Proceedings of Machine Learning Research vol vvv:1–22, 2025

# Iteration Complexity for Robust CMDP

**Sourav Ganguly** SG2786@NJIT.EDU
*Department of Electrical and Computer Engineering, NJIT*
**Arnob Ghosh** ARNOB.GHOSH@NJIT.EDU
*Department of Electrical and Computer Engineering, NJIT*

## Abstract

We consider the robust Constrained Markov decision (RCMDP) problem of learning a policy that will maximize the cumulative reward while satisfying a constraint against the worst possible stochastic model under the *unknown* uncertainty set. Such a problem is relevant when the simulated and the real environment differ. Such a problem poses significant additional challenges compared to the non-robust CMDP problem and the unconstrained robust MDP problem. We seek to characterize the number of iterations required to bound both the sub-optimality gap and the violations by at most $\epsilon$. We observe that the primal-dual-based approaches that achieves iteration complexity bounds for non-robust CMDP cannot achieve the same in the robust CMDP case. We consider a modified problem where we consider the convex hull of the policy-spaces and the decision becomes the simplex over the policy space. We propose a primal-dual based approach and show that $\epsilon$ suboptimality gap and violation bound can be achieved after $O(1/\epsilon^2)$ iterations. We also show that an extra-gradient based approach can achieve $\epsilon$ suboptimality gap and violation bound can be achieved after $O(1/\epsilon)$ iterations. This is the first such result for iteration complexity for the robust CMDP. Empirical evaluations show that our proposed approach can achieve feasible and yet optimal policies.

**Keywords:** CMDP, Primal-Dual, Optimistic Online Mirror Descent, Min-max game

## 1. Introduction

In many practical applications of online reinforcement learning (RL) (e.g., safety, resource constraints), there exist additional constraints on the learned policy in the sense that it also needs to ensure that the expected total utility (cost, resp.) exceeds a given threshold (is below a threshold, resp.). Such problems are formulated as constrained Markov Decision Processes (CMDPs) (Altman, 1999; Efroni et al., 2020).

$$\max_{\pi} V_r^{\pi,P}(x) \quad \text{subject to } V_g^{\pi,P}(x) \geq b \tag{1}$$

where $V_r^{\pi,P}(x) = \mathbb{E}_{\pi,P}[\sum_{h=1}^{H} r(x_h, a_h)|x_1 = x]$ and $V_g^{\pi,P}(x) = \mathbb{E}_{\pi,P}[\sum_{h=1}^{H} g(x_h, a_h)|x_1 = x]$ are the expected cumulative reward and the expected discounted cumulative utility respectively following the policy $\pi$. Algorithms are proposed with provable performance guarantee to solve for CMDP using both simulator or with online interaction Vaswani et al. (2022); Ding et al. (2020, 2021); Wei et al. (2022).

However, a feasible policy for CMDP found by training using a simulator may violate the constraints when employed in the real environment because of the mismatch between the models. Such mismatch may exist because of the non-stationarity, sim-to-real gap, or even because of the adversarial attacks. Standard RL algorithms even fail to adapt to the model-mismatch in the unconstrained setting Sünderhauf et al. (2018); Tobin et al. (2017). *Thus, it is important to find a feasible (nearly) and yet close to the optimal policy even when there is a model mismatch using only simulator data.*

$$\max_{\pi} \min_{P\in\mathcal{P}} V_r^{\pi,P}(x) \quad \text{subject to } \min_{P\in\mathcal{P}} V_g^{\pi,P}(x) \geq b \tag{2}$$

where the objective is to maximize the worst case cumulative reward ($\min_P V_r^{\pi,P}(x)$) subject to the constraint that the worst case cumulative utility ($\min_P V_g^P(x)$) is above a certain threshold. We consider that the uncertainty set $\mathcal{P}$ (defined in (5)) of the transition probabilities $P$ is unknown and is centered around the nominal model $P^0$.

While there are works that obtain optimal policy with provable guarantee for unconstrained robust MDP, *it still remains open for robust CMDP setup*. Finding algorithms with provable convergence rate guarantee using CMDP is inherently more challenging compared to the unconstrained case as one not only needs to find a policy that maximizes the worst-case, one also needs to ensure that the policy satisfies the constraint even when there is a model mismatch. While primal-dual based and LP-based approaches are proposed for obtaining convergence rate guarantee for non-robust CMDP, those approaches rely on the convexity of the state-action occupancy measure. However, Wang et al. (2022) shows that for the robust CMDP state-action occupancy measures are not convex. Hence, it is unclear whether the strong duality exists and whether the primal-dual algorithm can achieve the convergence rate unlike the non-robust CMDP. Recently, Zhang et al. (2024) considered a variant where they consider a class of mixed policies $Conv(\Pi)$ over a set of policies $\pi_i \in \Pi$, and then consider $p_i$, the probability to choose the policy $\pi$. In other words, the decision variable becomes to select $p_i$, a policy $\pi_i$ is sampled with probability $p_i$ at the start of the episode. Then the robust value function for a given distribution $p$ is $V_j^{\pi}(p) = \sum_i p_i \min_P V^{\pi,P}$ for $j = r, g$. Hence, the robust CMDP problem in (2) becomes

$$\max_p \sum_i p_i \min_P V_r^{\pi_i,P} \quad \text{subject to} \quad \sum_i p_i \min_P V_g^{\pi_i,P} \geq b \tag{3}$$

Zhang et al. (2024) showed that the above has a strong duality if Slater's condition is satisfied as the decision space is now over $p$ rather than $\pi$. However, Zhang et al. (2024) relied on best-response oracle for a given dual-variable to show the convergence of their approach. However, there does not exist any approach to effectively solve for $\pi$ given a dual variable $\lambda$ for most of the relevant uncertainty sets. Furthermore, convergence rate guarantee has not been provided. Hence, we seek to answer the following open question

*Can we design an algorithm that achieves provable iteration convergence rate guarantee for robust CMDP problem in (3)?*

**Definition 1** *We seek to obtain categorical distribution $\hat{p}_i$ such that after $T$ iterations*

$$\begin{aligned} \text{Sub} - \text{Opt}(\hat{p}) &:= \sum_i p_i^* \min_P V_r^{\pi_i,P}(x) - \sum_i \hat{p}_i \min_P V_r^{\pi_i,P}(x) \leq \epsilon \\ \text{Violation}(\hat{p}) &:= \sum_i p_i \min_P V_g^{\pi_i,P}(x) \geq b - \epsilon \end{aligned} \tag{4}$$

where $p_i^*$ is the optimal policy for (3). Hence, we want to find the policy $\hat{p}$ such that it will be sub-optimal by at most $\epsilon$ amount and will violate by only $\epsilon$ amount. Note that compared to the unconstrained case, here, one also needs to bound violation.

Our main contributions are the followings:

1. We show that we can achieve $\text{Sub} - \text{Opt}(\hat{p}) \leq \epsilon$, and $\text{Violation}(\hat{p}) \leq \epsilon$ after $\mathcal{O}(1/\epsilon^2)$ iterations. This is the first such result for robust CMDP. The algorithm is based on OMD update in the primal direction and the dual variable is updated using OGD method.

2. We also propose an OOMD-based approach to update $p$ and the dual-variable $\lambda$ which after $O(1/\epsilon)$ number of iterations achieve $\text{Sub}-\text{Opt}(\hat{p}) \leq \epsilon$, and $\text{Violation}(\hat{p}) \leq \epsilon$. This is the first $O(1/\epsilon)$ iteration complexity result for robust CMDP. The key difference is that we maintain additional copies of the primal and the dual variables which are like intermediate update steps, using those values we update the primal and dual variables. Thus, they are capable of tracking the change in the primal and dual variable while update resulting in faster convergence.

3. We empirically evaluate our proposed algorithms on constrained environments. Our empirical results indeed show that our proposed approaches achieve feasible and close to the optimal solution for robust CMDP setup.

### 1.1. Other Related Literature

**CMDP**: Primal-dual based approaches with provable performance guarantee have been proposed Vaswani et al. (2022); Ghosh et al. (2022); Wei et al. (2022); Ding et al. (2020, 2021, 2023); Ghosh et al. (2024); Xu et al. (2021); Efroni et al. (2020); Bura et al. (2022); Liu et al. (2021) to study non-robust CMDP with $O(1/\epsilon^2)$ iteration and sample complexity guarantee. The approaches used the strong duality result and the dynamic programming method for the combined value function. As we mentioned, both the approaches are not possible to apply for the robust CMDP.
**Robust Unconstrained MDP**: Robust MDP has been studied with *known* uncertainty set Iyengar (2005); Nilim and Ghaoui (2003); Wang and Zou (2021) and with unknown uncertainty set Shi et al. (2024); Panaganti et al. (2022); Xu et al. (2023); Yang and Wang (2019); Wang and Zou (2022); Wang et al. (2023); Zhou et al. (2021). Among these only Wang et al. (2023) obtained iteration complexity for the robust policy optimization.
**Robust CMDP**: Unlike the non-robust CMDP, there are limited work on robust CMDP. Primal-dual based approach has been proposed in Russel et al. (2020); Mankowitz et al. (2020); Wang et al. (2022). However, Wang et al. (2022) showed that robust CMDP may not have strong duality guarantee even when a strictly feasible policy exists (aka Slater's condition) unlike the non-robust CMDP. Naturally, those approaches are unable to provide theoretical sample complexity bound.

