you-go* approach, in which controllers are refined on a step by step basis as more data becomes readily available. One of the main tools aiding in the design of these controllers is Reinforcement Learning (RL) (Sutton and Barto, 2018; Bertsekas, 2019). This relatively new field has seen a rebirth in recent years, obtaining outstanding performance in certain domains, particularly when the algorithms are coupled with deep neural networks (Mnih et al., 2015) and tree-search methods (Silver et al., 2016). Notwithstanding, this super-human performance has been mostly obtained on setups where *i)* the domain is virtual, *ii)* transition dynamics are fairly simple and *iii)* training is computationally-intesive. There is huge promise, however, in the potential of RL to be extended to complex real-world tasks such as autonomous transportation or robot manipulation, where *safety is paramount*.

In the subfield of Safe RL, most of the current corpus relies on adding constraints in expectation to trade-off between the conflicting goals of achieving good performance while satisfying feasibility (Geibel, 2006; Miryoosefi et al., 2019), and commonly used methods rely on primal-dual algorithms that take into consideration both the reward function and the constraints to be met (Paternain et al., 2019; Ding et al., 2020). These methods, however, typically guarantee feasibility only asymptotically—with a possibly unbounded number of constraint violations during training, something highly undesirable in safety-critical systems. Other approaches include formal verification methods (Junges et al., 2016), which first deal with computing permissive (ie. feasible) strategies, restricting the actions agents can take at each step (Jansen et al., 2020).

In this work we argue that specifying hard constraints actually aids in the development of controllers, since feasible policies are easy to find. This is similar to what some authors have done in the field of deterministic finite automata, where low-complexity policies can be found rapidly (Stefansson and Johansson, 2021). Once safe policies are learnt—or equivalently, once unsafe states and actions are identified—the search for good performance can be done over a smaller set. For real-world applications with physical systems it is critical to keep track of the number of interactions between the agent and the environment. This has led to a drive to develop sample-complexity bounds (Agarwal et al., 2020), and most recently, sample-complexity bounds for learning policies with zero and bounded constraint violations (HasanzadeZonuzy et al., 2020; Liu et al., 2021).

Paper outline: In Section 2 we formulate the problem and illustrate why stationary policies are not sufficient under this setting. Section 3 contains the main results of the paper. We show a bijection between the original MDP and one that tracks—via a quantity called *budget*—how close the agent is to violating the constraint. We show feasible policies can be completely characterized in terms of the minimal required budget, which can be obtained as the solution of a fixed point iteration. This requires, however, knowledge of the transition kernel of the MDP. In Section 4 we improve on this result by showing that the budget can be learned if one knows an approximate kernel, and give sample-complexity bounds to construct it. Numerical experiments showing the different nature of our proposed constraint as opposed to the state of the art expectation-based counterparts are presented in Section 5, and we conclude in Section 6.

2. Problem formulation

Consider a finite state space, finite action space and infinite horizon Constrained Markov Decision Process (CMDP) defined as a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, p, r, d)$ where $\mathcal{S}$ is the set of states, $\mathcal{A}$ is the set of actions, p is the kernel that specifies the conditional transition probability $p(s', r, d|s, a)$, $r \in \mathbb{R}$ is the reward and $d \in \{0, 1\}$ is a binary-valued *damage indicator* used to model constraint violations. Consider also a user-specified *total damage budget* Δ. The goal in this case is to achieve the highest return while never allowing more than Δ units of damage in a single trajectory:

$$\max_{\pi \in \Pi_H} \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} R_{t+1} \,\middle|\, S_0 = s \right] \tag{1}$$

$$\text{s.t: } P_\pi \left(\sum_{t=0}^{\infty} D_{t+1} \leq \Delta \,\middle|\, S_0 = s \right) = 1 \tag{2}$$

where the initial state s is fixed and the maximization is carried over the set of general, history-dependent policies Π_H. For now the optimization is carried over this broad set, further down the line we will discuss the adequacy of other classes of policies for (1)–(2). We assume there is an absorbing termination state such that when the system enters this state it remains there with no further reward. We also assume that the structure of the MDP is such that this state is eventually reached under any policy, a common assumption for stochastic shortest path problems (Bertsekas, 2012).

Recalling that the damage D_t is a binary-valued random variable, in essence the quantity Δ serves as a *tolerance* to damage. A feasible policy is one that—almost surely—does not allow for more than Δ total damage along a single trajectory. The harshest case corresponds to a tolerance $\Delta = 0$, in which no damage is allowed. Under the zero tolerance case, it can be shown that the safety of a particular state-action pair can be encoded in a barrier function akin to the typical action-value function Q, under which feasible policies and high-return policies can be learned in parallel (Castellano et al., 2021).

In the original formulation (1)–(2) the maximization is carried out over the broad class of history-dependent policies Π_H. We define the history at time t as the collection of (S, A, S', R, D) tuples up to time t, that is $h_t = (s_0, a_0, r_1, d_1, s_1, \dots, s_{t-1}, a_{t-1}, r_t, d_t, s_t)$. Policies in this class induce a probability distribution over the set of actions conditioned on the history, i.e. $\pi(\cdot|h_t) : \mathcal{A} \to [0, 1]$.

The class of general policies is a very large set to work with, with the combination of possible histories growing exponentially as time increases. It is desirable, then, to avoid working with history-dependent policies and restrict the optimization over a simpler class that still attains optimal performance. Generally, the class of stationary policies $\pi(\cdot|s_t)$ is considered, in which the distribution over the actions is just a function of the current state.

It is a well-established fact that for unconstrained problems the stationary policies are *adequate*, in the sense that they can fully mimic the expected return obtained by any general, history-dependent policy (Altman, 1999). This result carries over to constrained problems where the constraint is cast as the expected value of a sum.

It is easy to check that the set of stationary policies is not adequate for solving (1)–(2), which is argued in the following proposition.

Proposition 1 (Stationary policies are not adequate for $\mathcal{M}$) *The set of stationary policies is not adequate for solving* (1)–(2).

Proof *As a proof by counterexample, consider the MDP of Figure 1. The episode starts with*

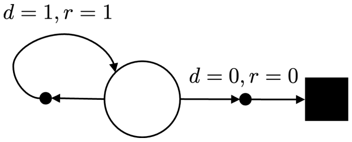

Figure 1: Example MDP: the episode starts in the circle state and ends upon reaching the square.

the agent in the white circle state and ends when the black square is reached. The two possible actions are `left` *and* `right`*. The optimal policy picks* `left` Δ *times (accruing both reward and damage) and then goes* `right`*. It is clear that this policy is non-stationary. Moreover, the only feasible stationary policy is the one that always picks* `right`*, obtaining the least return. The preceding example provides a hint on why general or history-dependent policies work for solving* (1)–(2)*: they keep track of both the rewards (useful for maximization) and the damage incurred so far (needed for feasibility). This fact will be used as a building block towards what will be developed in the next section. Namely, that endowing the controller with memory of the accumulated damage so far is sufficient for learning optimal behavior.* ■

3. Safe reinforcement learning with memory policies

Throughout this section we argue that in order to learn an optimal policy for (1)–(2) it suffices to consider the class of stationary policies that keep track of the accumulated damage along the trajectory. To this end, we consider an augmented MDP with a new state variable K_t that incorporates the accumulated damage so far, which we call budget, and show it to be equivalent to the original MDP. We finalize by showing stationary policies in the augmented MDP are adequate.

Definition 2 (Budget) *For the original MDP $\mathcal{M}$ define the budget at time t as the random variable*

$$K_t = \Delta - \sum_{\ell=0}^{t-1} D_{\ell+1} \qquad \forall\, t \geq 1 \tag{3}$$

with $K_0 = \Delta$.

This term can be seen as the *remaining damage budget*, that is to say, how many more units of damage the agent can suffer along the trajectory while still satisfying (2). From the definition in (3) it follows that $K_{t+1} = K_t - D_{t+1}$, so the budget between successive time steps either stays the same or decreases by one only if damage occurs. We argue that this magnitude K_t is a sufficient statistic for learning an optimal (feasible) policy, in the sense that stationary policies are adequate for a new MDP $\tilde{\mathcal{M}}$ with state variable $\tilde{S} = (S, K)$, which we define next.

