Reward Penalties on Augmented States for Solving Richly Constrained RL Effectively

Hao Jiang, Tien Mai, Pradeep Varakantham, Huy Hoang

Singapore Management University
haojiang.2021@phdcs.smu.edu.sg, atmai@smu.edu.sg, pradeepv@smu.edu.sg, mhhoang@smu.edu.sg

Abstract

Constrained Reinforcement Learning employs trajectory-based cost constraints (such as expected cost, Value at Risk, or Conditional VaR cost) to compute safe policies. The challenge lies in handling these constraints effectively while optimizing expected reward. Existing methods convert such trajectory-based constraints into local cost constraints, but they rely on cost estimates, leading to either aggressive or conservative solutions with regards to cost. We propose an unconstrained formulation that employs reward penalties over states augmented with costs to compute safe policies. Unlike standard primal-dual methods, our approach penalizes only infeasible trajectories through state augmentation. This ensures that increasing the penalty parameter always guarantees a feasible policy, a feature lacking in primal-dual methods. Our approach exhibits strong empirical performance and theoretical properties, offering a fresh paradigm for solving complex Constrained RL problems, including rich constraints like expected cost, Value at Risk, and Conditional Value at Risk. Our experimental results demonstrate superior performance compared to leading approaches across various constraint types on multiple benchmark problems.

1 Introduction

There are multiple objectives of interest when handling safety depending on the type of domain: (a) ensuring safety constraint is never violated; (b) ensuring safety constraint is not violated in expectation; (c) ensuring the chance of safety constraint violation is small (Value at Risk, VaR) (Lucas and Klaassen 1998); (d) ensuring the expected cost of violation is bounded (Conditional Value at Risk, CVaR) (Rockafellar, Uryasev et al. 2000; Yang et al. 2021); and others. One of the main models in Reinforcement Learning to ensure safety is Constrained RL, which employs objective (b) above. Our focus in this paper is also on Constrained RL but considering the four types of constraints mentioned above.

Constrained RL problems are of relevance in domains that can be represented using an underlying Constrained Markov Decision Problem (CMDP) (Altman 1999). The main challenge in solving Constrained RL problems is the expected cost constraint, which requires averaging over multiple trajectories from the policy. Such problems have many applications including but not limited to: (a) electric self driving cars reaching destination at the earliest while minimizing the risk of getting stranded on the road with no charge; (b) robots moving through unknown terrains to reach a destination, while having a threshold on the average risk of passing through unsafe areas (e.g., a ditch) (Low, Kumar, and Sanner 2023; Pankayaraj and Varakantham 2023). Broadly, they are also applicable to problems such as robot motion planning (Ono et al. 2015; Moldovan and Abbeel 2012; Chow et al. 2015a), resource allocation (Mastronarde and van der Schaar 2010; Junges et al. 2015; Chen et al. 2023; Ling et al. 2023), and financial engineering (Abe et al. 2010; Di Castro, Tamar, and Mannor 2012).

Research in Constrained RL: Many model free approaches have been proposed to solve Constrained RL problems. One of the initial approaches to be developed for addressing such constraints is the Lagrangian method (Chow et al. 2015b). However, such an approach does not provide either theoretical or empirical guarantees in ensuring the constraints are enforced. To counter the issue of safety guarantees, next set of approaches focused on transforming the cost constraint over trajectories into cost constraint over local decisions in many different ways (Gábor, Kalmár, and Szepesvári 1998; Achiam et al. 2017a; Chow et al. 2019b; Satija, Amortila, and Pineau 2020; As et al. 2022; Hogewind et al. 2022). In converting a trajectory based constraint to a local constraint, there is an estimation of cost involved in the trajectory. Due to such estimation, transformed cost constraints over individual decisions are error prone. In problems where the estimation is not close to the actual, results with such approaches with regards to cost constraint enforcement are poor (as we demonstrate in our experimental results). Since augmented state has information of cost incurred so far, our approach does not have to rely on cost estimation.

State Augmentation in Constrained RL: Prior research has explored state augmentation in Constrained RL. One study (Sootla et al. 2022b) focused on "hard" constraints, like ensuring safe landings with a probability of 1. Another work (Sootla et al. 2022a) added accumulated costs to states for policy training. A third approach (Calvo-Fullana et al. 2021) incorporated penalty coefficients from the Lagrangian into states for policy determination, similar to (Sootla et al. 2022a). Our approach has notable differences. First, we handle various "soft" cost constraints like expected cost and CVaR, where mere state augmentation is insufficient due to

limited information from a single trajectory violation. Second, our primary contribution is utilizing appropriate reward penalties over augmented states, leading to improved cost-constrained policies. These reward penalties enable the generalization of our approach to diverse constraint types, including expected cost and CVaR cost.

Contributions: We make four key contributions:

- We provide a re-formulation of the constrained RL problem through augmenting the state space with cost accumulated so far and employing reward penalties when cost constraint is violated. In contrast to conventional primal-dual techniques, which optimize an expectation by penalizing all trajectories, our approach focuses solely on penalizing infeasible trajectories using state augmentation. This distinctive strategy ensures that, as we escalate the penalty parameter, the attainment of a feasible policy is achieved faster (demonstrated empirically) and is a guaranteed outcome (as demonstrated by our Theorem 3.5), which is not the case when employing a primal-dual method.
- We show theoretically the difference between our reward penalties and Lagrangian penalties. We also show that the reward penalties employed in the new formulation are not ad-hoc and can equivalently represent different constraints mentioned in the first paragraph of the introduction, i.e. risk-neutral, chance constrained (or VAR) and CVaR constraints.
- We modify existing RL methods (DQN and SAC) to solve the re-formulated RL problem with augmented state space and more importantly employing reward penalties when constraints are violated. The knowledge of exact costs incurred so far (available within the state space) allows for assigning credit for cost constraint violations more precisely during learning compared to existing approaches.
- Finally, we demonstrate the utility of our approach by comparing against leading approaches for constrained RL on multiple benchmark problems for different types of constraints. We show that our approaches are able to outperform leading Constrained RL approaches from the literature either with respect to expected value or in enforcing different types of cost constraints or both.