## 2. Problem Formulation

**Constrained Markov Decision Problem**: A constrained Markov Decision Process (CMDP) is characterized by the tuple $\{S, A, R, G, P, H\}$ where $S$ is the state-space, $A$ is the action-space; $R = \{r(s,a)\}$ and $G = \{g(s,a)\}$ are respectively the collection of rewards and utility for state-action pair $(s,a)$. $P$ denotes the transition probability $P_{s,a}(s') = P(s'|s,a)$. Without loss of generality, we assume that $r$, and $g$ are deterministic, and $|r(x,a)| \leq 1$, and $|g(x,a)| \leq 1$. $H$ We denote $\pi_h(\cdot|x) \in \Delta(A)$ as the policy at step $h$. In a CMDP (Efroni et al., 2020; Ghosh et al., 2022) setup one seeks to solve the following optimization problem in (1).

**Example 1** *Consider the setup where the agent wants to maximize the reward while being at the safe state. In this case, the utility is $g(x) = 1$ if $x$ is safe and 0 otherwise. This problem can be cast as a CMDP.*

**Robust CMDP**: We often use a simulator to train our policy before implementing in the real-life. However, the simulator setup and the real-life environment are often different, hence, we need a robust policy so that the policy can perform reasonably well in the real-life setup. In particular, we seek to solve the robust CMDP problem described in (2). $\rho > 0$,

and is known. In (2), $\mathcal{P}$ denotes the set of all transition probabilities. In particular, different transition probability defines different set of randomness inherent in the true environment. *The objective of the robust CMDP formulation is that constraints are satisfied even if there are model mismatch the constraint is satisfied while maximizing the reward among the worst of all the transition probability models.* Such robustness guarantee is important for implementing RL algorithms in practice. Consider the example we described above, there, the solution in (2) ensures that the policy will still be safe even if there is a mismatch.

Note that our analysis and approach can be easily applicable to the setting where $\max_P V_g^{\pi,P} \leq b$ as well where $g$ denotes the cost instead of utility, and we are interested in the constraint such that the worst-case cost is below a certain threshold $b$. Further, we can easily extend our analysis for multiple constraints. *For notational simplicity, we interchangably denote* $V_j^{\pi}(x) = \min_{P\in\mathcal{P}} V_j^{\pi,P}(x)$ *for* $j = r, g$. Note that the worst case model $P$ indeed depends on the policy.

**Uncertainty Set on models**: We consider an uncertainty set $\mathcal{P}$ consisting of multiple transition probabilities. The popular choice is (Panaganti et al., 2022; Panaganti and Kalathil, 2022) a set of transition probability models within a ball centered around the nominal model $P^0_{s,a}$ $\forall (s,a) \in S \times A$. In particular, $\mathcal{P} = \bigotimes_{(s,a)\in S\times A} \mathcal{P}_{s,a}$ such that

$$\mathcal{P}_{s,a} = \{P \in \Delta(S) : D(P, P^0_{s,a}) \leq \rho\} \tag{5}$$

where $D$ is the distance metric between two probability measures, and $\rho$ is the radius of the uncertainty set. Note that some popular choices of $D$ are TV-distance, KL-divergence, and $\chi$-squared distribution. Interested readers can refer to Panaganti and Kalathil (2022); Panaganti et al. (2022); Yang et al. (2022) for the complete characterization.

Note that (5) corresponds to the $(s,a)$-rectangularity assumption. Without rectangularity assumption, even for unconstrained robust MDP, obtaining optimal policy is NP-hard problem Wiesemann et al. (2013). However, we only rely on the robust policy evaluation approach. Hence, if robust policy evaluation is computationally feasible, our algorithm and analysis can be extended.

**Why primal-dual methods do not work?**:

In the non-robust CMDP, one solve for the Lagrangian

$$\min_{\lambda\geq 0} \max_{\pi} V_r^{\pi,P}(x) + \lambda(V_g^{\pi,P}(x) - b) \tag{6}$$

It has been shown that strong duality holds if a strict feasible policy (a.k.a. Slater's condition) (Paternain et al., 2019) exists. Thus, one can solve in the dual-domain. For non-robust CMDP problems, Vaswani et al. (2022); Xu et al. (2021) demonstrate that primal-dual based approaches can achieve a feasible and an $\epsilon$-sub optimal policy with $\tilde{O}(1/\epsilon^2)$ iteration complexity. The key to obtain strong duality result in Paternain et al. (2019) is that the state-action occupancy measure $d^{\pi,P}(s,a)$ for a given transition probability measure $P$ is convex. In particular, consider $\pi_1$, and $\pi_2$ with the corresponding state-action occupancy measures $d^{\pi_1,P}$ and $d^{\pi_2,P}$, then there exists $\pi'$ such that $(1-\lambda)d^{\pi_1,P}(s,a) + \lambda d^{\pi_2,P}(s,a) = d^{\pi',P}(s,a)$ for a $\lambda \in (0,1)$. However, the robust state-action occupancy measure *may not be convex* as shown in Lemma 1 in Wang et al. (2022). Intuitively, the worst-case transition model depends on the policy, hence, the convexity is not assured. Thus, the robust CMDP may not admit strong duality even if there exists a strictly feasible policy. Hence, the traditional primal-dual based algorithms will be unable to obtain the sample complexity guarantees.

**Issue in applying robust value iteration:** In non-robust case, for a given $\lambda$, one can solve for the optimal $\pi$ using the standard Dynamic programming approach as it would

simply be an unconstrained problem with modified per-step reward $r + \lambda g$. Hence, the standard value-based approach can be applied for the composite value function. This is used to update the policy $\pi$ and the dual-variable. One might be tempted to apply robust value iteration to solve for $\pi$ for the Lagrangian $V_r^\pi(x) + \lambda(V_g^\pi(x) - b)$ similar to the non-robust CMDP. *However, this won't work as the worst case model for the reward value function and the worst-case model for the utility value function can be different Zhang et al. (2024).* In particular, the robust Bellman operator $L_{\mathcal{P}_{s,a}} V_r^\pi + L_{\mathcal{P}_{s,a}} V_g^\pi \neq L_{\mathcal{P}_{s,a}}[V_h^\pi + V_g^\pi]$ because of the *non-linearity* unlike the non-robust CMDP. Thus, one cannot apply the robust value iteration to the combined value function to solve for the lagrangian $V_r^\pi(x) + \lambda(V_g^\pi(x) - b)$ for the robust CMDP even for a given fixed $\lambda$.

### 2.1. Categorical Distribution

To solve the strong-duality issue, we consider a convex hull or mixed policies among some policy space $\Pi$. In particular, for a given finite policy space $\Pi$ with $I$ elements, we consider a mixture of policies $\pi(p)$ such that the policy $\pi_i$ is chosen with probability $p_i$. We thus consider a modified problem (3). Such convex hull relaxation is also common for non-robust CMDP problem as well (Le et al., 2019; Miryoosefi et al., 2019).

At the start of the episode $k$ $\pi_i$ is sampled with probability $p_i^k$. Then, for the episode $\pi_i$ is played. Note that the problem in (3) becomes a convex optimization problem and thus it has a strong duality if Slater's condition is satisfied. Hence, the strong duality holds. We can then solve using the Lagrangian as the strong duality holds Zhang et al. (2024) for (3).

$$\min_{\lambda} \max_{\pi} \sum_i p_i \min_P V_r^{\pi_i, P} + \lambda(\sum_i p_i \min_P V_g^{\pi_i, P} - b) \tag{7}$$

We seek to solve this min-max problem. Note that Zhang et al. (2024) considered an approach where for a given dual variable it relies on finding the best policy $\pi$. However, as we described and Zhang et al. (2024) pointed out, finding such a policy using robust dynamic programming is inherently challenging. We thus consider a novel approach to find $p$ based on the min-max game.

## 3. Algorithm and Results

The algorithm 1 is based on Online Mirror Descent (OMD) approach.

**Robust Policy evaluation Oracle**: Our algorithms rely on a robust policy evaluation oracle that evaluates $\min_P V_j^{\pi,P}$ for a given $\pi$. Note that for uncertainty set with $(s,a)$-rectangularity assumption (see (5)) and popular uncertainty measures there are efficient approaches to evaluate the robust policy Yang et al. (2022); Panaganti et al. (2022). Note that such oracles are standard even for the unconstrained case (Wang and Zou, 2022; Wang et al., 2022). One can certainly employ a neural-network to approximate the robust value function corresponding to a policy.