Definition 3 (Augmented MDP) *Given transition tuples $(S_t, A_t, S_{t+1}, R_{t+1}, D_{t+1})$ from $\mathcal{M}$, define the augmented MDP $\tilde{\mathcal{M}}$ as the one with tuples $(\tilde{S}_t, A_t, \tilde{S}_{t+1}, R_{t+1}, \tilde{D}_{t+1})$ where*

$$\tilde{S}_t = (S_t, K_t)\,, \qquad\qquad \tilde{D}_{t+1} = \mathbf{1}\{K_t - D_{t+1} < 0\}\,. \tag{4}$$

In $\tilde{\mathcal{M}}$ the state space is enlarged so as to consider the remaining damage budget K_t. Therefore the states $\tilde{S}$ now lie in $\tilde{\mathcal{S}} = \mathcal{S} \times \{\Delta, \Delta-1, \ldots, 0\}$. The binary damage signal $\tilde{D}_{t+1}$ is only one when the system is out of budget—i.e., it signifies failure to comply with (2). Figure 2 depicts the structure of this modified MDP. Each blob corresponds to a slice of the state space for fixed K, with transitions between states on the same slice occurring as long as the original damage signal $D_{t+1} = 0$. When $D_{t+1} = 1$ in the original MDP, the transition in the augmented MDP corresponds to decreasing K_t by one. At the slice $\mathcal{S} \times \{0\}$ the system is critically compromised—performing one more unsafe state transition leads to failure (encoded as $\tilde{D}_{t+1} = 1$ in the augmented MDP).

Consider the following optimization problem on $\tilde{\mathcal{M}}$:

$$\max_{\tilde{\pi} \in \tilde{\Pi}_H} \mathbb{E}_{\tilde{\pi}, \tilde{\mathcal{M}}} \left[\sum_{t=0}^{\infty} R_{t+1} \,\middle|\, (S_0, K_0) = (s, \Delta) \right] \tag{5}$$

$$\text{s.t: } P_{\tilde{\pi}}\left(\tilde{D}_{t+1} = 0\right) = 1 \qquad \forall t \geq 0 \tag{6}$$

where the first component of the initial state $\tilde{S}_0 = (S_0, K_0)$ is the same as in (1) and the second component is the total budget Δ in the original formulation. Maximization in this case is done over the set of history-dependent policies $\tilde{\Pi}_H$, whose elements are of the form $\tilde{\pi}(\cdot|\tilde{h}_t)$ with $\tilde{h}_t = (\tilde{s}_0, a_0, r_1, \tilde{d}_1, \tilde{s}_1, \ldots, \tilde{s}_{t-1}, a_{t-1}, r_t, \tilde{d}_t, \tilde{s}_t)$. We explicitly write $\mathbb{E}_{\tilde{\pi}, \tilde{\mathcal{M}}}[\cdot]$ in (5) to denote that the expectation is taken with respect to the trajectory induced by $\tilde{\mathcal{M}}$. When there is no room for confusion we use the shorthanded version $\mathbb{E}_{\tilde{\pi}}[\cdot]$ instead.

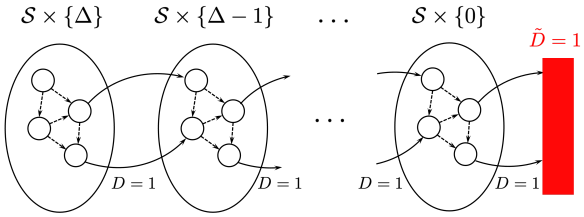
$\mathcal{S} \times \{\Delta\}$
$\mathcal{S} \times \{\Delta - 1\}$
...
$\mathcal{S} \times \{0\}$
$\tilde{D} = 1$
$D = 1$
$D = 1$
$D = 1$
$D = 1$

Figure 2: Illustration of transition dynamics in $\tilde{\mathcal{M}}$. Each disk corresponds to a partition of the state-space for fixed budget $K \in \{\Delta, \dots, 0\}$. At any time step the system either retains the current budget or decreases it by one (when $D_{t+1} = 1$), depicted by the solid arrows. Failure occurs when $K_t = 0$ and the agent encounters damage ($\tilde{D}_{t+1} = 1$ in red).

3.1. Adequacy of memory policies

We first argue that the problem in the extended and original MDPs are equivalent. Specifically, any given feasible general policy $\tilde{\pi}_h$ for $\tilde{\mathcal{M}}$ can be readily mapped to a corresponding policy in $\mathcal{M}$ and vice versa. Secondly, we claim that stationary policies are adequate for $\tilde{\mathcal{M}}$. In this sense K_t can be seen as a low-complexity descriptor of a policy, gathering all the relevant information in $\tilde{h}_t$.

Lemma 4 (Equivalence of MDPs) *The optimization problem* (5)–(6) *in the augmented MDP* $\tilde{\mathcal{M}}$ *is equivalent to the optimization problem* (1)–(2) *in the original MDP* $\mathcal{M}$ *.*

Proof [Sketch] We show a bijection $f : \Pi_H \to \tilde{\Pi}_H$ between the set of history-dependent polices for $\mathcal{M}$ and the one for $\tilde{\mathcal{M}}$ such that the expect return is matched under f. Moreover, $\pi \in \Pi_H$ is feasible for (2) if and only if $\tilde{\pi} = f(\pi)$ is feasible for (6). Hence the two optimization problems are equivalent. ■

Lemma 5 (Stationary policies are adequate for $\tilde{\mathcal{M}}$)
Proof *Given the fact that* $\tilde{D}_{t+1} \in \{0, 1\}$ *almost surely,* (6) *can be equivalently represented as* $E_{\tilde{\pi}} \left[\sum_{t=0}^{\infty} \tilde{D}_{t+1} \right] = 0$*. This constraint along with* (5) *lie in the usual formulation for shortest path problems in CMDPs, for which stationary policies are adequate (Ch.6 in Altman (1999)).* ■

We finish the section by decomposing the action-value function that rises from (5)–(6) in one term focused on return and the other focused on feasibility. Then we show that the feasible stationary policies in $\tilde{\mathcal{M}}$—and therefore the 1-memory policies in $\mathcal{M}$—can be completely characterized by this barrier.

3.2. Characterizing feasible policies with a barrier function

Consider for a stationary policy $\tilde{\pi}$ the extended action-value function:

$$Q_{\tilde{\pi}}(s, k, a) := \mathbb{E}_{\tilde{\pi}} \left[\sum_{\ell=t}^{\infty} R_{\ell+1} + \mathbb{I} \left\{ \sum_{\ell=t}^{\infty} D_{\ell+1} \leq K_t \right\} \Bigg| S_t = s, K_t = k, A_t = a \right], \tag{7}$$

where we introduce the barrier-indicator $\mathbb{I}\{x\} = 0$ if x is true and $\mathbb{I}\{x\} = -\infty$ otherwise. We can find an optimal policy for (1)–(2) by solving $\max_{\tilde{\pi}\in\tilde{\Pi}_S} \mathbb{E}_{a\sim\tilde{\pi}}\left[Q_{\tilde{\pi}}(s,\Delta,a)\right]$, where $\tilde{\Pi}_S$ is the set of stationary policies in $\tilde{\mathcal{M}}$. For simplicity, it is useful to specify a function that encodes for the feasibility of the whole state-action space under a policy. This is the barrier action-value function:

$$B_{\tilde{\pi}}(s,k,a) := \mathbb{E}_{\tilde{\pi}}\left[\mathbb{I}\left\{\sum_{l=t}^{\infty} D_{l+1} \leq K_t\right\} \,\middle|\, S_t = s, K_t = k, A_t = a\right] . \tag{8}$$

This function either takes values zero or $-\infty$, with zero indicating that policy $\tilde{\pi}$ is guaranteed to be feasible when starting from (s,k,a) and $-\infty$ meaning that, with positive probability, more than k units of damage will be seen along the trajectory. This might be a consequence of either having too small a budget or a poor policy. The usefulness for defining $B_{\tilde{\pi}}$ is that $Q_{\tilde{\pi}}$ can be decomposed in terms of itself and $B_{\tilde{\pi}}$, which decouples optimality and feasibility, as is shown next.

Lemma 6 (Barrier decomposition and Bellman equation) *Let $\tilde{\mathcal{M}}$ be an MDP with an absorbing state . Let $\tilde{\pi}$ be a policy in $\mathcal{M}$ such that under $\tilde{\pi}$ the absorbing state is eventually reached. If rewards R_{t+1} are bounded almost surely for all t, then*

$$Q_{\tilde{\pi}}(s,k,a) = Q_{\tilde{\pi}}(s,k,a) + B_{\tilde{\pi}}(s,k,a) . \tag{9}$$

Additionally, the optimal barrier function B_* satisfies the Bellman equation

$$B_*(s,k,a) = \mathbb{E}\left[\mathbb{I}\{\tilde{D}_{t+1}\} + \max_{a'\in\mathcal{A}} B_*(S_{t+1},K_{t+1},a')\right] . \tag{10}$$

This barrier function satisfies the following monotonicity properties:

$$B_\pi(s,k,a) = 0 \Longrightarrow B_\pi(s,k+i,a) = 0 \qquad \text{(Safe and more budget} \rightarrow \text{safe.)} \tag{11}$$

$$B_\pi(s,k,a) = -\infty \Longrightarrow B_\pi(s,k-i,a) = -\infty \qquad \text{(unsafe and less budget} \rightarrow \text{unsafe.)} \tag{12}$$

3.3. Characterizing feasible policies via minimal budget

As commented previously, B_* completely characterizes the feasibility of every (s,k,a) triplet. The safety of a state-action pair (s,a) is conditioned on the agent's remaining budget k. With this idea in mind we can define the minimal required budget at each (s,a) as follows.

