Constrained Upper Confidence Reinforcement Learning with Known Dynamics

Liyuan Zheng LIYUANZ8@UW.EDU
Department of Electrical & Computer Engineering, University of Washington

Lillian J. Ratliff RATLIFFL@UW.EDU
Department of Electrical & Computer Engineering, University of Washington

Editors: A. Bayen, A. Jadbabaie, G. J. Pappas, P. Parrilo, B. Recht, C. Tomlin, M. Zeilinger

Abstract

Constrained Markov Decision Processes are a class of stochastic decision problems in which the decision maker must select a policy that satisfies auxiliary cost constraints. This paper extends upper confidence reinforcement learning for settings in which the reward function and the constraints, described by cost functions, are unknown a priori but the transition kernel is known. Such a setting is well-motivated by a number of applications including exploration of unknown, potentially unsafe, environments. The algorithm, `C-UCRL`, is shown to have sub-linear regret ($O(T^{\frac{3}{4}}\sqrt{\log(T/\delta)})$) with respect to the reward while satisfying the constraints even while learning with probability $1-\delta$. An illustrative example is provided.

Keywords: constrained Markov decision process, upper confidence reinforcement learning, regret, online learning

1. Introduction

Markov Decision Processes (MDPs) have been successfully utilized to model sequential decision-making problems in stochastic environments. In the typical approach to learning a policy, the decision-maker trades off between exploration and exploitation, gradually improving their performance at the task as learning progresses. Reinforcement learning, a standard paradigm of learning in MDPs, has shown exceptional success in a variety of domains such as video games (Mnih et al., 2015), robotics (Lillicrap et al., 2015; Levine et al., 2016), recommender systems (Shani et al., 2005), and autonomous vehicles (Sallab et al., 2017). Yet, in many of these real-world applications there is additional constraints, or specifications that lead to constraints, on the learning problem.

For instance, a recommender system should avoid presenting offending items to users and autonomous vehicles must avoid crashing into others while navigating (Garcıa and Fernández, 2015). Building algorithms that respect safety constraints not only during normal operation, but also during the initial learning period, is a question of particular interest (Leike et al., 2017). This problem is known as the *safe exploration problem* (Moldovan and Abbeel, 2012; Amodei et al., 2016). In the standard MDP framework, an approach for baseline performance is risk-sensitive reinforcement learning (Coraluppi and Marcus, 1999; Garcıa and Fernández, 2015), where the optimization criterion is transformed in order to reflect a subjective measure balancing the return and the risk. On the other hand, in a safety-critical environment, it is more reasonable to separate the return and the risk criterion, and enforce constraint satisfaction in the learning procedure. A standard formulation for an environment with safety constraints is the constrained MDPs (CMDPs) (Altman, 1999). A

decision-maker facing a CMDP aims to maximize the total reward while satisfying the constraints on costs in expectation over the whole trajectory.

In recent literature, policy gradient-based reinforcement learning algorithms have been proposed as a means to learn a policy for a CMDP. The following are two constrained policy search algorithms with state-of-the-art performance guarantees: Lagrangian-based actor-critic algorithm (Bhatnagar and Lakshmanan, 2012; Chow et al., 2018a) and Constrained Policy Optimization (CPO) (Achiam et al., 2017). However, for these policy gradient-based methods, safety is only approximately guaranteed *after* a sufficient learning period. The fundamental issue is that without a model, safety must be learned via trial and error, which means it may be violated during initial learning interactions.

Model-based approaches have utilized Gaussian processes to model the state safety values or the dynamic uncertainties (Berkenkamp et al., 2017; Koller et al., 2018; Wachi et al., 2018; Cheng et al., 2019) or utilized Lyapunov-based methods (Chow et al., 2018b) to guarantee safety during learning. Although these methods guarantee constraint satisfaction during learning, an arguably valuable analysis of the regret is lacking. In unconstrained settings when the reward and transition kernel are unknown, upper confidence based reinforcement learning algorithms have been proposed—namely, `UCRL2` (Jaksch et al., 2010)—with sub-linear regret. The key idea is to build confidence intervals on the reward and transition kernel and iteratively solve for policies using value iteration.

