# Concave Utility Reinforcement Learning with Zero-Constraint Violations

**Anonymous authors**
**Paper under double-blind review**

## Abstract

We consider the problem of tabular infinite horizon concave utility reinforcement learning (CURL) with convex constraints. For this, we propose a model-based learning algorithm that also achieves zero constraint violations. Assuming that the concave objective and the convex constraints have a solution interior to the set of feasible occupation measures, we solve a tighter optimization problem to ensure that the constraints are never violated despite the imprecise model knowledge and model stochasticity. We use Bellman error-based analysis for tabular infinite-horizon setups which allows analyzing stochastic policies. Combining the Bellman error-based analysis and tighter optimization equation, for $T$ interactions with the environment, we obtain a high-probability regret guarantee for objective which grows as $\tilde{O}(1/\sqrt{T})$, excluding other factors. The proposed method can be applied for optimistic algorithms to obtain high-probability regret bounds and also be used for posterior sampling algorithms to obtain a loose Bayesian regret bounds but with significant improvement in computational complexity.

## 1 Introduction

In many applications where a learning agent uses reinforcement learning to find optimal policies, the agent optimizes a concave function of the expected rewards or the agent must satisfy certain constraints while maximizing an objective (Altman & Schwartz, 1991; Roijers et al., 2013). For example, in network scheduling, a controller can maximize fairness of the users using a concave function of the average rewards of each of the users (Chen et al., 2021). Consider a scheduler which allocates a resource to users. Each user obtains some reward based on their current state. The goal of the scheduler is to maximize fairness among the users. However, there are certain preferred users for which some service level agreements (SLA) must be made. For this setup, the scheduler aims to find a policy which maximize the fairness while ensuring the SLA constraints of the preferred users are met. Note that, here, he objective is a non-linear concave utility in the presence of constraints on service level agreement. Setups with constraints also exists in autonomous vehicles where the goal is to reach the destination quickly while ensuring the safety of the surroundings (Le et al., 2019; Tessler et al., 2018). Further, an agent may aim to efficiently explore the environment by maximizing the entropy, a concave function, of the distribution induced over the state and action space (Hazan et al., 2019).

Owing to the variety of the use cases, recently, there has been significant effort to make RL algorithms for setups with constraints, or concave utilties, or both. For episodic setup, works range from model based algorithms (Brantley et al., 2020; Yu et al., 2021) to primal-dual based model-free algorithms (Ding et al., 2021). Recently, there has been a thrust towards developing algorithms which can also achieve zero-constraint violations in the learning phase as well (Wei et al., 2022a; Liu et al., 2021b;a). However, for the episodic setup, the majority of the current works consider the weaker regret definition specified by (Efroni et al., 2020) and only achieve zero expected constraint violations. Further, these algorithms require the knowledge of a safe policy following which the agent does not violate constraints, or the knowledge of the Slater's gap $\delta$ which determines how far a safe policy is from the constraint boundary.

The definition which considers the average over time makes sense for an infinite horizon setup as the long-term average is naturally defined (Puterman, 2014). For a tabular infinite-horizon setup, (Singh et al., 2020) proposed an optimistic epoch-based algorithm. Much recently, (Chen et al., 2022) proposed an Optimistic Online Mirror Descent based algorithm. In this work, we consider the problem of maximizing concave utility

of the expected rewards while also ensuring that a set of convex constraints of the expected rewards are also satisfied. Moreover, we aim to develop algorithms that can also ensure that the constraints are not violated during the training phase as well. We work with tabular MDP with infinite horizon. For such setup, our algorithm updates policies as it learns the system model. Further, our approach also bounds the accumulated observed constraint violations as compared to the expected constraint violations.

For infinite horizon setups for non-constrained setup, various works exist (Fruit et al., 2018; Jaksch et al., 2010). However, we note that the dealing with constraints and non-linear setup requires additional attention because of the stochastic policies. Further, unlike episodic setup, the distribution at the epoch is not constant and hence the policy switching cost has to be accounted explicitly. Prior works in infinite horizon also faced this issue and provide some tools to overcome this limitation. (Singh et al., 2020) builds confidence intervals for transition probability for every next state given the current state-action pair and obtains a regret bound of $O(T^{2/3})$. (Chen et al., 2022) obtains a regret bound of $O(T_M\sqrt{T})$ with $O(T_M^2S^3A)$ constraint violations for ergodic MDPs with $T_M$ mixing time following an analysis which works with confidence intervals on both transition probability vectors and value functions.

To overcome the limitations mentioned in previous analysis and to obtain a tighter result, we propose an optimism based UC-CURL algorithm which proceeds in epochs $e$. At each epoch, we solve for an policy which considers constraints tighter by $\epsilon_e$ than the true bounds for the optimistic MDP in the confidence intervals for the transition probabilities. Further, as the knowledge of the model improves with increased interactions with the environment, we reduce this tightness. This $\epsilon_e$-sequence is critical to our algorithm as, if the sequence decays too fast, the constraints violations cannot be bounded by zero. And, if this sequences decays too slow, the objective regret may not decay fast enough. Further, using the $\epsilon_e$-sequence, we do not require the knowledge of the total time $T$ for which the algorithm runs.

We bound our regret by bounding the gap between the optimal policy in the feasible region and the optimal policy for the optimization problem with $\epsilon_e$ tight constraints. We bound this gap with a multiplicative factor of $O(1/\delta)$, where $\delta$ is Slater's parameter. Based on our analysis using the Slater's parameter $\delta$, we consider a case where a lower bound $T_l$ on the time horizon $T$ is known. This knowledge of $T_l$ allows us to relax our assumption on $\delta$.

Further, for the regret analysis of the proposed UC-CURL algorithm, we use Bellman error for infinite horizon setup to bound the difference between the performance of optimistic policy on the optimistic MDP and the true MDP. Compared to analysis of (Jaksch et al., 2010), this allows us to work with stochastic policies. We bound our regret as $\tilde{O}(\frac{1}{\delta}LdT_MS\sqrt{A/T} + CT_MS^2A/T(1-\rho))$ and constraint violations as 0, where $S$ and $A$ are the number of states and actions respectively, $L$ is the Lipschitz constant of the objective and constraint functions, $d$ is the number of costs the agent is trying to optimize, and $T_M$ is the mixing time of the MDP. The Bellman error based analysis along with Slater's slackness assumption also allows to develop posterior sampling based methods for constrained RL (see Appendix G) by showing feasibility of the optimization problem for the sampled MDPs.

To summarize our contributions, we improve prior results on infinite horizon concave utility reinforcement learning setup on multiple fronts. First, we consider convex function for objectives and constraints. Second, even with a non-linear function setup, we reduce the regret order to $O(T_MS\sqrt{A/T})$ and bound the constraint violations with 0. Third, our algorithm does not require the knowledge of the time horizon $T$, safe policy, or Slater's gap $\delta$. Lastly, we provide analysis for posterior sampling algorithm which improves both emprical performance and computational complexity.

## 2 Related Works

**Constrained RL:** (Altman, 1999) builds the formulation for constrained MDPs to study constrained reinforcement learning and provides algorithms for obtaining policies with known transition models. (Zheng & Ratliff, 2020) considered an episodic CMDP (Constrained Markov Decision Processes) and use an optimism based algorithm to bound the constraint violation as $\tilde{O}(1/T^{0.25})$ with high probability. (Kalagarla et al., 2020) also considered the episodic setup to obtain PAC-style bound for an optimism based algorithm. (Ding et al., 2021) considered the setup of $H$-episode length episodic CMDPs with $d$-dimensional linear function approximation to bound the constraint violations as $\tilde{O}(d\sqrt{H^5/T})$ by mixing the optimal policy with an exploration policy. (Efroni et al., 2020) proposes a linear-programming and primal-dual policy optimization

algorithm to bound the regret as $O(S\sqrt{H^3/T})$. (Wei et al., 2022a; Liu et al., 2021a) considered the problem of ensuring zero constraint violations using a model-free algorithm for tabular MDPs with linear rewards and constraints. However, for infinite horizon setups, the analysis from finite horizon algorithms does not directly holds. This is because finite horizon setups can update the policy after every episode. But this policy switch modifies the induced Markov chains which takes time to converge to stationary distribution.

