Provable Reset-free Reinforcement Learning by No-Regret Reduction

Hoai-An Nguyen [1 2] **Ching-An Cheng** [1]

Abstract

Reinforcement learning (RL) so far has limited real-world applications. One key challenge is that typical RL algorithms heavily rely on a reset mechanism to sample proper initial states; these reset mechanisms, in practice, are expensive to implement due to the need for human intervention or heavily engineered environments. To make learning more practical, we propose a generic no-regret reduction to systematically design reset-free RL algorithms. Our reduction turns the reset-free RL problem into a two-player game. We show that achieving sublinear regret in this two-player game would imply learning a policy that has both sublinear performance regret and sublinear total number of resets in the original RL problem. This means that the agent eventually learns to perform optimally and avoid resets. To demonstrate the effectiveness of this reduction, we design an instantiation for linear Markov decision processes, which is the first provably correct reset-free RL algorithm.

1. Introduction

Reinforcement learning (RL) enables an artificial agent to learn problem-solving skills directly through interactions. However, RL is notorious for its sample inefficiency. Successful stories of RL so far are mostly limited to applications where a fast and accurate simulator of the world is available for collecting large amounts of samples (like in games). Real-world RL, such as robot learning, remains a challenging open question.

One key obstacle to scaling up data collection in real-world RL problems is the need for resetting the agent. The ability to reset the agent to proper initial states plays an important role in typical RL algorithms, as it affects which region the agent can explore and whether the agent can recover from its past mistakes (Kakade & Langford, 2002). In most settings, completely avoiding resets without prior knowledge of the reset states or environment is impossible. In the absence of a reset mechanism, agents may get stuck in absorbing states, such as those where it has damaged itself or irreparably altered the learning environment. For instance, a robot learning to walk would inevitably fall before perfecting the skill, and timely intervention is needed to prevent damaging the hardware and to return the robot to a walkable configuration. Another example is a robot manipulator learning to stack three blocks on top of each other. Unrecoverable states would include the robot knocking a block off the table, or the robot smashing itself forcefully into the table. Reset would then reconfigure the scene to a meaningful initial state that is good for the robot to learn from.

Although resetting is necessary to real-world RL, it is non-trivial to implement. Unlike in simulation, a real-world agent (e.g., a robot) cannot be reset to an arbitrary initial state with a click of a button. Resetting in the real world is expensive as it usually requires constant human monitoring and intervention. Normally, a person would need to oversee the entire learning process and manually reset the agent (e.g., a robot) to a meaningful starting state before it enters an unrecoverable state where the problem can no longer be solved. Sometimes automatic resetting can be implemented by cleverly engineering the physical learning environment (Gupta et al., 2021), but it is not always feasible.

An approach we can take to make real-world RL more cost-efficient is through *reset-free* RL. The goal of reset-free RL is not to completely remove resets, but to have an agent learn how to perform well while minimizing the amount of external resets required. Some examples of problems that have been approached in reset-free RL include agents learning dexterity skills, such as picking up an item or inserting a pipe, and learning how to walk (Ha et al., 2020; Gupta et al., 2021). While there has been numerous works proposing reset-free RL algorithms using approaches such as multi-task learning (Gupta et al., 2021; Ha et al., 2020), learning a reset policy (Eysenbach et al., 2018; Sharma et al., 2022), and skill-space planning (Lu et al., 2020), to our knowledge, there has not been any work with provable guarantees.

In this work, we take the first step by providing a provably

[1]Microsoft Research [2]Rutgers University. Correspondence to: Hoai-An Nguyen <hoaian.nguyen@rutgers.edu>, Ching-An Cheng <chinganc@microsoft.com>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

correct framework to design reset-free RL algorithms. Our framework is based on the idea of a no-regret reduction. First, we reduce the reset-free RL problem to a sequence of constrained Markov decision processes (CMDPs) with an adaptive initial state sequence. Using the special structure of reset-free RL, we establish the existence of a common saddle-point for these CMDPs. Interestingly, we show such a saddle-point exists without using the typical Slater's condition for strong duality and despite the fact that CMDPs with different initial states in general do not share a common Markovian optimal policy. Then, we derive our main no-regret reduction, which further turns this sequence into a two-player game between a primal player (updating the Markovian policy) and a dual player (updating the Lagrange multiplier function of the CMDPs) to solve for the common saddle-point. We show that if no regret is achieved in this game, then the regret of the original RL problem and the total number of required resets are both provably sublinear. This means the agent eventually learns to perform optimally and avoid resets. Using this reduction, we design a reset-free RL algorithm instantiation under the linear MDP assumption, using learning with upper confidence bound as the baseline algorithm for the primal player and projected gradient descent for the dual player. Our algorithm achieves $\tilde{O}(\sqrt{d^3H^4K})$ regret and $\tilde{O}(\sqrt{d^3H^4K})$ resets with high probability, where d is the feature dimension, H is the length of an episode, and K is the total number of episodes.[1]

2. Related Work

Reset-free RL, despite being a promising avenue to tackle real-world RL, is a relatively new concept in the literature. The work thus far, to our knowledge, has been limited to approaches without theoretical guarantees but with only empirical verification. One such work takes the approach of learning a reset policy (Eysenbach et al., 2018; Sharma et al., 2022). The idea is to learn two policies concurrently: one to maximize reward, and one to bring the agent back to a reset-free initial state if they encounter a *reset state* (a state which normally requires human intervention). This approach prevents the need for manual resets; however, it requires the knowledge of the reset states (Eysenbach et al., 2018). Sharma et al. (2022) avoid this assumption but assume given demonstrations on how to accomplish the goal and a fixed initial state distribution.

Using multi-task learning is another way to perform resets without human intervention. Here, the agent learns to solve multiple tasks instead of just maximizing the reward of one task. The hope is that a combination of the learned tasks can achieve the main goal, and some tasks can perform natural resets for others. This approach breaks down the reset process and (possibly) makes it easier to learn. However, the order in which tasks should be learned needs to be provided manually (Gupta et al., 2021; Ha et al., 2020).

A related problem is infinite-horizon non-episodic RL with provable guarantees (see Wei et al. (2020; 2019); Dong et al. (2019) and the references within) as this problem is also motivated by not using resets. In this setting, there is only one episode that goes on indefinitely. The objective is to maximize cumulative reward, and progress is usually measured in terms of regret with the comparator being an optimal policy. However, compared with the reset-free RL setting we study here, extra assumptions, such as the absence or knowledge of absorbing states, are usually required to achieve sublinear regret. In addition, the objective does not necessarily lead to a minimization of resets as the agent can leverage reset transitions to maximize reward. In reset-free RL, minimizing/avoiding resets is a priority. Learning in infinite-horizon CMDPs (where one possible constraint could be minimizing resets) has been studied (Zheng & Ratliff, 2020; Jain et al., 2022), but to our knowledge, all such works make strong assumptions such as a fixed initial state distribution or known dynamics. None of these assumptions are feasible for most real-world RL settings. Enforcing a fixed initial state distribution or removing absorbing states in theory oftentimes requires physically resetting a real-world agent to satisfy those desired mathematical conditions. In this paper, we focus on an episodic setting of reset-free RL (see Section 3); a non-episodic formulation of reset-free RL could be an interesting one for further research.

Another related problem is safe RL, which involves solving the standard RL problem while adhering to some safety constraints. We can think of reset states in reset-free RL as unsafe states in safe RL. There has been a lot of work in safe RL, with approaches such as utilizing a baseline safe (but not optimal) policy (Huang et al., 2022; Garcia Polo & Fernandez Rebollo, 2011), pessimism (Amani & Yang, 2022), and shielding (Alshiekh et al., 2018; Wagener et al., 2021). These works have promising empirical results but usually require extra assumptions such as a given baseline policy or knowledge of unsafe states. There are also provable safe RL algorithms. To our knowledge, all involve framing safe RL as a CMDP. Here, the safety constraints are modeled as a cost, and the overall goal is to maximize performance while keeping the cost below a threshold. Some of the works explicitly study safe RL while others study learning in CMDPs more generally. They commonly have provable guarantees of either sublinear regret and constraint violations, or sublinear regret with zero constraint violation (Wei et al., 2021; HasanzadeZonuzy et al., 2021; Qiu et al., 2020; Wachi & Sui, 2020; Efroni et al., 2020; Ghosh et al., 2022; Ding et al., 2021). However, most works (including all the aforementioned ones), consider the episodic case

[1] In the tabular MDP setting, our bounds on the regret and total resets become $\tilde{O}(\sqrt{|\mathcal{S}|^3|\mathcal{A}|^3H^4K})$, where $|\mathcal{S}|$,$|\mathcal{A}|$ denote the size of the state and action spaces, respectively.

where the initial state distribution of each episode is *fixed*. This prevents a natural extension to reset-free learning as human intervention would be required to reset the environment at the end of each episode. In technical terms, this is the difference between solving a sequence of the *same* CMDP versus solving a sequence of *different* CMDPs. Works that allow for arbitrary initial states require fairly strong assumptions, such as knowledge (and the existence) of safe actions from each state (Amani et al., 2021). In our work, we utilize some techniques from provable safe RL for reset-free RL. However, it is important to note that safe RL and reset-free RL are fundamentally different, albeit related, problems. Safe RL aims to ensure the safety of an agent. On the other hand, reset-free RL aims to avoid reset states, which encompass not only unsafe states but all undesirable ones, for a varying sequence of initial states as the agent cannot be reset freely.

We weaken typical assumptions of current approaches in empirical reset-free RL, infinite-horizon RL, and safe RL by dropping any requirements on knowledge of undesirable states or for demonstrations, and by allowing *arbitrary* initial state sequences that admit reset-free policies. Considering arbitrary initial state sequences where initial states potentially are correlated with past behaviors is not only necessary to the reset-free RL setting, but also allows for extensions to both lifelong and multi-task learning. We achieve this important relaxation on the initial state sequence with a key observation that identifies a shared Markovian policy saddle-point across CMDPs where the constraint imposes *zero* resets. This observation is new to our knowledge, and it is derived from the particular structure of reset-free RL; we note that generally, CMDPs with different initial states do not admit shared Markovian policy saddle-points. Finally, by the analogy between safe states and reset states, on the technical side, our framework and algorithm can also be viewed as the first provable safe RL algorithm that allows for arbitrary initial state sequences without strong assumptions.

While our main contribution is a generic reduction technique to design reset-free RL algorithms, we also instantiate the framework and achieve regret and constraint violation bounds that are still comparable to the above works when specialized to their setting. Under the linear MDP assumption, our algorithm achieves $\tilde{O}(\sqrt{d^3H^4K})$ regret and violation (equivalently, the number of resets in reset-free RL), which is asymptotically equivalent to Ghosh et al. (2022) and comparable to the bounds of $\tilde{O}(\sqrt{d^2H^6K})$ from Ding et al. (2021) for a fixed initial state.

In summary, our contributions are as follows.

1. We create a framework to design provably correct reset-free RL algorithms via a reduction first to a sequence of CMDPs with an adaptive initial state sequence, and then to a two-player game. We prove that achieving sublinear regret in this two-player game implies learning a policy that achieves sublinear performance regret and sublinear number of resets in the original problem.