Definition 7 (Minimal budget) *The minimal required budget k_* that guarantees feasibility for an (s,a) pair is*

$$k_*(s,a) = \min_{0\leq k\leq\infty} k \;\; \textit{s.t.: } B_*(s,k,a) = 0 \tag{13}$$

This quantity k_* serves as a proxy for safety. A state-action pair is safe if the agent's budget k is at least k_*, and thus k_* completely characterizes the set of feasible, stationary policies:

Theorem 8 (Characterization of feasible, stationary policies) *The set of feasible, stationary policies for (5)–(6) is*

$$\tilde{\Pi}_S^F = \{\tilde{\pi} : \tilde{\pi}(a|s,k) = 0 \quad \forall a : k_*(s,a) > k\}. \tag{14}$$

Proof *[Sketch] The proof is straightforward and follows from the definition of k_* and monotonicity properties (11)–(12).* ■

Notice that k_* is intrinsic to the MDP. We focus on learning this quantity, first establishing a Bellman-like recursion for k_* and then deriving an Algorithm that provably converges to it.

Theorem 9 (Recursion for k_*) *For each (s,a), the minimal budget satisfies the recursion:*

$$k_*(s,a) = \max_{s':p(s'|s,a)>0} \left[\mathbf{1}_d(s,a,s') + \min_{a'} k_*(s',a') \right] \tag{15}$$

where $\mathbf{1}_d(s,a,s') := \mathbf{1}\{p(d=1 \mid s,a,s') > 0\}$ and $\mathbf{1}\{x\} = 1$ if x is true and 0 otherwise.
Proof *[Sketch.] The proof relies on decomposing $B_*(s,k,a)$ and evaluating it on $k_*(s,a)$.* ■

The recursion in (15) can be seen as a fixed point of an operator $\mathcal{T}_p$ that acts on budgets k (for a given transition kernel $p(s',d|s,a)$). We define this operator next and analyze some of its properties.

Definition 10 (The budget operator $\mathcal{T}_p$) *Given a transition kernel p, define the operator $\mathcal{T}_p$ acting on the extended natural vector $\bar{\mathbb{N}}^{\mathcal{S}\times\mathcal{A}}$ with $\bar{\mathbb{N}} = \mathbb{N} \cup \{\infty\}$ as $\mathcal{T}_p : \bar{\mathbb{N}}^{(\mathcal{S}\times\mathcal{A})} \to \bar{\mathbb{N}}^{(\mathcal{S}\times\mathcal{A})}$:*

$$(\mathcal{T}_p\, k)(s,a) := \max_{s':p(s'|s,a)>0} \left[\mathbf{1}_d(s,a,s') + \min_{a'} k(s',a') \right] \quad \forall (s,a) \in \mathcal{S}\times\mathcal{A} \tag{16}$$

Notice that here we are making explicit the dependency of $\mathcal{T}_p$ with the transition kernel p from $\mathcal{M}$. Later we will argue that k_* can be learned even if the learner has no access to p, as long as it knows a proper surrogate kernel $\hat{p}$. In that case this notation will allow for the difference of $\mathcal{T}_p$ and $\mathcal{T}_{\hat{p}}$. However, for the remainder of this section we spare the subscript and speak just of $\mathcal{T}$.

We would like to make use of this operator to learn k_*. The idea is straightforward: for a given kernel p, start from $k = 0$ and keep applying $\mathcal{T}_p$ until reaching a fixed point. But there are many fixed points of (16). Indeed, one can easily check that if $k^\dagger$ is a fixed point so is $k^\dagger + \mathbf{1}c, c \in \bar{\mathbb{N}}$. Therefore the question still remains as to whether this procedure converges to k_*. We will show this is the case, arguing that Algorithm 1 converges to k_*.

Algorithm 1 Fixed point budget iteration

```
Input: Transition kernel $p$ from $\mathcal{M}$
Result: $k_*$ for $\mathcal{M}$
$k_0(s,a) \leftarrow 0 \quad \forall (s,a) \in \mathcal{S}\times\mathcal{A}$
for $n = 0, 1, \dots$ do
    for $(s,a) \in \mathcal{S}\times\mathcal{A}$ do
        $k_{n+1}(s,a) \leftarrow \max_{s':p(s'|s,a)>0} [\mathbf{1}\{p(d=1 \mid s,a,s') > 0\} + \min_{a'} k_n(s',a')]$
    end
end
```

Theorem 11 (Convergence to k_*) *Define $k_\infty \in \bar{\mathbb{N}}^{\mathcal{S}\times\mathcal{A}}$ as the limit of Algorithm 1, that is $k_\infty(s,a) = \lim_{n\to\infty} k_n(s,a)$ for all $(s,a) \in \mathcal{S}\times\mathcal{A}$. Let the input of Algorithm 1 be the true transition kernel p of MDP $\mathcal{M}$. Then the iterates of this algorithm converge to k_*:*

$$k_\infty(s,a) := \lim_{n\to\infty} k_n(s,a) = k_*(s,a) \quad \forall (s,a) \in \mathcal{S}\times\mathcal{A}$$

Proof *[Sketch] The proof proceeds by induction, starting from the (s,a) pairs for which $k_*(s,a) = 0$.* ■

Unfortunately, it might be possible that we do not have access to the true kernel p. The next section focuses on learning k_* as long as one knows a *surrogate* kernel $\hat{p}$.

4. Sample complexity for learning the minimal budget

We start by defining a consistent kernel $\hat{p}$ of p and show that Algorithm 1 converges to k_* under input $\hat{p}$.

Definition 12 (Consistent kernel) *Given a transition kernel p, $\hat{p}$ is a consistent kernel of p if and only if* $\operatorname{sign}\big(\hat{p}(s',d|s,a)\big) = \operatorname{sign}\big(p(s',d|s,a)\big)\;\; \forall (s,a,s',d) \in \mathcal{S}\times\mathcal{A}\times\mathcal{S}\times\{0,1\}$.

Throughout what follows we assume access to a generative model or a sampler, which allows to sample transitions $(s',d) \sim p(\cdot|s,a)$. By collecting N samples at each state-action pair, we can build an empirical model $\hat{p}$ of the transition kernel p, counting the fraction of transitions to each (s',d) from (s,a), as depicted in Algorithm 2. The number of samples needed to build a consistent kernel with arbitrarily high probability is elucidated in Lemma 13.

```
Algorithm 2 Kernel builder
Input: Transition kernel $p$ from $\mathcal{M}$, number of sample queries $N$.
Result: Empirical kernel $\hat{p}$.
for $(s,a) \in \mathcal{S}\times\mathcal{A}$ do
    Sample $N$ transitions $(s',d) \sim p(\cdot|s,a)$
end
Build estimate kernel $\hat{p}(s',d|s,a) = \frac{\text{count}(s',d;s,a)}{N}$   $\forall s' \in \mathcal{S}, d \in \{0,1\}, s \in \mathcal{S}, a \in \mathcal{A}$
```

Lemma 13 (Sample complexity for Algorithm 2) *Assume that* $p(s',d|s,a) = 0$ *or* $p(s',d|s,a) \geq \mu > 0$, *for every* $(s,a,s',d) \in \mathcal{S}\times\mathcal{A}\times\mathcal{S}\times\{0,1\}$. *Then with probability at least* $1-\delta$, `Kernel builder` *produces a consistent kernel* $\hat{p}$ *of p, provided that*

$$N \geq \frac{1}{\mu} \log \frac{2|\mathcal{S}|^2|\mathcal{A}|}{\delta}\,. \tag{17}$$

Proof *[Sketch] Follows from taking a union bound on the probability that Algorithm 2 fails to produce a consistent kernel.* ■

It turns out building a consistent kernel is sufficient in order to learn k^* under the true MDP $\mathcal{M}$.

Lemma 14 (Consistent kernels are enough) *Let p be a transition kernel associated with an MDP $\mathcal{M}$ and let $\hat{p}$ be a consistent kernel of p. Then Algorithm 1 with input $\hat{p}$ converges to the minimal budget k^* of $\mathcal{M}$.*
Proof $\mathcal{T}_{\hat{p}} \equiv \mathcal{T}_p$ *if $\hat{p}$ is consistent with p.* ■

We conclude the main body of the paper with a couple remarks: firstly that the samples needed to learn k_* are small; lastly we discuss the utility of using k_* to learn optimal policies.