(Xu et al., 2020) consider an infinite horizon discounted setup with constraints and obtain global convergence using policy gradient algorithms. (Ding et al., 2020) also considers an infinite horizon discounted setup. They use a natural policy gradient to update the primal variable and sub-gradient descent to update the dual variable. (Singh et al., 2020) considered the setup of infinite-horizon ergodic CMDPs with long-term average constraints with an optimism based algorithm. (Gattami et al., 2021) analyzed the asymptotic performance for Lagrangian based algorithms for infinite-horizon long-term average constraints, however they only show convergence guarantees without explicit convergence rates. (Chen et al., 2022) provide an optimistic online mirror descent algorithm for ergodic MDPs which obtain a regret bound of $O(T_M S\sqrt{SAT})$, and (Wei et al., 2022b) provide a model free SARSA algorithm which obtains a regret bound of $O(\sqrt{SA}T^{5/6})$ for discounted constrained MDPs. (Liu et al., 2021b) proposed an algorithm which achieves $\tilde{O}(1/\epsilon^3)$ sample complexity to obtain zero constraint violations in infinite horizon discounted setup.

| Algorithm(s) | Setup | Regret | Constraint Violation | Non-Linear |
|---|---|---|---|---|
| CONRL (Brantley et al., 2020) | FH | $\tilde{O}(LH^{5/2}S\sqrt{A/K})$ | $O(H^{5/2}S\sqrt{A/K})$ | Yes |
| MOMA (Yu et al., 2021) | FH | $\tilde{O}(LH^{3/2}\sqrt{SA/K})$ | $\tilde{O}(H^{3/2}\sqrt{SA/K})$ | Yes |
| TripleQ (Wei et al., 2022a) | FH | $\tilde{O}(\frac{1}{\delta}H^4\sqrt{SA}K^{-1/5})$ | 0 | No |
| OptPess-LP (Liu et al., 2021a) | FH | $\tilde{O}(\frac{H^3}{\delta}\sqrt{S^3A/K})$ | 0 | No |
| OptPess-Primal Dual (Liu et al., 2021a) | FH | $\tilde{O}(\frac{H^3}{\delta}\sqrt{S^3A/K})$ | $\tilde{O}(H^4S^2A/\delta)$ | No |
| UCRL-CMDP (Singh et al., 2020) | IH | $\tilde{O}(\sqrt{SA}T^{-1/3})$ | $\tilde{O}(\sqrt{SA}/T^{1/3})$ | No |
| Chen et al. (Chen et al., 2022) | IH | $\tilde{O}(\frac{1}{\delta}T_M S\sqrt{SA/T})$ | $\tilde{O}(\frac{1}{\delta^2}T_M^2S^3A)$ | No |
| Wei et al. (Wei et al., 2022b) | IH | $\tilde{O}(\frac{1}{\delta}\sqrt{SA}T^{-1/6})$ | 0 | No |
| UC-CURL (This work) | IH | $\tilde{O}(\frac{1}{\delta}LT_M S\sqrt{A/T})$ | 0 | **Yes** |

Table 1: Overview of work for constrained reinforcement learning setups. For finite horizon (FH) setups, $H$ is the episode length and $K$ is the number of episodes for which the algorithm runs. For infinite horizon (IH) setups, $T_M$ denotes the mixing time of the MDP, and $T$ is the time for which algorithm runs. $L$ is the Lipschitz constant.

**Concave Utility RL:** Another major research area related to this work is concave utility RL (Hazan et al., 2019). Along this direction, (Cheung, 2019) considered a concave function of expected per-step vector reward and developed an algorithm using Frank-Wolfe gradient of the concave function for tabular infinite horizon MDPs. (Agarwal & Aggarwal, 2019) also considered the same setup using a posterior sampling based algorithm. Recently, (Brantley et al., 2020) combined concave utility reinforcement learning and constrained reinforcement learning for an episodic setup. (Yu et al., 2021) also considered the case of episodic setup with concave utility RL. However, both (Brantley et al., 2020) and (Yu et al., 2021) consider the weaker regret definition by (Efroni et al., 2020), and (Cheung, 2019) and (Yu et al., 2021) do not target the convergence of the policy. Further, these works do not target zero-constraint violations.

Compared to prior works, we consider the constrained reinforcement learning with convex constraints and concave objective function. Using infinite-horizon setup, we consider the tightest possible regret definition. Further, we achieve zero constraint violations with objective regret tight in $T$ using an optimization problem with decaying tightness. A comparative survey of prior works and our work is also presented in Table 1.

## 3 Problem Formulation

We consider an ergodic tabular infinite-horizon constrained Markov Decision Process $\mathcal{M} = (\mathcal{S}, \mathcal{A}, r, f, c_1, \cdots, c_d, g, P, \phi)$. $\mathcal{S}$ is finite set of $S$ states, and $\mathcal{A}$ is a finite set of $A$ actions. $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ denotes the transition probability distribution such that on taking action $a \in \mathcal{A}$ in state $s \in \mathcal{S}$, the system moves to state $s' \in \mathcal{S}$ with probability $P(s'|s, a)$. $r : \mathcal{S} \times \mathcal{A} \to [0, 1]$ and $c_i : \mathcal{S} \times \mathcal{A} \to [0, 1], i \in 1, \cdots, d$

denotes the average reward obtained and average costs incurred in state action pair $(s, a) \in \mathcal{S} \times \mathcal{A}$. Lastly, $\phi$ is the distribution over the initial state.

The agent interacts with $\mathcal{M}$ in time-steps $t \in 1, 2, \cdots$ for a total of $T$ time-steps. We note that $T$ is possibly unknown and $s_1 \sim \phi$. At each time $t$, the agent observes state $s_t$, plays action $a_t$. The agent selects an action on observing the state $s$ using a policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$, where $\Delta(\mathcal{A})$ is the probability simplex on the action space. On following a policy $\pi$, the long-term average reward of the agent is denoted as:

$$\lambda_\pi^P = \lim_{\tau \to \infty} \mathbb{E}_{\pi,P} \left[ \sum\nolimits_{t=1}^{\tau} r(s_t, a_t)/\tau \right] \tag{1}$$

where $\mathbb{E}_{\pi,P}[\cdot]$ denotes the expectation over the state and action trajectory generated from following $\pi$ on transitions $P$. The long-term average reward can also be represented as:

$$\lambda_\pi^P = \sum\nolimits_{s,a} \rho_\pi^P(s, a) r(s, a) = \lim_{\gamma \to 1} (1 - \gamma) V_\gamma^{\pi,P}(s) \ \forall s \in \mathcal{S}$$

where $V_\gamma^{\pi,P}(s)$ is the discounted cumulative reward on following policy $\pi$, and $\rho_\pi^P \in \Delta(\mathcal{S} \times \mathcal{A})$ is the steady-state occupancy measure generated from following policy $\pi$ on MDP with transitions $P$ (Puterman, 2014). Similarly, we also define the long-term average costs as follows:

$$\begin{aligned} \zeta_\pi^P(i) &= \lim_{\tau \to \infty} \mathbb{E}_{\pi,P} \left[ \sum\nolimits_{t=1}^{\tau} c_i(s_t, a_t)/\tau \right] = \lim_{\gamma \to 1} (1 - \gamma) V_\gamma^{\pi,P}(s; i) \quad \forall s \in \mathcal{S} \\ &= \sum\nolimits_{s,a} \rho_\pi^P(s, a) c_i(s, a) \end{aligned} \tag{2}$$