In this work, we are not only interested in learning the optimal policy that satisfies the constraints via interacting with the stochastic environment, but also in ensuring performance guarantees on the learning algorithm during learning. With some practical scenarios in mind, we make the assumption that the rewards and constraint costs are unknown. For instance, consider a robot navigation task; here may have an approximate model the dynamics of the robot (known with some uncertainty) and the reward and constraints which model the value of exploring the environment as unknown—e.g., constraints can be abstracted as costs which seek to limit the frequency of visiting a potentially hazardous states (El Chamie et al., 2019).

Motivated by upper confidence reinforcement learning (Jaksch et al., 2010), we introduce the constrained upper confidence reinforcement learning (`C-UCRL`) algorithm which combines elements of the classical `UCRL2` algorithm with robust linear programming[1]. We define our goals as follows: (1) maintain constraint satisfaction throughout the learning process with high probability, and (2) achieve sub-linear regret comparing the rewards collected by the algorithm during learning with the reward of an optimal stochastic policy.

Contributions. The contributions can be summarized as follows. Building on `UCRL2`, we introduce the `C-UCRL` algorithm (Algorithm 1). We show that `C-UCRL` is guaranteed to satisfy constraints during learning with probability at least $1-\delta$ (Theorem 4) and achieves $O(T^{\frac{3}{4}}\sqrt{\log(T/\delta)})$ reward regret (Theorem 8). Of independent interest, we note that when the state space is trivial, the setting we consider subsumes stochastic multi-armed bandits with per-round budget constraints, where the optimal policy is a randomized policy across arms[2].

2. Constrained Upper Confidence Reinforcement Learning Algorithm

An MDP is a tuple $(\mathcal{S}, \mathcal{A}, P, r)$, where $\mathcal{S}$ is the set of states, $\mathcal{A}$ is the set of actions, $P : \mathcal{S}\times\mathcal{A}\times\mathcal{A} \to [0,1]$ is the transition kernel such that $P(s'|s,a)$ is the probability of transitioning to state s' given

1. `UCRL2` assumes the transition kernel is unknown; we assume it is known and leave the extension to future work.
2. See the extended version (Zheng and Ratliff, 2020) for further discussion on the multi-armed bandit setting and related works.

that the previous state was s and the agent took action a in s, and $r : \mathcal{S} \times \mathcal{A} \to [0, 1]$ is the reward function. A stationary policy $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$ is a map from states to a probability distribution over actions, with $\pi(a|s)$ denoting the probability of selecting action a in state s. We consider the setting in which the transition kernel $P(s'|s, a)$ is known to the agent, but the reward and costs are stochastic and unknown. In the example of robot navigation, the agent (robot) is aware of the transition probability of next state based on its action, but the *safety quality* of each state is unknown. Let $S = |\mathcal{S}|$ and $A = |\mathcal{A}|$ where $|\cdot|$ is the cardinality of its argument. We use the notation $[\cdot] = \{1, \dots, \cdot\}$ for index sets.

A CMDP is an MDP augmented with 'cost' constraints that restrict the set of allowable policies for that MDP. For a given CMDP, we consider the performance measure to be the *infinite horizon average reward* which is given by

$$J(\pi) = \lim_{T\to\infty} \mathbb{E}_{\tau\sim\pi}\left[\frac{1}{T}\sum_{t=0}^{T-1} r(s_t, a_t)\right] \tag{1}$$

where τ denotes a trajectory $\tau = (s_0, a_0, s_1, \dots)$, and $\tau \sim \pi$ is shorthand for indicating that the distribution over trajectories depends on π: $s_0 \sim p(s_0), a_t \sim \pi(\cdot|s_t), s_{t+1} \sim P(\cdot|s_t, a_t)$. Similarly, define the *average constraint costs* by

$$C_i(\pi) = \lim_{T\to\infty} \mathbb{E}_{\tau\sim\pi}\left[\frac{1}{T}\sum_{t=0}^{T-1} c_i(s_t, a_t)\right]. \tag{2}$$

where $\{c_1, \dots, c_m\}$ with $c_i : \mathcal{S} \times \mathcal{A} \to [0, 1]$ are the cost constraints. The CMDP is then defined by

$$\max_\pi \{J(\pi)|\ C_i(\pi) \le d_i,\ \ \forall\, i \in [m]\} \tag{3}$$

where $\{d_1, \dots, d_m\}$ are upper bounds on the average constraint costs. Note that without loss of generality both the reward and costs are random variables with a distribution supported on $[0, 1]$.