The rest of the algorithm pertains to updating $p_i$, and updating the dual variable. We update $p$ at iteration $t$ as

$$p_t = \arg\max_p \sum_i p_i(\min_P V_r^{\pi_i, P} + \lambda_t \min_P V_g^{\pi_i, P}) - D(p||p_t) \tag{8}$$

where D is the KL-divergence. Dual variable is updated with the online gradient descent on the Lagrangian (7).

**Algorithm 1** Robust Safe Reinforcement Learning Algorithm (RSLA)

**Input:** Robust Policy evaluator, Finite Policy Class $\Pi$.
**Initialization**: $p_0 = 1/|\Pi|$.
**for** $t = 0, \dots, T-1$ **do**
  Evaluate $\min_P V_{j,P}^{\pi_i}$ for all $\pi_i \in \Pi$, for $j = r, c$ using the robust policy evaluator oracle.
  Update $p_{t+1}(i) \propto p_t(i) \exp(\alpha(\min_P V_{r,P}^{\pi_i} + \lambda_t \min_P V_{g,P}^{\pi_i}))$ for all $i$
  Update $\lambda_{t+1} = \min\{\max\{\lambda_t + \eta(b - \sum_i p_i \min_P V_{g,P}^{\pi_i}), 0\}, \xi\}$
Output $p = Uniform(p_1, \dots, p_T)$

### 3.1. Result and Analysis

We state the main result of this section here

**Theorem 2** *Algorithm 1 returns the probability distribution $p$ such that*

$$\text{Sub} - \text{Opt}(p) \leq C_1 \sqrt{\frac{H^2 \xi^2 \log(|\Pi)}{T}}, \quad \text{Violation}(p) \leq C_2 \sqrt{\frac{H^2 (1+\xi)^2 \log(|\Pi|)}{\xi T}} \tag{9}$$

*where $C_1$, and $C_2$*

Both the sub-opt and the violation bounds are $O(1/\sqrt{T})$. This is the first such result for the robust CMDP. Note that if $T = O(1/\epsilon^2)$, then both the $\text{Sub} - \text{Opt}$ and the Violation bound becomes $\epsilon$. Hence, the iteration complexity is $O(1/\epsilon^2)$. Xu et al. (2021) achieved the simialr iteration complexity for the non-robust CMDP, however, they did not consider the mixed-policy space.

### 3.2. Analysis (Proof Outline)

First note that we choose uniformly over $\{p_1, \dots, p_T\}$. Hence,

$$\sum_i (p_i^* V_r^{\pi_i} - p_i V_r^{\pi_i}) = \frac{1}{T} \sum_t \sum_i (p_i^* V_r^{\pi_i} - p_{i,t} V_r^{\pi_i}), \quad (b - \sum_i p_i V_g^{\pi_i}) = \frac{1}{T} \sum_t (b - \sum_i p_{i,t} V_g^{\pi_i})$$

We thus seek to bound the above.

In order to b, we use the following result

**Lemma 3** *For any $\lambda \in [0, \xi]$,*

$$\begin{aligned} &\sum_t \sum_i (p_i^* V_r^{\pi_i} - p_{i,t} V_r^{\pi_i}) + \lambda \sum_t (b - \sum_i p_{i,t} V_g^{\pi_i}) \\ &\leq \underbrace{\sum_t \sum_i (p_i^* V_r^{\pi_i} - p_{i,t} V_r^{\pi_i}) + \sum_t \lambda_t (\sum_i p_i^* V_g^{\pi_i} - \sum_i p_{i,t} V_g^{\pi_i})}_{\mathcal{T}_1} + \underbrace{\sum_t (\lambda - \lambda_t)(b - \sum_i p_{i,t} V_g^{\pi_i})}_{\mathcal{T}_2} \end{aligned}$$

We now bound $\mathcal{T}_1$ and $\mathcal{T}_2$.

We can bound $\mathcal{T}_1$ using the OMD update rule in (8), we have the following

**Lemma 4** $\sum_t (\sum_i p_i^* (V_r^{\pi_i} + \lambda_t V_g^{\pi_i}) - \sum_i p_{i,t} (V_r^{\pi_i} + \lambda_t V_g^{\pi_i})) \leq \dfrac{\log|\Pi|}{\alpha} + \alpha T H^2 (1+\xi)^2$

Note that with the choice of $\alpha = O(\sqrt{\frac{\log(|\Pi|)}{TH^2(1+\xi)^2)}})$, we have $\mathcal{T}_1 \leq O(\sqrt{(1+\xi)^2 H^2 T \log(|\Pi|)})$.

Now, we bound $\mathcal{T}_2$,

**Lemma 5** *For any* $\lambda \in [0, \xi]$

$$\sum_{t=1}^{T}(\lambda - \lambda_t)(\sum_i (b - p_{i,t} V_g^{\pi_i})) \leq \frac{\lambda^2}{2\eta} + \frac{\eta T H^2}{2} \tag{10}$$

*By proper choice of* $\eta = O(\sqrt{\frac{TH^2}{\xi^2}})$, *we can bound the above by* $O(\sqrt{TH^2/\xi^2})$.

Since Lemma 5 holds for $\lambda \in [0, \xi]$, it will hold for $\lambda = 0$. Hence, putting $\lambda = 0$, and using the fact that $\sum_i p_i^* V_g^{\pi_i} \geq b$, we obtain the following

$$\sum_{t=1}^{T}(\sum_i p_i^* V_r^{\pi_i} - p_{i,t} V_r^{\pi_i}) \leq O(\sqrt{T \log(|\Pi|)} H(1+\xi)) \tag{11}$$

which bounds the sub-optimality gap by dividing by $T$.

In order to prove the violation bound, we use the strong duality result and the details are in the supplementary file.

## 4. Faster Algorithm

**Algorithm 2** Fast Robust Safe Reinforcement Learning Algorithm (FAST-RSLA)

**Input:** Robust Policy evaluator, Finite Policy Class $\Pi$.
**Initialization**: $p_0 = 1/|\Pi|$, $\hat{p}_0 = 1/|\Pi|$.
**for** $t = 0, \ldots, T-1$ **do**
  Evaluate $\min_P V_{j,P}^{\pi_i}$ for all $\pi_i \in \Pi$, for $j = r, c$ using the robust policy evaluator oracle.
  Update $p_t(i) \propto \hat{p}_t(i) \exp(\alpha(\min_P V_{r,P}^{\pi_i} + \lambda_{t-1} \min_P V_{g,P}^{\pi_i}))$ for all $i$
  Update $\lambda_t = \min\{\max\{\hat{\lambda}_t + \eta(b - \sum_i p_{i,t-1} \min_P V_{g,P}^{\pi_i}), 0\}, \xi\}$
  Update $\hat{p}_{t+1}(i) \propto \hat{p}_t(i) \exp(\alpha(\min_P V_{r,P}^{\pi_i} + \lambda_t \min_P V_{g,P}^{\pi_i}))$
  Update $\hat{\lambda}_{t+1} = \min\{\max\{\lambda_t + \eta(b - \sum_i p_{i,t} \min_P V_{g,P}^{\pi_i}), 0\}, \xi\}$
Output $p = Uniform(p_1, \ldots, p_T)$

Here, we use the OOMD algorithm or extra gradient method (Syrgkanis et al., 2015; Zhang et al., 2022; Daskalakis et al., 2020) to update $p$ and $\lambda$. We show that such an approach (Algorithm 2) has a faster convergence compared to Algorithm 1.

The algorithm maintains two copies of the primal and dual variables. For example, $p_t$ is updated based on the last-iteration's dual variable $\lambda_{t-1}$, however, it also maintains $\hat{p}_{t+1}$ which is updated based on $\lambda_t$. Hence, $\hat{p}_{t+1}$ takes into account of the dual-variable update while updating the value. Then, $\hat{p}_t$ is used as a regularization for $p_{t+1}$. The intuition is that as the primal variable change the optimal dual variable should change, thus $\hat{p}_t$ tracks the

change by taking another extra step in gradient. The updates are given by the following:

$$p_t = \arg\max_p \sum_i p_i(\min_P V_r^{\pi_i,P} + \lambda_{t-1} \min_P V_g^{\pi_i,P}) - D(p||\hat{p}_t)$$
$$\hat{p}_{t+1} = \arg\max_p \sum_i p_i(\min_P V_r^{\pi_i,P} + \lambda_t \min_P V_g^{\pi_i,P}) - D(p||\hat{p}_t)$$

In a similar fashion, the dual variable $\lambda_t$, and dummy dual variable $\hat{\lambda}_t$ are updated. Now, we state the main result of this section

**Theorem 6** *Algorithm 2 returns $p$ such that*

$$\text{Sub} - \text{Opt}(p) \leq C_1 \frac{H \log(|\Pi|)\xi}{T}, \quad \text{Violation}(p) \leq C_2 \frac{H \log(|\Pi|)\xi}{\xi T} \tag{12}$$

*for absolute constants $C_1$ and $C_2$.*

Note that the iteration complexity is $O(1/T)$. This is the first result with $O(1/T)$ iteration complexity for robust CMDP problem. Hence, we only need $O(1/\epsilon)$ number of iterations to bound both the sub-opt and violation bound by $\epsilon$.