The agent interacts with the CMDP $\mathcal{M}$ for $T$ time-steps in an online manner and aims to maximize a function $f : [0, 1] \to \mathbb{R}$ of the average per-step reward. Further, the agent attempts to ensure that a function of average per-step costs $g : [0, 1]^d \to \mathbb{R}$ is at most 0. In the hindsight, the agents wants to play a policy $\pi^*$ which which satisfies:

$$\max_\pi f\left(\lambda_\pi^P\right) \quad s.t. \quad g\left(\zeta_\pi^P(1), \cdots, \zeta_\pi^P(d)\right) \leq 0, \tag{3}$$

Let $P_{\pi,s}^t = \prod_{t'=1}^{t} P_\pi$ denote the $t$-step transition probability on following policy $\pi$ in MDP $\mathcal{M}$ starting from some state $s$ where $P_\pi(\cdot|s) = \sum_a \pi(a|s) P(\cdot|s, a)$. Also, let $T_{s \to s'}^\pi$ denote the time taken by the Markov chain induced by the policy $\pi$ to hit state $s'$ starting from state $s$. Then let $T_M = T_M := \max_\pi \mathbb{E}[T_{s \to s'}^\pi]$ be the mixing time of the MDP $\mathcal{M}$. We now introduce our assumptions on the MDP $\mathcal{M}$.

**Assumption 3.1.** The MDP $\mathcal{M}$ is ergodic, or $\|P_{\pi,s}^t - P_\pi\| \leq C\rho^t$ with $P_\pi$ being the long-term steady state distribution induced by policy $\pi$, and $C > 0$ and $\rho < 1$ are problem specific constants. And, the mixing time of the MDP $\mathcal{M}$ if finite or $T_M < \infty$.

**Assumption 3.2.** The rewards $r(s, a)$, the costs $c_i(s, a); \forall\ i$ and the functions $f$ and $g$ are known to the agent.

**Assumption 3.3.** The scalarization function $f$ is jointly concave and the constraints $g$ are jointly convex. Hence for any arbitrary distributions $\mathcal{D}_1$ and $\mathcal{D}_2$, the following holds.

$$f\left(\mathbb{E}_{x \sim \mathcal{D}_1}[x]\right) \geq \mathbb{E}_{x \sim \mathcal{D}_1}[f(x)] \tag{4}$$

$$g\left(\mathbb{E}_{\mathbf{x} \sim \mathcal{D}_2}[\mathbf{x}]\right) \leq \mathbb{E}_{\mathbf{x} \sim \mathcal{D}_2}[g(\mathbf{x})]; \ \mathbf{x} \in \mathbb{R}^d \tag{5}$$

**Assumption 3.4.** The function $f$ and $g$ are assumed to be a $L-$ Lipschitz function, or

$$|f(x) - f(y)| \leq L|x - y|; \ x, y \in \mathbb{R} \tag{6}$$

$$|g(\mathbf{x}) - g(\mathbf{y})| \leq L \|\mathbf{x} - \mathbf{y}\|_1; \ \mathbf{x}, \mathbf{y} \in \mathbb{R}^d \tag{7}$$

*Remark* 3.5. We consider a standard setup of concave and the Lipschitz function as considered by (Cheung, 2019; Brantley et al., 2020; Yu et al., 2021). Note that the analysis in this paper directly works for $f : \mathbb{R}^K \to \mathbb{R}$, where the function takes as input $K$ average per-step rewards for $K$ objectives.

*Remark* 3.6. For non-Lipshitz continuous functions such as entropy, we can obtain maximum entropy exploration if choose function $f = -\sum_k \lambda_k \log(\lambda_k + \eta)$ with $r_k(s,a) = \mathbf{1}_{\{s_k,a_k\}}$ for a particular state action pair $s_k, a_k$ and choosing $K = S \times A$ to cover all state-action pairs and a regularizer $\eta$ (Hazan et al., 2019).

**Assumption 3.7.** There exists a policy $\pi$, and one constant $\delta > LdST_M\sqrt{(A\log T)/T} + (CSA\log T)/(T(1-\rho))$ such that

$$g\left(\zeta_\pi^P(1), \cdots, \zeta_\pi^P(d)\right) \leq -\delta \tag{8}$$

This assumption is again a standard assumption in the constrained RL literature (Efroni et al., 2020; Ding et al., 2021; 2020; Wei et al., 2022a). $\delta$ is referred as Slater's constant. (Ding et al., 2021) assumes that the Slater's constant $\delta$ is known. (Wei et al., 2022a) assumes that the number of iterations of the algorithm is at least $\tilde{\Omega}(SAH/\delta)^5$ for episode length $H$. On the contrary, we simply assume the existence of $\delta$ and a lower bound on the value of $\delta$ which gets relaxed as the agent acquires more time to interact with the environment.

Any online algorithm starting with no prior knowledge will require to obtain estimates of transition probabilities $P$ and obtain reward $r$ and costs $c_k, \forall\ k \in \{1, \cdots, d\}$ for each state action pair. Initially, when algorithm does not have good estimate of the model, it accumulates a regret as well as violates constraints as it does not know the optimal policy. We define reward regret $R(T)$ as the difference between the average cumulative reward obtained vs the expected rewards from running the optimal policy $\pi^*$ for $T$ steps, or

$$R(T) = f\left(\lambda_{\pi^*}^P\right) - f\left(\sum\nolimits_{t=1}^T r(s_t, a_t)/T\right)$$

Additionally, we define constraint regret $C(T)$ as the gap between the constraint function and incurred and constraint bounds, or

$$C(T) = \left(g\left(\sum\nolimits_{t=1}^T c_1(s_t, a_t)/T, \cdots, \sum\nolimits_{t=1}^T c_d(s_t, a_t)/T\right)\right)_+, \text{ where}(x)_+ = \max(0, x).$$

In the following section, we present a model-based algorithm to obtain this policy $\pi^*$, and reward regret and the constraint regret accumulated by the algorithm.

## 4 Algorithm

We now present our algorithm UC-CURL and the key ideas used in designing the algorithm. Note that if the agent is aware of the true transition $P$, it can solve the following optimization problem for the optimal feasible policy.

$$\max_{\rho(s,a)} f\big(\sum\nolimits_{s,a} r(s,a)\rho(s,a)\big) \tag{9}$$

with the following set of constraints,

$$\sum\nolimits_{s,a} \rho(s,a) = 1, \quad \rho(s,a) \geq 0 \tag{10}$$

$$\sum\nolimits_{a\in\mathcal{A}} \rho(s',a) = \sum\nolimits_{s,a} P(s'|s,a)\rho(s,a) \tag{11}$$

$$g\big(\sum\nolimits_{s,a} c_1(s,a)\rho(s,a), \cdots, \sum\nolimits_{s,a} c_d(s,a)\rho(s,a)\big) \leq 0 \tag{12}$$

for all $s' \in \mathcal{S},\ \forall\ s \in \mathcal{S}$, and $\forall\ a \in \mathcal{A}$. Equation (11) denotes the constraint on the transition structure for the underlying Markov Process. Equation (10) ensures that the solution is a valid probability distribution. Finally, Equation (12) are the constraints for the constrained MDP setup which the policy must satisfy. Using the solution for $\rho$, we can obtain the optimal policy as:

$$\pi^*(a|s) = \frac{\rho(s,a)}{\sum_{b\in\mathcal{A}} \rho(s,b)} \forall\ s, a \tag{13}$$