2. On the technical side, we show that such a reduction can be constructed without using the typical Slater's condition for strong duality and despite the fact that CMDPs with different initial states in general do *not* share a common Markovian optimal policy.

3. We instantiate the framework under the linear MDP setting as a proof of concept, creating the first provably correct reset-free RL algorithm to our knowledge that achieves sublinear regret and resets.

3. Preliminary

We consider episodic reset-free RL: in each episode, the agent aims to optimize for a fixed-horizon return starting from the last state of the previous episode or some state that the agent was reset to if reset occurred.

Problem Setup and Notation Formally, we can define episodic reset-free RL as a Markov decision process (MDP), $(\mathcal{S}, \mathcal{A}, P, r, H)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $P = \{P_h\}_{h=1}^H$ is the transition dynamics, $r = \{r_h\}_{h=1}^H$ is the reward function, and H is the task horizon. We assume P and r are unknown. We allow $\mathcal{S}$ to be large or continuous but assume A is relatively small so that $\max_{a\in\mathcal{A}}$ can be performed. We designate the set of reset states, or any states that human intervention normally would have been required, as $\mathcal{S}_{reset} \subseteq \mathcal{S}$; we do not assume that the agent has knowledge of $\mathcal{S}_{reset}$. We also do not assume that there is a reset-free action at each state, as opposed to Amani et al. (2021). Therefore, the agent needs to plan for the long-term to avoid resets. We assume $r_h : \mathcal{S} \times \mathcal{A} \to [0, 1]$, and for simplicity, we assume r_h is deterministic. However, we note that it would be easy to extend this to the setting where rewards are stochastic.

The agent interacts with the environment for K total episodes. Following the convention of episodic problems, we suppose the state space $\mathcal{S}$ is layered, and a state $s_t \in \mathcal{S}$ at time t is factored into two components $s_t = (\bar{s}, t)$ where $\bar{s}$ denotes the time-invariant part. Reset happens at some time t if the time-invariant part of s_t, $\bar{s} \in \mathcal{S}_{reset}$. The initial state of the next episode will be $s_1 = (\bar{s}', 1)$ where $\bar{s}'$ is sampled from an unknown state distribution. In a given episode, if reset happens, we designate the time step this occurs as $t = \tau$. If there is no reset in an episode, the initial state of the next episode is the last state of the current episode, i.e., for episode $k+1$, $s_1^{k+1} = (\bar{s}, 1)$ if $s_H^k = (\bar{s}, H)$ in episode k.[2] Therefore, the initial state sequence is adaptive. This

[2]This setup covers reset-free multi-task or lifelong RL problems that are modeled as contextual MDPs. We can treat each state as

sequence is necessary to consider since in reset-free RL, we want to avoid resetting, including after each episode.

We denote the set of Markovian policies as Δ, and a policy $\pi \in \Delta$ as $\pi = \{\pi_h(a_h|s_h)\}_{h=1}^H$. We define the state value function and the state-action value function under π as

$$V_{r,h}^\pi(s) := \mathbb{E}_\pi\Big[\textstyle\sum_{t=h}^{\min(H,\tau)} r_t(s_t,a_t)|s_h = s\Big] \tag{1}$$

$$Q_{r,h}^\pi(s,a) := r_h(s,a) + \mathbb{E}\Big[V_{r,h+1}^\pi(s_{h+1})|s_h = s, a_h = a\Big],$$

where $h \leq \tau$, and we recall τ is the time step when the agent enters $\mathcal{S}_{reset}$ (if at all).

Objective The overall goal is for the agent to learn a Markovian policy to maximize its cumulative reward while avoiding resets. Therefore, our performance measures are as follows (we seek to minimize both quantities):

$$\text{Regret}(K) = \max_{\pi\in\Delta_0(K)} \textstyle\sum_{k=1}^K V_{r,1}^\pi(s_1^k) - V_{r,1}^{\pi^k}(s_1^k) \tag{2}$$

$$\text{Resets}(K) = \textstyle\sum_{k=1}^K \mathbb{E}_{\pi^k}\Big[\sum_{h=1}^{\min(H,\tau)} \mathbb{1}[s_h \in \mathcal{S}_{reset}]\Big| s_1 = s_1^k\Big] \tag{3}$$

where $\Delta_0(K) \subseteq \Delta$ denotes the set of Markovian policies that avoid resets for all episodes, and π^k is the policy used by the agent in episode k. Note that by the reset mechanism $\sum_{h=1}^{\min(H,\tau)} \mathbb{1}[s_h \in \mathcal{S}_{reset}] \in \{0,1\}$.

Notice that the initial states in our regret and reset measures are determined by the learner. Given the motivation behind reset-free RL (see Section 1), we can expect that the initial states here are meaningful for performance comparison by construction. Otherwise, a reset would have occurred to set the learner to a meaningful state. Note that this means by the reset mechanism, all "bad" absorbing states are in $\mathcal{S}_{reset}$, and hence, the agent cannot hide in a "bad" absorbing state to achieve small regret. In addition, since the learner does not receive any reward within an episode after being reset, the learner cannot leverage resets to gain an advantage over the optimal policy (which never resets) to minimize or even achieve negative regret.

To make the problem feasible, we assume achieving no resets is possible. We state this assumption formally below.

Assumption 3.1. For any sequence $\{s_1^k\}_{k=1}^K$, the set $\Delta_0(K)$ is not empty. That is, there is a Markovian policy $\pi \in \Delta$ such that $\mathbb{E}_\pi[\sum_{h=1}^H \mathbb{1}[s_h \in \mathcal{S}_{reset}]|s_1 = s_1^k] = 0$.

This assumption is simply stating that every episode of learning admits a reset-free policy. As discussed in Section 2, this assumption is weaker than existing assumptions in the literature. We note that an alternate assumption that only s_1^1 (i.e., the initial state of the first episode) admits a reset-free policy is insufficient to make reset-free RL feasible; since the transition dynamics are unknown to the agent, under this assumption alone, for any algorithm, there is a problem[3] such that the number of resets must be linear.

We highlight that Assumption 3.1 is a reasonable assumption in practice. It does not require a fixed initial state. In addition, if reset happens, in practice, the agent is usually set to a state where it can continue to operate in without reset; if the agent is at a state where no such reset-free policy exists, reset should happen. This assumption is similar to the assumption on the existence of a perfectly safe policy in safe RL literature, which is a common and relatively weak assumption (Ghosh et al., 2022; Ding et al., 2021). If there were to be initial states that inevitably lead to a reset, the problem would be infeasible and does not follow from the motivation of reset-free RL.

4. A No-Regret Reduction for Reset-Free RL

In this section, we present our main reduction of reset-free RL to regret minimization in a two-player game. In the following, we first show that reset-free RL can be framed as a sequence of CMDPs with an adaptive initial state sequence. Then we design a two-player game based on a primal-dual analysis of this sequence of CMDPs. Finally, we show achieving sublinear regret in this two-player game implies sublinear regret and resets in the original reset-free RL problem in (2), and we discuss how to leverage this framework to systematically design reset-free RL algorithms.

Our reduction differs from standard reductions involving bounding the Nash gap with the sum of two players' regret in the literature of constrained optimization and online learning. First, we show a reduction for a *sequence* of saddle-point problems instead of for a single fixed saddle-point problem. There are CMDP methods (e.g. (Ghosh et al., 2022) that implicitly use the two players' regret to bound the Nash gap of a CMDP in their analysis. However, those proofs are applicable to only a single CMDP with a fixed initial state distribution. And importantly, they fundamen-

$s_\tau = (\bar{s}, c, \tau)$, where c denotes the context that stays constant within an episode. If no reset happens, the initial state of episode $k+1$ is $s_1^{k+1} = (\bar{s}, c^{k+1}, 1)$ if $s_H^k = (\bar{s}, c^k, H)$ in episode k, where the new context c^{k+1} may depend on the current context c^k.

[3] Consider a state space $\mathcal{S}$ which can be separated into a reset-free state $s^\dagger$, a reset state $\bar{s}$, and a subset $\tilde{\mathcal{S}} = \mathcal{S} \setminus s^\dagger, \bar{s}$. The subset $\tilde{\mathcal{S}}$ contains reset states and reset-free states. For all actions taken at $s^\dagger$, the agent will land at the reset state $\bar{s}$. Given a learning algorithm, let $\tilde{\mathcal{A}}$ be a subset of actions that the learning algorithm has constant probability of taking at s_1^1 (the very first initial state). Since the learning agent has no knowledge of the MDP, without violating the assumption that s_1^1 admits a reset-free policy, we can construct an MDP such that taking actions in $\tilde{\mathcal{A}}$ would lead to a reset state and a reset mechanism which resets the agent to the reset-free state $s^\dagger$ whenever the agent enters a reset state. As a result, the learning agent has a constant probability of incurring a linear number of resets over the total number of episodes.

tally rely on Slater's condition assumption (i.e., requiring a strictly feasible policy), which does not hold for the CMDPs considered here. Additionally, unlike the typical bound for a convex-concave saddle-point problem in the optimization literature, (Wang & Li, 2020; Kovalev & Gasnikov, 2022; Boyd & Vandenberghe, 2014) the saddle-point problem of a CMDP is non-concave in terms of the policy (and convex in terms of the dual function λ). In this paper, we take a different analysis to bypass the interplaying complexities due to CMDP sequences, non-concavity, and the lack of Slater's condition. The complete proofs for this section are in Appendix A.1.

4.1. Reset-free RL as a Sequence of CMDPs

The first step of our reduction is to cast the reset-free RL problem in Section 3 to a sequence of CMDP problems which share the same rewards, constraints, and dynamics, but have different initial states. Each problem instance in this sequence corresponds to an episode of the reset-free RL problem, and its constraint describes the probability of the agent entering a state that requires reset.

Specifically, we denote these constrained MDPsas $\{(\mathcal{S}, \mathcal{A}, P, r, H, c, s_1^k)\}_{k=1}^K$: in episode k, the CMDP problem is defined as

$$\max_{\pi\in\Delta} V_{r,1}^{\pi}(s_1^k), \text{ s.t. } V_{c,1}^{\pi}(s_1^k) \leq 0 \tag{4}$$

where we define the cost as

$$c_h(s,a) \coloneqq \mathbb{1}[s \in \mathcal{S}_{reset}]$$

and $V_{c,1}^{\pi}$, defined similarly to (1), is the state value function with respect to the cost c . We note that the initial state, s_1^k, depends on the past behaviors of the agent, and that Assumption 3.1 ensures each CMDP in (4) is a feasible problem (i.e., there is a Markovian policy satisfying the constraint). The objective of the agent in each CMDP is to be reward-maximizing while adhering to the constraint, namely that the number of resets should be 0. Requiring zero constraint violation instead of a small constraint violation will be crucial for our reduction.

Since CMDPs are typically defined without early episode termination unlike (1), with abuse of notation, we extend the definitions of P, $\mathcal{S}$, r, c as follows so that the CMDP definition above is consistent with the common literature. We introduce a fictitious absorbing state denoted as $s^\dagger$ in $\mathcal{S}$, where $r_h(s^\dagger, a) = 0$ and $c_h(s^\dagger, a) = 0$; once the agent enters $s^\dagger$, it stays there until the end of the episode. We extend the definition P such that, after the agent is in a state $s \in \mathcal{S}_{reset}$, any action it takes brings it to $s^\dagger$ in the next time step. In this way, we can write the value function, e.g. for reward, as $V_{r,h}^{\pi}(s) = \mathbb{E}_\pi\left[\sum_{t=h}^{H} r_t(s_t, a_t) | s_h = s\right]$ in terms of this extended dynamics. We note that these two formulations are mathematically the same for the purpose of learning; when the agent enters $s^\dagger$, it means that the agent is reset in the episode.