Remark 15 (Learning k_* is sample-efficient.) *The last two lemmas indicate that the minimal budget k_* can be learned with very few samples. Indeed, the number of interactions with the environment is $\tilde{\mathcal{O}}\left(\frac{|\mathcal{S}||\mathcal{A}|}{\mu}\right)$ disregarding logarithmic terms. Contrast this, for example, with the $\tilde{\mathcal{O}}\left(\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^3\epsilon^2}\right)$ dependency needed to learn an ϵ-optimal policy with a generative model (Agarwal et al., 2020). There is no accuracy (i.e. ϵ) requirement in our case, nor the sample complexity depends on the effective horizon $(1-\gamma)^{-1}$: the focus is on detecting transitions rather than estimating them, which makes the problem easier, despite requiring a richer set of policies.*

Remark 16 (Using k_* to learn optimal policies) *Throughout this paper we have shown that the minimal budget k_* (which is intrinsic to the MDP) can be efficiently learned. The utility of knowing this quantity is that it characterizes the set of feasible, stationary memory-one policies (as was argued in Theorem 8), or, to put it in another way, it specifies the region of the state space that is Δ-unsafe:*

$$\mathcal{S}^{\Delta}_{\texttt{unsafe}} = \{s \in \mathcal{S} : k_*(s,a) > \Delta \ \forall a \in \mathcal{A}\}\,.$$

From this point of view knowing k_* effectively "trims" the region of interest of the MDP, meaning the search for an optimal policy is now constrained to a smaller state space, and will therefore require less samples and less computations.

5. Experiments

We illustrate the difference between the proposed constraint (2) and expectation-like constraints using the simple MDP of Figure 1. We recall the optimal policy π^*_Δ under (2) chooses action `left` Δ times, getting both Δ damage and reward, and then goes `right`. The optimal policy π^*_c under constraint $\mathbb{E}_{\pi_c}\left[\sum_{t=0}^{\infty} D_{t+1}\right] \leq c$ chooses action `left` with probability $\frac{c}{1+c}$ (this is the probability that makes the expected damage constraint hold with equality). It also achieves c expected return. The top half of Figure 3 shows a histogram of the total damage per episode for these optimal policies under different values of c, for fixed $\Delta = 5$. While the optimal policy for the probability-one constraint π^*_Δ always achieves $\Delta = 5$ damage, the damage incurred by the other policy is highly variable. This hints at one of the shortcomings of this type of constraints: even if an optimal policy can be learned, when it is deployed it can have very poor performance in terms of safety. As a second example, consider a modified version of the same MDP in which the probability of observing damage is $P(d = 1|s, \texttt{left}) = 0.6$. While the optimal policy π^*_Δ remains unchanged, now π^*_c takes `left` more frequently, now with probability $\frac{c}{P(d=1|\texttt{left})+c}$. The bottom half of Figure 3 shows the return (sum of rewards in the episode) under both optimal policies. Notice that π^*_c is essentially insensitive to c, the only difference being that slightly longer tails (not shown on the figure) are observed as c gets larger. The results for the observed damage are similar to those of Experiment I, so we omit them.

6. Conclusions

In this work we formulate the problem of Safe Reinforcement Learning under constraints that must be satisfied with probability one. The type of constraint in consideration being that the agent encounters less than Δ units of damage along a trajectory, where damage is a binary signal. We show that *i)* stationary policies are not adequate for solving this type of problems, *ii)* a sufficiently rich class of policies can be learned if one tracks the damage incurred along the trajectory. The minimal

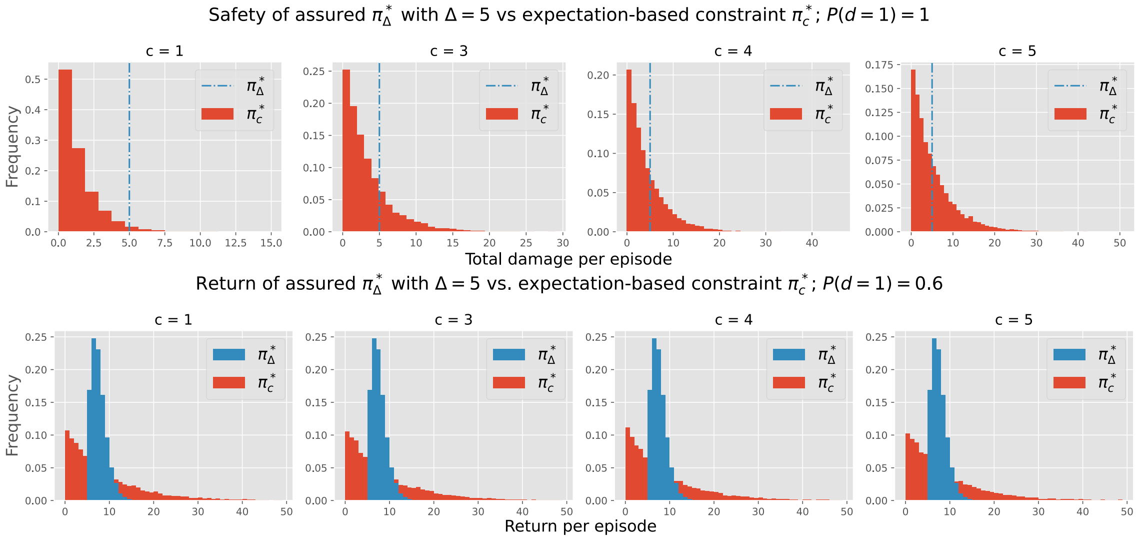

Figure 3: Top: Damage per episode for the optimal policies of the MDP of Figure 1 under different types of constraints. For each panel, the red histogram corresponds to the violations per episode for π_c^* under constraint $\mathbb{E}_{\pi_c}\left[\sum_{t=0}^{\infty} D_{t+1}\right] \leq c$. The assured policy π_Δ^* with (2) always attains $\Delta = 5$ total damage. Bottom: Returns per episode for the optimal policies of the (modified) MDP of Figure 1 where $P(d = 1|s, \texttt{left}) = 0.6$. The policy under our proposed scheme (in blue) always achieves a return of at least 5, with returns tightly concentrated around 10.

required budget (which is intrinsic to each MDP) can be learned by solving for the fixed point of a newly defined operator, provided one knows a consistent approximation of the transition probabilities. Learning this minimal budget is essentially the same as learning a set of feasible policies. Thus it simplifies the exploration for optimal or near-optimal policies, reducing it to a search within the smaller set of feasible states and actions. Our experiments illustrate in a simple setup the different nature of probability one constraints as contrasted with expectation-like constraints.

References

Alekh Agarwal, Sham Kakade, and Lin F. Yang. Model-based reinforcement learning with a generative model is minimax optimal, 2020.

Eitan Altman. *Constrained Markov decision processes*, volume 7. CRC Press, 1999.

Dimitri Bertsekas. *Dynamic programming and optimal control: Volume I*, volume 1. Athena scientific, 2012.

Dimitri Bertsekas. *Reinforcement learning and optimal control*. Athena Scientific, 2019.

Agustin Castellano, Hancheng Min, Juan Bazerque, and Enrique Mallada. Learning to act safely with limited exposure and almost sure certainty, 2021.

Dongsheng Ding, Kaiqing Zhang, Tamer Basar, and Mihailo R Jovanovic. Natural policy gradient primal-dual method for constrained markov decision processes. In *NeurIPS*, 2020.

Peter Geibel. Reinforcement learning for mdps with constraints. In *European Conference on Machine Learning*, pages 646–653. Springer, 2006.

Aria HasanzadeZonuzy, Dileep M Kalathil, and Srinivas Shakkottai. Learning with safety constraints: Sample complexity of reinforcement learning for constrained mdps. *arXiv preprint arXiv:2008.00311*, 2020.

Nils Jansen, Bettina Könighofer, Sebastian Junges, Alex Serban, and Roderick Bloem. Safe Reinforcement Learning Using Probabilistic Shields (Invited Paper). In *31st International Conference on Concurrency Theory (CONCUR 2020)*, volume 171 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 3:1–3:16, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

Sebastian Junges, Nils Jansen, Christian Dehnert, Ufuk Topcu, and Joost-Pieter Katoen. Safety-constrained reinforcement learning for mdps. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 130–146. Springer, 2016.

Tao Liu, Ruida Zhou, Dileep Kalathil, P. R. Kumar, and Chao Tian. Learning policies with zero or bounded constraint violation for constrained mdps, 2021.

Sobhan Miryoosefi, Kianté Brantley, Hal Daume III, Miro Dudik, and Robert E Schapire. Reinforcement learning with convex constraints. In *Advances in Neural Information Processing Systems*, pages 14070–14079, 2019.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

Santiago Paternain, Miguel Calvo-Fullana, Luiz FO Chamon, and Alejandro Ribeiro. Learning safe policies via primal-dual methods. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 6491–6497. IEEE, 2019.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484, 2016.