2 Constrained Markov Decision Process

A Constrained Markov Decision Process (CMDP) (Altman 1999) is defined using tuple $\langle S, A, r, p, d, s_0, c_{max}\rangle$, where S is set of states with initial state as s_0, A is set of actions, $r : S\times A \to \mathbb{R}$ is reward with respect to each state-action pair, $p : S\times A \to P$ is transition probability of each state. $d : S \to d(S)$ is the cost function and c_{max} is the maximum allowed cumulative cost. Here, we assume that $d(s) \geq 0$ for all $s \in S$. This assumption is not restrictive as one can always add positive amounts to $d(s)$ and c_{max} to meet the assumption. The objective in a risk-neutral CMDP is to compute a policy, $\pi : S \times A \to [0, 1]$, which maximizes reward over a finite horizon T while ensuring the cumulative cost does not exceed the maximum allowed cumulative cost.

$$\begin{aligned} \max_{\pi} & \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t r(s_t, a_t)|s_0, \pi\right] \\ \text{s.t.} \quad & \mathbb{E}\left[\sum_{t=0}^{T} d(s_t)|s_0, \pi\right] \leq c_{max}. \end{aligned} \tag{RN-CMDP}$$

In the constraint above, we remove discount factor for notational simplicity. It can be easily included with minor changes to the extended MDP formulation.

The literature has seen other types of constraints, e.g., chance constraints requiring that $P_\pi(D(\tau) > c_{max}) \leq \alpha$ for a risk level $\alpha \in [0, 1]$, or CVaR ones of the form $\mathbb{E}_\pi[(D(\tau) - c_{max})^+] \leq \beta$, where $D(\tau) = \sum_{s\in\tau} d(s)$ is the cumulative cost in trajectory τ. Handling different types of constraints would require different techniques. In the next section, we present our approach based on augmented state and reward penalties that assembles all the aforementioned constraint types into one single framework.

3 Cost Augmented Formulation for Safe RL

We first present our extended MDP reformulation and provide several theoretical findings that connect our extended formula with different variants of CMDP. We focus on the case of single-constrained MDP. Extension to multiple-constrained MDP will be discussed in the appendix.

3.1 Extended MDP Reformulation

We introduce our approach to track the accumulated cost at each time period, which allows us to determine states that potentially lead to high-cost trajectories. To this end, let us define a new MDP with an extended state space $\left\langle \widetilde{S}, A, \widetilde{r}, \widetilde{p}, d, s_0, c_{max}\right\rangle$ where $\widetilde{S} = \{(s, c)|\ s \in S, c \in \mathbb{R}_+\}$. That is, each state s' of the extended MDP includes an original state from S and information about the accumulated cost. We then define the transition probabilities between states in the extended space.

$$\widetilde{p}((s'_{t+1}, c'_{t+1})|(s_t, c_t), a_t) = \begin{cases} p(s'_{t+1}|s_t, a_t) \\ \qquad \text{if } c'_{t+1} = c_t + d(s_t) \\ 0 \text{ otherwise} \end{cases}$$

and new rewards with penalties

$$\widetilde{r}(s_t, a_t, c_t) = \begin{cases} r(s_t, a_t) \\ \text{ if } c_t \leq c_{max} \text{ and } c_t + d(s_t) \leq c_{max} \\ r(s_t, a_t) - \lambda(c_t + d(s_t))/\gamma^t \\ \quad \text{if } c_t \leq c_{max} \text{ and } c_t + d(s_t) > c_{max} \\ r(s_t, a_t) - \lambda d(s_t)/\gamma^t \text{ if } c_t > c_{max} \end{cases} \tag{1}$$

where λ is a positive scalar and $\lambda d(s_t)$ and $\lambda(c_t + d(s_t))$ are penalties given to the agent if the accumulated cost exceeds the upper bound c_{max}. In the second case of (1) (for stages right before exceeding the upper bound c_{max}), we add a penalty λc_t to capture the accumulated cost until those stages. Under the reward penalties specified in the second and the third cases of (1), the accumulated reward

for each trajectory $\tau = \{(s_0, a_0), \ldots, (s_T, a_T)\}$ can be written as $\widetilde{R}(\tau) = \sum_t \gamma^t r(s_t, a_t)$ if $D(\tau) \leq c_{max}$ and $\widetilde{R}(\tau) = \sum_t \gamma^t r(s_t, a_t) - \lambda D(\tau)$ if $D(\tau) > c_{max}$, where $D(\tau)$ is the total cost of trajectory τ, i.e., $D(\tau) = \sum_{s_t \in \tau} d(s_t)$.

State augmentation is essential for us to monitor the accumulated cost and determine where to apply reward penalties. It's worth noting that the primary focus of our contribution is not on the state augmentation itself. Instead, our main innovation lies in our approach to penalizing rewards using augmented states and cost functions. This approach allows us to establish a connection between our reformulation and various types of constraints.

In the reward definition, we penalize *every trajectory* that violates the cost constraint. This allows for the fine grained control that ensures cost constraint is enforced correctly, while also allowing for expected reward maximization. Overall, we have the following unconstrained objective which handles the constraints in a relaxed manner through penalties:

$$\max_{\pi} \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t \widetilde{r}(s_t, a_t, c_t) \Big| (s_0, c_0), \pi\right] \qquad \text{(EMDP)}$$

where $c_0 = 0$. There are also other ways to penalize the rewards, allowing us to establish equivalence between the extended MDP and other risk-averse CMDPs, which we will discuss later in the next section.