Denote the mean of reward and cost constraint functions as $\bar{r}(s, a) = \mathbb{E}[r(s, a)]$, $\bar{c}_i = \mathbb{E}[c_i(s, a)]$ where the expectation is taken with respect to the distribution of the reward and cost function of that state-action pair (s, a). If the transition kernel $P(s'|s, a)$, the mean of the reward function $\bar{r}(s, a)$, and mean cost functions $\bar{c}_i(s, a)$ are all given, them we can solve the CMDP by solving the following linear program in matrix form (Altman, 1999):

$$\max_y\{\, \bar{r}^\top y \mid I_o y = Py,\ \mathbf{1}^\top y = 1,\ y \ge 0,\ \bar{c}^\top y \le d\} \tag{4}$$

where $\bar{r} \in \mathbb{R}^{SA}, y \in \mathbb{R}^{SA}, \bar{c} \in \mathbb{R}^{SA\times m}, d \in \mathbb{R}^m$, $P \in \mathbb{R}^{S\times SA}$, and $I_o \in \mathbb{R}^{S\times SA}$ is a sparse matrix built by placing S row blocks of length A in a block diagonal fashion, where each row block consists of all ones. Here, $y \in \mathbb{R}^{S\times A}$ represents the steady-state occupation measure (Altman, 1999) defined by

$$y(s, a) = \lim_{T\to\infty} \mathbb{E}_{\tau\sim\pi}\left[\frac{1}{T}\sum_{t=0}^{T-1} \mathbf{1}\{s_t = s, a_t = a\}\right]. \tag{5}$$

With $\bar{y}$ the solution of this linear program, the optimal stationary policy is

$$\bar{\pi}(a|s) = \bar{y}(s, a)/(\textstyle\sum_{a\in\mathcal{A}} \bar{y}(s, a)). \tag{6}$$

Remark 1 *It is worth noting that unlike in tabular MDPs without constraints, where the optimal policy is always deterministic, the optimal policy in CMDPs could be stochastic (Puterman, 2014). It is, in fact, trivial to solve the CMDP if the optimal policy in CMDPs is deterministic because that means the constraints are not active.*

Since the reward and constraint cost functions are unknown, motivated by UCRL2, we introduce C−UCRL (Algorithm 1). In general, the C−UCRL algorithm follows a principle of "optimism in the face of reward uncertainty; pessimism in the face of cost uncertainty." That is, it defines confidence intervals for the reward and cost of each state-action pair given the observations so far, and solves for the optimistic policy that satisfies the constraints. More specifically, in C−UCRL, given the current confidence interval estimates, we use a robust linear program (Luenberger et al., 1984) formulation to find a policy using the confidence intervals as determined at the current iteration. In particular, in episode k, we define estimates of the reward and costs by $\hat{r}_k(s,a) = \frac{R_k(s,a)}{\max\{1,N_k(s,a)\}}$ and $\hat{c}_{i,k}(s,a) = \frac{C_{i,k}(s,a)}{\max\{1,N_k(s,a)\}}$, respectively, where $N_k(s,a)$, $R_k(s,a)$, and $C_{i,k}(s,a)$ are the state-action count, and cumulative reward and costs, respectively, as defined in Algorithm 1. Using these estimates, we define the following:

$$\tilde{r}_k(s,a) = \min\left\{\hat{r}_k(s,a) + \left(\frac{\log(SA(m+1)\pi^2 t_k^3/3\delta)}{2\max\{1,N_k(s,a)\}}\right)^{1/2}, 1\right\} \tag{7}$$

$$\tilde{c}_{i,k}(s,a) = \min\left\{\hat{c}_{i,k}(s,a) + \left(\frac{\log(SA(m+1)\pi^2 t_k^3/3\delta)}{2\max\{1,N_k(s,a)\}}\right)^{1/2}, 1\right\}, \tag{8}$$

Using the above, we solve the following robust linear program:

$$\max_y\{\ \tilde{r}_k^\top y \mid I_o y = P y,\ \mathbf{1}^\top y = 1,\ y \geq 0,\ \tilde{c}_k^\top y \leq d\} \tag{RLP}$$