However, the agent does not have the knowledge of $P$ to solve this optimization problem, and thus starts learning the transitions with an arbitrary policy. We first note that if the agent does not have complete knowledge of the transition $P$ of the true MDP $\mathcal{M}$, it should be conservative in its policy to allow room to violate constraints. Based on this idea, we formulate the $\epsilon$-tight optimization problem by modifying the constraint in Equation (12) as.

$$g\big(\sum\nolimits_{s,a} c_1(s,a)\rho_\epsilon(s,a), \cdots, \sum\nolimits_{s,a} c_d(s,a)\rho_\epsilon(s,a)\big) \leq -\epsilon \tag{14}$$

Let $\rho_\epsilon$ be the solution of the $\epsilon$-tight optimization problem, then the optimal conservative policy becomes:

$$\pi^*_\epsilon(a|s) = \frac{\rho_\epsilon(s,a)}{\sum_{b\in\mathcal{A}} \rho_\epsilon(s,b)} \forall\ s,a \tag{15}$$

We are now ready to design our algorithm UC-CURL which is based on the optimism principle (Jaksch et al., 2010). The UC-CURL algorithm is presented in Algorithm 1. The algorithm proceeds in epochs $e$. The algorithm maintains three key variables $\nu_e(s,a)$, $N_e(s,a)$, and $\hat{P}(s,a,s')$ for all $s,a$. $\nu_e(s,a)$ stores the number of times state-action pair $(s,a)$ are visited in epoch $e$. $N_e(s,a)$ stores the number of times $(s,a)$ are visited till the start of epoch $e$. $\hat{P}(s,a,s')$ stores the number of times the system transitions to state $s'$ after taking action $a$ in state $s$. Another key parameter of the algorithm is $\epsilon_e = K\sqrt{(\log t_e)/t_e}$ where $t_e$ is the start time of the epoch $e$ and $K$ is a configurable constant. Using these variables, the agent solves for the optimal $\epsilon_e$-conservative policy for the optimistic MDP by replacing the constraints in Equation (11) by:

$$\sum\nolimits_{a\in\mathcal{A}} \rho(s',a) \leq \sum\nolimits_{s,a} \tilde{P}_e(s'|s,a)\rho(s,a) \tag{16}$$

$$\tilde{P}_e(s'|s,a) > 0, \sum\nolimits_{s'} \tilde{P}_e(s'|s,a) = 1 \tag{17}$$

$$\|\tilde{P}_e(\cdot|s,a) - \frac{\hat{P}(s,a,\cdot)}{1 \vee N_e(s,a)}\|_1 \leq \sqrt{\frac{14S\log(2At)}{1 \vee N_e(s,a)}} \tag{18}$$

for all $s' \in \mathcal{S}, \forall s \in \mathcal{S}$, and $\forall a \in \mathcal{A}$ and $x \vee y = \max(x,y)$. Equation (18) ensures that the agent searches for optimistic policy in the confidence intervals of the transition probability estimates.

Combining the right hand side of (16) with (10) gives

$$\sum\nolimits_{s'} \sum\nolimits_{s,a} \tilde{P}_e(s'|s,a)\rho(s,a) = 1 = \sum\nolimits_{s',a} \rho(s',a)$$

Thus, joint with (16), we see that equality in (16) will be satisfied at the boundary as $\sum_a \rho(s',a)$ for some $s'$ can never exceed the boundary to compensate for another $s'$ and hence, for all $s'$, $\sum_a \rho(s',a)$ will lie on the boundary. In other words, the above constraints give $\sum_{a\in\mathcal{A}} \rho(s',a) = \sum_{s,a} \tilde{P}_e(s'|s,a)\rho(s,a)$. Further, we note that the region for the constraints is convex. This is because the set $\{x,y,z : xy \geq z\}$ is convex when $x,y,z \geq 0$. We note that even though the optimization problem may look non-convex due to constraints having product of two variables, we see Equations (9), (14), and (16)-(18) form a convex optimization problem. We expand more on this in Appendix B. We note that (Rosenberg & Mansour, 2019) provide another approach to obtain a convex optimization problem for optimistic MDP.

Let $\rho_e$ be the solution for $\epsilon_e$-tight optimization equation for the optimistic MDP. Then, we obtain the optimal conservative policy for epoch $e$ as:

$$\pi_e(a|s) = \frac{\rho_e(s,a)}{\sum_{b\in\mathcal{A}} \rho_e(s,b)} \forall\ s,a \tag{19}$$

The agent plays the optimistic conservative policy $\pi_e$ for epoch $e$. Note that the conservative parameter $\epsilon_e$ decays with time. As the agent interacts with the environment, the system model improves and the agent does not need to be as conservative as before. This allows us to bound both constraint violations and the objective regret. Further, if during the initial iterations of the algorithms a conservative solution is not feasible, we can ignore the constraints completely. We will show that the conservation behavior is required when $t = \Theta(T)$ to compensate for the violations in the initial period of the algorithm E.2.

**Algorithm 1** UC-CURL

**Parameters**: $K$
**Input**: $S$, $A$, $r$, $d$, $c_i$ $\forall$ $i \in [d]$

Let $t = 1$, $e = 1, \epsilon_e = K\sqrt{\frac{\ln t}{t}}$
**for** $(s, a) \in \mathcal{S} \times \mathcal{A}$ **do**
  $\nu_e(s, a) = 0, N_e(s, a) = 0, \widehat{P}(s', a, s) = 0 \forall\ s' \in \mathcal{S}$
**end for**
Solve for policy $\pi_e$ using Eq. (19)
**for** $t \in \{1, 2, \cdots\}$ **do**
  Observe $s_t$, and play $a_t \sim \pi_e(\cdot|s_t)$
  Observe $s_{t+1}$, $r(s_t, a_t)$ and $c_i(s_t, a_t)$ $\forall$ $i \in [d]$
  $\nu_e(s_t, a_t) = \nu_e(s_t, a_t) + 1$
  $\widehat{P}(s_t, a_t, s_{t+1}) = \widehat{P}(s_t, a_t, s_{t+1}) + 1$
  **if** $\nu_e(s, a) = \max\{1, N_e(s, a)\}$ for any $s, a$ **then**
    **for** $(s, a) \in \mathcal{S} \times \mathcal{A}$ **do**
      $N_{e+1}(s, a) = N_e(s, a) + \nu_e(s, a)$
      $e = e + 1$, $\nu_e(s, a) = 0$
    **end for**
    $\epsilon_e = K\sqrt{\frac{\ln t}{t}}$
    Solve for policy $\pi_e$ using Eq. (19)
  **end if**
**end for**

For the UC-CURL algorithm described in Algorithm 1, we choose $\{\epsilon_e\} = \{K\sqrt{(\log t_e)/t_e}\}$. However, if the agent has access to a lower bound $T_l$ (Assumption 3.7) on the time horizon $T$, the algorithm can change the $\epsilon_e = K\sqrt{(\ln(t_e \vee T_l))/(t_e \vee T_l)} \leq \delta$ in each epoch $e$ as follows. Note that if $T_l = 0$, $\epsilon_e$ becomes as specified in Algorithm 1 and if $T_l = T$, $\epsilon_e$ becomes constant for all epochs $e$.