3.2 Theory: Reward Penalty vs Lagrangian Penalty

We first discuss a connection between our extended formulation and Lagrange-based methods. In a Lagrange-based algorithm, the following Lagrange dual is considered

$$L(\pi, \lambda) = E_\pi\left[\sum_t \gamma^t r(s_t, a_t)\right] - \lambda\left(E_\pi\left[\sum_t \gamma^t d(s_t)\right]\right)$$

It can be seen that in a Lagrange-based method, all trajectories are penalized, while only violated trajectories are penalized in our (EMDP) formulation. To further understand the effect of this difference, we compare policies obtained via solving $\max_\pi L(\pi, \lambda)$ and (EMDP). Under a penalty parameter λ, let π_λ^{EX} be an optimal policy for (EMDP) and π_λ^{Lag} is optimal for $\max_\pi L(\lambda, \pi)$. We have the following result

Proposition 3.1 (Comparison with Lagrange-based methods)**.** *For any $\lambda > 0$, we have*

$$\mathbb{E}_{\pi_\lambda^{Lag}}[D(\tau) | D(\tau) \leq c_{max}] \leq \mathbb{E}_{\pi_\lambda^{EX}}[D(\tau) | D(\tau) \leq c_{max}]$$

Moreover, if the expected constraint is replaced by a chance constraint $P_\pi(D(\tau) > c_{max}) \leq \alpha$, then

$$P_{\pi_\lambda^{EX}}(D(\tau) > c_{max}) \leq P_{\pi_\lambda^{Lag}}(D(\tau) > c_{max})$$

Proposition 3.1 implies that π_λ^{EX} will offer a higher expected cost over feasible trajectories compared to π_λ^{Lag}. To achieve this, intuitively, π_λ^{EX} would need to assign lower probabilities to violated trajectories, consequently yielding a reduced expected cost. This becomes clearer if we consider a chance-constrained formulation, as Proposition 3.1 indicates that π_λ^{EX} assigns lower probabilities to violated trajectories in comparison to π_λ^{Lag}. All these indicate that, under the same λ, π_λ^{EX} tends to offer a lower cost value than that from π_λ^{Lag}. As a result, a Lagrange-based method would need a higher penalty λ to bring the policy to the safe area. Because of the lower value of penalty in our case and a direct connection of reward penalty to specific trajectories in our approach, this can potentially help with faster convergence to a safe area (as can be observed in our experiments).

3.3 Theory: Generality of Reward Penalties

To demonstrate the generality in the representation of the reward penalties along with state augmentation in the unconstrained MDP (EMDP), we provide theoretical properties that map (EMDP) to CMDP under different types of constraints (expected cost, VaR, CVaR, Worst-case cost):

(i) Proposition 3.2 states that if the penalty parameter $\lambda = 0$, then (EMDP) becomes the classical unconstrained MDP.

(ii) Theorem B.1 (in appendix) shows that if $\lambda = \infty$, then (EMDP) is equivalent to a worst-case constrained MDP

(iii) Theorem 3.5 establishes a lower bound on λ from which any solution to (EMDP) will satisfy the risk-neutral constraint in (RN-CMDP).

(iv) Theorem A.1 (in appendix) connects (EMDP) with chance-constrained MDP by providing a lower bound for λ from which any solution to (EMDP) will satisfy a VaR constraint $P(\sum_t d(s_t) \leq c_{max}) \leq \alpha$.

(v) Theorems A.1 (in appendix) and 3.6 further strengthen the above results by showing that, under some different reward settings, (EMDP) is equivalent to a chance-constrained (or VaR) or equivalent to a CVaR CMDP.

We now describe our theoretical results in detail. All the proofs can be found in the appendix. We also extend the results to Constrained MDPs with multiple constraints (e.g., a combination of expected cost on one cost measure and CVaR on another cost measure) in the appendix. We first state, in Proposition 3.2, a quite obvious result saying that if we set the penalty parameter $\lambda = 0$, then the MDP with augmented state space becomes the original unconstrained MDP.

Proposition 3.2. *If $\lambda = 0$, then* (EMDP) *is equivalent to the unconstrained MDP* $\max_\pi \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t r(s_t, a_t) | s_0, \pi\right]$.

It can be seen that increasing λ will set more penalties to trajectories whose costs exceed the maximum cost allowed $c_{\max}$, which also implies that (EMDP) would lower the probabilities of taking these trajectories. So, intuitively, if we raise λ to infinity, then (EMDP) will give policies that yield *zero* probabilities to violating trajectories. We state this result in Theorem B.1 in the appendix.

We further establish a bound for λ from which (EMDP) gives a feasible to (RN-CMDP). To this end, let us define Ψ^* as the optimal value of the unconstrained MDP problem $\Psi^* = \max_\pi \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t r(s_t, a_t) | s_0, \pi\right]$. and $\overline{\Psi}$ be the optimal value of the worst-case CMDP. We define a conditional expectation $\Phi_\pi\left[D(\tau) | \, D(\tau) \leq c_{\max}\right] =$

$\sum_{\tau|\ D(\tau)\leq c_{max}} P_\pi(\tau)D(\tau)$ where $P_\pi(\tau)$ is the probability of τ under policy π. Before presenting the bound, we first need two lemmas. Lemma 3.3 establishes a condition under which a policy π is feasible to the (RN-CMDP).

Lemma 3.3. *Let* $\phi^* = c_{max} - \max_\pi \{\Phi_\pi[D(\tau)|\ D(\tau) \leq c_{max}]\}$. *Given any policy* π, *if* $\Phi_\pi[D(\tau)|\ D(\tau) > c_{max}] \leq \phi^*$, *then* $\mathbb{E}_\pi[D(\tau)] \leq c_{max}$.

Lemma 3.4 below further provides an upper bound for the expected cost of violating trajectories under an optimal policy given by the extended MDP reformulation (EMDP).

Lemma 3.4. *Given* $\lambda > 0$, *let* π^* *be an optimal solution to* (EMDP). *We have* $\Phi_{\pi^*}[D(\tau)|\ D(\tau) > c_{max}] \leq \frac{\Psi^* - \overline{\Psi}}{\lambda}$.