Elis Stefansson and Karl H Johansson. Computing complexity-aware plans using kolmogorov complexity. *arXiv preprint arXiv:2109.10303*, 2021.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Appendix

Proof of Lemma 4

Proof As we stated in the proof sketch, we show a bijection $f : \Pi_H \to \tilde{\Pi}_H$ between the set of history-dependent polices for $\mathcal{M}$ and the one for $\tilde{\mathcal{M}}$ such that $\forall \pi \in \Pi_H$, we have

$$\mathbb{E}_{\pi,\mathcal{M}}\left[\sum_{t=0}^{\infty} R_{t+1} \;\middle|\; S_0 = s\right] = \mathbb{E}_{\tilde{\pi},\tilde{\mathcal{M}}}\left[\sum_{t=0}^{\infty} R_{t+1} \;\middle|\; (S_0, K_0) = (s, \Delta)\right], \text{where } \tilde{\pi} = f(\pi)\,.$$

Moreover, $\pi \in \Pi_H$ is feasible for (2) if and only if $\tilde{\pi} = f(\pi)$ is feasible for (6).

First, for the MDP $\mathcal{M}$, any $\pi \in \Pi_H$ is determined by set of probability measure on the action space $\{\pi_t(\cdot|h_t) : t \geq 0\}$, where we define the history up to time t as

$$h_t = (s_0, a_0, r_1, d_1, s_1, \ldots, s_{t-1}, a_{t-1}, r_t, d_t, s_t) \in \mathcal{H}_t\,.$$

Now for the augmented MDP $\tilde{\mathcal{M}}$, the history is defined as

$$\tilde{h}_t = (s_0, k_0, a_0, r_1, \tilde{d}_1, s_1, \ldots, s_{t-1}, k_{t-1}, a_{t-1}, r_t, \tilde{d}_t, s_t) \in \tilde{\mathcal{H}}_t\,.$$

The following mapping $g_t : \mathcal{H}_t \to \tilde{\mathcal{H}}_t$ is defined from the construction of the augmented MDP

$$g(h_t) = \left(s_0, \overbrace{\Delta - \sum_{l=0}^{-1} d_l}^{k_0}, a_0, r_1, \overbrace{\mathbb{1}\{\sum_{l=0}^{-1} d_l > \Delta\}}^{\tilde{d}_0}, s_1, \ldots, s_{t-1}, \overbrace{\Delta - \sum_{l=0}^{t-1} d_l}^{k_{t-1}}, a_{t-1}, r_t, \overbrace{\mathbb{1}\{\sum_{l=0}^{t-1} d_l > \Delta\}}^{\tilde{d}_t}, s_t \right),$$

This map has well-defined inverse $g_t^{-1} : \tilde{\mathcal{H}}_t \to \mathcal{H}_t$:

$$g^{-1}(\tilde{h}_t) = \left(s_0, a_0, r_1, \overbrace{k_0 - k_1}^{d_1}, s_1, \ldots, s_{t-1}, a_{t-1}, r_t, \overbrace{k_{t-1} - k_t}^{d_t}, s_t \right)\,.$$

Now for any $\pi = \{\pi_t(\cdot|h_t) : t \geq 0\} \in \Pi_H$, let $f(\pi) = \{\pi_t(\cdot|g_t^{-1}(\tilde{h}_t)) : t \geq 0\} \in \tilde{\Pi}_H$. f has well-defined inverse $f^{-1} : \tilde{\Pi}_H \to \Pi_H$, $f^{-1}(\tilde{\pi}) = \{\tilde{\pi}_t(\cdot|g_t(h_t)) : t \geq 0\}$, as one can check:

$$\begin{aligned} f \circ f^{-1}(\tilde{\pi}) &= \{\tilde{\pi}_t(\cdot|g_t^{-1}(g_t(\tilde{h}_t)) : t \geq 0\} = \{\tilde{\pi}_t(\cdot|\tilde{h}_t) : t \geq 0\} = \pi\,, \\ f^{-1} \circ f(\pi) &= \{\pi_t(\cdot|g_t(g_t^{-1}(\tilde{h}_t)) : t \geq 0\} = \{\pi_t(\cdot|h_t) : t \geq 0\} = \tilde{\pi}\,. \end{aligned}$$

This shows $f : \Pi_H \to \tilde{\Pi}_H$ is a bijection.

Now we left to prove f preserves the expected return and feasibility. It suffices to prove that

$$p_{\pi,\mathcal{M}}(h_t|S_0 = s) = p_{f(\pi),\tilde{\mathcal{M}}}(g(h_t)|S_0 = s, K_0 = \Delta),\ \forall t \geq 0, \forall h_t \in \mathcal{H}_t\,, \tag{18}$$

that is, the distribution of the history is matched between $\mathcal{M}$ under π and $\tilde{M} = f(\pi)$ under $\tilde{\pi}$ through the bijection g. With (18), we have

$$
\begin{aligned}
&\mathbb{E}_{\pi,\mathcal{M}}\left[\sum_{t=0}^{\infty} R_{t+1} \middle| S_0 = s\right] \\
&= \sum_{t=0}^{\infty} \mathbb{E}_{\pi_t,\mathcal{M}}\left[R_{t+1} \middle| S_0 = s\right] \\
&= \sum_{t=0}^{\infty} \int_{\mathcal{H}_{t+1}} r_{t+1} p_{\pi,\mathcal{M}}(h_{t+1}|S_0 = s) dh_{t+1}
\end{aligned}
$$

(18)

$$
\begin{aligned}
&= \sum_{t=0}^{\infty} \int_{\mathcal{H}_{t+1}} r_{t+1} p_{\tilde{\pi},\tilde{\mathcal{M}}}(g(h_{t+1})|S_0 = s, K_0 = \Delta) dh_{t+1} \\
&= \sum_{t=0}^{\infty} \int_{\tilde{\mathcal{H}}_{t+1}} r_{t+1} p_{\tilde{\pi},\tilde{\mathcal{M}}}(\tilde{h}_{t+1}|S_0 = s, K_0 = \Delta) d\tilde{h}_{t+1} \\
&= \mathbb{E}_{\tilde{\pi},\tilde{\mathcal{M}}}\left[\sum_{t=0}^{\infty} R_{t+1} \middle| (S_0, K_0) = (s, \Delta)\right].
\end{aligned}
$$

Similarly for the constraint, since $\left\{\sum_{t=0}^{M-1} D_{t+1} \leq \Delta\right\}$, $M \geq 1$ is a decreasing sequence of events, we have, for $\pi \in \Pi_H$ and $\tilde{\pi} = f(\pi)$,

$$
\begin{aligned}
&\mathbb{P}_{\pi,\mathcal{M}}\left(\sum_{t=0}^{\infty} D_{t+1} \leq \Delta \middle| S_0 = s\right) \\
&= 1 - \mathbb{P}_{\pi,\mathcal{M}}\left(\sum_{t=0}^{\infty} D_{t+1} > \Delta \middle| S_0 = s\right) \\
&= 1 - \lim_{M\to\infty} \mathbb{P}_{\pi,\mathcal{M}}\left(\sum_{t=0}^{M-1} D_{t+1} > \Delta \middle| S_0 = s\right) \\
&= 1 - \lim_{M\to\infty} \int_{\mathcal{H}_M} \mathbb{1}\left\{\sum_{t=0}^{M-1} d_{t+1} > \Delta\right\} p_{\pi,\mathcal{M}}(h_M|S_0 = s) dh_M
\end{aligned}
$$

((18), and the definition of $\tilde{d}_t$)

$$
\begin{aligned}
&= 1 - \lim_{M\to\infty} \int_{\mathcal{H}_M} \tilde{d}_M p_{\tilde{\pi},\tilde{\mathcal{M}}}(g(h_M)|S_0 = s, K_0 = \Delta) dh_M \\
&= 1 - \lim_{M\to\infty} \int_{\tilde{\mathcal{H}}_M} \tilde{d}_M p_{\tilde{\pi},\tilde{\mathcal{M}}}(\tilde{h}_M)|S_0 = s, K_0 = \Delta) d\tilde{h}_M \\
&= 1 - \lim_{M\to\infty} \mathbb{P}_{\tilde{\pi},\tilde{\mathcal{M}}}\left(\tilde{D}_M = 1|S_0 = s, K_0 = \Delta\right) \\
&= 1 - \sup_M \mathbb{P}_{\tilde{\pi},\tilde{\mathcal{M}}}\left(\tilde{D}_M = 1|S_0 = s, K_0 = \Delta\right),
\end{aligned}
$$

where the last equality is due to the fact that $\tilde{D}_t \leq \tilde{D}_{t+1}, \forall t \geq 1$ almost surely, which means $\mathbb{P}_{\tilde{\pi},\tilde{\mathcal{M}}}\left(\tilde{D}_M = 1 | S_0 = s, K_0 = \Delta\right)$ is increasing with respect to M. Therefore, π is feasible,

$$\mathbb{P}_{\pi,\mathcal{M}}\left(\sum_{t=0}^{\infty} D_{t+1} \leq \Delta \,\middle|\, S_0 = s\right),$$

if and only if

$$\sup_M \mathbb{P}_{\tilde{\pi},\tilde{\mathcal{M}}}\left(\tilde{D}_M = 1 | S_0 = s, K_0 = \Delta\right) = 0\,,$$

or equivalently,

$$\mathbb{P}_{\tilde{\pi},\tilde{\mathcal{M}}}\left(\tilde{D}_t = 0 | S_0 = s, K_0 = \Delta\right) = 1, \forall t\,,$$

and this exactly means $\tilde{\pi} = f(\pi)$ is feasible.