As in (7) and (8), the confidence intervals of reward and cost are positively correlated with the cardinality of the CMDP, time steps, and negatively correlated with the visit times of that state-action pair. To trade-off between exploration and exploitation, we assume we are given a safe baseline policy as a input of C−UCRL. It is common to assume a initial safe policy before learning procedure (Achiam et al., 2017) and assume under such policy, the Markov chain resulting from the CMDP is irreducible and aperiodic (Bhatnagar et al., 2009). This baseline policy could be obtained by some prior information about which states are safe to start the conservative exploration. We execute the baseline policy in the early learning stage to build estimation of reward and cost by (7) and (8) until there is a feasible solution to (RLP)[3]. Using the solution to (RLP), namely $\tilde{y}_k$. we recover the policy $\tilde{\pi}_k$ via (6). Thus, for each episode of C−UCRL, we execute the baseline policy for h steps, estimate the reward and costs, and then execute $\tilde{\pi}_k$ for a linearly increasing (in the number of epochs) number of steps $(k-1)h$, making kh the total duration of episode k.

3. Analysis

In this section, we show that C−UCRL has guarantees on constraint satisfaction during learning. Then, we provide regret analysis with respect to the reward, showing that the regret is sub-linear. To capture constraint satisfaction, we leverage the notion of δ-safety.

Definition 2 (δ-safe) *An algorithm is δ-safe if, with probability at least $1-\delta$, for all time steps t, the policy executed by the algorithm satisfies $C_i(\pi_t) \leq d_i$, $\forall i \in [m]$.*

3. The heuristic for choosing h to have 'sufficient' exploration is based on the mixing time of the Markov chain induced by π_0 given the known transition kernel for the CMDP. See (Zheng and Ratliff, 2020) for more discussion.

Algorithm 1 Constrained UCRL (C−UCRL) algorithm

Input: safety parameter $\delta \in (0,1)$, baseline policy $\pi_0(a|s)$, episode length h.
Initialization: set $t=1$, observe the initial state s_1
for episodes $k = 1, 2, \ldots, K$ **do**
  $t_k = t$; // initialize start time of episode k
  while $t \leq t_k + h$; // Execute baseline policy h times for exploration
  do
    Draw action $a_t \sim \pi_0(\cdot|s_t)$
    Observe reward r_t, costs $c_{i,t}$, and the next state s_{t+1}
    $t \leftarrow t+1$
  end
  $N_k(s,a) = \sum_{t'=1}^{t} \mathbf{1}(s_{t'}=a, a_{t'}=a),\ \forall (s,a) \in \mathcal{S}\times\mathcal{A}$; // set the state-action count
  $R_k(s,a) = \sum_{t'=1}^{t} r_{t'}\mathbf{1}(s_{t'}=a, a_{t'}=a)$; // compute cumulative reward
  $C_{i,k}(s,a) = \sum_{t'=1}^{t} c_{i,t'}\mathbf{1}(s_{t'}=a, a_{t'}=a)$; // compute the cumulative costs
  $\hat{r}_k(s,a) = \frac{R_k(s,a)}{\max\{1,N_k(s,a)\}}, \quad \hat{c}_{i,k}(s,a) = \frac{C_{i,k}(s,a)}{\max\{1,N_k(s,a)\}}$; // compute estimates
  $\tilde{y}_k \leftarrow$ arg max of (RLP) using $\tilde{r}_k(s,a)$ and $\tilde{c}_{i,k}(s,a)$ in (7) and (8), resp.
  $\tilde{\pi}_k \leftarrow \tilde{y}_k(s,a)/(\sum_{a\in\mathcal{A}} \tilde{y}_k(s,a))$; // recover policy
  while $t \leq t_k + kh$; // Execute $\tilde{\pi}_k$ policy $(k-1)h$ times
  do
    Draw action $a_t \sim \tilde{\pi}_k(\cdot|s_t)$
    Observe reward r_t, costs $c_{i,t}$, and the next state s_{t+1}
    $t \leftarrow t+1$
  end
end

Lemma 3 *With probability at least $1-\delta$, for every state-action pair (s,a), cost c_i and episode k,* C−UCRL *satisfies the following:*

$$|\hat{r}_k(s,a) - \bar{r}(s,a)| \leq \left(\frac{\log(SA(m+1)\pi^2 t_k^3/3\delta)}{2\max\{1,N_k(s,a)\}}\right)^{1/2}, \tag{9}$$

$$|\hat{c}_{i,k}(s,a) - \bar{c}_i(s,a)| \leq \left(\frac{\log(SA(m+1)\pi^2 t_k^3/3\delta)}{2\max\{1,N_k(s,a)\}}\right)^{1/2} \tag{10}$$

The proof follows by applying Hoeffding's inequality and finding union bound over all states, actions and episodes. For the full proof, please see Section 4, Lemma 2 in (Zheng and Ratliff, 2020).