### 4.1. Analysis: Proof Outline

Similar to Theorem 2, we seek to obtain the bound in Lemma 3. Here, we use the Regret bounded by variation in utilities RVU (Syrgkanis et al., 2015) property for OOMD–

$$\sum_{t=1}^{T} \sum_i p_i^*(V_r^{\pi_i} + \lambda_t V_g^{\pi_i}) - p_{i,t}(V_r^{\pi_i} + \lambda_t V_g^{\pi_i})$$
$$\leq \log|\Pi|/\alpha + \beta \sum_t ||(V_r^{\pi} + \lambda_t V_g^{\pi}) - (V_r^{\pi} + \lambda_{t-1} V_g^{\pi})||_\infty - \gamma \sum_t ||p_t - p_{t-1}||_1 \tag{13}$$

Similarly, from the dual update, we have $\sum_{t=1}^{T}(\lambda - \lambda_t)(b - \sum_i p_{i,t} V_g^{\pi_t}) \leq \frac{\xi^2}{\eta} + \beta \sum_t |\sum_i p_{i,t} V_g^{\pi_i} - \sum_i p_{i,t-1} V_g^{\pi_i}| - \gamma \sum_t |\lambda_t - \lambda_{t-1}|$. Here, $\beta = \eta$, and $\gamma = 1/(4\eta)$ for the update rule.

Now, $\sum_t ||(V_r^{\pi} + \lambda_t V_g^{\pi}) - (V_r^{\pi} + \lambda_{t-1} V_g^{\pi})||_\infty \leq \sum_t |\lambda_t - \lambda_{t-1}| ||V_g^{\pi}||_\infty = \sum_t |\lambda_t - \lambda_{t-1}| H$. Thus, plugging in the dual-update step, we have $\sum_t |\sum_i p_{i,t} V_g^{\pi_i} - \sum_i p_{i,t-1} V_g^{\pi_i}| \leq ||V_g^{\pi}||_\infty \sum_t ||p_t - p_{t-1}||_1 \leq H \sum_t ||p_t - p_{t-1}||$.

Hence, by combining all we have

$$\sum_t (\sum_i p_i^*(V_r^{\pi_i} + \lambda_t V_g^{\pi_i}) - p_{i,t}(V_r^{\pi_i} + \lambda_t V_g^{\pi_i})) + \sum_t (\lambda - \lambda_t)(b - \sum_i p_{i,t} V_g^{\pi_i})$$
$$\leq \log(|\Pi|)/\alpha + \frac{\xi^2}{\eta} + \eta \sum_t |\lambda_t - \lambda_{t-1}| H - \gamma \sum_t ||p_t - p_{t-1}||_1 + H\beta \sum_t ||p_t - p_{t-1}||_1 - \gamma \sum_t |\lambda_t - \lambda_{t-1}|$$

By replacing $\eta = \alpha = 1/(4H)$, the right-hand side can be bounded by constant $\log|\Pi|/\alpha + \xi^2/\alpha$. The rest of the proof is similar to that of Theorem 2.

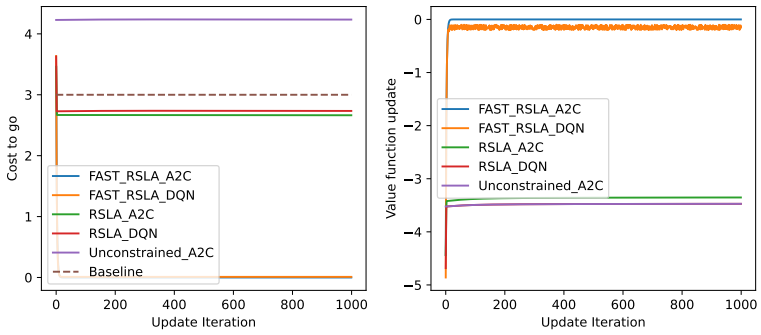


Figure 1: Comparison of the performance of various algorithms on Machine Replacement environment. The left plot corresponds to safety constraint cost and the right hand-side plot corresponds to the Value function. The x-axis denote iteration number and the y-axis denotes the safety-cost function in the left plot and the value function in the right plot.

### 4.2. Limitations and Discussions

Note that we consider the finite policy space. We do consider mixture of the policy space hence we indeed consider stochastic policies. It remains an open question on how to extend to infinite policy space. Recent studies Zhan et al. (2022); Zhang et al. (2022) show that for min-max Markov game sub-linear iteration complexity may not be achievable. Note that one can certainly train a policy for the non-robust CMDP using the nominal model, and then consider a perturbation around that policy to address the robustness guarantee to reduce the policy.

## 5. Experiments

In this section, we discuss about the experiments performed for the validation of our theory[1]. The two environments we choose to test our algorithms are (i) Constrained Machine replacement problem (CMR) and (ii) Constrained River-swim (CRS) problem. In this section we first briefly discuss the environments followed by the steps taken to perturb the environment. We conclude this section by presenting the results obtained.

**Constrained Machine Replacement**: For this environment we consider 4 states and in each state there are 2 actions (which are continue action and replace action) available for the agent to take. The states denote the operating conditions of a radioactive machine. If we consider $s_0$ state as the best state and $s_3$ as the worst state, the agent can either move to a higher state value or stay in the same state but can never move to a better state for continue ($a_0$) action. However, at any state the agent might decide to perform replacement of the present machine ($a_1$) with an identical one thus, moving from present state to the best state $s_0$. For performing replacement action, the agent receives an additional replacement cost and for safely disposing the radioactive materials, an additional safety cost. The long term safety cost should not exceed a safety threshold level as that might be hazardous. Note that the worse the state the worse is the cost to replace and the worse is the safety cost.

**Constrained River-swim environment**: The River-Swim environment consists of 6 states representing islands in a water body. A swimmer starts at any island and aims to reach either end of the river to earn a reward. At each state, the swimmer can choose to swim left($a_0$) or right ($a_1$). Rewards are only given at the end states, while intermediate states yield no reward. If we consider $s_0$ and $s_5$ as the left-most and right-most banks, the

1. The code and the experimental results can be found in the Github https://github.com/Sourav1429/RCMDP.git

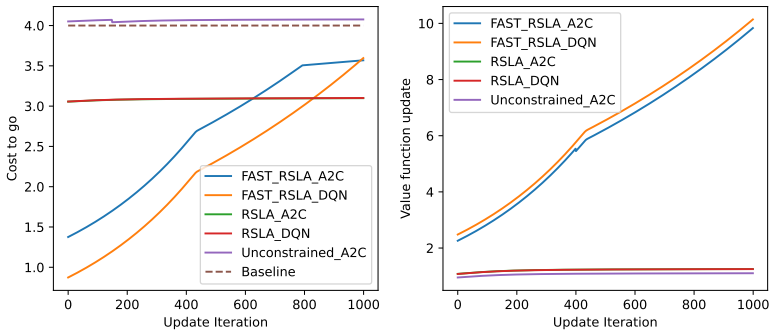


Figure 2: Comparison of the performance of various algorithms on Machine Replacement environment. The left plot corresponds to safety constraint cost and the right hand-side plot corresponds to the Value function. The x-axis denote iteration number and the y-axis denotes the safety-cost function in the left plot and the value function in the right plot

depth of the waterbody and presence of dangerous whirlpools increase as we move from $s_0$ to $s_5$. This is characterized by a safety constraint cost attached at each state. In particular, the cost is minimum when the swimmer is at state $s_0$ and maximum when the swimmer is at state $s_5$. The objective is to find which direction the swimmer should swim at each state without violating the safety threshold value while maximizing the reward.

### 5.1. Results

The Constrained machine replacement (CMR) environment is essentially a cost based environment that is, the objective of the agent is reduce the long term expected cost. On the other hand constrained Riverswim (CRS) environment is a reward based enviornment as is depicted by the figures 1 and 2 respectively. A safety threshold constraint, represented by a dashed line, is included in both environments. We set $\xi$ at 0.5. We perturb the transition probabilities by $\rho$ amount randomly for each state-action pairs. In order to have a probability distribution, we use softmax. Note that we need to have a policy space $\Pi$. We thus first train using a DQN, and Advantage Actor Critic (A2C) method to obtain a sub-optimal policy $\hat{\pi}$. We then create the policy space $\Pi$ as $\{\pi||\pi - \hat{\pi}||_2 \leq \delta\}$. We discretize to have the space finite.