By the construction above, we can write

$$\text{Resets}(K) = \textstyle\sum_{k=1}^{K} V_{c,1}^{\pi^k}(s_1^k)$$

which is the same as the number of total constraint violations across the CMDPs. Because we do not make any assumptions about the agent's knowledge of the constraint function (e.g., the agent does not know states $\in \mathcal{S}_{reset}$), we allow the agent to reset during learning while minimizing the total number of resets over all K episodes.

4.2. Reduction to Two-Player Game

From the problem formulation above, we see that the major difficulty of reset-free RL is the coupling between an adaptive initial state sequence and the constraint on reset probability. If we were to remove either of them, we could use standard algorithms, since the problem would become a single CMDP problem (Altman, 1999) or an episodic RL problem with varying initial states (Jin et al., 2019).

We propose a reduction to systematically design algorithms for this sequence of CMDPs and therefore for reset-free RL. The main idea is to approximately solve the saddle-point problem of each CMDP in (4), i.e.,

$$\max_{\pi\in\Delta} \min_{\lambda\geq 0} V_{r,1}^{\pi}(s_1^k) - \lambda V_{c,1}^{\pi}(s_1^k) \tag{5}$$

where λ denotes the dual variable (i.e., the Lagrange multiplier). Each CMDP can be framed as a linear program (Hernández-Lerma & Lasserre, 2002) whose primal and dual optimal values match (see section 8.1 in Hazan et al. (2016)). Therefore, for each CMDP, $\max_{\pi\in\Delta}\min_{\lambda\geq 0} V_{r,1}^{\pi}(s_1^k) - \lambda V_{c,1}^{\pi}(s_1^k) = \min_{\lambda\geq 0}\max_{\pi\in\Delta} V_{r,1}^{\pi}(s_1^k) - \lambda V_{c,1}^{\pi}(s_1^k)$.

While using a primal-dual algorithm to solve for the saddle-point of a *single* CMDP is known, using this approach for a sequence of CMDPs is not obvious. This is because in general, each CMDP's optimal policy (and Lagrange multiplier) can be a function of its initial state distribution (Altman, 1999). An easy way to see this is that an optimal policy satisfying a constraint on an initial state s_1 may be infeasible for a constraint on another initial state s_1', if, e.g, running this policy starting from s_1 does not reach s_1' and therefore can have arbitrary behavior at s_1'. This behavior is different from that of unconstrained MDPs, where there is an optimal Markovian policy across all the initial states.

Therefore, in general, a common saddle-point of *Markovian* polices and Lagrange multipliers does not necessarily exist for a sequence of CMDPs with varying initial states.[4] As a

[4] A shared saddle-point with a *non-Markovian* policy always

result, it is unclear if there exists a primal-dual algorithm to solve this sequence, especially given that the initial states here are adaptively chosen.

Existence of a Shared Saddle-Point Fortunately, we show that there is a shared saddle-point with a Markovian policy across all the CMDPs considered in (4) due to the special structure of reset-free RL. It is a proof that does not use Slater's condition for strong duality, unlike similar literature, but attains the desired property. Instead, it follows from the fact that we impose a constraint that requires zero constraint violations. In addition, we also use Assumption 3.1 and the fact that the cost c is non-negative. We formalize this below.

Theorem 4.1. *There exists a function $\hat{\lambda}(\cdot)$ where for each s,*

$$\hat{\lambda}(s) \in \arg\min_{y \geq 0} \left(\max_{\pi \in \Delta} V_{r,1}^{\pi}(s) - y V_{c,1}^{\pi}(s) \right),$$

and a Markovian policy $\pi^* \in \Delta$, such that $(\pi^*, \hat{\lambda})$ is a saddle-point to the CMDPs

$$\max_{\pi \in \Delta} V_{r,1}^{\pi}(s_1), \text{ s.t. } V_{c,1}^{\pi}(s_1) \leq 0$$

for all *initial states $s_1 \in \mathcal{S}$ such that the CMDP is feasible. That is, for all $\pi \in \Delta$, $\lambda : \mathcal{S} \to \mathbb{R}$, and $s_1 \in \mathcal{S}$,*

$$\begin{aligned} V_{r,1}^{\pi^*}(s_1) - \lambda(s_1) V_{c,1}^{\pi^*}(s_1) &\geq V_{r,1}^{\pi^*}(s_1) - \hat{\lambda}(s_1) V_{c,1}^{\pi^*}(s_1) \\ &\geq V_{r,1}^{\pi}(s_1) - \hat{\lambda}(s_1) V_{c,1}^{\pi}(s_1). \quad (6) \end{aligned}$$

Corollary 4.2. *For π^* in Theorem 4.1, it holds that Regret$(K) = \sum_{k=1}^{K} V_{r,1}^{\pi^*}(s_1^k) - V_{r,1}^{\pi^k}(s_1^k)$.*

We prove for ease of construction that the pair (π^*, λ^*) where $\lambda^*(\cdot) = \hat{\lambda}(\cdot) + 1$ is also a saddle-point.

Corollary 4.3. *For any saddle-point to the CMDPs*

$$\max_{\pi \in \Delta} V_{r,1}^{\pi}(s_1), \text{ s.t. } V_{c,1}^{\pi}(s_1) \leq 0$$

of $(\pi^*, \hat{\lambda})$ from Theorem 4.1, $(\pi^*, \lambda^*) =: (\pi^*, \hat{\lambda} + 1)$ is also a saddle-point as defined in (6).

Therefore, the pair (π^*, λ^*) in Corollary 4.3 is a saddle-point to all the CMDPs the agent faces. This makes potentially designing a two-player game reduction possible. Now we give the details of our construction.

Two-Player Game Our two-player game proceeds iteratively in the following manner: in episode k, a dual player determines a state value function $\lambda^k : \mathcal{S} \to \mathbb{R}$, and then, a primal player determines a policy π^k which can depend on λ^k. The primal and dual players then receive losses $\mathcal{L}^k(\pi^k, \lambda)$ and $-\mathcal{L}^k(\pi, \lambda^k)$, respectively, where $\mathcal{L}^k(\pi, \lambda)$ is a Lagrangian function defined as

$$\mathcal{L}^k(\pi, \lambda) := V_{r,1}^{\pi}(s_1^k) - \lambda(s_1^k) V_{c,1}^{\pi}(s_1^k). \quad (7)$$

The regret of these two players are defined as follows.

exists on the other hand.

Definition 4.4. Let π_c and λ_c be comparators. The regret of the primal and the dual players are

$$\begin{aligned} R_p(\{\pi^k\}_{k=1}^K, \pi_c) &:= \textstyle\sum_{k=1}^K \mathcal{L}^k(\pi_c, \lambda^k) - \mathcal{L}^k(\pi^k, \lambda^k), \\ R_d(\{\lambda^k\}_{k=1}^K, \lambda_c) &:= \textstyle\sum_{k=1}^K \mathcal{L}^k(\pi^k, \lambda^k) - \mathcal{L}^k(\pi^k, \lambda_c). \end{aligned}$$

We present our main reduction theorem below.

Theorem 4.5. *Under Assumption 3.1, for any sequences $\{\pi^k\}_{k=1}^K$ and $\{\lambda^k\}_{k=1}^K$, it holds that*

$$\begin{aligned} \textit{Regret}(K) &\leq R_p(\{\pi^k\}_{k=1}^K, \pi^*) + R_d(\{\lambda^k\}_{k=1}^K, 0) \\ \textit{Resets}(K) &\leq R_p(\{\pi^k\}_{k=1}^K, \pi^*) + R_d(\{\lambda^k\}_{k=1}^K, \lambda^*) \end{aligned}$$

where (π^*, λ^*) is the saddle-point defined in Corollary 4.3.

Designing Reset-free RL Algorithms by Reduction By Theorem 4.5, if both players have sublinear regret in the two-player game, then the resulting policy sequence will have sublinear performance regret and a sublinear number of resets in the original RL problem. Minimization of regret in a two-player game and solving saddle-point problems have been very well studied (see, e.g., (Ho-Nguyen & Kılınç-Karzan, 2018; Benzi et al., 2005; Hazan et al., 2016)). Therefore, our reduction gives a constructive and provably correct template for designing reset-free RL algorithms. It should be noted however, that the resultant two-player game here is slightly different from the classical full-information online learning problems. Unlike in these problems, the learning agent needs to actively explore the environment in each episode to collect information about the unknown dynamics, reward, and cost (i.e., which states are reset-free). Specifically Theorem 4.5 needs the primal (policy) player to solve an online MDP sequence (with the same transition dynamics but varying reward functions due to changing λ), and the dual player to solve an online linear problem. In the next section, we will give an example algorithm under this reduction for linear MDPs.

4.3. Proof Sketches

Proof Sketch of Theorem 4.1 Let $Q_c^*(s,a) = \min_{\pi \in \Delta} Q_c^{\pi}(s,a)$ and $V_c^*(s) = \min_{\pi \in \Delta} V_c^{\pi}(s)$. We define π^* in Theorem 4.1 as the optimal policy to the following MDP: $(\mathcal{S}, \overline{\mathcal{A}}, P, r, H)$, where we define a state-dependent action space $\overline{\mathcal{A}}$ as

$$\overline{\mathcal{A}}_s = \{a \in \mathcal{A} : Q_c^*(s,a) \leq V_c^*(s)\}.$$

By definition, $\overline{\mathcal{A}}_s$ is non-empty for all s.

We also define a shorthand notation: we write $\pi \in \overline{\mathcal{A}}(s)$ if $\mathbb{E}_{\pi}[\sum_{t=1}^H \mathbb{1}\{a_t \notin \overline{\mathcal{A}}_{s_t}\} | s_1 = s] = 0$. We have the following lemma, which is an application of the performance difference lemma (see Lemma 6.1 in (Kakade & Langford, 2002) and Lemma A.1 in (Cheng et al., 2021)).

Lemma 4.6. *For any $s_1 \in \mathcal{S}$ such that $V_c^*(s_1) = 0$ and any $\pi \in \Delta$, it is true that $\pi \in \overline{\mathcal{A}}(s_1)$ if and only if $V_c^\pi(s_1) = 0$.*

We prove our main claim, (6), below. Because $V_{c,1}^{\pi^*}(s_1) = 0$, the first inequality is trivial: $V_{r,1}^{\pi^*}(s_1) - \lambda(s_1)V_{c,1}^{\pi^*}(s_1) = V_{r,1}^{\pi^*}(s_1) = V_{r,1}^{\pi^*}(s_1) - \hat{\lambda}(s_1)V_{c,1}^{\pi^*}(s_1)$.