Given that, for each episode, we can bound the gaps between the estimated reward (respectively, costs) and the mean reward (respectively, mean costs), with probability $1-\delta$, we can provide an assurance on C−UCRL being δ-safe.

Theorem 4 C−UCRL *is δ-safe.*

Proof According to Lemma 3, with probability at least $1-\delta$, $\bar{c}_i(s,a) \leq \tilde{c}_{i,k}(s,a)$. The occupation measure $\tilde{y}_k$ obtained at each episode via (RLP) satisfies $\sum_{s,a} \tilde{c}_{i,k}(s,a)\tilde{y}_k(s,a) \leq d_i$. Hence, $C_i(\tilde{\pi}_k) = \sum_{s,a} \bar{c}_i(s,a)\tilde{y}_k(s,a) \leq d_i$ with probability $1-\delta$. ■

Given that we have shown that C−UCRL is δ-safe, we now analyze the reward regret. In episode k of C−UCRL, we execute a baseline policy π_0 for h times and policy $\tilde{\pi}_k$ for $(k-1)h$ times. The

pseudo-regret of episode k is given by

$$\Delta_k = h[J(\bar{\pi}) - J(\pi_0)] + (k-1)h[J(\bar{\pi}) - J(\tilde{\pi}_k)] = h\bar{r}^\top(\bar{y} - y_0) + (k-1)h\bar{r}^\top(\bar{y} - \tilde{y}_k).$$

We first upper bound the per-step pseudo-regret of executing policy $\tilde{\pi}_k$, $\bar{r}^\top(\bar{y} - \tilde{y}_k)$, where the first term is the expected average reward under the optimal policy $\bar{\pi}$ and the second term is the sub-optimal expected average reward under policy $\tilde{\pi}_k$.

Define the following two linear programs:

$$\max_y\{r^\top y | Ay = 0, \mathbf{1}^\top y = 1, y \geq 0, c^\top y \leq d\} \tag{11}$$

$$\max_y\{(r+\epsilon_r)^\top y | Ay = 0, \mathbf{1}^\top y = 1, y \geq 0, (c+\epsilon_c)^\top y \leq d\}. \tag{12}$$

where $0 \leq r \leq \mathbf{1}$, $0 \leq c \leq \mathbf{1}$, $\epsilon_r \geq 0$, and $\epsilon_c \geq 0$ hold element wise.

Lemma 5 *Assuming the domains of* (11) *and* (12) *are not empty, let y_1 and y_2 be solutions for each of the problems, respectively. If, for some constant $\alpha > 0$ and $\beta > 0$, there exist $y_0 \in \{y | Ay = 0, \mathbf{1}^\top y = 1, y \geq 0, (c+\epsilon_c)^\top y \leq d\}$ such that $r^\top(y_1 - y_0) = \alpha > 0$ and $c^\top(y_1 - y_0) = \beta > 0$, then $r^\top(y_1 - y_2) \leq \frac{2\alpha}{\beta}\|\epsilon_c\|_1 + \|\epsilon_r\|_1$.*

The intuition of the proof is to define y_3 as the solution to $\max_y\{r^\top y | Ay = 0, \mathbf{1}^\top y = 1, y \geq 0, (c+\epsilon_c)^\top y \leq d\}$. We first find the upper bound of $r^\top(y_1 - y_3)$ and then find the bound of $r^\top(y_3 - y_2)$. By summing the absolute value of these two bounds, we have the bound of $r^\top(y_1 - y_2)$. For the full proof, please see Section 4, Lemma 4 in (Zheng and Ratliff, 2020).

We can use Lemma 5 to get a bound on the pseudo-regret.