From the experiments we can infer that both Fast_RSLA and RSLA algorithm are able to find the optimal policy while maintaining feasibility irrespective of training by DQN or by A2C even when the nominal model is perturbed. Note that the unconstrained A2C method is unable to satisfy the constraints for both the environments. As the theory predicted FAST_RSLA is faster compared to the RSLA algorithm for both the environments. This is because FAST_RSLA keeps an estimate of the next step dual variable using OOMD. As shown in Cen et al. (2022), keeping an estimate expedites the convergence to the equilibrium point. The results of the experiments corroborate our theoretical findings stated in the previous sections.

## 6. Conclusions and Future work

We propose RSLA and Fast RSLA with $O(1/\epsilon^2)$ and $O(1/\epsilon)$ iteration complexity respectively. Our empirical results show that our proposed approaches are able to find policies that are close to optimal and yet feasible even when there is a model-mismatch. Extending the work for infinite state-action space constitutes an interesting future research direction. Extending the experiment to large-scale real-life system also constitutes an interesting future research direction.

## References


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

Archana Bura, Aria HasanzadeZonuzy, Dileep Kalathil, Srinivas Shakkottai, and Jean-Francois Chamberland. Dope: Doubly optimistic and pessimistic exploration for safe reinforcement learning. *Advances in Neural Information Processing Systems*, 35:1047–1059, 2022.

Shicong Cen, Yuejie Chi, Simon S Du, and Lin Xiao. Faster last-iterate convergence of policy optimization in zero-sum markov games. *arXiv preprint arXiv:2210.01050*, 2022.

Constantinos Daskalakis, Dylan J Foster, and Noah Golowich. Independent policy gradient methods for competitive reinforcement learning. *Advances in neural information processing systems*, 33:5527–5540, 2020.

Dongsheng Ding, Kaiqing Zhang, Tamer Basar, and Mihailo R Jovanovic. Natural policy gradient primal-dual method for constrained markov decision processes. In *NeurIPS*, 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*, pages 3304–3312. PMLR, 2021.

Dongsheng Ding, Chen-Yu Wei, Kaiqing Zhang, and Alejandro Ribeiro. Last-iterate convergent policy gradient primal-dual methods for constrained mdps. *arXiv preprint arXiv:2306.11700*, 2023.

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

Arnob Ghosh, Xingyu Zhou, and Ness Shroff. Provably efficient model-free constrained rl with linear function approximation. *Advances in Neural Information Processing Systems*, 35:13303–13315, 2022.

Arnob Ghosh, Xingyu Zhou, and Ness Shroff. Towards achieving sub-linear regret and hard constraint violation in model-free rl. In *International Conference on Artificial Intelligence and Statistics*, pages 1054–1062. PMLR, 2024.

Garud N Iyengar. Robust dynamic programming. *Mathematics of Operations Research*, 30 (2):257–280, 2005.

Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

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

Xin Liu, Bin Li, Pengyi Shi, and Lei Ying. An efficient pessimistic-optimistic algorithm for stochastic linear bandits with general constraints. *Advances in Neural Information Processing Systems*, 34, 2021.

Daniel J Mankowitz, Dan A Calian, Rae Jeong, Cosmin Paduraru, Nicolas Heess, Sumanth Dathathri, Martin Riedmiller, and Timothy Mann. Robust constrained reinforcement learning for continuous control with model misspecification. *arXiv preprint arXiv:2010.10644*, 2020.

Sobhan Miryoosefi, Kianté Brantley, Hal Daume III, Miro Dudik, and Robert E Schapire. Reinforcement learning with convex constraints. *Advances in neural information processing systems*, 32, 2019.

Arnab Nilim and Laurent Ghaoui. Robustness in markov decision problems with uncertain transition matrices. *Advances in neural information processing systems*, 16, 2003.

Kishan Panaganti and Dileep Kalathil. Sample complexity of robust reinforcement learning with a generative model. In *International Conference on Artificial Intelligence and Statistics*, pages 9582–9602. PMLR, 2022.

Kishan Panaganti, Zaiyan Xu, Dileep Kalathil, and Mohammad Ghavamzadeh. Robust reinforcement learning using offline data. *Advances in neural information processing systems*, 35:32211–32224, 2022.

Santiago Paternain, Luiz Chamon, Miguel Calvo-Fullana, and Alejandro Ribeiro. Constrained reinforcement learning has zero duality gap. *Advances in Neural Information Processing Systems*, 32, 2019.

Reazul Hasan Russel, Mouhacine Benosman, and Jeroen Van Baar. Robust constrained-mdps: Soft-constrained robust policy optimization under model uncertainty. *arXiv preprint arXiv:2010.04870*, 2020.

Laixi Shi, Gen Li, Yuting Wei, Yuxin Chen, Matthieu Geist, and Yuejie Chi. The curious price of distributional robustness in reinforcement learning with a generative model. *Advances in Neural Information Processing Systems*, 36, 2024.

Niko Sünderhauf, Oliver Brock, Walter Scheirer, Raia Hadsell, Dieter Fox, Jürgen Leitner, Ben Upcroft, Pieter Abbeel, Wolfram Burgard, Michael Milford, et al. The limits and potentials of deep learning for robotics. *The International journal of robotics research*, 37 (4-5):405–420, 2018.

Vasilis Syrgkanis, Alekh Agarwal, Haipeng Luo, and Robert E Schapire. Fast convergence of regularized learning in games. *Advances in Neural Information Processing Systems*, 28, 2015.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 23–30. IEEE, 2017.

Sharan Vaswani, Lin F Yang, and Csaba Szepesvári. Near-optimal sample complexity bounds for constrained mdps. *arXiv preprint arXiv:2206.06270*, 2022.

Qiuhao Wang, Chin Pang Ho, and Marek Petrik. Policy gradient in robust mdps with global convergence guarantee. In *International Conference on Machine Learning*, pages 35763–35797. PMLR, 2023.

Yue Wang and Shaofeng Zou. Online robust reinforcement learning with model uncertainty. *Advances in Neural Information Processing Systems*, 34:7193–7206, 2021.

Yue Wang and Shaofeng Zou. Policy gradient method for robust reinforcement learning. In *International conference on machine learning*, pages 23484–23526. PMLR, 2022.

Yue Wang, Fei Miao, and Shaofeng Zou. Robust constrained reinforcement learning. *arXiv preprint arXiv:2209.06866*, 2022.

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*, pages 3274–3307. PMLR, 2022.

Wolfram Wiesemann, Daniel Kuhn, and Berç Rustem. Robust markov decision processes. *Mathematics of Operations Research*, 38(1):153–183, 2013.

Tengyu Xu, Yingbin Liang, and Guanghui Lan. Crpo: A new approach for safe reinforcement learning with convergence guarantee. In *International Conference on Machine Learning*, pages 11480–11491. PMLR, 2021.

Zaiyan Xu, Kishan Panaganti, and Dileep Kalathil. Improved sample complexity bounds for distributionally robust reinforcement learning. In *International Conference on Artificial Intelligence and Statistics*, pages 9728–9754. PMLR, 2023.

Lin Yang and Mengdi Wang. Sample-optimal parametric q-learning using linearly additive features. In *International Conference on Machine Learning*, pages 6995–7004. PMLR, 2019.

Wenhao Yang, Liangyu Zhang, and Zhihua Zhang. Toward theoretical understandings of robust markov decision processes: Sample complexity and asymptotics. *The Annals of Statistics*, 50(6):3223–3248, 2022.

Wenhao Zhan, Jason D Lee, and Zhuoran Yang. Decentralized optimistic hyperpolicy mirror descent: Provably no-regret learning in markov games. *arXiv preprint arXiv:2206.01588*, 2022.

Mengxiao Zhang, Peng Zhao, Haipeng Luo, and Zhi-Hua Zhou. No-regret learning in time-varying zero-sum games. In *International Conference on Machine Learning*, pages 26772–26808. PMLR, 2022.

Zhengfei Zhang, Kishan Panaganti, Laixi Shi, Yanan Sui, Adam Wierman, and Yisong Yue. Distributionally robust constrained reinforcement learning under strong duality. *arXiv preprint arXiv:2406.15788*, 2024.

Zhengqing Zhou, Zhengyuan Zhou, Qinxun Bai, Linhai Qiu, Jose Blanchet, and Peter Glynn. Finite-sample regret bound for distributionally robust offline tabular reinforcement learning. In *International Conference on Artificial Intelligence and Statistics*, pages 3331–3339. PMLR, 2021.