## 5 Regret Analysis

After describing UC-CURL algorithm, we now perform the regret and constraint violation analysis. We note that the standard analysis for infinite horizon tabular MDPs of UCRL2 (Jaksch et al., 2010) cannot be directly applied as the policy $\pi_e$ is possibly stochastic for every epoch. Another peculiar aspect of the analysis of the infinite horizon MDPs is that the regret grows linearly with the number of epochs (or policy switches). This is because a new policy induces a new Markov chain and this chain take time to converge to the stationary distribution. The analysis still bounds the regret by $\tilde{O}(T_M S\sqrt{A/T})$ as the number of epochs are bounded by $O(SA\log T)$.

Before diving into the details, we first define few important variables which are key to our analysis. The first variable is the standard $Q$-value function. We define $Q_\gamma^{\pi,P}$ as the long term expected reward on taking action $a$ in state $s$ and then following policy $\pi$ for the MDP with transition $P$. Mathematically, we have

$$Q_\gamma^{\pi,P}(s,a) = r(s,a) + \gamma \sum\nolimits_{s' \in \mathcal{S}} P(s'|s,a) V_\gamma^{\pi,P}(s'); V_\gamma^{\pi,P}(s) = \mathbb{E}_{a\sim\pi}\left[Q_\gamma^{\pi,P}(s,a)\right].$$

We also define Bellman error $B^{\pi,\tilde{P}}(s,a)$ for the infinite horizon MDPs as the difference between the cumulative expected rewards obtained for deviating from the system model with transition $\tilde{P}$ for one step by taking action $a$ in state $s$ and then following policy $\pi$. We have:

$$B^{\pi,\tilde{P}}(s,a) = \lim_{\gamma\to 1}\left(Q_\gamma^{\pi,\tilde{P}}(s,a) - r(s,a) - \gamma \sum\nolimits_{s' \in \mathcal{S}} P(s'|s,a) V_\gamma^{\pi,\tilde{P}}(s,a)\right) \tag{20}$$

After defining the key variables, we can now jump into bounding the objective regret $R(T)$. Intuitively, the algorithm incurs regret on three accounts. First source is following the conservative policy which we require

to limit the constraint violations. Second source of regret is solving for the policy which is optimal for the optimistic MDP. Third source of regret is the stochastic behavior of the system. We also note that the constraints are violated because of the imperfect MDP knowledge and the stochastic behavior. However, the conservative behavior actually allows us to violate the constraints within some limits which we will discuss in the later part of this section.

We start by stating our first lemma which relates the regret because we solve for a conservative policy. We define $\epsilon_e$-tight optimization problem as optimization problem for the true MDP with transitions $P$ with $\epsilon = \epsilon_e$. We bound the gap between the value of function $f$ at the long-term expected reward of the policy for $\epsilon_e$-tight optimization problem and the true optimization problem (Equation (9)-(12)) in the following lemma.

**Lemma 5.1.** *Let $\lambda^P_{\pi^*}$ be the long-term average reward following the optimal feasible policy $\pi^*$ for the true MDP $\mathcal{M}$ and let $\lambda^P_{\pi_e}$ be the long-term average rewards following the optimal policy $\pi_e$ for the $\epsilon_e$ tight optimization problem for the true MDP $\mathcal{M}$, then for $\epsilon_e \leq \delta$, we have,*

$$f\left(\lambda^P_{\pi^*}\right) - f\left(\lambda^P_{\pi_e}\right) \leq 2L\epsilon_e/\delta \tag{21}$$

*Proof Sketch.* We construct a policy for which the steady state distribution is the weighted average of two steady state distributions. First distribution is for the optimal policy for the true optimization problem. Second distribution is for the policy which satisfies Assumption 3.7. Now, we show that this constructed policy satisfies the $\epsilon_e$-tight constraints. Now, using Lipschitz continuity, we convert the difference between function values into the difference between the long-term average rewards to obtain the required result. The detailed proof is provided in Appendix C. □

Lemma 5.1 and our construction of $\epsilon_e$ sequence allows us to limit the growth of regret because of conservative policy by $\tilde{O}(LdT_M S\sqrt{A/T})$.

To bound the regret from the second source, we use a Bellman error based analysis. In our next lemma, we show that the difference between the performance of a policy on two different MDPs is bounded by long-term averaged Bellman error. Formally, we have:

**Lemma 5.2.** *The difference of long-term average rewards for running the optimistic policy $\pi_e$ on the optimistic MDP, $\lambda^{\tilde{P}_e}_{\pi_e}$, and the average long-term average rewards for running the optimistic policy $\pi_e$ on the true MDP, $\lambda^P_{\pi_e}$, is the long-term average Bellman error as*

$$\lambda^{\tilde{P}_e}_{\pi_e} - \lambda^P_{\pi_e} = \sum\nolimits_{s,a} \rho^P_{\pi_e} B^{\pi_e,\tilde{P}_e}(s,a) \tag{22}$$

*Proof Sketch.* We start by writing $Q^{\pi_e,\tilde{P}_e}_\gamma$ in terms of the Bellman error. Now, subtracting $V^{\pi_e,P}_\gamma$ from $V^{\pi_e,\tilde{P}_e}_\gamma$ and using the fact that $\lambda^P_{\pi_e} = \lim_{\gamma\to 1} V^{\pi,P}_\gamma$ and $\lambda^{\tilde{P}_e}_{\pi_e} = \lim_{\gamma\to 1} V^{\pi,\tilde{P}_e}_\gamma$ we obtain the required result. A complete proof is provided in Appendix D.3. □

After relating the gap between the long-term average rewards of policy $\pi_e$ on the two MDPs, we now want to bound the sum of Bellman error over an epoch. For this, we first bound the Bellman error for a particular state action pair $s, a$ in the form of following lemma. We have,

**Lemma 5.3.** *With probability at least $1 - 1/t_e^6$, the Bellman error $B^{\pi_e,\tilde{P}_e}(s,a)$ for state-action pair $s, a$ in epoch $e$ is upper bounded as*

$$B^{\pi_e,\tilde{P}_e}(s,a) \leq \sqrt{\frac{14S\log(2AT)}{1 \vee N_e(s,a)}}\|\tilde{h}\|_\infty \tag{23}$$

*where $N_e(s,a)$ is the number of visitations to $s, a$ till epoch $e$ and $\tilde{h}$ is the bias of the MDP with transition probability $\tilde{P}_e$.*

*Proof Sketch.* We start by noting that the Bellman error essentially bounds the impact of the difference in value obtained because of the difference in transition probability to the immediate next state. We bound the difference in transition probability between the optimistic MDP and the true MDP using the result from

(Weissman et al., 2003). This approach gives the required result. A complete proof is provided in Appendix D.3. □

We use Lemma 5.2 and Lemma 5.3 to bound the regret because of the imperfect knowledge of the system model. We bound the expected Bellman error in epoch $e$ starting from state $s_{t_e}$ and action $a_{t_e}$ by constructing a Martingale sequence with filtration $\mathcal{F}_t = \{s_1, a_1, \cdots, s_{t-1}, a_{t-1}\}$ and using Azuma's inequality (Bercu et al., 2015). Using the Azuma's inequality, we can also bound the deviations because of the stochasticity of the Markov Decision Process. The result is stated in the following lemma with proof in Appendix D.

**Lemma 5.4.** *With probability at least $1 - T^{-5/4}$, the regret incurred from imperfect model knowledge and process stochastics is bounded by*

$$O(T_M S\sqrt{A(\log AT)/T} + (CT_M S^2 A \log T)/(1-\rho)) \tag{24}$$

The regret analysis framework also prepares us to bound the constraint violations as well. We again start by quantifying the reasons for constraint violations. The agent violates the constraint because **1.** it is playing with the imperfect knowledge of the MDP and **2.** the stochasticity of the MDP which results in the deviation from the average costs. We note that the conservative policy $\pi_e$ for every epoch does not violate the constraints, but instead allows the agent to manage the constraint violations because of the imperfect model knowledge and the system dynamics.