Proposition 6 *Denote $\mathcal{Y} = \{y | (I_o - P)y = 0, \mathbf{1}^\top y = 1, y \geq 0\}$. If there exists $y_0 \in \mathcal{Y}$ such that $\bar{r}^\top(\bar{y} - y_0) = \alpha > 0, \bar{c}^\top(\bar{y} - y_0) = \beta > 0$, then with probability at least $1 - \delta$,*

$$\bar{r}^\top(\bar{y} - \tilde{y}_k) \leq 2\left(\tfrac{2\alpha m}{\beta} + 1\right)\textstyle\sum_{s,a}\left(\frac{\log(SA(m+1)\pi^2 t_k^3/3\delta)}{2\max\{1, N_k(s,a)\}}\right)^{1/2}. \tag{13}$$

The proof is constructed by constructing a sequence of subproblems in which one constraint is adding at a time. Lemma 5 provides the bound of two solutions to two LPs, with one constraint difference. Here, we construct a sequence of subproblems and each two subproblems differ from one constraint. Applying Lemma 5, we have bounds for each of the two solution pairs. Summing the respective bounds, we have the bound of $\bar{r}^\top(\bar{y} - \tilde{y}_k)$. For the full proof, please see Section 4, Proposition 5 in (Zheng and Ratliff, 2020).

Note that according to Proposition 6, with probability at least $1 - \delta$, the per-step pseudo-regret of executing policy $\tilde{\pi}_k$ depends on the confidence intervals of reward and costs of all state-action pairs. This is intuitive since in order for the policy $\tilde{\pi}_k$ to be close to the optimal policy $\bar{\pi}$, we need to have good approximations of the reward and costs for all state-action pairs. To ensure this, we need to constantly explore the CMDP so that $N_k(s,a)$ is not 'too small' for any state-action pair. Since the Markov chain resulting from the baseline policy is irreducible and aperiodic, the steady state occupation measure $y_0(s,a)$ corresponding to the baseline policy $\pi_0(a|s)$ has the property that $y_0(s,a) > 0, \forall s, a$. Due to this universal exploration demand, we execute the baseline policy π_0 for a constant number of times in each linear increasing episode in the `C-UCRL` algorithm.

To have a upper bound on the regret derived in Proposition 6, we need to have a lower bounds on $N_k(s,a)$. Given our assumptions on the baseline policy as discussed above, define $\rho > 0$ such that $y_0(s,a) \geq \rho > 0$ for all state-action pairs $(s,a) \in \mathcal{S} \times \mathcal{A}$. The following lemma gives a lower bound on the number of times each state-action pair is visited in episode k.

Lemma 7 *With probability at least* $1-\delta$*, for every state-action pair* (s,a) *and episode* k*,*

$$N_k(s,a) \geq (k-1)\rho h - (k-1)\left(72\xi\rho h \log\left(\tfrac{\varphi \cdot SAK}{\delta}\right)\right)^{1/2} \tag{14}$$

where ξ *the the mixing time of the Markov chain induced by policy* π_0*,* $\rho > 0$ *is such that* $y_0(s,a) \geq \rho > 0$ *for all state-action pairs* $(s,a) \in \mathcal{S} \times \mathcal{A}$*, and* $\varphi = \sum_{s,a} \frac{y'(s,a)^2}{y_0(s,a)}$*, where* y' *is the initial state action distribution and* y_o *is the steady state action distribution under the baseline policy.*

The proof follows by apply union bound over states, actions, and episodes using the inequality in (Chung et al., 2012, Theorem 3). For a more detailed proof, please see Section 4, Lemma 6 in (Zheng and Ratliff, 2020).

Combining Proposition 6 and Lemma 7 and summing over K episodes, we obtain the total regret bound for C-UCRL. For a detailed proof of the theorem below, please see Section 4, Theorem 7 in (Zheng and Ratliff, 2020).

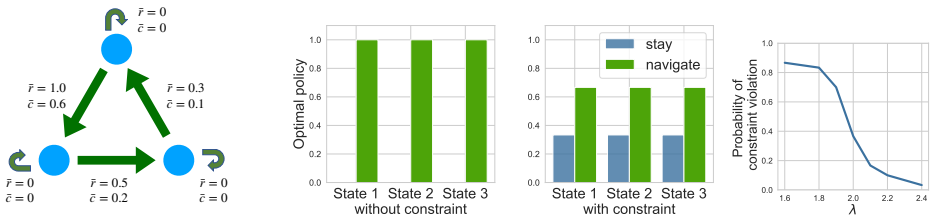
Theorem 8 *Suppose that* $\delta \leq \varphi SAK \exp(-\frac{\rho h}{288\xi})$*. Under the assumptions of Proposition 6, with probability at least* $1-\delta$*,* C-UCRL *has total pseudo-regret* $\Delta(T) = O(T^{\frac{3}{4}}\sqrt{\log(T/\delta)})$*.*