# Appendices

## Appendix A. Environments used

The two environments where we test our algorithms are as given below

### A.1. Modified Machine Replacement

The Machine Replacement environment is a standard reinforcement learning benchmark introduced for operations control research. In this environment, the agent operates in $n_S$ states $(0, 1, \ldots, n_S - 1)$, where at each state, the agent can take one of two actions: 0 (continue working with the current machine) or 1 (replace the current machine with a new one of the same specifications). A cost $r(s, a)$ is associated with operating the machine in state $s$ under action $a$, where $r(s, a)$ is defined as a function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$.
State 0 represents the best operational state for a machine and serves as the initial state for a newly replaced machine. At any state $s \in [n_S]$, the agent can remain in the same state or transition to a worse state but cannot transition to a better state. If the states 0 to $n_S - 1$ are ordered by increasing operational cost, an agent in state $k$ cannot transition to any state $< k$. If the replace action $(a = 1)$ is taken, the agent transitions to state 0. Additionally, a replacement cost is incurred, making the total cost for this action $replacement\_cost + \mathcal{R}(s = 0, a = 0)$, where $a = 0$ denotes the "continue operation" action.
This work modifies the problem by assuming the machine operates using nuclear fuel. As the machine deteriorates, its efficiency decreases, requiring more fuel to maintain production. This increased fuel consumption introduces higher safety risks associated with fuel storage and leakage. These risks are quantified as a safety cost $c(s, a)$, where $c(s, a)$ is defined as a function $\mathcal{C} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. If the agent chooses the replace action $(a = 1)$, the machine must be properly disposed of to prevent radioactive contamination. Consequently, an additional safety cost is incurred, making the total cost for this action $safety\_cost + \mathcal{C}(s = 0, a = 0)$, where $a = 0$ again denotes the "continue operation" action[2].
The probability function, replacement funtion and safety function is given as follows

$$\mathcal{P}(a_k, s_i, s_j) = \begin{cases} \begin{bmatrix} 0.1 & 0.2 & 0.3 & 0.4 \\ 0 & 0.225 & 0.325 & 0.45 \\ 0 & 0 & 0.4 & 0.6 \\ 0 & 0 & 0 & 1 \end{bmatrix}, & k = 0 \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, & k = 1 \end{cases} \tag{14}$$

The replacement function can be given as

$$\mathcal{R}(s_i, a_1) = c_i \ \forall \ 1 \leq \ i \ \leq n \tag{15}$$
$$\mathcal{R}(s_i, a_2) = r + c_1 \ \forall \ 1 \leq \ i \ \leq n \tag{16}$$

2. The complete code can be found at https://github.com/Sourav1429/RCMDP.git

| State | Action | Next state |
|---|---|---|
| $s_0$ | $a_0$ | $s_0$:0.9, $s_1$:0.1 |
| $s_i \underset{i\in[1,5]}{\forall}$ | $a_0$ | $s_i$:0.6, $s_{i-1}$ : 0.3, $s_{i+1}$ : 0.1 |
| $s_i \underset{i\in[0,4]}{\forall}$ | $a_1$ | $s_i$:0.6, $s_{i-1}$ : 0.1, $s_{i+1}$ : 0.3 |
| $s_5$ | $a_1$ | $s_5$:0.9, $s_4$:0.1 |

Table 1: Transition probability of River-swim environment

$$\mathcal{R}(s_i, a_k) = \begin{cases} \begin{bmatrix} 0.1 & 0.397 & 0.693 & 0.99 \end{bmatrix}, & k = 0 \\ \begin{bmatrix} 0.8 & 0.8 & 0.8 & 0.8 \end{bmatrix}, & k = 1 \end{cases} \tag{17}$$

where $replacement\_cost = 0.7 + \mathcal{R}(s = 0, a = 0) = 0.7 + 0.1 = 0.8$ The safety cost function can be given as

$$\mathcal{C}(s_i, a_1) = \mathbf{c}_i \ \forall \ 1 \leq \ i \ \leq n \tag{18}$$

$$\mathcal{C}(s_i, a_2) = s + \mathbf{c}_1 \ \forall \ 1 \leq \ i \ \leq n \tag{19}$$

$$\mathcal{C}(s_i, a_k) = \begin{cases} \begin{bmatrix} 0.2 & 0.46333333 & 0.72666667 & 0.99 \end{bmatrix}, & k = 0 \\ \begin{bmatrix} 0.6 & 0.6 & 0.6 & 0.6 \end{bmatrix}, & k = 1 \end{cases} \tag{20}$$

where $safety\_cost[s_i, a_1] = 0.4 + \mathcal{C}(s = 0, a = 0) = 0.4 + 0.2 = 0.6 \ \ i \in [0, n_S - 1]$

### A.2. River-swim

The River-swim environment is a widely studied benchmark in optimization theory and stochastic control. This environment assumes six states, conceptualized as islands in a large water body. A swimmer is initially dropped on one of these six landmasses and must navigate to one of the river's endpoints to reach a goal state and receive a reward. At each state, the swimmer can choose to swim either left or right. Rewards are only assigned at the end states, while intermediate states yield no reward (see Table 1).

The state transition diagram for this environment is illustrated in Figure 3. Transition probabilities are depicted in white text on a plain background, representing the likelihood of moving from one state to another (as indicated by the arrow). Rewards are shown in orange text with a magenta background, denoting the deterministic reward obtained at each state.

A key challenge in this environment arises from the increasing depth and presence of dangerous whirlpools as the swimmer transitions to higher states, posing significant risks. The problem thus involves identifying the optimal policy for the swimmer that balances the rewards while accounting for these risks. The deterministic constraints for each state are highlighted as red text on a yellow background in Figure 3.

## Appendix B. Performing policy perturbation

A policy is defined as a mapping from the state space to the action space for a given environment $\pi : \mathcal{S} \to \mathbb{P}(\mathcal{A})$. To peturb any function, we need to find a epsilon bound on the range space of the function. Thus, we know $\pi(s_a) = \mathbb{P}(a \ \in \mathcal{A}|s = s_a)$. Thus, after

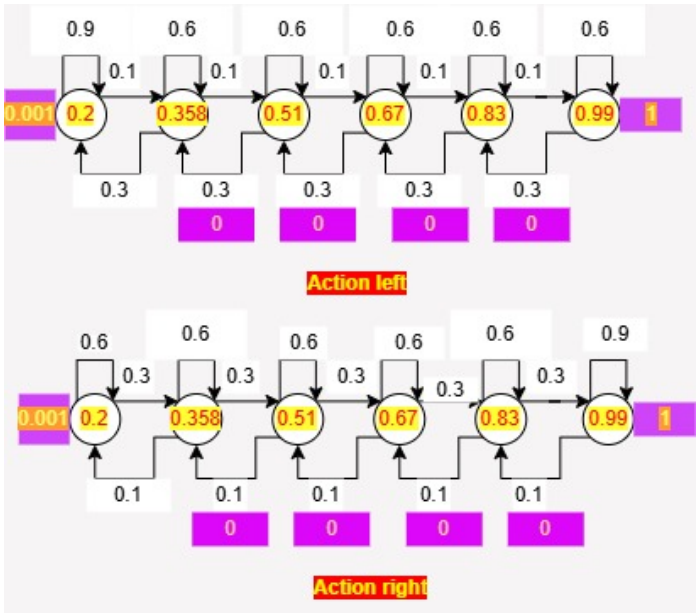


Figure 3: State diagram of Riverswim environment

soft-starting with DQN or A2C, the policy that is returned is $p = \pi(s)$. Now $p$ is probability distribution. We need to find all the policies $\Pi : \|\pi - \hat{\pi}\| \leq \epsilon$. If we discretize this policy ball with $n$ concentric circles, then $k$-th circle will have a radius of value $k.\frac{\epsilon}{2.n} \underset{k=1}{\overset{n}{\forall}}$. Hence for any circle, we need to find the policy $\hat{\pi}$ such that $\|\pi - \hat{\pi}\| \leq \epsilon = \pi - \epsilon \leq \hat{\pi} \leq \pi + \epsilon$. Thus, we perturb each value $\epsilon$ that is we create a matrix $\hat{P(s)} = P(s) + \epsilon \times I_{|\mathcal{A}|\times|\mathcal{A}|}$ where $P(s) = \pi(s) \times I_{|\mathcal{A}|\times|\mathcal{A}|}$ $\forall s \in \mathcal{S}$. where $I$ is a unit matrix or identity matrix. Finally we can treat each of the rows of $\hat{P(s)}$ as a perturbed policy of $\pi(s)$. However, it is possible that after perturbation, the rows of $\hat{P}(s)$ do not sum upto 1. Hence we should divide each row of $\hat{P}(s)$ by the sum of the values in that row. This process is repeated for all the $s \in \mathcal{S}$ for all the n circles by replacing $\epsilon$ with $(k.\frac{\epsilon}{2.n})$. This process is computationally expensive and deciding upon $n$ is difficult $|\hat{\Pi} = \{\hat{\pi} : \|\pi - \hat{\pi}\| \leq \epsilon\}| = \tilde{O}(|\mathcal{S}|.|\mathcal{A}|.n)$ and not feasible for continuous state environments