Using Lemmas 3.3 and 3.4, we are ready to state the main result in Theorem 3.5 below.

Theorem 3.5 (Connection to the risk-neutral CMDP)**.** *For any* $\lambda \geq \frac{\Psi^* - \overline{\Psi}}{\phi^*}$, *a solution to* (EMDP) *is always feasible to the* (RN-CMDP).

To prove Lemmas 3.3, 3.4, we leverage that objective of (EMDP) can be written equivalently as

$$\mathbb{E}_\pi\left[\sum_t \gamma^t r(s_t, a_t)\right] - \lambda\Phi_\pi\left[D(\tau)|\ D(\tau) > c_{\max}\right] \quad (2)$$

which allows us to establish a relation between λ and $\Phi_{\pi^*}[D(\tau)|\ D(\tau) > c_{\max}]$, where π^* is an optimal policy of (EMDP). The bounds then come from this relation. We refer the reader to the appendix for detailed proofs.

We further connect (EMDP) with a risk-averse CMDP that has a CVaR intuition. The theorem below shows that, by slightly changing the reward penalties, (EMDP) actually solves a risk-averse CMDP problem.

Theorem 3.6 (CVaR CMDP equivalence)**.** *If we modify the reward penalties as*

$$\widetilde{r}(s_t, a_t, c_t) = \begin{cases} r(s_t, a_t) \text{ if } c_t + d(s_t) \leq c_{max} \\ r(s_t, a_t) - \lambda(c_t + d(s_t) - c_{max})/\gamma^t \\ \quad \text{if } c_t \leq c_{max} \text{ and } c_t + d(s_t) > c_{max} \\ r(s_t, a_t) - \lambda d(s_t)/\gamma^t \text{ if } c_t > c_{max} \end{cases}$$

then for any $\lambda > 0$, *there is* $\beta^\lambda \in \left[0; \frac{\Psi^* - \overline{\Psi}}{\lambda}\right]$ *(*β^λ *is dependent of* λ*) such that any optimal solution to the extended CMDP* (EMDP) *is also optimal to the following risk-averse CMDP*

$$\max_\pi \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t r(s_t, a_t)|s_0, \pi\right] \quad \text{(CVaR-CMDP)}$$
$$s.t. \mathbb{E}_{\tau\sim\pi}\left[(D(\tau) - c_{max})^+\right] \leq \beta^\lambda.$$

Moreover, $\lim_{\lambda\to\infty} \beta^\lambda = 0$.

It can be also shown that, under a chance-constrained (or VaR) CMDP, there is also a lower bound for λ from which any solution to (EMDP) always satisfies a chance constraint (or VaR). Furthermore, by changing the reward penalties, we can show that (EMDP) is equivalent to a chance-constrained CMDP. Due to limited space, we present these results in the appendix.

In practice, since λ is just a scalar, one can just gradually increase it from 0 to get feasible policies or decrease it from a large value if the policy becomes too conservative.

Theorem 3.6 (and theorems related to Var-CMDP) are just to show that there are confidence levels where the equivalence happens. In other words, we always can find penalty parameters such that solving our extended MDP yields feasible solutions to any risk-aware formulation, under any predefined confidence level.

This indicates the generality of the unconstrained extended MDP formulation (EMDP). In summary, we show that (EMDP) brings risk-neutral, worst-case and VaR and CVaR CMDPs under one umbrella.

3.4 Example

To further illustrate the advantage of our reward penalization approach, let us consider the following deterministic MDP (the MDP is made simple for the sake of illustration). There are 6 states, indexed from 0 to 5. Their rewards are $r(0) = 0, r(1) = r(2) = r(3) = 3$, $r(4) = 5$, $c(1) = c(2) = c(3) = 3$, $c(4) = 2$, $c_{max} = 4$. Since the MDP is deterministic, actions are treated as next states. There are three trajectories from state 0 to 5, denotes as $\tau_1 = \{0, 1, 4, 5\}, \tau_2 = \{0, 2, 4, 5\}, \tau_3 = \{0, 3, 5\}$. Let $R(\tau) = \sum_{s\in\tau} r(s)$ Under a Lagrange-based method, the objective is $L(\pi, \lambda) = \pi(1|0)(R(\tau_1) - \lambda D(\tau_1))$ $+\pi(2|0)(R(\tau_2) - \lambda D(\tau_1)) + (\pi(3|0)R(\tau_3) - \lambda D(\tau_3))$. It can be seen that τ_1 and τ_2 are violated w.r.t c_{max}. Therefore, our reward penalty approach considers the following objective, where only τ_1, τ_2 are penalized: $F(\pi, \lambda) = \pi(1|0)(R(\tau_1) - \lambda D(\tau_1)) + \pi(2|0)(R(\tau_2) - \lambda D(\tau_1)) + (\pi(3|0)R(\tau_3))$. To encourage random policies, we set a lower bound of 0.1 for any $\pi(s'|s)$ (otherwise optimal policies will become degenerated). We also include the *hard-constrained approach* in (Sootla et al. 2022b) for the sake of comparison. In (Sootla et al. 2022b), states are also augmented as we do, but when the accumulated cost exceeds c_{max}, the rewards of augmented states are set to infinity (for this approach we remove the policy lower bound, otherwise the objective will become infinite for any given policy). Under this hard-constrained method, τ_1 and τ_2 are violated, thus take $-\infty$ rewards. Consequently, the unique optimal policy is $\pi^{\text{hard}}(3|0) = 1, \pi^{\text{hard}}(1|0) = \pi^{\text{hard}}(2|0) = 0$, which yields an expected reward of 3 and expected cost of 3. For the Lagrange-based method, we increase λ from 0 and solve