To prove the second inequality, we use Lemma 4.6:

$$\begin{aligned}
&V_{r,1}^{\pi}(s_1) - \hat{\lambda}(s_1)V_{c,1}^{\pi}(s_1)\\
\leq &\max_{\pi\in\Delta} V_{r,1}^{\pi}(s_1) - \hat{\lambda}(s_1)V_{c,1}^{\pi}(s_1)\\
= &\min_{y\geq 0}\max_{\pi\in\Delta} V_{r,1}^{\pi}(s_1) - yV_{c,1}^{\pi}(s_1)\\
= &\max_{\pi\in\overline{\mathcal{A}_c}(s_1)} V_{r,1}^{\pi}(s_1) \qquad \text{(By Lemma 4.6)}\\
= &V_{r,1}^{\pi^*}(s_1) = V_{r,1}^{\pi^*}(s_1) - \hat{\lambda}(s_1)V_{c,1}^{\pi^*}(s_1).
\end{aligned}$$

Proof Sketch of Theorem 4.5 We first establish the following intermediate result that will help us with our decomposition.

Lemma 4.7. *For any primal-dual sequence $\{\pi^k, \lambda^k\}_{k=1}^K$, $\sum_{k=1}^K(\mathcal{L}^k(\pi^*, \lambda') - \mathcal{L}^k(\pi^k, \lambda^k)) \leq R_p(\{\pi\}_{k=1}^K, \pi^*)$, where (π^*, λ') is the saddle-point defined in either Theorem 4.1 or Corollary 4.3.*

Then we upper bound Regret(K) and Resets(K) by $R_p(\{\pi^k\}_{k=1}^K, \pi_c)$ and $R_d(\{\lambda^k\}_{k=1}^K, \lambda_c)$ for suitable comparators. This decomposition is inspired by the techniques used in Ho-Nguyen & Kılınç-Karzan (2018). We first bound Resets(K).

Lemma 4.8. *For any primal-dual sequence $\{\pi^k, \lambda^k\}_{k=1}^K$, $\sum_{k=1}^K V_{c,1}^{\pi^k}(s_1^k) \leq R_p(\{\pi\}_{k=1}^K, \pi^*) + R_d(\{\lambda\}_{k=1}^K, \lambda^*)$, where (π^*, λ^*) is the saddle-point defined in Corollary 4.3.*

Proof. Notice $\sum_{k=1}^K V_{c,1}^{\pi^k}(s_1^k) = \sum_{k=1}^K \mathcal{L}^k(\pi^k, \hat{\lambda}) - \mathcal{L}^k(\pi^k, \lambda^*)$ where $(\pi^*, \hat{\lambda})$ is the saddle-point defined in Theorem 4.1. By (6), and adding and subtracting $\sum_{k=1}^K \mathcal{L}^k(\pi^k, \lambda^k)$, we can bound this difference by

$$\sum_{k=1}^K \mathcal{L}^k(\pi^*, \hat{\lambda}) - \mathcal{L}^k(\pi^k, \lambda^k) + \mathcal{L}^k(\pi^k, \lambda^k) - \mathcal{L}^k(\pi^k, \lambda^*).$$

Using Lemma 4.7 and Definition 4.4 to upper bound the above, we get the desired result. □

Lastly, we bound Regret(K) with the lemma below and Corollary 4.2.

Lemma 4.9. *For any primal-dual sequence $\{\pi^k, \lambda^k\}_{k=1}^K$, $\sum_{k=1}^K(V_{r,1}^{\pi^*}(s_1^k) - V_{r,1}^{\pi^k}(s_1^k)) \leq R_p(\{\pi\}_{k=1}^K, \pi^*) + R_d(\{\lambda\}_{k=1}^K, 0)$, where (π^*, λ^*) is the saddle-point defined in Corollary 4.3.*

Proof. Note that $\mathcal{L}(\pi^*, \lambda^*) = \mathcal{L}(\pi^*, 0)$ since $V_{c,1}^{\pi^*} = 0$ for all $k \in [K] = \{1, ..., K\}$. Since by definition, for any π, $\mathcal{L}^k(\pi, 0) = V_{r,1}^{\pi}(s_1^k)$, we have the following:

$$\begin{aligned}
&\textstyle\sum_{k=1}^K V_{r,1}^{\pi^*}(s_1^k) - V_{r,1}^{\pi^k}(s_1^k) = \sum_{k=1}^K \mathcal{L}^k(\pi^*, \lambda^*) - \mathcal{L}^k(\pi^k, 0)\\
&= \textstyle\sum_{k=1}^K \mathcal{L}^k(\pi^*, \lambda^*) - \mathcal{L}^k(\pi^k, \lambda^k) + \mathcal{L}^k(\pi^k, \lambda^k) - \mathcal{L}^k(\pi^k, 0)\\
&\leq R_p(\{\pi\}_{k=1}^K, \pi^*) + R_d(\{\lambda\}_{k=1}^K, 0)
\end{aligned}$$

where the last inequality follows from Lemma 4.7 and Definition 4.4. □

5. Reset-Free Learning for Linear MDP

To demonstrate the utility of our reduction, we design a provably correct reset-free algorithm instantiation for linear MDP. This example serves to ground our abstract framework and to illustrate concretely how an algorithm instantiating our framework might operate. We also aim to show that despite generality, our framework does not lose any optimality in the rates of regret and total number of resets when compared to similar but specialized works.

We consider a linear MDP setting, which is common in the RL theory literature (Jin et al., 2019; Amani et al., 2021; Ghosh et al., 2022).

Assumption 5.1. We assume (S, A, P, r, c, H) is linear with a known feature map $\phi : S \times A \to \mathbb{R}^d$: for any $h \in [H]$, there exists d unknown signed measures $\mu_h = \{\mu_h^1, ..., \mu_h^d\}$ over S such that for any $(s, a, s') \in S \times A \times S$, we have

$$P_h(s'|a) = \langle \phi(s, a), \mu_h(s') \rangle,$$

and there exists unknown vectors $\omega_{r,h}, \omega_{c,h} \in \mathbb{R}^d$ such that for any $(s, a) \in S \times A$,

$$r_h(s, a) = \langle \phi(s, a), \omega_{r,h} \rangle, \quad c_h(s, a) = \langle \phi(s, a), \omega_{c,h} \rangle.$$

We assume, for all $(s, a, h) \in S \times A \times [H]$, $||\phi(s, a)||_2 \leq 1$, and $\max\{||\mu_h(s)||_2, ||\omega_{r,h}||_2, ||\omega_{c,h}||_2\} \leq \sqrt{d}$.

Note that the above assumption on the cost function does not imply knowledge of the reset states nor some hidden structure of the reset states. At the high level, it merely asks that the feature is expressive enough to separate the reset states and reset-free states in a classification problem.

In addition, we make a linearity assumption on the function λ^* defined in Theorem 4.1.

Assumption 5.2. We assume the knowledge of a feature $\xi : \mathcal{S} \to \mathbb{R}^d$ such that $\forall s \in \mathcal{S}$, $||\xi(s)||_2 \leq 1$ and $\lambda^*(s) = \langle \xi(s), \theta^* \rangle$ for some unknown vector $\theta^* \in \mathbb{R}^d$. In addition, we assume the knowledge of a convex set[5] $\mathcal{U} \subseteq \mathbb{R}^d$ such that $\theta^*, 0 \in \mathcal{U}$ and $\forall \theta \in \mathcal{U}$, $||\theta||_2 \leq B$ and $\langle \xi(s), \theta \rangle \geq 0$. [6]

[5] Such a set can be constructed by upper bounding the values using scaling and ensuring non-negativity by, e.g., sum of squares.

[6] From the previous section, we can see that the optimal function for the dual player is not necessarily unique. So, we assume bounds on at least one optimal function that we designate as $\lambda^*(s)$.

For a linear CMDP with a fixed initial state distribution, it is standard to assume that the optimal Lagrange multiplier (i.e., $\lambda^*(s_1)$) is bounded and non-negative. In this regard, our assumption is a mild and natural linear extension of this boundedness assumption for varying initial states, and is true when the feature map is expressive enough.

5.1. Algorithm

The basis of our algorithm lies between the interaction between the primal and dual players. We adopt the common no-regret plus best-response approach (Wang et al., 2021) to the no-regret two-player game in designing algorithms for these two players. We let the dual player perform (no-regret) projected gradient descent and the primal player update policies based on upper confidence bound (UCB) with knowledge of the dual player's decision (i.e., the best-response scheme to the dual player).

As discussed above, the environment is unknown to the agent. Therefore, as discussed in Section 4, we need to modify the actions of the primal player as compared to in the generic saddle-point problem to instantiate our framework. We do this using UCB, where the agent views unknown actions/states in a more positive ("optimistic")[7] way than the ones that have been observed to drive exploration in unknown environments.

Specifically, in each episode, upon receiving the initial state, we execute actions according to the policy based on a softmax (lines 5-8). Then, we perform the dual update through projected gradient descent. The dual player plays for the next round, $k+1$, after observing its loss after the primal player plays for the current round, k. The projection is to a l_2 ball containing $\lambda^*(\cdot)$ (lines 9-11). Finally, we perform the update of the primal player by computing the Q-functions for both the reward and cost with a bonus to encourage exploration (lines 12-20).

This algorithm uses Ghosh et al. (2022) as the baseline for the primal player. However, we emphasize that our algorithm can handle the adaptive initial state sequence that is seen in reset-free RL Theorems 4.1 and 4.5.

5.2. Analysis

We show below that our algorithm achieves regret and number of resets that are sublinear in the total number of time steps, KH, using Theorem 4.5. This result is asymptotically equivalent to Ghosh et al. (2022) and comparable to the bounds of $\tilde{O}(\sqrt{d^2H^6K})$ from Ding et al. (2021). Therefore, we do not make sacrifices in terms of the regret and total number of resets when specializing our abstract framework.

Theorem 5.3. *Under Assumptions 3.1, 5.1, and 5.2, with high probability, Regret$(K) \le \tilde{O}((B+1)\sqrt{d^3H^4K})$ and Resets$(K) \le \tilde{O}((B+1)\sqrt{d^3H^4K})$.*

Proof Sketch of Theorem 5.3 We provide a proof sketch here and defer the complete proof to Appendix A.2. We first bound the regret of $\{\pi^k\}_{k=1}^K$ and $\{\lambda^k\}_{k=1}^K$ and then use this to prove the bounds on our algorithm's regret and number of resets with Theorem 4.5.

We first bound the regret of $\{\lambda^k\}_{k=1}^K$.

Lemma 5.4. *Consider $\lambda_c(s) = \langle \xi(s), \theta_c \rangle$ for some $\theta_c \in \mathcal{U}$. Then it holds that $R_d(\{\lambda^k\}_{k=1}^K, \lambda_c) \le 1.5B\sqrt{K} + \sum_{k=1}^K (\lambda^k(s_1^k) - \lambda_c(s_1^k))(V_{c,1}^k(s_1^k) - V_{c,1}^{\pi^k}(s_1^k))$.*

Proof. We notice first an equality.

$$\begin{aligned}
&R_d(\{\lambda^k\}_{k=1}^K, \lambda_c) = \textstyle\sum_{k=1}^K \mathcal{L}^k(\pi^k, \lambda^k) - \mathcal{L}^k(\pi^k, \lambda_c) \\
&= \textstyle\sum_{k=1}^K \lambda_c(s_1^k) V_{c,1}^{\pi^k}(s_1^k) - \lambda^k(s_1^k) V_{c,1}^{\pi^k}(s_1^k) \\
&= \textstyle\sum_{k=1}^K (\lambda^k(s_1^k) - \lambda_c(s_1^k))(-V_{c,1}^k(s_1^k)) \\
&+ \textstyle\sum_{k=1}^K (\lambda^k(s_1^k) - \lambda_c(s_1^k))(V_{c,1}^k(s_1^k) - V_{c,1}^{\pi^k}(s_1^k)).
\end{aligned}$$

We observe that the first term is an online linear problem for θ^k (the parameter of $\lambda^k(\cdot)$). In episode $k \in [K]$, λ^k is played, and then the loss is revealed. Since the space of θ^k is convex, we use standard results (see Lemma 3.1 in Hazan et al. (2016)) to show that updating θ^k through projected gradient descent results in an upper bound for $\sum_{k=1}^K (\lambda^k(s_1^k) - \lambda_c(s_1^k))(-V_{c,1}^k(s_1^k))$. □

We now bound the regret of $\{\pi\}_{k=1}^K$.