We note that the Lipschitz continuity of the constraint function $g$ allows us to convert the function of $d$ averaged costs to the sum of $d$ averaged costs. Further, we note that we can treat the cost similar to rewards (Brantley et al., 2020). This property allows us to bound the cost incurred incurred in a way similar to how we bound the gap from the optimal reward by $LdT_M S\sqrt{A(\log AT)/T}$. We now want that the slackness provided by the conservative policy should allow $LdT_M S\sqrt{A(\log AT)/T}$ constraint violations. This is ensured by our chosen $\epsilon_e$ sequence. We formally state that result in the following lemma proven in parts in Appendix D and Appendix E.

**Lemma 5.5.** *The cumulative sum of the $\epsilon_e$ sequence is upper and lower bounded as,*

$$\sum\nolimits_{e=1}^{E} (t_{e+1} - t_e)\epsilon_e = \Theta\left(K\sqrt{T\log T}\right) \tag{25}$$

After giving the details on bounds on the possible sources of regret and constraint violations, we can formally state the result in the form of following theorem.

**Theorem 5.6.** *For all $T$ and $K = \Theta(LdT_M S\sqrt{A} + CSA/(1-\rho))$, the regret $R(T)$ of UC-CURL algorithm is bounded by*

$$R(T) = O\left(\frac{1}{\delta} LdT_M S\sqrt{A\frac{\log AT}{T}} + \frac{CT_M S^2 A \log T}{(1-\rho)T}\right) \tag{26}$$

*and the constraints are bounded as $C(T) = 0$, with probability at least $1 - \frac{1}{T^{5/4}}$.*

### 5.1 Posterior Sampling Algorithm

We can also modify the analysis to obtain Bayesian regret for a posterior sampling version of the UC-CURL algorithm using Lemma 1 of (Osband et al., 2013). In the posterior sampling algorithm, instead of finding the optimistic MDP, we sample the transition probability $\tilde{P}_e$ using an updated posterior. This sampling allows to reduce the complexity of the optimization problem by eliminating Eq. (17) and Eq. (18). The complete algorithm is described in Appendix G. We note that optimization problem for the UC-CURL algorithm is feasible because the true MDP lies in the confidence interval. However, for the sampled MDP obtaining the feasibility requires a stronger Slater's condition.

### 5.2 Further Modifications

The proposed algorithm, and the analysis can be easily extended to $M$ convex constraints $g_1, \cdots, g_M$ by applying union bounds. Further, our analysis uses Proposition 1 of (Jaksch et al., 2010) to bound the epochs

by $O(SA\log_2 T)$. However, we can improve the empirical performance of the UC-CURL algorithm by modifying the epoch trigger condition (Line 11 of Algorithm 1). Triggering a new episode whenever $\nu_e(s,a)$ becomes $\max\{1, \nu_{e-1}(s,a)+1\}$ for any state-action pair results in a linearly increasing episode length with total epochs bounded by $O(SA+\sqrt{SAT})$. This modification results in a better empirical performance (See Appendix 6 for simulations) at the cost of a higher theoretical regret bound and computation complexity for obtaining a new policy at every epoch.

## 6 Simulation Results

To validate the performance of the UC-CURL algorithm and the PS-CURL algorithm, we run the simulation on the flow and service control in a single-serve queue, which is introduced in (Altman & Schwartz, 1991). Along with validating the performance of the proposed algorithms, we also compare the algorithms against the algorithms proposed in (Singh et al., 2020) and in (Chen et al., 2022) for model-based constrained reinforcement learning for infinite horizon MDPs. Compared to these algorithms, we note that our algorithm is also designed to handle concave objectives of expected rewards with convex constraints on costs with 0 constraint violations.

In the queue environment, a discrete-time single-server queue with a buffer of finite size $L$ is considered in this case. The number of customers waiting in the queue is considered as the state in this problem and thus $|S| = L+1$. Two kinds of the actions, service and flow, are considered in the problem and control the number of customers together. The action space for service is a finite subset $A$ in $[a_{min}, a_{max}]$, where $0 < a_{min} \leq a_{max} < 1$. Given a specific service action $a$, the service a customer is successfully finished with the probability $b$. If the service is successful, the length of the queue will reduce by 1. Similarly, the space for flow is also a finite subsection $B$ in $[b_{min}, b_{max}]$. In contrast to the service action, flow action will increase the queue by 1 with probability $b$ if the specific flow action $b$ is given. Also, we assume that there is no customer arriving when the queue is full. The overall action space is the Cartesian product of the $A$ and $B$. According to the service and flow probability, the transition probability can be computed and is given in the Table 2.

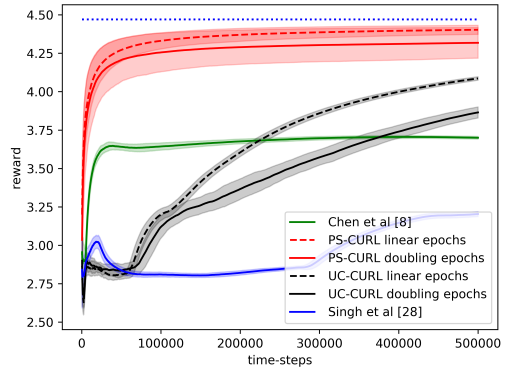
Table 2: Transition probability of the queue system

| Current State | $P(x_{t+1} = x_t - 1)$ | $P(x_{t+1} = x_t)$ | $P(x_{t+1} = x_t + 1)$ |
|---|---|---|---|
| $1 \leq x_t \leq L-1$ | $a(1-b)$ | $ab + (1-a)(1-b)$ | $(1-a)b$ |
| $x_t = L$ | $a$ | $1-a$ | $0$ |
| $x_t = 0$ | $0$ | $1-b(1-a)$ | $b(1-a)$ |

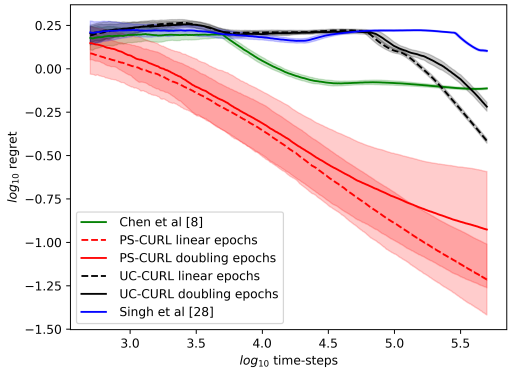
Define the reward function as $r(s,a,b)$ and the constraints for service and flow as $c^1(s,a,b)$ and $c^2(s,a,b)$, respectively. Define the stationary policy for service and flow as $\pi_a$ and $\pi_b$, respectively. Then, the problem can be defined as

$$\begin{aligned} \max_{\pi_a,\pi_b} \quad & \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} r(s_t, \pi_a(s_t), \pi_b(s_t)) \\ s.t. \quad & \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} c^1(s_t, \pi_a(s_t), \pi_b(s_t)) \geq 0 \\ & \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} c^2(s_t, \pi_a(s_t), \pi_b(s_t)) \geq 0 \end{aligned} \tag{27}$$

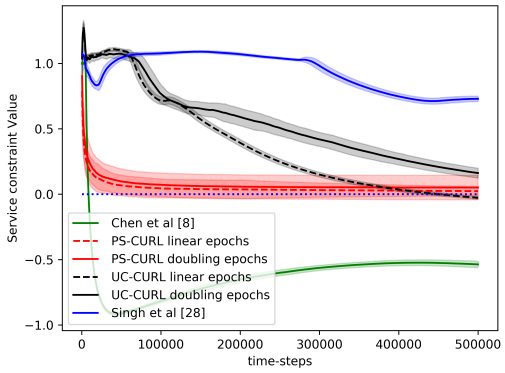
According to the discussion in (Altman & Schwartz, 1991), we define the reward function as $r(s,a,b) = 5-s$, which is an decreasing function only dependent on the state. It is reasonable to give higher reward when the number of customers waiting in the queue is small. For the constraint function, we define $c^1(s,a,b) = -10a+6$ and $c^2 = -8(1-b)^2+2$, which are dependent only on service and flow action, respectively. Higher constraint value is given if the probability for the service and flow are low and high, respectively.