## Appendix C. Obtaining the uncertainity set

$$D(P, P_0(s,a)) \leq \delta \tag{21}$$

We have a nominal model P_0 with us which we chose to be uniform distribution for our experiments. The most basic technique to get the uncertainity is by using $D(P, P_0(s,a)) = \|P - P_0(s,a)\| \leq \delta$. Unlike policy perturbation, this time we have a vector to deal with.

$$\begin{aligned} \|P - P_0(s,a)\| &\leq \delta \\ -\delta \leq (P - P_0(s,a)) &\leq \delta \\ P_0(s,a) - \delta \leq P &\leq P_0(s,a) + \delta \end{aligned} \tag{22}$$

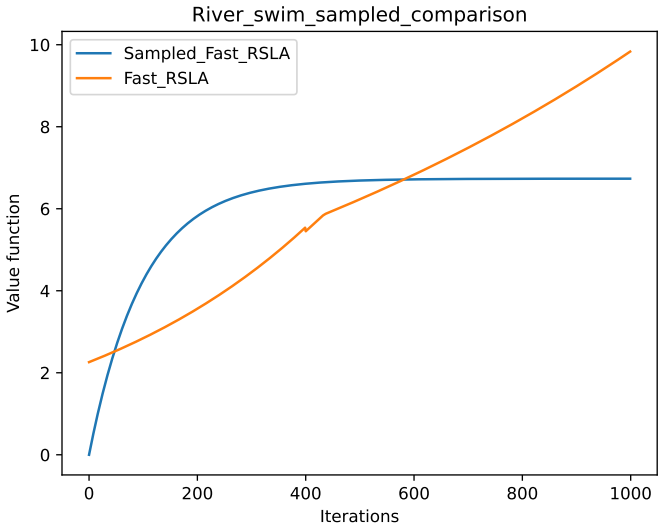


Figure 4: Comparison of the value functions between the Fast_RSLA and the Sampled Fast_RSLA variants on the CRS environment soft-started using A2C *(Time taken for FAST_RSLA is 156 minutes approximately and Sampled_FAST_RSLA is 10 minutes approximately)*

Thus, our aim is to get $P$, we perturb each element $\delta$ once and $-\delta$ once to form a set of uncertainity set. But it might be possible that after perturbation, $P$ does not form a distribution. To do that, we divide $P$ by sum of elements in $P$ or taking softmax of each element in $P$. This gives us a distribution over the next state given current state and action take. This process is repeated for all the state-action pairs to get the final uncertainity set $\mathcal{U} = \{u : D(P, P_0(s,a)) \leq \delta)\} \forall (s,a) \in (\mathcal{S}, \mathcal{A})$. $|\mathcal{U}| = \tilde{O}(|\mathcal{S}|^2.|\mathcal{A}|)$

## Appendix D. Results obtained upon implementation

We soft started the algorithm with some policy gradient technique such as DQN, Actor-critic, Advantage Actor-Critic (A2C) etc. Here we used DQN and A2C for our experiments. The results obtained by RSLA (algorithm RSLA) on Machine-Replacement and River-swim environments are as shown in figures 1 and 2 respectively

In what we follows, we extend the study to our second algorithm that is aimed to expedite the learning process by estimating the value of the upcoming probability measures of the policies. The results obtained after applying the FAST_RSLA algorithm (Algorithm FAST_RSLA) is as shown in the figure 2.

As shown in figure 8, the Unconstrained case visits state 6 for the maximum times since the reward in that state is maximum however, it does not follow he constraint. RSLA which obeys the constraint function initially explores the bad states and the quickly realises that State 1 is the safest state to be in with a nominal reward so it stays there for the rest of the time.

A key limitation of the vanilla FAST_RSLA (or RSLA) is its computational intensity in larger environments, since it requires evaluating the worst-case value function over all policies within the $\delta$-ball around the nominal policy. This task is highly challenging and resource-intensive. To address this, we approximate the evaluation by sampling a fixed number of policies from the $\delta$-ball (200 in our experiments), thereby reducing the

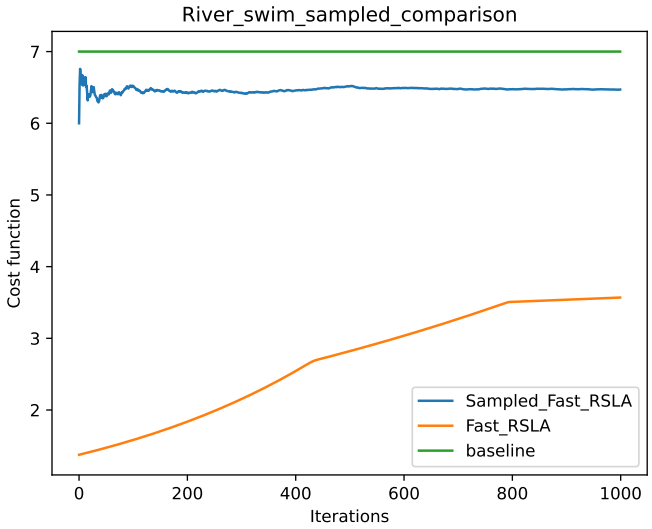


Figure 5: Comparison of the expected cost functions between the Fast_RSLA and the Sampled Fast_RSLA variants on the CRS environment soft-started using A2C

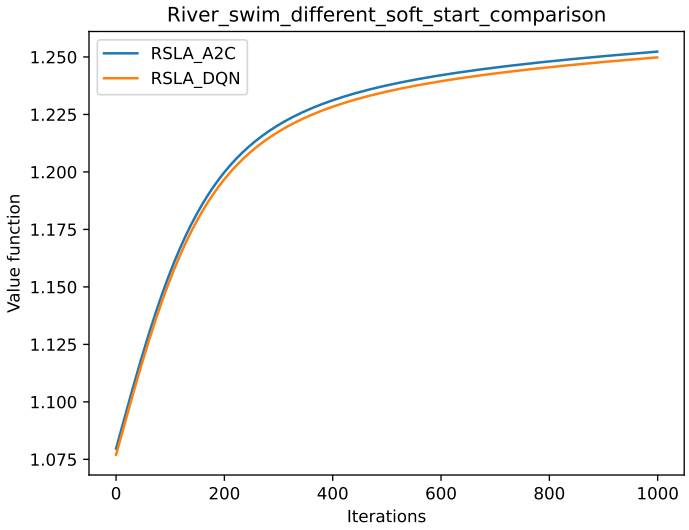


Figure 6: Value function comparison between RSLA algorithms softstarted by DQN and A2C

computational burden. Figures 4 and 5 compare the value and cost functions of the sampled variant (Sampled_FAST_RSLA) with the vanilla FAST_RSLA. As shown, Sampled_FAST_RSLA incurs a slight performance loss due to the restricted policy set, but achieves a significant reduction in running time, making it more suitable for scaling to larger environments.

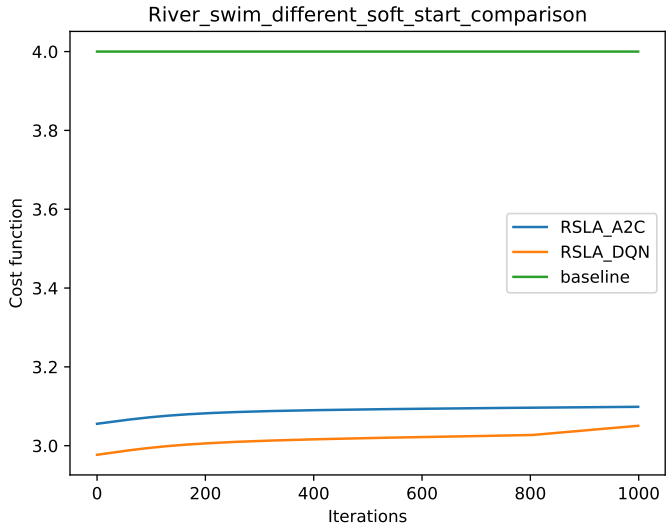


Figure 7: Cost function comparison between RSLA algorithms soft-started by DQN and A2C