With that, we conclude that optimization problem (1)–(2) is equivalent to

$$\begin{aligned} \max_{f(\pi)\in\tilde{\Pi}_H} \ & \mathbb{E}_{f(\pi),\tilde{\mathcal{M}}}\left[\sum_{t=0}^{\infty} R_{t+1} \,\middle|\, S_0 = s, K_0 = \Delta\right] \\ \text{s.t: } & \mathbb{P}_{f(\pi),\tilde{\mathcal{M}}}\left(\tilde{D}_t = 0 | S_0 = s, K_0 = \Delta\right) = 1, \forall t\,. \end{aligned}$$

This is exactly (5)–(6).

What is left to show is equation (18). Suppose $\tilde{\pi} = f(\pi)$. By induction, for $h_0 = (s, a_0)$, $g(h_0) = (s, \Delta, a_0)$, we have

$$p_{\pi,\mathcal{M}}(h_0|S_0 = s) = \pi_0(a_0|S_0 = s) = \tilde{\pi}_0(a_0|S_0 = s, K_0 = \Delta) = p_{\tilde{\pi},\tilde{\mathcal{M}}}(g(h_0)|S_0 = s, K_0 = \Delta)\,,$$

Now assume for some $t \geq 1$, we have

$$p_{\pi,\mathcal{M}}(h_{t-1}|S_0 = s) = p_{\tilde{\pi},\tilde{\mathcal{M}}}(g(h_{t-1})|S_0 = s, K_0 = \Delta)\,,$$

Then $\forall h_t \in \mathcal{H}_t$

$$\begin{aligned} p_{\pi,\mathcal{M}}(h_t|S_0 = s) &= p_{\pi,\mathcal{M}}(h_t|h_{t-1}) p_{\pi,\mathcal{M}}(h_{t-1}|S_0 = s) \\ &= p_{\pi,\mathcal{M}}(s_t, a_{t-1}, r_t, d_t|h_{t-1}) p_{\pi,\mathcal{M}}(h_{t-1}|S_0 = s) \\ &= \pi_t(a_{t-1}|h_{t-1}) p_{\mathcal{M}}(s_t, r_t, d_t|h_{t-1}, a_{t-1}) p_{\pi,\mathcal{M}}(h_{t-1}|S_0 = s)\,, \end{aligned}$$

For the first term, we have

$$\pi_t(a_{t-1}|h_{t-1}) = \tilde{\pi}_t(a_{t-1}|g(h_{t-1}))\,. \tag{19}$$

For the second term, we have

$$\begin{aligned} p_{\pi,\mathcal{M}}(s_t, r_t, d_t|h_{t-1}) &= p_{\mathcal{M}}(s_t, r_t, d_t|s_{t-1}, a_{t-1}, k_{t-1}) \\ &\quad \text{(From the construction of augmented MDP)} \\ &= p_{\tilde{\mathcal{M}}}(s_t, k_{t-1} - d_t, r_t, , \mathbb{1}\left\{k_{t-1} - d_t < 0\right\} | s_{t-1}, a_{t-1}, k_{t-1}) \end{aligned} \tag{20}$$

For the last term, we have, from the induction assumption,

$$p_{\pi,\mathcal{M}}(h_{t-1}|S_0 = s) = p_{\tilde{\pi},\tilde{\mathcal{M}}}(g(h_{t-1})|S_0 = s, K_0 = \Delta)\,. \tag{21}$$

Using (19)(20)(21), we have

$$\begin{aligned}
& p_{\pi,\mathcal{M}}(h_t|S_0=s) \\
&= \tilde{\pi}_t(a_{t-1}|g(h_{t-1}))p_{\tilde{\mathcal{M}}}(s_t, r_t, k_{t-1}-d_t, \mathbb{1}\left\{k_{t-1}-d_t<0\right\}|s_{t-1}, a_{t-1}, k_{t-1}) \\
&\qquad\qquad p_{\tilde{\pi},\tilde{\mathcal{M}}}(g(h_{t-1})|S_0=s, K_0=\Delta) \\
&= p_{\tilde{\pi},\tilde{\mathcal{M}}}(s_t, k_{t-1}-d_t, a_{t-1}, r_t, \mathbb{1}\left\{k_{t-1}-d_t<0\right\}|g(h_{t-1}))p_{\tilde{\pi},\tilde{\mathcal{M}}}(g(h_{t-1})|S_0=s, K_0=\Delta) \\
&= p_{\tilde{\pi},\tilde{\mathcal{M}}}(g(h_t)|g(h_{t-1}))p_{\tilde{\pi},\tilde{\mathcal{M}}}(g(h_{t-1})|S_0=s, K_0=\Delta) \\
&= p_{\tilde{\pi},\tilde{\mathcal{M}}}(g(h_t)|S_0=s, K_0=\Delta)\,.
\end{aligned}$$

Induction on t gives us (18), which completes the proof. ■

Proof of (11)–(12) (properties of B_π)

- Safe and more budget → safe

  $$B_\pi(s,k,a)=0 \Longrightarrow B_\pi(s,k+i,a)=0$$

  Proof

  $$\begin{aligned}
  B_\pi(s,k,a) =& \mathbb{E}_\pi\left[-\mathbb{I}\left\{\sum_{l=t}^{\infty} D_{l+1}\le k \,\middle|\, S_t=s, A_t=a\right\}\right]=0 \iff \\
  &\iff P_\pi\left\{\sum_{l=t}^{\infty} D_{l+1}\le k \,\middle|\, S_t=s, A_t=a\right\}=1
  \end{aligned}$$

  Event inclusion:

  $$\left\{\sum_{l=t}^{\infty} D_{l+1}\le k \,\middle|\, S_t=s, A_t=a\right\} \subset \left\{\sum_{l=t}^{\infty} D_{l+1}\le k+i \,\middle|\, S_t=s, A_t=a\right\} \quad i\ge 0$$

  Monotonicity of P:

  $$1=P_\pi\left\{\sum_{l=t}^{\infty} D_{l+1}\le k \,\middle|\, S_t=s, A_t=a\right\} \le P_\pi\left\{\sum_{l=t}^{\infty} D_{l+1}\le k+i \,\middle|\, S_t=s, A_t=a\right\} \Longrightarrow$$
  $$B_\pi(s,k+i,a)=0$$

  ■

- unsafe and less budget → unsafe

  $$B_\pi(s,k,a)=-\infty \Longrightarrow B_\pi(s,k-i,a)=-\infty \qquad \text{(unsafe and less slack} \to \text{unsafe.)}$$

  Proof

  $$B_\pi(s,k,a)=-\infty \iff P_\pi\left\{\sum_{l=t}^{\infty} D_{l+1}\le k \,\middle|\, S_t=s, A_t=a\right\}<1$$

Monotonicity:

$$1 > P_\pi \left\{ \sum_{l=t}^{\infty} D_{l+1} \leq k \,\middle|\, S_t = s, A_t = a \right\} > P_\pi \left\{ \sum_{l=t}^{\infty} D_{l+1} \leq k - i \,\middle|\, S_t = s, A_t = a \right\} \implies$$

$$B_\pi(s, k-i, a) = -\infty$$

■

Proof of Theorem 9

Proof Consider an (s,a)-pair with $k_*(s,a) = K^* \geq 1$. We prove (15) by contradiction.