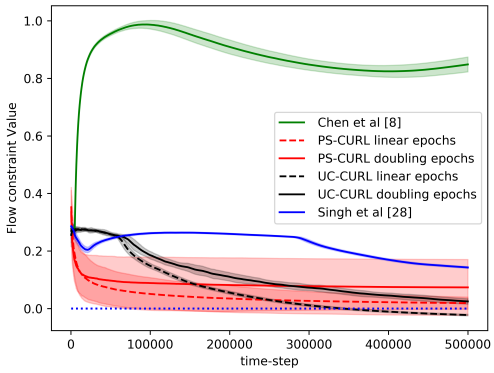

(a) Reward growth *w.r.t.* time




(b) Regret *w.r.t.* time




(c) Service constraints *w.r.t.* time




(d) Flow constraints *w.r.t.* time

Figure 1: Performance of the proposed UC-CURL and PS-CURL algorithms on a flow and service control problem for a single queue with doubling epoch lengths and linearly increasing epoch lengths. The algorithms are compared against Chen et al. (2022) and Singh et al. (2020)

.

In the simulation, the length of the buffer is set as $L = 5$. The service action space is set as $[0.2, 0.4, 0.6, 0.8]$ and the flow action space is set as $[0.4, 0.5, 0.6, 0.7]$. We use the length of horizon $T = 5 \times 10^5$ and run 50 independent simulations of all algorithms. The experiments were run on a 36 core Intel-i9 CPU @3.00 GHz with 64 GB of RAM. The result is shown in the Figure 1. The average values of the cumulative reward and the constraint functions are shown in the solid lines. Also, we plot the standard deviation around the mean value in the shadow to show the random error. In order to compare this result to the optimal, we assume that the full information of the transition dynamics is known and then use Linear Programming to solve the problem. The optimal cumulative reward for the constrained optimization is calculated to be 4.48 with both flow constraint and service contraint values to be 0. Also, the optimal cumulative reward for the unconstrained optimization is 4.8 with service constraint being $-2$ and flow constraint being $-0.88$.

We now discuss the performance of all the algorithms starting with our algorithms UC-CURL and PS-CURL. In Figure 1, we observe that the proposed UC-CURL algorithm in Algorithm 1 does not perform well initially. We observed that this is because the confidence interval radius $\sqrt{S \log(At)/N(s,a)}$ for any $s, a$ are not tight enough in the initial period. After the algorithms collects sufficient samples to construct tight confidence intervals around the transition probabilities, the algorithm starts converging towards the optimal policy. We also note that the linear epoch modification of the algorithm works better than the doubling epoch algorithm presented in Algorithm 1. This is because the linear epoch variant updates the policy quickly whereas the doubling epoch algorithm works with the same policy for too long and thus looses the advantages of collected samples. For our implementation, we choose the value of parameter $K$ in Algorithm 1 as $K = 1$, using which we observe that the constraint values start converging towards zero.

We now analyse the performance of the PS-CURL algorithm. For our implementation of the PS-CURL algorithm, we sample the transition probabilities using Dirichlet distribution. Note that the true transition probabilities were not sampled from a Dirichlet distribution and hence this experiment also shows the robustness against misspecified priors. We observe that the algorithm quickly brings the reward close to the optimal rewards. The performance of the PS-CURL algorithm is significantly better than the UC-CURL algorithm. We suspect this is because the UC-CURL algorithm wastes a large-number of steps to find optimistic policy with a large confidence interval. This observation aligns with the TDSE algorithm (Ouyang et al., 2017), where they show that the Thompson sampling algorithm with $O(\sqrt{SAT})$ epochs performs empirically better than the optimism based UCRL2 algorithm (Jaksch et al., 2010) with $O(\sqrt{SA\log T})$ epochs. (Osband et al., 2013) also made a similar observation where their PSRL algorithm worked better than the UCRL2 algorithm. Again, we set the value of parameter $K$ as 1 and with $K = 1$, the algorithm does not violate constraints. We also observe that the standard deviation of the rewards and constraints are higher for the PS-CURL algorithm as compared to the UC-CURL algorithm as the PS-CURL algorithm has an additional stochastic component which arises from sampling the transition probabilities.

After analysing the algorithms presented in this paper, we now analyse the performance of the algorithm by Chen et al. (2022). They provide an optimistic online mirror descent algorithm which also works with conservative parameter to tightly bound constraint violations. Their algorithm also obtains a $O(\sqrt{T}$ regret bound. However, their algorithm is designed for a linear reward/constraint setup with a single constraint, and empirically the algorithm is difficult to tune as it requires additional knowledge of $T_M$, $\rho$, $\delta$, and $T$ to fine tune parameters used in their algorithm. We set the value of the learning rate $\theta$ for online mirror descent as $5 * 10^{-2}$ with an episode length of $5 \times 10^3$. Further, we scale the rewards and costs to ensure that they lie between 0 and 1. We analyze the behavior of the optimistic online mirror descent algorithm in Figure 1b. We observe that the algorithm has three phases. The first phase is the first episodes where the algorithm uses a uniform policy which is the initial flat area till first 5000 steps. In the second phase, the algorithm updates the policy for the first time and starts converging to the optimal policy with a convergence rate which matches to that of the PS-CURL algorithm. However, after few policy updates, we observe that the algorithm has oscillatory behavior which is because the dual variable updates require online constraint violations.

Finally, we analyze the the algorithm by Singh et al. (2020). They also provide an algorithm which proceeds in epochs and solves an optimization problem at every epoch. The algorithm considers a fixed epoch length $T^{1/3}$. Further, the algorithm considers a confidence interval on each estimate of $P(s'|s,a)$ for all $s, a, s'$ triplet. The algorithm does not perform well even though it updates the policy most frequently because of creating confidence intervals on individual transition probabilities $P(s'|s,a)$ instead of the probability vector $P(s'|s,a)$.

From the experimental observations, we note that the proposed UC-CURL algorithm is suitable in cases where the parameter tuning is not possible and the system requires tighter bounds on deviation of the performance of the algorithm. The PS-CURL algorithm can be used in cases where the variance in algorithm's performance can be tolerated or computational complexity is a constraint. Further, for both the algorithms, it is beneficial to use the linear increasing epoch lengths. Additionally, the algorithm by (Chen et al., 2022) is suitable for cases where solving an optimization equation is not feasible, for example an embedded system, as the algorithm updates policy using exponential function which can be easily computed. However, This algorithm is only applicable in applications with linear reward/constraint and single constraint.