Lemma 5.5. *Consider any π_c. With high probability, $R_p(\{\pi\}_{k=1}^K, \pi_c) \le 2H(1+B+H) + \sum_{k=1}^K V_{r,1}^k(s_1^k) - V_{r,1}^{\pi^k}(s_1^k) + \lambda^k(s_1^k)(V_{c,1}^{\pi^k}(s_1^k) - V_{c,1}^k(s_1^k))$.*

Proof. First we expand the regret into two terms.

$$\begin{aligned}
&R_p(\{\pi\}_{k=1}^K, \pi_c) = \textstyle\sum_{k=1}^K \mathcal{L}^k(\pi_c, \lambda^k) - \mathcal{L}^k(\pi^k, \lambda^k) \\
&= \textstyle\sum_{k=1}^K V_{r,1}^{\pi_c}(s_1^k) - \lambda^k(s_1^k) V_{c,1}^{\pi_c}(s_1^k) - [V_{r,1}^{\pi^k}(s_1^k) - \lambda^k(s_1^k) V_{c,1}^{\pi^k}(s_1^k)] \\
&= \textstyle\sum_{k=1}^K V_{r,1}^{\pi_c}(s_1^k) - \lambda^k(s_1^k) V_{c,1}^{\pi_c}(s_1^k) - [V_{r,1}^k(s_1^k) - \lambda^k(s_1^k) V_{c,1}^k(s_1^k)] \\
&\quad + \textstyle\sum_{k=1}^K V_{r,1}^k(s_1^k) - V_{r,1}^{\pi^k}(s_1^k) + \lambda^k(s_1^k)(V_{c,1}^{\pi^k}(s_1^k) - V_{c,1}^k(s_1^k)).
\end{aligned}$$

To bound the first term, we use Lemma 3 from Ghosh et al. (2022), which characterizes the property of upper confidence bound. □

Lastly, we derive a bound on $R_d(\{\lambda^k\}_{k=1}^K, \lambda_c) + R_p(\{\pi^k\}_{k=1}^K, \pi_c)$, which directly implies our final upper bound on Regret(K) and Resets(K) in Theorem 5.3 by Theorem 4.5. Combining the upper bounds in Lemma 5.4 and Lemma 5.5, we have a high-probability upper bound of

$$\begin{aligned}
&R_d(\{\lambda^k\}_{k=1}^K, \lambda_c) + R_p(\{\pi^k\}_{k=1}^K, \pi_c) \\
&\le 1.5B\sqrt{K} + 2H(1+B+H) + \\
&+ \textstyle\sum_{k=1}^K V_{r,1}^k(s_1^k) - V_{r,1}^{\pi^k}(s_1^k) + \lambda_c(s_1^k)(V_{c,1}^{\pi^k}(s_1^k) - V_{c,1}^k(s_1^k))
\end{aligned}$$

[7] Note that the term "optimism" here refers to the use of overestimation in exploration, which is different from the usage in, e.g., optimistic mirror descent (Mertikopoulos et al., 2018).

Algorithm 1 Primal-Dual Reset-Free RL Algorithm for Linear MDP with Adaptive Initial States

Input: Feature maps ϕ and ξ. Failure probability p. Some universal constant $\mathcal{C}$.
Initialization: $\theta^1 = 0$, $w_{r,h} = 0$, $w_{c,h} = 0$, $\alpha = \dfrac{\log(|\mathcal{A}|)K}{2(1+B+H)}$, $\beta = \mathcal{C}dH\sqrt{\log(4\log|\mathcal{A}|dKH/p)}$
for episodes $k = 1, ...K$ **do**
  Observe the initial state s_1^k.
  for step $h = 1, ..., H$ **do**
    Compute $\pi_{h,k}(a|\cdot) \leftarrow \dfrac{\exp(\alpha(Q_{r,h}^k(\cdot,a) - \lambda^k(s_1^k)Q_{c,h}^k(\cdot,a)))}{\sum_a \exp(\alpha(Q_{r,h}^k(\cdot,a) - \lambda^k(s_1^k)Q_{c,h}^k(\cdot,a)))}$.
    Take action $a_h^k \sim \pi_{h,k}(\cdot|s_h^k)$ and observe s_{h+1}^k.
  end for
  $\eta_k \leftarrow B/\sqrt{k}$
  Update $\theta^{k+1} \leftarrow \text{Proj}_{\mathcal{U}}(\theta^k + \eta_k \cdot \xi(s_1^k)V_{c,1}^k(s_1^k))$
  $\lambda^{k+1}(\cdot) \leftarrow \langle \theta^{k+1}, \xi(\cdot)\rangle$
  for step $h = H, ..., 1$ **do**
    $\Lambda_h^{k+1} \leftarrow \sum_{i=1}^{k} \phi(s_h^i, a_h^i)\phi(s_h^i, a_h^i)^T + \lambda\mathbb{I}$.
    $w_{r,h}^{k+1} \leftarrow (\Lambda_h^{k+1})^{-1}[\sum_{i=1}^{k} \phi(s_h^i, a_h^i)[r_h(s_h^i, a_h^i) + V_{r,h+1}^{k+1}(s_{h+1}^i)]]$
    $w_{c,h}^{k+1} \leftarrow (\Lambda_h^{k+1})^{-1}[\sum_{i=1}^{k} \phi(s_h^i, a_h^i)[c_h(s_h^i, a_h^i) + V_{c,h+1}^{k+1}(s_{h+1}^i)]]$
    $Q_{r,h}^{k+1}(\cdot,\cdot) \leftarrow \max\{\min\{\langle w_{r,h}^{k+1}, \phi(\cdot,\cdot)\rangle + \beta(\phi(\cdot,\cdot)^T(\Lambda_h^{k+1})^{-1}\phi(\cdot,\cdot))^{1/2}, H-h+1\}, 0\}$
    $Q_{c,h}^{k+1}(\cdot,\cdot) \leftarrow \max\{\min\{\langle w_{c,h}^{k+1}, \phi(\cdot,\cdot)\rangle - \beta(\phi(\cdot,\cdot)^T(\Lambda_h^{k+1})^{-1}\phi(\cdot,\cdot))^{1/2}, 1\}, 0\}$
    $V_{r,h}^{k+1}(\cdot) = \sum_a \pi_{h,k}(a|\cdot)Q_{r,h}^{k+1}(\cdot,a)$
    $V_{c,h}^{k+1}(\cdot) = \sum_a \pi_{h,k}(a|\cdot)Q_{c,h}^{k+1}(\cdot,a)$
  end for
end for

where the last term is the overestimation error due to optimism. Note that for all $k \in [K]$, $V_{r,1}^k(s_1^k)$ and $V_{c,1}^k(s_1^k)$ are as defined in Algorithm 1 and are optimistic estimates of $V_{r,1}^{\pi^*}(s_1^k)$ and $V_{c,1}^{\pi^*}(s_1^k)$. To bound this term, we use Lemma 4 from (Ghosh et al., 2022).

5.3. Other Possible Instantiations

We demonstrated above that our general framework can be instantiated to achieve sublinear regret and total number of resets. Importantly, our algorithm serves as an example of how our general framework can be used to systematically design new algorithms for reset-free RL. We can leverage the multitude of existing algorithms that aim to minimize regret in a two-player game. An example of a different strategy is using a no-regret algorithm like optimistic mirror descent (Mertikopoulos et al., 2018) for the dual player. We can also replace UCB for the primal player with an online MDP no-regret algorithm such as a variation of POLITEX (Abbasi-Yadkori et al., 2019). Further studying different combinations of baseline algorithms is an interesting future research direction, which perhaps could even improve existing algorithms for more specialized settings.

6. Conclusion

We propose a generic no-regret reduction for designing provable reset-free RL algorithms. Our reduction casts reset-free RL into the regret minimization problem of a two-player game, for which many existing no-regret algorithms are available. As a result, we can reuse these techniques, and future better techniques, to systematically build new reset-free RL algorithms. In particular, we design a reset-free RL algorithm for linear MDPs using our new reduction techniques, taking the first step towards designing provable reset-free RL algorithms. Extending these techniques to nonlinear function approximators and verifying their effectiveness empirically are important future research directions.

Acknowledgements

Part of this work was done during Hoai-An Nguyen's internship at Microsoft Research.

References

Abbasi-Yadkori, Y., Bartlett, P. L., Bhatia, K., Lazic, N., Szepesvari, C., and Weisz, G. POLITEX: regret bounds for policy iteration using expert prediction. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3692–3702. PMLR, 2019. URL http://proceedings.mlr.press/v97/lazic19a.html.

Alshiekh, M., Bloem, R., Ehlers, R., Könighofer, B., Niekum, S., and Topcu, U. Safe reinforcement learning via shielding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Altman, E. *Constrained Markov decision processes: stochastic modeling*. Routledge, 1999.

Amani, S. and Yang, L. F. Doubly pessimistic algorithms for strictly safe off-policy optimization. In *Annual Conference on Information Sciences and Systems*, pp. 113–118, 2022. doi: 10.1109/CISS53076.2022.9751158.

Amani, S., Thrampoulidis, C., and Yang, L. Safe reinforcement learning with linear function approximation. In *International Conference on Machine Learning*, pp. 243–253. PMLR, 2021.

Benzi, M., Golub, G. H., and Liesen, J. Numerical solution of saddle point problems. *Acta Numerica*, 14:1–137, 2005. doi: 10.1017/S0962492904000212.

Boyd, S. P. and Vandenberghe, L. *Convex Optimization*. Cambridge University Press, 2014. ISBN 978-0-521-83378-3. doi: 10.1017/CBO9780511804441. URL https://web.stanford.edu/%7Eboyd/cvxbook/.

Cheng, C.-A., Kolobov, A., and Swaminathan, A. Heuristic-guided reinforcement learning. *Advances in Neural Information Processing Systems*, 34:13550–13563, 2021.

Ding, D., Wei, X., Yang, Z., Wang, Z., and Jovanovic, M. Provably efficient safe exploration via primal-dual policy optimization. In *International Conference on Artificial Intelligence and Statistics*, pp. 3304–3312. PMLR, 2021.

Dong, K., Wang, Y., Chen, X., and Wang, L. Q-learning with UCB exploration is sample efficient for infinite-horizon MDP. *CoRR*, abs/1901.09311, 2019. URL http://arxiv.org/abs/1901.09311.