- Assume

  $$K^* = k_*(s,a) > \max_{s' : p(s'|s,a) > 0} \left[\mathbb{1}_d(s,a,s') + \min_{a'} k_*(s',a') \right] .$$

  This suggests that $\forall s' : p(s' \mid s,a) > 0$, we have

  $$K^* - \mathbb{1}_d(s,a,s') > \min_{a'} k_*(s',a') \implies K^* - \mathbb{1}_d(s,a,s') - 1 \geq \min_{a'} k_*(s',a') \,,$$

  and which, by definition of k^*, it is equivalent to,

  $$\mathbb{P}_{\pi^*}\left(\sum_{t=0}^{\infty} D_{t+1} \leq K^* - 1 - \mathbb{1}_d(s,a,s') \mid S_0 = s' \right) = 1 \,.$$

  Then we have

  $$\begin{aligned} &\mathbb{P}_{\pi^*}\left(\sum_{t=0}^{\infty} D_{t+1} \leq K^* - 1 \mid S_0 = s, A_0 = a \right) \\ &= \sum_{s'} \mathbb{P}_{\pi^*}\left(\sum_{t=1}^{\infty} D_{t+1} \leq K^* - 1 - D_1 \mid S_0 = s, A_0 = a, S_1 = s' \right) p(s' \mid s,a) \\ &\geq \sum_{s'} \mathbb{P}_{\pi^*}\left(\sum_{t=1}^{\infty} D_{t+1} \leq K^* - 1 - \mathbb{1}_d(s,a,s') \mid S_0 = s, A_0 = a, S_1 = s' \right) p(s' \mid s,a) \\ &= \sum_{s'} \mathbb{P}_{\pi^*}\left(\sum_{t=1}^{\infty} D_{t+1} \leq K^* - 1 - \mathbb{1}_d(s,a,s') \mid S_1 = s' \right) p(s' \mid s,a) = 1 \,. \end{aligned}$$

  This is equivalent to $k_*(s,a) \leq K^* - 1$, a contradiction.

  $$k_*(s,a) \leq \max_{s' : p(s'|s,a) > 0} \left[\mathbb{1}_d(s,a,s') + \min_{a'} k_*(s',a') \right] . \tag{22}$$

- Assume

  $$K^* = k_*(s,a) < \max_{s' : p(s'|s,a) > 0} \left[\mathbb{1}_d(s,a,s') + \min_{a'} k_*(s',a') \right] .$$

This suggests that there exists an s' with $p(s' \mid s, a) > 0$, for which we have

$$K^* - \mathbb{1}_d(s, a, s') < \min_{a'} k_*(s', a')\,,$$

and which, by definition of k^*, it is equivalent to,

$$\mathbb{P}_\pi\left(\sum_{t=0}^{\infty} D_{t+1} > K^* - \mathbb{1}_d(s, a, s') \mid S_0 = s'\right) > 0, \quad \forall \pi\,,$$

and it leads to

$$\mathbb{P}_\pi\left(\sum_{t=0}^{\infty} D_{t+1} > K^* - \mathbb{1}_d(s, a, s') \mid S_0 = s'\right) p(d = \mathbb{1}_d(s, a, s') | s, a, s') > 0\,.$$

Then we have for any policy π

$$\begin{aligned}
&\mathbb{P}_\pi\left(\sum_{t=0}^{\infty} D_{t+1} > K^* \mid S_0 = s, A_0 = a\right) \\
&= \sum_{s'}\Bigg(\mathbb{P}_\pi\left(\sum_{t=1}^{\infty} D_{t+1} > K^* - 1 \mid S_0 = s, A_0 = a, S_1 = s'\right) p(d = 1 | s, a, s') p(s' \mid s, a) \\
&\qquad + \mathbb{P}_\pi\left(\sum_{t=1}^{\infty} D_{t+1} > K^* \mid S_0 = s, A_0 = a, S_1 = s'\right) p(d = 0 | s, a, s') p(s' \mid s, a)\Bigg) \\
&= \sum_{s'}\Bigg(\mathbb{P}_\pi\left(\sum_{t=1}^{\infty} D_{t+1} > K^* - 1 \mid S_1 = s'\right) p(d = 1 | s, a, s') p(s' \mid s, a) \\
&\qquad + \mathbb{P}_\pi\left(\sum_{t=1}^{\infty} D_{t+1} > K^* \mid S_1 = s'\right) p(d = 0 | s, a, s') p(s' \mid s, a)\Bigg) > 0\,.
\end{aligned}$$

This is equivalent to $k_*(s, a) > K^*$, a contradiction. Therefore one must have

$$k_*(s, a) \geq \max_{s' : p(s'|s,a) > 0}\left[\mathbb{1}_d(s, a, s') + \min_{a'} k_*(s', a')\right]\,. \tag{23}$$

Combining the two inequalities (22)(23), we have

$$k_*(s, a) = \max_{s' : p(s'|s,a) > 0}\left[\mathbb{1}_d(s, a, s') + \min_{a'} k_*(s', a')\right]\,,$$

for $k_*(s, a) = K^* \geq 1$.

Lastly, for the case $k_*(s, a) = 0$, repeat the proof for (23), we have

$$0 = k_*(s, a) \geq \max_{s' : p(s'|s,a) > 0}\left[\mathbb{1}_d(s, a, s') + \min_{a'} k_*(s', a')\right]\,,$$

then the right hand side must be 0, which equals $k_*(s, a)$. ■

Properties of $\mathcal{T}$

Definition 17 *For two vectors $k_1, k_2 \in \bar{\mathbb{N}}^{\mathcal{S}\times\mathcal{A}}$, we write $k_1 \leq k_2$ if $k_1(s,a) \leq k_2(s,a), \forall (s,a) \in \mathcal{S}\times\mathcal{A}$. Then $(\leq)$ defines a partial order relation on $\bar{\mathbb{N}}^{\mathcal{S}\times\mathcal{A}}$.*

Proposition 18 (Properties of $\mathcal{T}$) *The operator $\mathcal{T}$ in (16) satisfies the following properties:*

1. *$\mathcal{T}$ preserves the partial ordering $(\leq)$ on $\bar{\mathbb{N}}^{\mathcal{S}\times\mathcal{A}}$:* $\quad k_1 \leq k_2 \implies \mathcal{T}k_1 \leq \mathcal{T}k_2\,.$
2. *k_* is a fixed point of $\mathcal{T}$:* $\quad \mathcal{T}k_* = k_*\,.$

Proof

1. *Using that $k_1(s',a') \leq k_2(s',a') \quad \forall (s,a) \in \mathcal{S}\times\mathcal{A}$ the following inequality holds:*

$$\begin{aligned}(\mathcal{T}\, k_1)(s,a) &= \max_{s':p(s'|s,a)>0}\left[\mathbb{1}_d(s,a,s') + \min_{a'} k_1(s',a')\right]\\ &\leq \max_{s':p(s'|s,a)>0}\left[\mathbb{1}_d(s,a,s') + \min_{a'} k_2(s',a')\right]\\ &= (\mathcal{T}\, k_2)(s,a)\end{aligned}$$

2. *This is the result of Theorem 9.*

■

Proof of Theorem 11

Proof Firstly notice Algorithm 1 starts iterating at $k_0 = 0$. Since (by definition) $0 \leq k_*$, then $k_0 \leq k_*$. Since $\mathcal{T}$ preserves partial ordering we have $k_n = \mathcal{T}^{(n)}k_0 \leq \mathcal{T}^{(n)}k_* = k_*\ \forall n$. This means

$$k_\infty(s,a) \leq k_*(s,a) \quad \forall (s,a) \in \mathcal{S}\times\mathcal{A} \tag{24}$$

Through the rest of this proof we show (24) holds with equality.
We proceed by induction, first by showing

(Induction base): $k_*(s,a) = 0 \iff k_\infty(s,a) = 0$.
$(\implies)$ We invoke the monotonicity of $\mathcal{T}$ to show

$$\begin{aligned} 0 \leq k_0 &\implies \mathcal{T}^{(n)}0 \leq \mathcal{T}^{(n)}k_0 = k_n\,, \forall n \geq 0 &&\implies 0 \leq k_\infty\\ 0 = k_0 \leq k_* &\implies k_n = \mathcal{T}^{(n)}k_0 \leq \mathcal{T}^{(n)}k_* = k_*\,, \forall n \geq 0 &&\implies k_\infty \leq k_*\end{aligned}$$

Then $0 \leq k_\infty \leq k_*$, so for all $(s,a) : k_*(s,a) = 0 \implies k_\infty(s,a) = 0$.
$(\impliedby)$ Suppose for some $(s,a) \in \mathcal{S}\times\mathcal{A}$, we have

$$k_\infty(s,a) = \max_{s':p(s'|s,a)>0}\left[\mathbb{1}\{p(d=1 \mid s,a,s')\} + \min_{a'} k_\infty(s',a')\right] = 0$$

Hence for all s' with $p(s'|s,a) > 0$, we must have $p(d = 1|s,a,s') = 0$ and $\min_{a'} k_\infty(s',a') = 0$.

Now consider the policy $\pi(\cdot)$ that, at time t, takes actions $A_t = \pi(S_t) = \mathrm{argmin}_a\, k_\infty(S_t, a)$. If $k_\infty(S_t, A_t) = 0$, then by our preceding argument, we have

$$\mathbb{P}_\pi(D_{t+1} = 1 | S_t, A_t, S_{t+1}) = 0,\ \mathbb{P}_\pi(\min_a k_\infty(S_{t+1}, a) = 0 | S_t, A_t) = 1\,.$$

That is, given $k_\infty(S_t, A_t) = 0$, taking action A_t at S_t guarantees that, with probability 1, the MDP transitions to a state S_{t+1} with $\min_a k_\infty(S_{t+1}, a) = 0$ and does not incur damage. Furthermore, $\min_a k_\infty(S_{t+1}, a) = 0$ restricts us to take action $A_{t+1} = \pi(S_{t+1})$ with $k_\infty(S_{t+1}, A_{t+1}) = 0$.