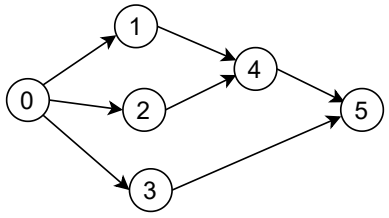

Figure 1: Example

$\max_\pi L(\pi, \lambda)$ for each value of λ. We then observe that only when $\lambda \geq 2.6$, the maximization offers a safe policy $\pi^{\text{Lag}}(1|0) = \pi^{\text{Lag}}(2|0) = 0.1$ and $\pi^{\text{Lag}}(3|0) = 0.8$, which yield an expected reward of 4, and expected cost of 3.8. For our method, we also vary λ and solve $\max_\pi F(\pi, \lambda)$ for each value of λ. We then see that a safe policy $\pi^{\text{EX}} = \pi^{\text{Lag}}$ is achieved when $\lambda = 1.1$, with an expected reward of 4, and an expected cost of 3.8. Looking at all the approaches together, we see that π^{hard} is safe, but worst than those from the other methods (so too conservative), and the Lagrange-based method needs a significantly higher λ to get a safe policy, compared to our method.

4 Safe RL Algorithms

In this section, we update existing RL methods to effectively utilize the extended state space and reward penalties, while considering RN-CMDP. Due to the theoretical findings in the previous section, by altering regular RL methods to consider different reward penalties on augmented state space, we can solve Constrained MDPs with different types of constraints, e.g., expected cost, VaR or CVar CMDPs.

4.1 Safe DQN

Deep Q Network (DQN) (Mnih et al. 2015) is an efficient method to learn in primarily discrete action RL problems. However, the original DQN does not consider safety constraints and cannot be applied to any of the CMDP variants. The main modifications in the updated algorithm, referred to as Safe DQN are with regards to exploiting the extended state space and the reward penalties based on constraint violations. The pseudo code for the Safe DQN algorithm is provided in the appendix.

The impact of extended state space on the algorithm can be observed in almost every line of the algorithm When selecting an action, Safe DQN does not consider the feasibility of the action with respect to cost. Instead, like in the original DQN, it is purely based on the current Q value. The assumption is that the penalties accrued due to violation will be sufficient to force the agent away from cost-infeasible actions. Once the new rewards are obtained (based on considering reward penalties), the Q network is updated using the mean square error loss . To avoid conservative decisions, we dynamically change the constraint penalty λ . We set an initial value for λ. During the training, we evaluate the maximum final cost in the recent few episodes, if the maximum final cost exceeds c_{max}, we make no change to λ, otherwise, we set λ to be $0.95 \times \lambda$ to diminish the conservativeness of the policy. However, the value of λ cannot keep decreasing, so we set a lower bound for it. We refer the reader to the appendix for more details.

4.2 Safe SAC

Soft Actor-Critic (SAC) (Haarnoja et al. 2018) is an off-policy algorithm that learns a stochastic policy for discrete and continuous action RL problems. SAC employs policy entropy in conjunction with value function to ensure a better tradeoff between exploration and exploitation. The Q value function in SAC is defined as follows:

$$Q(s,a) = \mathbb{E}\Big[\sum_{t=0}^{\infty} \gamma^t (r(s_t,a_t) + \alpha H_\pi(s_t)) | s_0 = s, a_0 = a\Big] \tag{3}$$

where $H_\pi(.)$ denotes the entropy of the action distribution for a given state s_t. SAC also employs a double Q-trick, i.e., two target Q-functions ($Q^{targ,i}(.),\ i \in 1,2$) are set for both Q-functions ($Q^i(.),\ i \in 1,2$). The minimum value of target Q-functions are set as target to avoid overestimation, the loss function can thus be defined as following:

$$\begin{aligned} L^i &= \mathbb{E}[Q^i(s,a) - y(s',d)] \\ y(s',d) &= r(s,a) + \gamma\Big(\min_{i=1,2} Q^{targ,i}(s',\tilde{a}') \\ &\qquad - \alpha \log \pi(\tilde{a}'|s')\Big) \end{aligned} \tag{4}$$

where L^i is the loss function for Q^i, $\tilde{a}' \sim \pi(\cdot|s')$.

Our algorithm that handles safety constraints is referred to as Safe SAC. It builds on SAC by having an extended state space and a new action selection strategy that exploits the extended state space. In Safe DQN, we primarily rely on the violation of constraints, so as to learn about the bad trajectories and avoid them. While such an approach works well for discrete action spaces and in an off-policy setting, it is sample inefficient and can be slow for on-policy (actor-critic) settings. In Safe SAC, apart from the reward penalty, we also focus on learning feasible actions, which are generated through the use of the cost accumulated so far (available as part of the state space) and an estimate of Q value on the future cost.

Formally, we define the optimization to select safe actions (at each decision epoch) in Equation (5). Extending on the double Q trick for reward, we also have double Q for future cost, referred to as $\{Q_D^i\}_{i\in 1,2}$. At each step, the objective is to pick an action that would maximize the reward Q value for the extended state and action minus the weighted entropy of the action. The constraint ensures that we only pick those actions that will not violate the cost constraint. Specifically, in the left-hand side of the constraint, we calculate the overall expected cost using : (a) (estimate) of the future cost, from the current state; (b) (actual) cost incurred so far; and (c) subtracting the (actual) cost incurred at the current step, as it is part of both (a) and (b);

$$\begin{aligned} &\arg\max_a \min_{i=1,2} Q^i((s,c),a) - \alpha \log \pi(a|(s,c)) \\ &\quad \textbf{s.t.} \max_{i=1,2} Q_D^i((s,c),a) + c - d((s,c)) \leq c_{max}, \forall (s,c) \end{aligned} \tag{5}$$

The detailed pseudocode for Safe SAC is provided in the appendix.

5 Experimental Results

We experimentally answer the following questions with regards to our approaches :

- Solving RN-CMDP: How do Safe SAC and Safe DQN compare with regards to expected reward and expected cost against leading baseline approaches for Constrained

RL? We compare against the BVF (Satija, Amortila, and Pineau 2020), Lyapunov (Chow et al. 2019a) and original DQN (Mnih et al. 2015) in discrete action environments. As for continuous action environments in Safety Gym, we compare against more suited approaches namely FOCOPS (Zhang, Vuong, and Ross 2020), CUP (Yang et al. 2022) and CPO (Achiam et al. 2017b).