Efroni, Y., Mannor, S., and Pirotta, M. Exploration-exploitation in constrained mdps. *arXiv preprint arXiv:2003.02189*, 2020.

Eysenbach, B., Gu, S., Ibarz, J., and Levine, S. Leave no trace: Learning to reset for safe and autonomous reinforcement learning. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=S1vuO-bCW.

Garcia Polo, F. J. and Fernandez Rebollo, F. Safe reinforcement learning in high-risk tasks through policy improvement. In *IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning*, pp. 76–83, 2011. doi: 10.1109/ADPRL.2011.5967356.

Ghosh, A., Zhou, X., and Shroff, N. Provably efficient model-free constrained rl with linear function approximation. *arXiv preprint arXiv:2206.11889*, 2022.

Gupta, A., Yu, J., Zhao, T. Z., Kumar, V., Rovinsky, A., Xu, K., Devlin, T., and Levine, S. Reset-free reinforcement learning via multi-task learning: Learning dexterous manipulation behaviors without human intervention. In *IEEE International Conference on Robotics and Automation*, pp. 6664–6671. IEEE, 2021.

Ha, S., Xu, P., Tan, Z., Levine, S., and Tan, J. Learning to walk in the real world with minimal human effort. *Conference on Robot Learning*, 2020.

HasanzadeZonuzy, A., Bura, A., Kalathil, D., and Shakkottai, S. Learning with safety constraints: Sample complexity of reinforcement learning for constrained mdps. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 7667–7674, 2021.

Hazan, E. et al. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.

Hernández-Lerma, O. and Lasserre, J. B. *The Linear Programming Approach*, pp. 377–407. Springer US, Boston, MA, 2002. ISBN 978-1-4615-0805-2. doi: 10.1007/978-1-4615-0805-2_12. URL https://doi.org/10.1007/978-1-4615-0805-2_12.

Ho-Nguyen, N. and Kılınç-Karzan, F. Primal–dual algorithms for convex optimization via regret minimization. *IEEE Control Systems Letters*, 2(2):284–289, 2018. doi: 10.1109/LCSYS.2018.2831721.

Huang, R., Yang, J., and Liang, Y. Safe exploration incurs nearly no additional sample complexity for reward-free rl, 2022. URL https://arxiv.org/abs/2206.14057.

Jain, A., Vaswani, S., Babanezhad, R., Szepesvári, C., and Precup, D. Towards painless policy optimization for constrained MDPs. In Cussens, J. and Zhang, K. (eds.), *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, volume 180 of *Proceedings of*

Machine Learning Research, pp. 895–905. PMLR, 01–05 Aug 2022. URL https://proceedings.mlr.press/v180/jain22a.html.

Jin, C., Yang, Z., Wang, Z., and Jordan, M. I. Provably efficient reinforcement learning with linear function approximation. *CoRR*, abs/1907.05388, 2019. URL http://arxiv.org/abs/1907.05388.

Kakade, S. and Langford, J. Approximately optimal approximate reinforcement learning. In *International Conference on Machine Learning*. Citeseer, 2002.

Kovalev, D. and Gasnikov, A. The first optimal algorithm for smooth and strongly-convex-strongly-concave minimax optimization. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=pD5Pl5hen_g.

Lu, K., Grover, A., Abbeel, P., and Mordatch, I. Reset-free lifelong learning with skill-space planning. *arXiv preprint arXiv:2012.03548*, 2020.

Mertikopoulos, P., Lecouat, B., Zenati, H., Foo, C.-S., Chandrasekhar, V., and Piliouras, G. Optimistic mirror descent in saddle-point problems: Going the extra (gradient) mile. *arXiv preprint arXiv:1807.02629*, 2018.

Qiu, S., Wei, X., Yang, Z., Ye, J., and Wang, Z. Upper confidence primal-dual optimization: Stochastically constrained markov decision processes with adversarial losses and unknown transitions. *arXiv preprint arXiv:2003.00660*, 2020.

Sharma, A., Ahmad, R., and Finn, C. A state-distribution matching approach to non-episodic reinforcement learning. *arXiv preprint arXiv:2205.05212*, 2022.

Wachi, A. and Sui, Y. Safe reinforcement learning in constrained markov decision processes. In *International Conference on Machine Learning*, pp. 9797–9806. PMLR, 2020.

Wagener, N. C., Boots, B., and Cheng, C.-A. Safe reinforcement learning using advantage-based intervention. In *International Conference on Machine Learning*, pp. 10630–10640. PMLR, 2021.

Wang, J.-K., Abernethy, J., and Levy, K. Y. No-regret dynamics in the fenchel game: A unified framework for algorithmic convex optimization. *arXiv preprint arXiv:2111.11309*, 2021.

Wang, Y. and Li, J. Improved algorithms for convex-concave minimax optimization, 2020.

Wei, C., Jafarnia-Jahromi, M., Luo, H., Sharma, H., and Jain, R. Model-free reinforcement learning in infinite-horizon average-reward markov decision processes. *CoRR*, abs/1910.07072, 2019. URL http://arxiv.org/abs/1910.07072.

Wei, C., Jafarnia-Jahromi, M., Luo, H., and Jain, R. Learning infinite-horizon average-reward mdps with linear function approximation. *CoRR*, abs/2007.11849, 2020. URL https://arxiv.org/abs/2007.11849.

Wei, H., Liu, X., and Ying, L. A provably-efficient model-free algorithm for constrained markov decision processes. *arXiv preprint arXiv:2106.01577*, 2021.

Zheng, L. and Ratliff, L. Constrained upper confidence reinforcement learning. In Bayen, A. M., Jadbabaie, A., Pappas, G., Parrilo, P. A., Recht, B., Tomlin, C., and Zeilinger, M. (eds.), *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, volume 120 of *Proceedings of Machine Learning Research*, pp. 620–629. PMLR, 10–11 Jun 2020. URL https://proceedings.mlr.press/v120/zheng20a.html.

A. Appendix

A.1. Missing Proofs for Section 4

A.1.1. Proof of Theorem 4.1

Theorem 4.1. *There exists a function $\hat{\lambda}(\cdot)$ where for each s,*

$$\hat{\lambda}(s) \in \arg\min_{y\geq 0}\left(\max_{\pi\in\Delta} V_{r,1}^{\pi}(s) - yV_{c,1}^{\pi}(s)\right),$$

and a Markovian policy $\pi^* \in \Delta$, such that $(\pi^*, \hat{\lambda})$ is a saddle-point to the CMDPs

$$\max_{\pi\in\Delta} V_{r,1}^{\pi}(s_1),\ \textit{s.t.}\ V_{c,1}^{\pi}(s_1) \leq 0$$

for all *initial states $s_1 \in \mathcal{S}$ such that the CMDP is feasible. That is, for all $\pi \in \Delta$, $\lambda : \mathcal{S} \to \mathbb{R}$, and $s_1 \in \mathcal{S}$,*

$$\begin{aligned} V_{r,1}^{\pi^*}(s_1) - \lambda(s_1)V_{c,1}^{\pi^*}(s_1) &\geq V_{r,1}^{\pi^*}(s_1) - \hat{\lambda}(s_1)V_{c,1}^{\pi^*}(s_1) \\ &\geq V_{r,1}^{\pi}(s_1) - \hat{\lambda}(s_1)V_{c,1}^{\pi}(s_1). \end{aligned} \tag{6}$$

For policy π^*, we define it by the following construction (we ignore writing out the time dependency for simplicity): first, we define a cost-based MDP $\mathcal{M}_c = (\mathcal{S}, \mathcal{A}, P, c, H)$. Let $Q_c^*(s,a) = \min_{\pi\in\Delta} Q_c^{\pi}(s,a)$ and $V_c^*(s) = \min_{\pi\in\Delta} V_c^{\pi}(s)$ be the optimal values, where we recall V_c^{π} and Q_c^{π} are the state and state-action values under policy π with respect to the cost. Now we construct another reward-based MDP $\overline{\mathcal{M}} = (\mathcal{S}, \overline{\mathcal{A}}, P, r, H)$, where we define the state-dependent action space $\overline{\mathcal{A}}$ as

$$\overline{\mathcal{A}}_s = \{a \in \mathcal{A} : Q_c^*(s,a) \leq V_c^*(s)\}.$$

By definition, $\overline{\mathcal{A}}_s$ is non-empty for all s. We define a shorthand notation: we write $\pi \in \overline{\mathcal{A}}(s)$ if $\mathbb{E}_{\pi}[\sum_{t=1}^{H} \mathbb{1}\{a_t \notin \overline{\mathcal{A}}_{s_t}\}|s_1 = s] = 0$. Then we have the following lemma, which is a straightforward application of the performance difference lemma.

Lemma 4.6. *For any $s_1 \in \mathcal{S}$ such that $V_c^*(s_1) = 0$ and any $\pi \in \Delta$, it is true that $\pi \in \overline{\mathcal{A}}(s_1)$ if and only if $V_c^{\pi}(s_1) = 0$.*

Proof. By performance difference lemma (Kakade & Langford, 2002), we can write

$$V_c^{\pi}(s_1) - V_c^*(s_1) = \mathbb{E}_{\pi}\left[\sum_{t=1}^{H} Q_c^*(s_t,a_t) - V_c^*(s_t)|s_1 = s_1\right].$$

If for some $s_1 \in \mathcal{S}$, $\pi \in \overline{\mathcal{A}}(s_1)$, then $\mathbb{E}_{\pi}\left[\sum_{t=1}^{H} Q_c^*(s_t,a_t) - V_c^*(s_t)\right] \leq 0$, which implies $V_c^{\pi}(s_1) \leq V_c^*(s_1)$. But since V_c^* is optimal, $V_c^{\pi}(s_1) = V_c^*(s_1)$. On the other hand, suppose $V_c^{\pi}(s_1) = 0$. It implies $\mathbb{E}_{\pi}\left[\sum_{t=1}^{H} Q_c^*(s_t,a_t) - V_c^*(s_t)\right] = 0$ since $V_c^*(s_1) = 0$. Because by definition of optimality $Q_c^*(s_t,a_t) - V_c^*(s_t) \geq 0$, this implies $\pi \in \overline{\mathcal{A}}(s_1)$. □

We set our candidate policy π^* as the optimal policy of this $\overline{\mathcal{M}}$. By Lemma 4.6, we have $V_c^{\pi^*}(s) = V_c^*(s)$, so it is also an optimal policy to $\mathcal{M}_c$. We prove our main claim of Theorem 4.1 below:

$$V_{r,1}^{\pi^*}(s_1) - \lambda(s_1)V_{c,1}^{\pi^*}(s_1) \geq V_{r,1}^{\pi^*}(s_1) - \hat{\lambda}(s_1)V_{c,1}^{\pi^*}(s_1) \geq V_{r,1}^{\pi}(s_1) - \hat{\lambda}(s_1)V_{c,1}^{\pi}(s_1).$$

Proof. Because $V_{c,1}^{\pi^*}(s_1) = 0$ (for an initial state s_1 such that the CMDP is feasible), the first inequality is trivial:

$$V_{r,1}^{\pi^*}(s_1) - \lambda(s_1)V_{c,1}^{\pi^*}(s_1) = V_{r,1}^{\pi^*}(s_1) = V_{r,1}^{\pi^*}(s_1) - \hat{\lambda}(s_1)V_{c,1}^{\pi^*}(s_1).$$