## 7 Conclusion

We considered the problem of Markov Decision Process with concave objective and convex constraints. For this problem, we proposed UC-CURL algorithm which works on the principle of optimism. To bound the constraint violations, we solve for a conservative policy using an optimistic model for an $\epsilon$-tight optimization problem. Using an analysis based on Bellman error for infinite-horizon MDPs, we show the UC-CURL algorithm achieves 0 constraint violations with a regret bound of $\tilde{O}(LdT_M S\sqrt{A/T} + (CSA\log T)/(T(1-\rho))$. Further, to reduce the computation complexity of finding optimistic MDP, we also propose a posterior sampling algorithm which finds the optimal policy for a sampled MDP. We provide a Bayesian regret bound of $\tilde{O}(LdT_M S\sqrt{A/T} + (CT_M S^2 A\log T)/(T(1-\rho))$ for the posterior sampling algorithm by considering a stronger Slater's condition to solve for constrained optimization for sampled MDPs as well. As part of

potential future works, we consider dynamically configuring $K$ to be an interesting and important direction to reduce the requirement of problem parameters.

## References


Mridul Agarwal and Vaneet Aggarwal. Reinforcement learning for joint optimization of multiple rewards. *arXiv preprint arXiv:1909.02940*, 2019.

E. Altman and A. Schwartz. Adaptive control of constrained markov chains. *IEEE Transactions on Automatic Control*, 36(4):454–462, 1991. doi: 10.1109/9.75103.

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

Bernard Bercu, Bernard Delyon, and Emmanuel Rio. *Concentration inequalities for sums and martingales*. Springer, 2015.

Kianté Brantley, Miroslav Dudik, Thodoris Lykouris, Sobhan Miryoosefi, Max Simchowitz, Aleksandrs Slivkins, and Wen Sun. Constrained episodic reinforcement learning in concave-convex and knapsack settings. *arXiv preprint arXiv:2006.05051*, 2020.

Sébastien Bubeck et al. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

Jingdi Chen, Yimeng Wang, and Tian Lan. Bringing fairness to actor-critic reinforcement learning for network utility optimization. In *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*, pp. 1–10. IEEE, 2021.

Liyu Chen, Rahul Jain, and Haipeng Luo. Learning infinite-horizon average-reward markov decision processes with constraints. *arXiv preprint arXiv:2202.00150*, 2022.

Wang Chi Cheung. Regret minimization for reinforcement learning with vectorial feedback and complex objectives. *Advances in Neural Information Processing Systems*, 32:726–736, 2019.

Dongsheng Ding, Kaiqing Zhang, Tamer Basar, and Mihailo Jovanovic. Natural policy gradient primal-dual method for constrained markov decision processes. *Advances in Neural Information Processing Systems*, 33, 2020.

Dongsheng Ding, Xiaohan Wei, Zhuoran Yang, Zhaoran Wang, and Mihailo Jovanovic. Provably efficient safe exploration via primal-dual policy optimization. In *International Conference on Artificial Intelligence and Statistics*, pp. 3304–3312. PMLR, 2021.

Yonathan Efroni, Shie Mannor, and Matteo Pirotta. Exploration-exploitation in constrained mdps. *arXiv preprint arXiv:2003.02189*, 2020.

Ronan Fruit, Matteo Pirotta, Alessandro Lazaric, and Ronald Ortner. Efficient bias-span-constrained exploration-exploitation in reinforcement learning. In *International Conference on Machine Learning*, pp. 1578–1586. PMLR, 2018.

Ather Gattami, Qinbo Bai, and Vaneet Aggarwal. Reinforcement learning for constrained markov decision processes. In *International Conference on Artificial Intelligence and Statistics*, pp. 2656–2664. PMLR, 2021.

Seyed Kamyar Seyed Ghasemipour, Richard Zemel, and Shixiang Gu. A divergence minimization perspective on imitation learning methods. In *Conference on Robot Learning*, pp. 1259–1277. PMLR, 2020.

Elad Hazan, Sham Kakade, Karan Singh, and Abby Van Soest. Provably efficient maximum entropy exploration. In *International Conference on Machine Learning*, pp. 2681–2691. PMLR, 2019.

Thomas Jaksch, Ronald Ortner, and Peter Auer. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11(Apr):1563–1600, 2010.

Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M Kakade, and Michael I Jordan. How to escape saddle points efficiently. In *International Conference on Machine Learning*, pp. 1724–1732. PMLR, 2017.

Krishna C Kalagarla, Rahul Jain, and Pierluigi Nuzzo. A sample-efficient algorithm for episodic finite-horizon mdp with constraints. *arXiv preprint arXiv:2009.11348*, 2020.

Raymond Kwan, Cyril Leung, and Jie Zhang. Proportional fair multiuser scheduling in lte. *IEEE Signal Processing Letters*, 16(6):461–464, 2009.

J Langford and S Kakade. Approximately optimal approximate reinforcement learning. In *Proceedings of ICML*, 2002.

Hoang Le, Cameron Voloshin, and Yisong Yue. Batch policy learning under constraints. In *International Conference on Machine Learning*, pp. 3703–3712. PMLR, 2019.

Tao Liu, Ruida Zhou, Dileep Kalathil, Panganamala Kumar, and Chao Tian. Learning policies with zero or bounded constraint violation for constrained mdps. *Advances in Neural Information Processing Systems*, 34:17183–17193, 2021a.

Tao Liu, Ruida Zhou, Dileep Kalathil, PR Kumar, and Chao Tian. Fast global convergence of policy optimization for constrained mdps. *arXiv preprint arXiv:2111.00552*, 2021b.

Ian Osband, Daniel Russo, and Benjamin Van Roy. (more) efficient reinforcement learning via posterior sampling. In *Advances in Neural Information Processing Systems*, pp. 3003–3011, 2013.

Yi Ouyang, Mukul Gagrani, Ashutosh Nayyar, and Rahul Jain. Learning unknown markov decision processes: A thompson sampling approach. *Advances in neural information processing systems*, 30, 2017.

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Diederik M Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research*, 48:67–113, 2013.

Aviv Rosenberg and Yishay Mansour. Online convex optimization in adversarial markov decision processes. In *International Conference on Machine Learning*, pp. 5478–5486. PMLR, 2019.

Rahul Singh, Abhishek Gupta, and Ness B Shroff. Learning in markov decision processes under constraints. *arXiv preprint arXiv:2002.12435*, 2020.

Chen Tessler, Daniel J Mankowitz, and Shie Mannor. Reward constrained policy optimization. In *International Conference on Learning Representations*, 2018.

Honghao Wei, Xin Liu, and Lei Ying. Triple-q: A model-free algorithm for constrained reinforcement learning with sublinear regret and zero constraint violation. In *International Conference on Artificial Intelligence and Statistics*, pp. 3274–3307. PMLR, 2022a.

Honghao Wei, Xin Liu, and Lei Ying. A provably-efficient model-free algorithm for infinite-horizon average-reward constrained markov decision processes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022b.

Tsachy Weissman, Erik Ordentlich, Gadiel Seroussi, Sergio Verdu, and Marcelo J Weinberger. Inequalities for the l1 deviation of the empirical distribution. *Hewlett-Packard Labs, Tech. Rep*, 2003.

Tengyu Xu, Yingbin Liang, and Guanghui Lan. A primal approach to constrained policy optimization: Global optimality and finite-time analysis. *arXiv preprint arXiv:2011.05869*, 2020.

Tiancheng Yu, Yi Tian, Jingzhao Zhang, and Suvrit Sra. Provably efficient algorithms for multi-objective competitive rl. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12167–12176. PMLR, 18–24 Jul 2021.

Junyu Zhang, Alec Koppel, Amrit Singh Bedi, Csaba Szepesvari, and Mengdi Wang. Variational policy gradient method for reinforcement learning with general utilities. *Advances in Neural Information Processing Systems*, 33:4572–4583, 2020.

Liyuan Zheng and Lillian Ratliff. Constrained upper confidence reinforcement learning. In *Learning for Dynamics and Control*, pp. 620–629. PMLR, 2020.