- Solving CVaR-CMDP: How do we compare with regards to CVaR cost constraint? In this case, there is only one benchmark approach, namely WCSAC (Yang et al. 2021).
- Ablation analysis: How important is augmenting states by itself in comparison to having state augmentation and reward penalty simultaneously.
- Impact of reward penalty: How to set and modify the reward penalty λ in ensuring the right trade-off between expected reward performance and cost constraint enforcement?

The performance values (expected cost and expected reward) along with the standard deviation in each experiment are averaged over 5 runs. We did not provide any results with Var-CMDP because there are no existing approaches to compare against that have been developed for it.

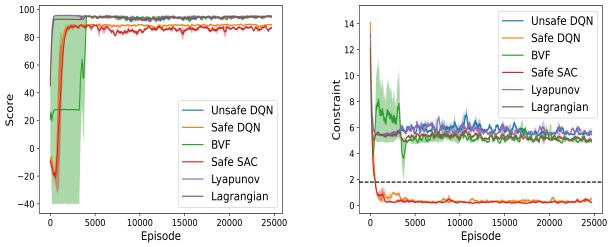

Figure 2: Comparison on Stochastic Gridworld Environment

5.1 RN-CMDP

For a discrete state and discrete action environment, we consider the stochastic 2D grid world problem introduced in previous CMDP works (Leike et al. 2017; Chow et al. 2018; Satija, Amortila, and Pineau 2020; Jain, Khetarpal, and Precup 2021). We employ a modified more challenging Gridworld in the experiment and show the details in the appendix. We set the expected cost threshold, $c_{max} = 2$, meaning agent could pass at most one pit.

Figure 2 shows the performance of each method with respect to expected reward (score) and expected cost (constraint): (a) With respect to expected reward, among safe approaches, Lyapunov achieves the highest reward. However, it violates the expected cost constraint by more than twice the cost constraint value. This is because the cost threshold is low and Lyapunov is typically a bit aggressive with regards to pursuing returns and thereby can violate cost constraints. (b) Safe SAC and Safe DQN achieve similar expected reward values, though Safe SAC converges faster. This high expected reward value is achieved while satisfying the expected cost constraint after 1000 episodes. (c) The other constrained RL approach, BVF is the last to converge while not being able to satisfy the expected cost constraint. (d) As expected, Unsafe DQN achieved the highest expected reward but was unable to satisfy the expected cost constraint.

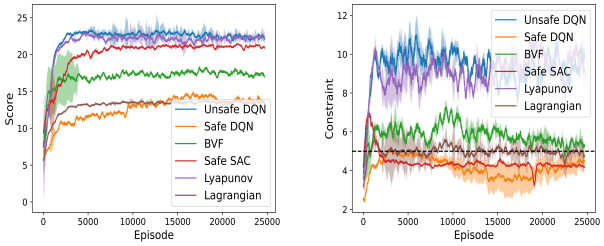

Figure 3: Comparison on Highway environment

Next, we consider the highway environment, which is a continuous state, discrete action environment. We set the $c_{max} = 8$. Details of highway environment are available in appendix. Inspired by experiment in GPIRL (Levine, Popovic, and Koltun 2011), we test our safe methods in the highway environment (Leurent 2018) of Figure 3. Safe SAC is able to get high expected rewards while satisfying the expected cost constraint. We also provide more results for RN-CMDP in appendix.

We then compare Safe SAC with recent safe methods for continuous action spaces on the two environments - SafetyPointGoal1-v0, SafetyCarGoal1-v0 from Safety Gymnasium (Ji et al. 2023). The goal of both environments is navigation to the goal with different types of agent (Point or Car). We set $c_{max} = 15$ show the performances in Figure 4 and Figure 5. Safe SAC performs the best in terms of achieving the highest expected reward while satisfying the cost constraints. While CPO performs on par with Safe SAC on expected reward, it is unable to satisfy the constraint. CUP and FOCOPS satisfy the constraints but do not do very well on the expected reward.

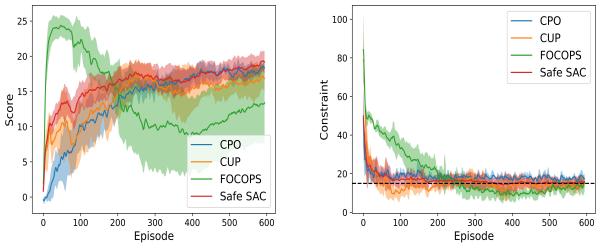

Figure 4: Comparison on SafetyPointGoal environment

5.2 CVaR-CMDP

We introduce CVaR constraint to our Safe SAC method and compare it against WCSAC (Yang et al. 2021), which is a leading algotrithm for CVaR constraint. As WCSAC is limited to continuous space, we only do the comparison with

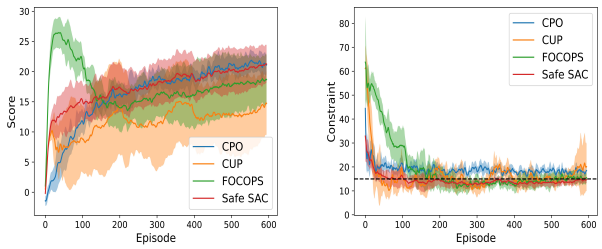

Figure 5: Comparison on SafetyCarGoal1 environment

continuous environments. Figure 6 shows the comparisons on the merge benchmark problem from GPIRL (Levine, Popovic, and Koltun 2011). Safe SAC was able to marginally perform better than WCSAC both with respect to expected reward and expected cost. As explained earlier due to reward penalties, our algorithm is able to get to the safe region (with regards to cost constraint) faster. ***This performance is noteworthy as we use the same algorithm to handle both RN-CMDP and CVaR-CMDP, as opposed to WCSAC which is specialized for CVaR constraint***. We provide more results for CVaR-CMDP in appendix.