Repeating this argument for the entire trajectory $\tau = \{S_t, A_t\}_{t=0}^{\infty}$ induced by π starting from $S_0 = s, A_0 = a$. Then with probability 1, we have

$$k_\infty(S_t, A_t) = 0,\ D_{t+1} = 0\,, \forall t \geq 0\,.$$

Then we find a policy π such that $\mathbb{P}_\pi\left(\sum_{t=0}^{\infty} D_{t+1} \leq 0 | S_0 = s, A_0 = a\right) = 0$, then $B_*(s, 0, a) = 0$, which is equivalent to $k_*(s, a) = 0$. This completes the induction base.

(Induction step): Now given some $L : 0 \leq L < \infty$, we assume

$$k_*(s, a) = \ell \iff k_\infty(s, a) = \ell\,,\ \forall \ell \leq L\,,$$

and show $k_*(s, a) = L + 1 \iff k_\infty(s, a) = L + 1$.
$(\implies)$ Given some (s, a) with $k_*(s, a) = L + 1$, by monotonicity of $\mathcal{T}$, we have $k_\infty(s, a) \leq k_*(s, a) = L+1$. Now suppose $k_\infty(s, a) = \ell$ for some $\ell < L+1$, then by our inductive assumption, one would have $k_*(s, a) = k_\infty(s, a) = l$, contradicting the fact that $k_*(s, a) = L + 1$. $(\impliedby)$ We show the final step, namely $k_\infty(s, a) = L + 1 \implies k_*(s, a) = L + 1$. First, we must have $k_*(s, a) \geq L + 1$, otherwise by our inductive assumption $k_\infty(s, a) = l$ for some $l \leq L$, which contradicts that $k_\infty(s, a) = L + 1$.

Now we show $k_*(s, a) \leq L + 1$. Notice that

$$k_\infty(s, a) = \max_{s' : p(s'|s,a) > 0} \left[\mathbb{1}\{p(d = 1 \mid s, a, s')\} + \min_{a'} k_\infty(s', a')\right] = L + 1\,.$$

Consider policy $\pi(\cdot)$ that takes actions $A_t = \pi(S_t) = \arg\min_a k_\infty(S_t, a)$ and trajectories $\tau = \{S_t, A_t\}_{t=0}^{\infty}$ such that $S_0 = s, A_0 = a$. We carefully examine the k_∞ value along the trajectory τ under π.

First of all, with probability one, $k_\infty(S_t, A_t) \geq k_\infty(S_{t+1}, A_{t+1}), \forall t \geq 0$. Because

$$\begin{aligned}
&\mathbb{1}\{p(d = 1 \mid S_t, A_t, S_{t+1})\} + k_\infty(S_{t+1}, A_{t+1}) \\
&= \mathbb{1}\{p(d = 1 \mid S_t, A_t, S_{t+1})\} + \min_{a'} k_\infty(S_{t+1}, a') \\
&\leq \max_{s' : p(s'|S_t, A_t) > 0} \left[\mathbb{1}\{p(d = 1 \mid S_t, A_t, s')\} + \min_{a'} k_\infty(s', a')\right] \leq k_\infty(S_t, A_t)\,.
\end{aligned}$$

Given a trajectory $\tau = \{S_t, A_t\}_{t=0}^{\infty}$ realized by π, let t_0 be the first $t \geq 0$ such that $L + 1 = k_\infty(S_t, A_t) > k_\infty(S_{t+1}, A_{t+1})$, then $k_\infty(S_{t_0+1}, A_{t_0+1}) = l$ for some $l \leq L$. For trajectories such that t_0 does not exist, we must have that for $t \geq 0$, $L + 1 = k_\infty(S_t, A_t) = k_\infty(S_{t+1}, A_{t+1})$, and $\mathbb{1}\{p(d = 1 \mid S_t, A_t, S_{t+1})\} = 0$, i.e. this trajectory incurs no damage at all. Now we consider trajectories such that t_0 exists.

Prior to t_0, for $t < t_0$, we must also have $\mathbb{1}\{p(d = 1 \mid S_t, A_t, S_{t+1})\} = 0$, since $L + 1 = k_\infty(S_t, A_t) = k_\infty(S_{t+1}, A_{t+1})$. At t_0, the MDP could incur damage depending on the value of $\mathbb{1}\{p(d = 1 \mid S_{t_0}, A_{t_0}, S_{t_0+1})\}$.

After t_0, by our inductive assumption, we have $k_*(S_{t_0+1}, A_{t_0+1}) = k_\infty(S_{t_0+1}, A_{t_0+1}) = l$, and $k_*(S_{t_0+t}, A_{t_0+t}) = k_\infty(S_{t_0+t}, A_{t_0+t}), t \geq 1$ for the remaining trajectory after t_0. That is, upon reaching (S_{t_0+1}, A_{t_0+1}), the remaining trajectory can be also viewed as a trajectory generated by $\pi^*(s) = \arg\min_a k_*(s, a)$, and π^* will incur at most l damage, starting from (S_{t_0+1}, A_{t_0+1}), since $k_*(S_{t_0+1}, A_{t_0+1}) = l$.

Overall, this trajectory incurs no more than $l + 1 \leq L + 1$ damage. This holds true for any trajectory generated by π, i.e.

$$\mathbb{P}_\pi\left(\sum_{t=0}^{\infty} D_{t+1} \leq L + 1 \mid S_0 = s, A_0 = a\right) = 1\,.$$

This implies $B^*(s, L + 1, a) = 0$, which is equivalent to $k_*(s, a) \leq L + 1$. Since we have shown $k_*(s, a) \geq L + 1$, one finally conclude $k_*(s, a) = L + 1$.

By induction we shown that $k_*(s, a) = l \iff k_\infty(s, a) = l, \forall 0 \leq l < \infty$. The final step is showing $k_*(s, a) = \infty \iff k_\infty(s, a) = \infty$.
$(\impliedby)$ For every (s, a) such that $k_\infty(s, a) = \infty$ apply the monotonicity of $\mathcal{T}$ to get $k_*(s, a) \geq k_\infty(s, a) = \infty$, so it must be $k_*(s, a) = \infty$.
$(\implies)$ Suppose $k_*(s, a) = \infty$, then $k_\infty(s, a)$ must be ∞. Because $k_\infty(s, a) = l$ for any value other than ∞ leads to $k_*(s, a) = l$, a contradiction. ■

Proof of Lemma 13

Proof For every (s, a, s', d) with $p(s', d|s, a) \geq \mu > 0$, the estimated kernel has $\hat{p}(s', d|s, a) > 0$ when transition $(s, a) \to (s', d)$ is observed at least once in our samples. Let $F_{(s,a)\to(s',d)}$ be the event of sampling the transition (s', d) from (s, a) (where there are at most $2\mathcal{S}$ possible transitions for fixed (s, a)). Let $X_{(s,a)\to(s',d)} = \mathbb{1}\left(F_{(s,a)\to(s',d)}\right)$. Then the probability that Algorithm 2 fails to produce a consistent kernel satisfies:

$$\begin{aligned}
\mathbb{P}\left(\bigcup_{(s,a)\in\mathcal{S}\times\mathcal{A}} \bigcup_{\substack{(s',d):\\ \mathbb{P}(s',d|s,a)>0}} \{F_{(s,a)\to(s',d)}\}^C\right) &\leq \sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} \sum_{\substack{(s',d):\\ \mathbb{P}(s',d|s,a)>0}} \mathbb{P}\left(X_{(s,a)\to(s',d)} = 0\right) \\
&= \sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} \sum_{\substack{(s',d):\\ \mathbb{P}(s',d|s,a)>0}} (1-\mu)^N \\
&\leq 2|\mathcal{S}|^2|\mathcal{A}|(1-\mu)^N\,. \\
&\leq 2|\mathcal{S}|^2|\mathcal{A}|\exp(N\log(1-\mu)) \\
&\leq 2|\mathcal{S}|^2|\mathcal{A}|\exp(-\mu N) \\
&= 2|\mathcal{S}|^2|\mathcal{A}|\exp\left(-\log\frac{2|\mathcal{S}|^2|\mathcal{A}|}{\delta}\right) = \delta\,,
\end{aligned}$$

where the last inequality uses the fact that $\log(1-\mu) \leq -\mu$. ■

Proof of Lemma 14

Proof Notice that Algorithm 1 under input $\hat{p}$ performs updates as

$$k_{n+1}(s,a) \leftarrow \max_{s':\hat{p}(s'|s,a)>0} \left[\mathbb{1}\{\hat{p}(d=1 \mid s,a,s')\} + \min_{a'} k_n(s',a') \right].$$

If $\hat{p}$ is consistent with p then for all $(s,a) \in \mathcal{S} \times \mathcal{A}$ the outer maximization is carried over the same set of s' and $\mathbb{1}\{\hat{p}(d=1 \mid s,a,s') = 1 \iff \mathbb{1}\{p(d=1 \mid s,a,s')$. Then the updates of Algorithm 1 under both inputs are identical. Since this algorithm converges under p to k^* (Theorem 11), the same is true under $\hat{p}$. ■