For the second inequality, we use Lemma 4.6:

$$\begin{aligned} V_{r,1}^{\pi}(s_1) - \hat{\lambda}(s_1)V_{c,1}^{\pi}(s_1) &\leq \max_{\pi\in\Delta} V_{r,1}^{\pi}(s_1) - \hat{\lambda}(s_1)V_{c,1}^{\pi}(s_1) \\ &= \min_{y\geq 0}\max_{\pi\in\Delta} V_{r,1}^{\pi}(s_1) - yV_{c,1}^{\pi}(s_1) \\ &= \max_{\pi\in\overline{\mathcal{A}_c}(s_1)} V_{r,1}^{\pi}(s_1) && \text{(By Lemma 4.6)} \\ &= V_{r,1}^{\pi^*}(s_1) \\ &= V_{r,1}^{\pi^*}(s_1) - \hat{\lambda}(s_1)V_{c,1}^{\pi^*}(s_1). \end{aligned}$$

□

A.1.2. Proof of Corollary 4.2

Corollary 4.2. *For π^* in Theorem 4.1, it holds that Regret$(K) = \sum_{k=1}^K V_{r,1}^{\pi^*}(s_1^k) - V_{r,1}^{\pi^k}(s_1^k)$.*

Proof. To prove Regret$(K) = \sum_{k=1}^K V_{r,1}^{\pi^*}(s_1^k) - V_{r,1}^{\pi^k}(s_1^k)$, it suffices to prove $\sum_{k=1}^K V_{r,1}^{\pi^*}(s_1^k) = \max_{\pi\in\Delta_0(K)} \sum_{k=1}^K V_{r,1}^{\pi}(s_1^k)$. By Lemma 4.6 and under Assumption 3.1, we notice that $\max_{\pi\in\Delta_0(K)} \sum_{k=1}^K V_{r,1}^{\pi}(s_1^k) = \max_{\pi\in\overline{\mathcal{A}}(s_1^k),\forall k\in[K]} \sum_{k=1}^K V_{r,1}^{\pi}(s_1^k)$. This is equal to $\sum_{k=1}^K V_{r,1}^{\pi^*}(s_1^k)$ by the definition of π^* in the proof of Theorem 4.1. □

A.1.3. Proof of Corollary 4.3

Corollary 4.3. *For any saddle-point to the CMDPs*

$$\max_{\pi\in\Delta} V_{r,1}^{\pi}(s_1),\ \textit{s.t.}\ V_{c,1}^{\pi}(s_1) \le 0$$

of $(\pi^*, \hat{\lambda})$ from Theorem 4.1, $(\pi^*, \lambda^*) =: (\pi^*, \hat{\lambda}+1)$ is also a saddle-point as defined in (6).

Proof. We prove that eq (6) holds for (π^*, λ^*), that is

$$V_{r,1}^{\pi^*}(s_1) - \lambda(s_1)V_{c,1}^{\pi^*}(s_1) \ge V_{r,1}^{\pi^*}(s_1) - \lambda^*(s_1)V_{c,1}^{\pi^*}(s_1) \ge V_{r,1}^{\pi}(s_1) - \lambda^*(s_1)V_{c,1}^{\pi}(s_1).$$

Because $V_{c,1}^{\pi^*}(s_1) = 0$ (for an initial state s_1 such that the CMDP is feasible), the first inequality is trivial:

$$V_{r,1}^{\pi^*}(s_1) - \lambda(s_1)V_{c,1}^{\pi^*}(s_1) = V_{r,1}^{\pi^*}(s_1) = V_{r,1}^{\pi^*}(s_1) - \lambda^*(s_1)V_{c,1}^{\pi^*}(s_1).$$

For the second inequality, we use Theorem 4.1:

$$\begin{aligned} V_{r,1}^{\pi}(s_1) - \lambda^*(s_1)V_{c,1}^{\pi}(s_1) \le& V_{r,1}^{\pi}(s_1) - \hat{\lambda}(s_1)V_{c,1}^{\pi}(s_1) \\ \le& V_{r,1}^{\pi^*}(s_1) - \hat{\lambda}(s_1)V_{c,1}^{\pi^*}(s_1) \\ =& V_{r,1}^{\pi^*}(s_1) - \lambda^*(s_1)V_{c,1}^{\pi^*}(s_1) \end{aligned}$$

where the first step is because $V_{c,1}^{\pi}(s_1)$ by definition is in $[0,1]$ and $\lambda^* = \hat{\lambda}+1$, and the second step is by Theorem 4.1. □

A.1.4. Proof of Theorem 4.5

Theorem 4.5. *Under Assumption 3.1, for any sequences $\{\pi^k\}_{k=1}^K$ and $\{\lambda^k\}_{k=1}^K$, it holds that*

$$\begin{aligned} \textit{Regret}(K) &\le R_p(\{\pi^k\}_{k=1}^K, \pi^*) + R_d(\{\lambda^k\}_{k=1}^K, 0) \\ \textit{Resets}(K) &\le R_p(\{\pi^k\}_{k=1}^K, \pi^*) + R_d(\{\lambda^k\}_{k=1}^K, \lambda^*) \end{aligned}$$

where (π^*, λ^*) is the saddle-point defined in Corollary 4.3.

We first establish the following intermediate result that will help us with our decomposition.

Lemma 4.7. *For any primal-dual sequence $\{\pi^k, \lambda^k\}_{k=1}^K$, $\sum_{k=1}^K(\mathcal{L}^k(\pi^*, \lambda') - \mathcal{L}^k(\pi^k, \lambda^k)) \le R_p(\{\pi\}_{k=1}^K, \pi^*)$, where (π^*, λ') is the saddle-point defined in either Theorem 4.1 or Corollary 4.3.*

Proof. We derive this lemma by Theorem 4.1 and Corollary 4.3. First notice by Theorem 4.1 and Corollary 4.3 that for $\lambda' = \lambda^*, \hat{\lambda}$,

$$\begin{aligned} \sum_{k=1}^K \mathcal{L}^k(\pi^*, \lambda') &= \sum_{k=1}^K V_{r,1}^{\pi^*}(s_1^k) - \lambda'(s_1^k)V_{c,1}^{\pi^*}(s_1^k) \\ &\le \sum_{k=1}^K V_{r,1}^{\pi^*}(s_1^k) - \lambda^k(s_1^k)V_{c,1}^{\pi^*}(s_1^k) = \sum_{k=1}^K \mathcal{L}^k(\pi^*, \lambda^k). \end{aligned}$$

Then we can derive

$$\sum_{k=1}^{K}(\mathcal{L}^k(\pi^*,\lambda') - \mathcal{L}^k(\pi^k,\lambda^k)) = \sum_{k=1}^{K}\mathcal{L}^k(\pi^*,\lambda') - \mathcal{L}^k(\pi^*,\lambda^k) + \mathcal{L}^k(\pi^*,\lambda^k) - \mathcal{L}^k(\pi^k,\lambda^k)$$
$$\leq \sum_{k=1}^{K}\mathcal{L}^k(\pi^*,\lambda^k) - \mathcal{L}^k(\pi^k,\lambda^k) = R_p(\{\pi\}_{k=1}^K,\pi^*)$$

which finishes the proof. □

Then we upper bound Regret(K) and Resets(K) by $R_p(\{\pi^k\}_{k=1}^K,\pi_c)$ and $R_d(\{\lambda^k\}_{k=1}^K,\lambda_c)$ for suitable comparators. This decomposition is inspired by the techniques used in Ho-Nguyen & Kılınç-Karzan (2018).

We first bound Resets(K).

Lemma 4.8. *For any primal-dual sequence $\{\pi^k,\lambda^k\}_{k=1}^K$, $\sum_{k=1}^K V_{c,1}^{\pi^k}(s_1^k) \leq R_p(\{\pi\}_{k=1}^K,\pi^*) + R_d(\{\lambda\}_{k=1}^K,\lambda^*)$, where (π^*,λ^*) is the saddle-point defined in Corollary 4.3.*

Proof. Notice $\sum_{k=1}^K V_{c,1}^{\pi^k}(s_1^k) = \sum_{k=1}^K \mathcal{L}^k(\pi^k,\hat{\lambda}) - \mathcal{L}^k(\pi^k,\lambda^*)$ where $(\pi^*,\hat{\lambda})$ is the saddle-point defined in Theorem 4.1. This is because, as defined, $\lambda^* = \hat{\lambda}+1$. Therefore, we bound the RHS. We have

$$\sum_{k=1}^{K}\mathcal{L}^k(\pi^k,\hat{\lambda}) - \mathcal{L}^k(\pi^k,\lambda^*) = \sum_{k=1}^{K}\mathcal{L}^k(\pi^k,\hat{\lambda}) - \mathcal{L}^k(\pi^k,\lambda^k) + \mathcal{L}^k(\pi^k,\lambda^k) - \mathcal{L}^k(\pi^k,\lambda^*)$$
$$\leq \sum_{k=1}^{K}\mathcal{L}^k(\pi^*,\hat{\lambda}) - \mathcal{L}^k(\pi^k,\lambda^k) + \mathcal{L}^k(\pi^k,\lambda^k) - \mathcal{L}^k(\pi^k,\lambda^*)$$
$$\leq R_p(\{\pi\}_{k=1}^K,\pi^*) + R_d(\{\lambda\}_{k=1}^K,\lambda^*)$$

where second inequality is because $\sum_{k=1}^K \mathcal{L}^k(\pi^*,\hat{\lambda}) \geq \sum_{k=1}^K \mathcal{L}^k(\pi^k,\hat{\lambda})$ by Theorem 4.1, and the first inequality follows from Lemma 4.7 and Definition 4.4. □

Lastly, we bound Regret(K) with the lemma below and Corollary 4.2.

Lemma 4.9. *For any primal-dual sequence $\{\pi^k,\lambda^k\}_{k=1}^K$, $\sum_{k=1}^K(V_{r,1}^{\pi^*}(s_1^k) - V_{r,1}^{\pi^k}(s_1^k)) \leq R_p(\{\pi\}_{k=1}^K,\pi^*) + R_d(\{\lambda\}_{k=1}^K,0)$, where (π^*,λ^*) is the saddle-point defined in Corollary 4.3.*

Proof. Note that $\mathcal{L}(\pi^*,\lambda^*) = \mathcal{L}(\pi^*,0)$ since $V_{c,1}^{\pi^*}(s_1^k) = 0$ for all $k \in [K]$. Since by definition, for any π, $\mathcal{L}^k(\pi,0) = V_{r,1}^{\pi}(s_1^k)$, we have the following:

$$\sum_{k=1}^{K} V_{r,1}^{\pi^*}(s_1^k) - V_{r,1}^{\pi^k}(s_1^k) = \sum_{k=1}^{K}\mathcal{L}^k(\pi^*,\lambda^*) - \mathcal{L}^k(\pi^k,0)$$
$$= \sum_{k=1}^{K}\mathcal{L}^k(\pi^*,\lambda^*) - \mathcal{L}^k(\pi^k,\lambda^k) + \mathcal{L}^k(\pi^k,\lambda^k) - \mathcal{L}^k(\pi^k,0)$$
$$\leq R_p(\{\pi\}_{k=1}^K,\pi^*) + R_d(\{\lambda\}_{k=1}^K,0)$$

where the last inequality follows from Lemma 4.7 and Definition 4.4. □

A.2. Missing Proofs for Section 5

A.2.1. Proof of Theorem 5.3

Theorem 5.3. *Under Assumptions 3.1, 5.1, and 5.2, with high probability, Regret$(K) \leq \tilde{O}((B+1)\sqrt{d^3H^4K})$ and Resets$(K) \leq \tilde{O}((B+1)\sqrt{d^3H^4K})$.*

We first bound the regret of $\{\pi^k\}_{k=1}^K$ and $\{\lambda^k\}_{k=1}^K$, and then use this to prove the bounds on our algorithm's regret and number of resets with Theorem 4.5.