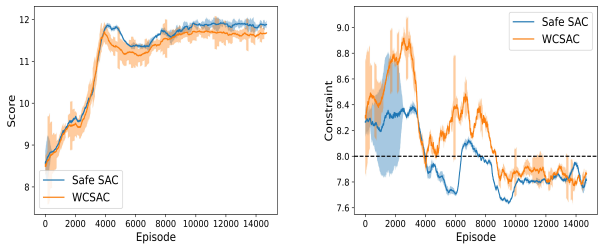

Figure 6: Experiment with CVaR Constraint in Merge Environment

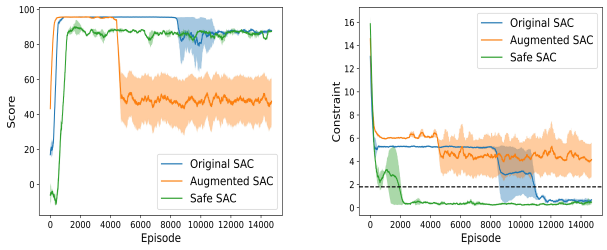

Figure 7: Ablation Analysis with GridWorld

5.3 Ablation Analysis

To investigate the impact of state augmentation and reward penalty, we conduct an ablation analysis using GridWorld and Highway environments. We compare the performance of Original SAC, SAC with only state augmentation (Augmented SAC), and SAC with state augmentation and reward penalty (Safe SAC) on GridWorld in Figure 7. In Augmented SAC, the agent chooses an action using (5). If no action could satisfy the constraint, it chooses the action with minimum future cost. More details and results are provided in appendix.

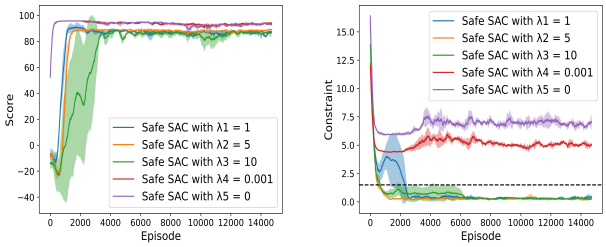

Figure 8: Experiment in GridWorld with Different Reward Penalties

5.4 Impact of Reward Penalty λ

To investigate the impact of different reward penalty values on the performance, we conduct experiments on GridWorld using Safe SAC, with $\lambda_1 = 1, \lambda_2 = 5\lambda_1, \lambda_3 = 10\lambda_1$, a small $\lambda_4 = 0.001$ and $\lambda_5 = 0$, we show the results in Figure 8. In this experiment, we fix λ so that it would be conservative in some cases. We can see that $\lambda_1, \lambda_2, \lambda_3$ in GridWorld could be good choices to achieve good performance, implying that only with an appropriate λ value, the agent can receive high rewards while satisfies the constraint with fast convergence speed. That is consistent with our theory and is why we set a threshold for λ and dynamically change it during the safe algorithms. Moreover, cases with a small λ are close to those without λ. We also provide the results on Highway environment in the appendix.

6 Conclusion

In this paper, we have provided a very generic and scalable mechanism for handling a wide variety of cost constraints. Lagrangian-based approaches, which penalize with respect to expected cost are unable to assign credit appropriately for a cost constraint violation, as expected cost averages over all trajectories. Instead, we propose to penalize with respect to individual trajectory while maintaining a cost-augmented state, thereby providing precise credit assignment with regard to cost constraint violations. We theoretically demonstrate that this reward penality approach on cost augmented states can enable faster convergence to safer areas (with regards to satisfying cost constraints) than Lagrangian-based approaches using lower costs. Furthermore, we also show that this approach is generalizable to a wide variety of cost constraints (worst case, expected, VaR, CVaR). We then provide Safe DQN and Safe SAC which are able to outperform leading expected-cost constrained RL approaches (FOCOPS, CUP, CPO, Lyapunov and BVF) while, at the same time (using the same approach), providing slightly better performance than the leading approach for CVaR constrained RL (WCSAC). While not extensively experimented here, we can also dynamically tune the the initial value and threshold of reward penalty for each environment to achieve better performance.

Acknowledgments

This research/project is supported by the National Research Foundation Singapore and DSO National Laboratories under the AI Singapore Programme (AISG Award No: AISG2-RP-2020-017)

References

Abe, N.; Melville, P.; Pendus, C.; Reddy, C. K.; Jensen, D. L.; Thomas, V. P.; Bennett, J. J.; Anderson, G. F.; Cooley, B. R.; Kowalczyk, M.; Domick, M.; and Gardinier, T. 2010. Optimizing Debt Collections Using Constrained Reinforcement Learning. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, 75–84. New York, NY, USA: Association for Computing Machinery. ISBN 9781450300551.

Achiam, J.; Held, D.; Tamar, A.; and Abbeel, P. 2017a. Constrained Policy Optimization. *CoRR*, abs/1705.10528.

Achiam, J.; Held, D.; Tamar, A.; and Abbeel, P. 2017b. Constrained policy optimization. In *International conference on machine learning*, 22–31. PMLR.

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

As, Y.; Usmanova, I.; Curi, S.; and Krause, A. 2022. Constrained policy optimization via bayesian world models. *arXiv preprint arXiv:2201.09802*.

Calvo-Fullana, M.; Paternain, S.; Chamon, L. F.; and Ribeiro, A. 2021. State augmented constrained reinforcement learning: Overcoming the limitations of learning with rewards. *arXiv preprint arXiv:2102.11941*.

Chen, C.; Karunasena, R.; Nguyen, T. H.; Sinha, A.; and Varakantham, P. 2023. Generative Modelling of Stochastic Actions with Arbitrary Constraints in Reinforcement Learning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Chow, Y.; Nachum, O.; Duenez-Guzman, E.; and Ghavamzadeh, M. 2018. A lyapunov-based approach to safe reinforcement learning. *Advances in neural information processing systems*, 31.