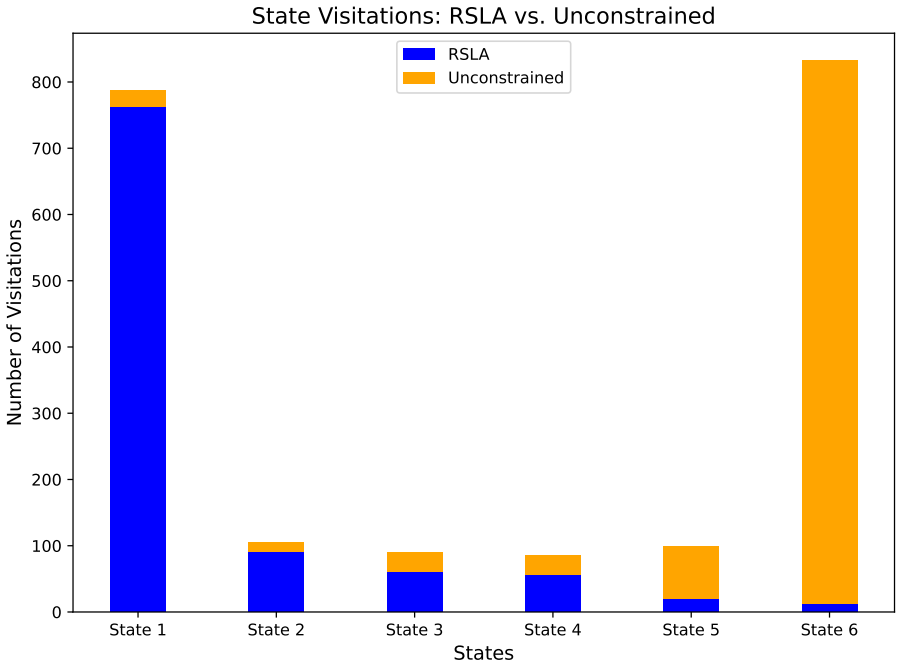


Figure 8: State visitation plot for River-swim environment

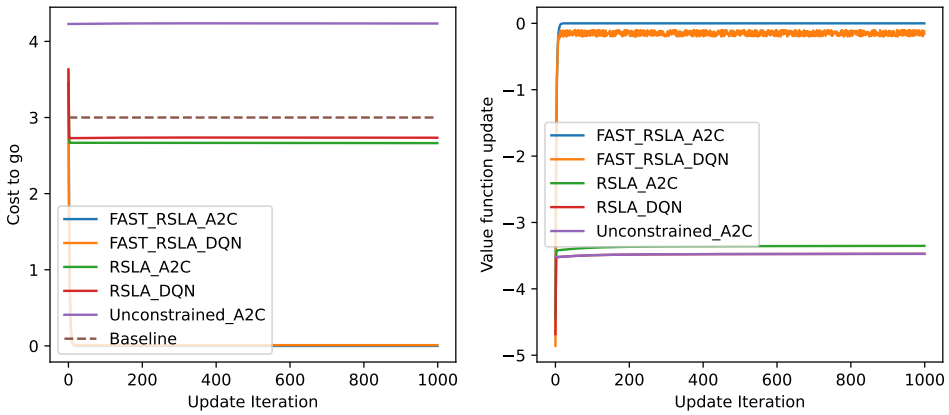


Figure 9: Comparison of the performance of various algorithms on Machine Replacement environment. The left plot corresponds to safety constraint cost and the right hand-side plot corresponds to the Value function

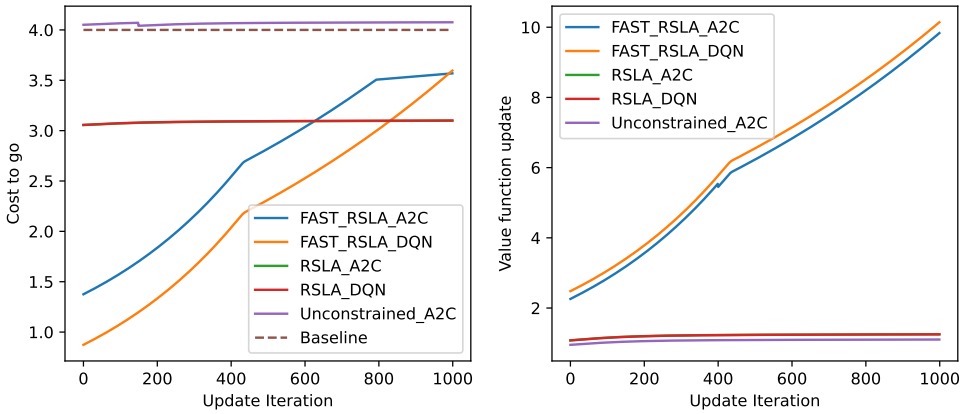


Figure 10: Comparison of the performance of various algorithms on Machine Replacement environment. The left plot corresponds to safety constraint cost and the right hand-side plot corresponds to the Value function. The x-axis denote iteration number at which the probability of selecting each policy is updated and the y-axis denotes the safety-cost function in the left plot and the value function in the right plot

## Appendix E. Detailed Proofs of Missing Results

### E.1. Proof of Lemma 3

$$\begin{aligned}
&\sum_t \sum_i (p_i^* V_r^{\pi_i} - p_{i,t} V_r^{\pi_i}) + \lambda \sum_t (b - \sum_i p_{i,t} V_g^{\pi_i}) \\
&= \sum_t \sum_i (p_i^* V_r^{\pi_i} - p_{i,t} V_r^{\pi_i}) + \sum_t \lambda_t (b - \sum_i p_{i,t} V_g^{\pi_i}) \\
&+ \sum_t (\lambda - \lambda_t)(b - \sum_i p_{i,t} V_g^{\pi_i}) \\
&\leq \sum_t \sum_i (p_i^* V_r^{\pi_i} - p_{i,t} V_r^{\pi_i}) + \sum_t \lambda_t (\sum_i p_i^* V_g^{\pi_i} - \sum_i p_{i,t} V_g^{\pi_i}) \\
&+ \sum_t (\lambda - \lambda_t)(b - \sum_i p_{i,t} V_g^{\pi_i}) \qquad (23)
\end{aligned}$$

Where we use the fact that $\sum_i p_i^* V_g^{\pi_i} \geq b$.

### E.2. Supporting Results

Lemma 4 directly follows from Online mirror descent Lattimore and Szepesvári (2020) result. Lemma 5 is adapted from Ghosh et al. (2022); Ding et al. (2021).

We use the following lemma proved in Efroni et al. (2020) (Theorem 42 there) to bound the constraint violation. Consider a constrained convex optimization problem $f_{opt} = \min_{x \in \mathcal{X}} \{f(x) : g(x) \leq 0\}$.

**Lemma 7** *Let $Y^*$ be the optimal dual variable, and $C \geq 2Y^*$, then, for any $\tilde{x} \in \mathcal{X}$*

$$f(\tilde{x}) - f_{opt} + C(g(\tilde{x}))_+ \leq \delta \qquad (24)$$

*then*

$$g(\tilde{x})_+ \leq \frac{2\delta}{C}. \qquad (25)$$

### E.3. Violation Bound for Theorem 2

Using Lemmas 4 and 5, we obtain

$$\begin{aligned}
&\sum_t \sum_i (p_i^* V_r^{\pi_i} - p_{i,t} V_r^{\pi_i}) + \lambda \sum_t (b - \sum_i p_{i,t} V_g^{\pi_i}) \\
&\leq O(\sqrt{(1+\xi)^2 H^2 T \log(|\Pi|)}) + O(\sqrt{T H^2 / \xi^2}) \qquad (26)
\end{aligned}$$

Now, note that by Algorithm 1, $p$ is uniform across the $T$ iterations, hence,

$$\sum_i (p_i^* V_r^{\pi_i} - p_i V_r^{\pi_i}) = \frac{1}{T} \sum_t \sum_i (p_i^* V_r^{\pi_i} - p_{i,t} V_r^{\pi_i}), \quad (b - \sum_i p_i V_g^{\pi_i}) = \frac{1}{T} \sum_t (b - \sum_i p_{i,t} V_g^{\pi_i})$$

Now, from (26),

$$\sum_i (p_i^* V_r^{\pi_i} - p_i V_r^{\pi_i}) + \lambda (b - \sum_i p_i V_g^{\pi_i}) \leq O(\sqrt{(1+\xi)^2 H^2 \log(|\Pi|)/T}) + O(\sqrt{H^2/\xi^2/T}) \qquad (27)$$

Now, consider that $\lambda = \xi$ if $(b - \sum_i p_i V_g^{\pi_i}) \geq 0$, and 0 otherwise. Then, we have

$$\sum_i (p_i^* V_r^{\pi_i} - p_i V_r^{\pi_i}) + \xi (b - \sum_i p_i V_g^{\pi_i})_+ \leq O(\sqrt{(1+\xi)^2 H^2 \log(|\Pi|)/T}) + O(\sqrt{H^2/\xi^2/T}) \tag{28}$$

Then, we can invoke Lemma 7 with $\xi \geq 2\lambda^*$. Hence, the result follows.

### E.4. Proof of Theorem 6

Proof of Theorem 6 follows similar manner as for the Proof of Theorem 2. Lemma 3 is already proved. We invoke the RVU property for the manner the primal and dual variables are updated in Algorithm 2. The regret is obtained by plugging $\lambda = 0$, and the violation bound is obtained similar to the Violation bound in Theorem 2.