We first bound the regret of $\{\lambda^k\}_{k=1}^K$.

Lemma 5.4. *Consider $\lambda_c(s) = \langle \xi(s), \theta_c \rangle$ for some $\theta_c \in \mathcal{U}$. Then it holds that $R_d(\{\lambda^k\}_{k=1}^K, \lambda_c) \leq 1.5B\sqrt{K} + \sum_{k=1}^K (\lambda^k(s_1^k) - \lambda_c(s_1^k))(V_{c,1}^k(s_1^k) - V_{c,1}^{\pi^k}(s_1^k))$.*

Proof. We notice first an equality.

$$
\begin{aligned}
R_d(\{\lambda^k\}_{k=1}^K, \lambda_c) &= \sum_{k=1}^K \mathcal{L}^k(\pi^k, \lambda^k) - \mathcal{L}^k(\pi^k, \lambda_c) \\
&= \sum_{k=1}^K \lambda_c(s_1^k) V_{c,1}^{\pi^k}(s_1^k) - \lambda^k(s_1^k) V_{c,1}^{\pi^k}(s_1^k) \\
&= \sum_{k=1}^K \lambda_c(s_1^k) V_{c,1}^{\pi^k}(s_1^k) - \lambda^k(s_1^k) V_{c,1}^{\pi^k}(s_1^k) \\
&\quad + \sum_{k=1}^K \lambda_c(s_1^k) V_{c,1}^k(s_1^k) - \lambda_c(s_1^k) V_{c,1}^k(s_1^k) + \lambda^k(s_1^k) V_{c,1}^k(s_1^k) - \lambda^k(s_1^k) V_{c,1}^k(s_1^k) \\
&= \sum_{k=1}^K (\lambda^k(s_1^k) - \lambda_c(s_1^k))(-V_{c,1}^k(s_1^k)) + \sum_{k=1}^K (\lambda^k(s_1^k) - \lambda_c(s_1^k))(V_{c,1}^k(s_1^k) - V_{c,1}^{\pi^k}(s_1^k)).
\end{aligned}
$$

We observe that the first term is an online linear problem for θ^k (the parameter of $\lambda^k(\cdot)$). In episode $k \in [K]$, λ^k is played, and then the loss is revealed. Since the space of θ^k is convex, we use standard results (Lemma 3.1 (Hazan et al., 2016)) to show that updating θ^k through projected gradient descent results in an upper bound for $\sum_{k=1}^K (\lambda^k(s_1^k) - \lambda_c(s_1^k))(-V_{c,1}^k(s_1^k))$. We restate the lemma here.

Lemma A.1 (Lemma 3.1 (Hazan et al., 2016))**.** *Let $\mathcal{S} \subseteq \mathbb{R}^d$ be a bounded convex and closed set in Euclidean space. Denote D as an upper bound on the diameter of $\mathcal{S}$, and G as an upper bound on the norm of the subgradients of convex cost functions f_k over $\mathcal{S}$. Using online projected gradient descent to generate sequence $\{x_k\}_{k=1}^K$ with step sizes $\{\eta_k = \frac{D}{G\sqrt{k}}, k \in [K]\}$ guarantees, for all $K \geq 1$:*

$$
\textit{Regret}_K = \max_{x^* \in \mathcal{K}} \sum_{k=1}^K f_k(x^k) - f_k(x^*) \leq 1.5GD\sqrt{K}.
$$

Let us bound D. By Assumption 5.2, $\lambda^* = \langle \xi(s), \theta^* \rangle$ and $||\theta^*||_2 \leq B$. Since the comparator we use is λ^*, we can set D to be B. To bound G, we observe that the subgradient of our loss function is $\xi(s) V_{c,1}^k(s_1^k)$ for each $k \in [K]$. Therefore, since $V_{c,1}^k(s_1^k) \in [0, 1]$ and $||\xi(s)||_2 \leq 1$ by Assumption 5.2, we can set G to be 1. □

We now bound the regret of $\{\pi\}_{k=1}^K$.

Lemma 5.5. *Consider any π_c. With high probability, $R_p(\{\pi\}_{k=1}^K, \pi_c) \leq 2H(1 + B + H) + \sum_{k=1}^K V_{r,1}^k(s_1^k) - V_{r,1}^{\pi^k}(s_1^k) + \lambda^k(s_1^k)(V_{c,1}^{\pi^k}(s_1^k) - V_{c,1}^k(s_1^k))$.*

Proof. First we expand the regret into two terms.

$$
\begin{aligned}
R_p(\{\pi\}_{k=1}^K, \pi_c) =& \sum_{k=1}^K \mathcal{L}^k(\pi_c, \lambda^k) - \mathcal{L}^k(\pi^k, \lambda^k) \\
=& \sum_{k=1}^K V_{r,1}^{\pi_c}(s_1^k) - \lambda^k(s_1^k) V_{c,1}^{\pi_c}(s_1^k) - [V_{r,1}^{\pi^k}(s_1^k) - \lambda^k(s_1^k) V_{c,1}^{\pi^k}(s_1^k)] \\
=& \sum_{k=1}^K V_{r,1}^{\pi_c}(s_1^k) - \lambda^k(s_1^k) V_{c,1}^{\pi_c}(s_1^k) - [V_{r,1}^{\pi^k}(s_1^k) - \lambda^k(s_1^k) V_{c,1}^{\pi^k}(s_1^k)] \\
&+ \sum_{k=1}^K [V_{r,1}^k(s_1^k) - \lambda^k(s_1^k) V_{c,1}^k(s_1^k)] - [V_{r,1}^k(s_1^k) - \lambda^k(s_1^k) V_{c,1}^k(s_1^k)] \\
=& \sum_{k=1}^K V_{r,1}^{\pi_c}(s_1^k) - \lambda^k(s_1^k) V_{c,1}^{\pi_c}(s_1^k) - [V_{r,1}^k(s_1^k) - \lambda^k(s_1^k) V_{c,1}^k(s_1^k)] \\
&+ \sum_{k=1}^K V_{r,1}^k(s_1^k) - V_{r,1}^{\pi^k}(s_1^k) + \lambda^k(s_1^k)(V_{c,1}^{\pi^k}(s_1^k) - V_{c,1}^k(s_1^k)).
\end{aligned}
$$

To bound the first term, we use Lemma 3 from Ghosh et al. (2022), which characterize the property of upper confidence bound. For completeness, we re-write the lemma here. [8]

Lemma A.2 (Lemma 3 (Ghosh et al., 2022))**.** *With probability* $1-p/2$*, it holds that* $\mathcal{T}_1 = \sum_{k=1}^K \left(V_{r,1}^{\pi_c}(s_1^k) - \lambda^k V_{c,1}^{\pi_c}(s_1^k)\right) - \left(V_{r,1}^k(s_1^k) - \lambda^k V_{c,1}^k(s_1^k)\right) \leq KH\log(|\mathcal{A}|)/\alpha$*. Hence, for* $\alpha = \dfrac{\log(|\mathcal{A}|)K}{2(1+C+H)}$, $\mathcal{T}_1 \leq 2H(1+C+H)$*, where* C *is such that* $\lambda^k \leq C$*.*

In our problem setting, we can set $C = B$ in the lemma above. Therefore, the first term is bounded by $2H(1+B+H)$. □

Lastly, we derive a bound on $R_d(\{\lambda^k\}_{k=1}^K, \lambda_c) + R_p(\{\pi^k\}_{k=1}^K, \pi_c)$, which directly implies our final upper bound on Regret(K) and Resets(K) in Theorem 5.3 by Theorem 4.5.

Lemma A.3. *For any* π_c *and* $\lambda_c(s) = \langle \xi(s), \theta_c \rangle$ *such that* $\|\theta_c\| \leq B$*, we have with probability* $1-p$*,* $R_d(\{\lambda^k\}_{k=1}^K, \lambda_c) + R_p(\{\pi^k\}_{k=1}^K, \pi_c) \leq 1.5B\sqrt{K} + 2H(1+B+H) + O((B+1)\sqrt{d^3H^4K\iota^2})$ *where* $\iota = \log[\log(|\mathcal{A}|)4dKH/p]$*.*

Proof. Combining the upper bounds in Lemma 5.4 and Lemma 5.5, we have an upper bound of

$$
\begin{aligned}
R_d(\{\lambda^k\}_{k=1}^K, \lambda_c) + R_p(\{\pi^k\}_{k=1}^K, \pi_c) =& 1.5B\sqrt{K} + \sum_{k=1}^K (\lambda^k(s_1^k) - \lambda_c(s_1^k))(V_{c,1}^k(s_1^k) - V_{c,1}^{\pi^k}(s_1^k)) \\
&+ 2H(1+B+H) + \sum_{k=1}^K V_{r,1}^k(s_1^k) - V_{r,1}^{\pi^k}(s_1^k) + \lambda^k(s_1^k)(V_{c,1}^{\pi^k}(s_1^k) - V_{c,1}^k(s_1^k)) \\
=& 1.5B\sqrt{K} + 2H(1+B+H) + \\
&+ \sum_{k=1}^K V_{r,1}^k(s_1^k) - V_{r,1}^{\pi^k}(s_1^k) + \lambda_c(s_1^k)(V_{c,1}^{\pi^k}(s_1^k) - V_{c,1}^k(s_1^k))
\end{aligned}
$$

where the last term is the overestimation error due to optimism. To bound this term, we use Lemma 4 from Ghosh et al. (2022). We re-write the lemma here.

[8] Note that Ghosh et al. (2022) use a utility function rather than a cost function to denote the constraint on the MDP (cost is just $-1\times$ utility). Also note that their Lemma 3 is proved for an arbitrary initial state sequence and for any comparator (which includes π^*).

Lemma A.4 (Lemma 4 (Ghosh et al., 2022)). *WIth probability at least* $1-p/2$*, for any* $\lambda \in [0, C]$*,* $\sum_{k=1}^{K}\left(V_{r,1}^{k}(s_1^k) - V_{r,1}^{\pi^k}(s_1^k)\right) + \lambda\sum_{k=1}^{K}\left(V_{c,1}^{\pi^k}(s_1^k) - V_{c,1}^{k}(s_1^k)\right) \leq O((\lambda+1)\sqrt{d^3H^4K\iota^2})$ *where* $\iota = \log[\log(|\mathcal{A}|)4dKH/p]$.

Since we have a bound on all $\lambda_c(s_1^k)$ of B for all $k \in [K]$, we have a bound of $O((B+1)\sqrt{d^3H^4K\iota^2})$. □