Chow, Y.; Nachum, O.; Faust, A.; Duenez-Guzman, E.; and Ghavamzadeh, M. 2019a. Lyapunov-based safe policy optimization for continuous control. *arXiv preprint arXiv:1901.10031*.

Chow, Y.; Nachum, O.; Faust, A.; Ghavamzadeh, M.; and Duéñez-Guzmán, E. A. 2019b. Lyapunov-based Safe Policy Optimization for Continuous Control. *CoRR*, abs/1901.10031.

Chow, Y.; Pavone, M.; Sadler, B. M.; and Carpin, S. 2015a. Trading Safety Versus Performance: Rapid Deployment of Robotic Swarms with Robust Performance Constraints. *CoRR*, abs/1511.06982.

Chow, Y.; Tamar, A.; Mannor, S.; and Pavone, M. 2015b. Risk-Sensitive and Robust Decision-Making: a CVaR Optimization Approach. *CoRR*, abs/1506.02188.

Di Castro, D.; Tamar, A.; and Mannor, S. 2012. Policy Gradients with Variance Related Risk Criteria.

Gábor, Z.; Kalmár, Z.; and Szepesvári, C. 1998. Multi-criteria reinforcement learning. In *ICML*, volume 98, 197–205.

Haarnoja, T.; Zhou, A.; Hartikainen, K.; Tucker, G.; Ha, S.; Tan, J.; Kumar, V.; Zhu, H.; Gupta, A.; Abbeel, P.; et al. 2018. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*.

Hogewind, Y.; Simao, T. D.; Kachman, T.; and Jansen, N. 2022. Safe reinforcement learning from pixels using a stochastic latent representation. *arXiv preprint arXiv:2210.01801*.

Jain, A.; Khetarpal, K.; and Precup, D. 2021. Safe option-critic: learning safety in the option-critic architecture. *The Knowledge Engineering Review*, 36.

Ji, J.; Zhang, B.; Pan, X.; Zhou, J.; Dai, J.; and Yang, Y. 2023. Safety-Gymnasium. *GitHub repository*.

Junges, S.; Jansen, N.; Dehnert, C.; Topcu, U.; and Katoen, J. 2015. Safety-Constrained Reinforcement Learning for MDPs. *CoRR*, abs/1510.05880.

Leike, J.; Martic, M.; Krakovna, V.; Ortega, P. A.; Everitt, T.; Lefrancq, A.; Orseau, L.; and Legg, S. 2017. AI safety gridworlds. *arXiv preprint arXiv:1711.09883*.

Leurent, E. 2018. An Environment for Autonomous Driving Decision-Making. https://github.com/eleurent/highway-env.

Levine, S.; Popovic, Z.; and Koltun, V. 2011. Nonlinear inverse reinforcement learning with gaussian processes. *Advances in neural information processing systems*, 24.

Ling, J.; Schuler, M. L.; Kumar, A.; and Varakantham, P. 2023. Knowledge Compilation for Constrained Combinatorial Action Spaces in Reinforcement Learning. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, 860–868.

Low, S. M.; Kumar, A.; and Sanner, S. 2023. Safe MDP Planning by Learning Temporal Patterns of Undesirable Trajectories and Averting Negative Side Effects. *arXiv preprint arXiv:2304.03081*.

Lucas, A.; and Klaassen, P. 1998. Extreme returns, downside risk, and optimal asset allocation. *Journal of Portfolio Management*, 25(1): 71.

Mastronarde, N.; and van der Schaar, M. 2010. Fast Reinforcement Learning for Energy-Efficient Wireless Communications. *CoRR*, abs/1009.5773.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *nature*, 518(7540): 529–533.

Moldovan, T. M.; and Abbeel, P. 2012. Safe Exploration in Markov Decision Processes. *CoRR*, abs/1205.4810.

Ono, M.; Pavone, M.; Kuwata, Y.; and Balaram, J. 2015. Chance-Constrained Dynamic Programming with Application to Risk-Aware Robotic Space Exploration. *Auton. Robots*, 39(4): 555–571.

Pankayaraj, P.; and Varakantham, P. 2023. Constrained reinforcement learning in hard exploration problems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 15055–15063.

Rockafellar, R. T.; Uryasev, S.; et al. 2000. Optimization of conditional value-at-risk. *Journal of risk*, 2: 21–42.

Satija, H.; Amortila, P.; and Pineau, J. 2020. Constrained markov decision processes via backward value functions. In *International Conference on Machine Learning*, 8502–8511. PMLR.

Sootla, A.; Cowen-Rivers, A.; Wang, J.; and Bou Ammar, H. 2022a. Enhancing safe exploration using safety state augmentation. *Advances in Neural Information Processing Systems*, 35: 34464–34477.

Sootla, A.; Cowen-Rivers, A. I.; Jafferjee, T.; Wang, Z.; Mguni, D. H.; Wang, J.; and Ammar, H. 2022b. Sauté rl: Almost surely safe reinforcement learning using state augmentation. In *International Conference on Machine Learning*, 20423–20443. PMLR.

Yang, L.; Ji, J.; Dai, J.; Zhang, L.; Zhou, B.; Li, P.; Yang, Y.; and Pan, G. 2022. Constrained update projection approach to safe policy optimization. *Advances in Neural Information Processing Systems*, 35: 9111–9124.

Yang, Q.; Simão, T. D.; Tindemans, S. H.; and Spaan, M. T. 2021. WCSAC: Worst-Case Soft Actor Critic for Safety-Constrained Reinforcement Learning. In *AAAI*, 10639–10646.

Zhang, Y.; Vuong, Q.; and Ross, K. 2020. First order constrained optimization in policy space. *Advances in Neural Information Processing Systems*, 33: 15338–15349.