4. Experiments

To demonstrate the performance of C-UCRL, we consider a simple three state CMDP.[4] As show in Figure 1(*a*), the CMDP we consider has three states and two actions. An agent can take either a *risky* exploratory action in which they navigate to another state or they can take the *safe* action and remain in the current state. There is no reward or cost for staying in the current state but there is a stochastic reward and cost for moving. The reward and cost of each state-action pair are each draw from a binomial distribution, with the means defined in the labels on edges in Figure 1(*a*). Unconstrained, the optimal policy is to continually navigate between all states. We introduce a constraint such that in expectation the average cost should be less than 0.2. This constraint prevents the agents from continuously navigating between all states. As shown in Figure 1(*b*), the constrained optimal policy is a randomized policy that has positive probability on the safe action in each state. The relatively conservative baseline policy we use in C-UCRL for exploration is staying in the current state and navigating to the next state with probability 0.8 and 0.2, respectively.

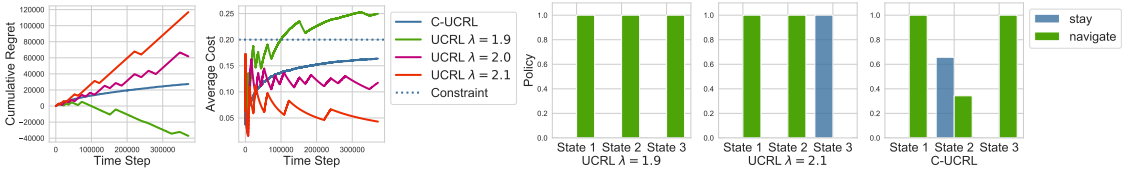
We compare our approach with the UCRL2 algorithm. However, UCRL2 does not allow for constraints or multiple reward/cost criteria. Hence, we leverage the idea of risk sensitive reinforcement learning (Coraluppi and Marcus, 1999; Leike et al., 2017), where we treat a linear combination of reward and cost—i.e., $r - \lambda c$—as the reward for the UCRL2 algorithm. The hyperparameter λ represents the trade off between the reward and cost, the combination of which represents the reward in the classical implementation of UCRL2; we refer to risk-sensitive UCRL2 by RS-UCRL2. Figure 1(*c*) shows the constraint violation probability in 30 training episodes by RS-UCRL2 algorithm with different λ. Figure 2(*a*) shows the cumulative regret and average cost of the C-UCRL and RS-UCRL2 algorithms. As we can see, when the cost value is underestimated ($\lambda = 1.9$), applying RS-UCRL2 directly leads to a 'good' reward (i.e., the regret is negative as it gets more reward than the optimal randomized policy), yet the constraints are violated. On the other hand, when the costs are overestimated ($\lambda = 2.1$), RS-UCRL2 is too conservative about the cost and, thus, receives high

4. Additional examples can be found in (Zheng and Ratliff, 2020).



(*a*) (*b*) (*c*)

Figure 1: Simple CMDP. (a) CMDP structure; (b) optimal policy computed with the true mean reward and mean cost, with and without the constraint on cost, $d = 0.2$; (c) probability of constraint violation in 30 training episodes by risk-sensitive UCRL2 (RS-UCRL2).



(*a*) (*b*)

Figure 2: C-UCRL vs. RS-UCRL2: (a) Cumulative regret and average cost for C-UCRL and risk sensitive UCRL2; (b) Policy learned by C-UCRL and RS-UCRL2.

regret. We can observe that C-UCRL does not violate the constraint during learning though in this experiment, δ is set to be 0.1, meaning that with probability at least 0.9, the constraint will not be violated in all episodes.

The fundamental problem with RS-UCRL2 is that with only one criterion, the policy it learns will always be a deterministic policy, while in this CMDP, the optimal policy is randomized. Figure 2(*b*) shows the policy learned by C-UCRL and RS-UCRL2. When $\lambda = 1.9$, RS-UCRL2 learn the optimal policy as there is no constraint, which leads to constraint violation. When $\lambda = 2.1$, the policy learned by RS-UCRL2 is to stay in one state forever. On the contrary, the policy learned by C-UCRL algorithm converges to the optimal randomized policy.

5. Conclusion

We formulate the problem of safe reinforcement learning when the transition kernel is known but the reward and constraint costs are unknown a priori as a CMDP and propose a C-UCRL algorithm to learn the optimal policy. Theoretically, we show that C-UCRL is guaranteed to satisfy the constraints during learning with probability at least $1-\delta$ and achieves $O(T^{\frac{3}{4}}\sqrt{\log(T/\delta)})$ reward regret. Empirically, we provide examples which demonstrate two key properties relative to comparable algorithms: 1) C-UCRL is able to learn the optimal policy which in general is a randomized policy as opposed to a deterministic policy, and 2) C-UCRL has high-probability guarantees on remaining safe while learning.

References

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. *arXiv preprint arXiv:1705.10528*, 2017.

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

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

Felix Berkenkamp, Matteo Turchetta, Angela Schoellig, and Andreas Krause. Safe model-based reinforcement learning with stability guarantees. In *Advances in Neural Information Processing Systems*, pages 908–918, 2017.

Shalabh Bhatnagar and K Lakshmanan. An online actor–critic algorithm with function approximation for constrained markov decision processes. *Journal of Optimization Theory and Applications*, 153(3):688–708, 2012.

Shalabh Bhatnagar, Richard S Sutton, Mohammad Ghavamzadeh, and Mark Lee. Natural actor–critic algorithms. *Automatica*, 45(11):2471–2482, 2009.

Richard Cheng, Gábor Orosz, Richard M Murray, and Joel W Burdick. End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks. *arXiv preprint arXiv:1903.08792*, 2019.

Yinlam Chow, Mohammad Ghavamzadeh, Lucas Janson, and Marco Pavone. Risk-constrained reinforcement learning with percentile risk criteria. *Journal of Machine Learning Research*, 18 (167):1–51, 2018a.

Yinlam Chow, Ofir Nachum, Edgar Duenez-Guzman, and Mohammad Ghavamzadeh. A lyapunov-based approach to safe reinforcement learning. *arXiv preprint arXiv:1805.07708*, 2018b.

Kai-Min Chung, Henry Lam, Zhenming Liu, and Michael Mitzenmacher. Chernoff-hoeffding bounds for markov chains: Generalized and simplified. *arXiv preprint arXiv:1201.0559*, 2012.

Stefano P Coraluppi and Steven I Marcus. Risk-sensitive and minimax control of discrete-time, finite-state markov decision processes. *Automatica*, 35(2):301–309, 1999.

Mahmoud El Chamie, Yue Yu, Behçet Açıkmeşe, and Masahiro Ono. Controlled markov processes with safety state constraints. *IEEE Transactions on Automatic Control*, 64(3):1003–1018, 2019.

Javier Garcıa and Fernando Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(1):1437–1480, 2015.

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

Torsten Koller, Felix Berkenkamp, Matteo Turchetta, and Andreas Krause. Learning-based model predictive control for safe exploration. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 6059–6066. IEEE, 2018.

Jan Leike, Miljan Martic, Victoria Krakovna, Pedro A Ortega, Tom Everitt, Andrew Lefrancq, Laurent Orseau, and Shane Legg. Ai safety gridworlds. *arXiv preprint arXiv:1711.09883*, 2017.

Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

David G Luenberger, Yinyu Ye, et al. *Linear and nonlinear programming*, volume 2. Springer, 1984.

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

Teodor Mihai Moldovan and Pieter Abbeel. Safe exploration in markov decision processes. *arXiv preprint arXiv:1205.4810*, 2012.

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

Ahmad EL Sallab, Mohammed Abdou, Etienne Perot, and Senthil Yogamani. Deep reinforcement learning framework for autonomous driving. *Electronic Imaging*, 2017(19):70–76, 2017.

Guy Shani, David Heckerman, and Ronen I Brafman. An mdp-based recommender system. *Journal of Machine Learning Research*, 6(Sep):1265–1295, 2005.

Akifumi Wachi, Yanan Sui, Yisong Yue, and Masahiro Ono. Safe exploration and optimization of constrained mdps using gaussian processes. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Liyuan Zheng and Lillian J Ratliff. Constrained upper confidence reinforcement learning. *arXiv preprint arXiv:2001.09377*, 2020.