# Robust Constrained MDPs

**Arushi Jain**[*]
McGill University & Mila
arushi.jain@mail.mcgill.ca

**Sharan Vaswani**[*]
Simon Fraser University

**Reza Babanezhad**
SAIT AI lab, Montreal

**Doina Precup**
McGill University & Mila

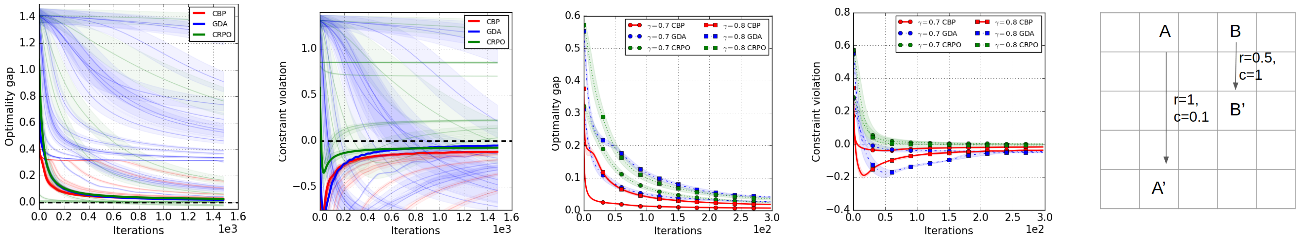
**Csaba Szepesvari**
Amii, University of Alberta

## Abstract

In many safety-critical applications, e.g., robotics, finance, autonomous driving, agents must subjected satisfy certain constraints on a cost function. The Constrained Markov decision process (CMDPs) (Altman, 1999) is a natural framework for modelling such constraints. The typical objective for CMDP is to maximize a cumulative function of the reward (like in unconstrained MDPs), while (approximately) satisfying the constraints. In this paper, we study incremental learning and planning with linear function approximation in infinite-horizon, discounted constrained Markov decision process (CMDP). We propose a generic primal-dual optimization framework, which allows us to bound the sub-optimality gap and constraint violation in terms of the primal and dual regret for arbitrary algorithms. We instantiate this framework in a way which allows us to use coin-betting algorithms from online linear optimization to control both the primal and dual regret. We call the resulting algorithm **Coin Betting Politex (CBP)**. Assuming that the action-value functions are $\varepsilon$-close to the span of $d$-dimensional state-action features and $\gamma$ is the discount factor, $T$ iterations of CBP result in an $O\left(\frac{1}{(1-\gamma)^3\sqrt{T}} + \frac{\varepsilon\sqrt{d}}{1-\gamma}\right)$ bound on the reward sub-optimality and a constraint violation of $O\left(\frac{1}{(1-\gamma)^2\sqrt{T}} + \varepsilon\sqrt{d}\right)$. Unlike gradient descent-ascent and primal-only methods, our proposed *CBP is robust to the choice of hyper-parameters*. We empirically demonstrate the superior performance and robustness of CBP in both tabular and linear function approximation setting, on both gridworld environments and OpenAI gym tasks.

**Keywords:** reinforcement learning, constrained MDPs, primal-dual framework, coin-betting algorithm

[*] Equal authorship



(a) Hyperparameter sensitivity OG

(b) Hyperparameter sensitivity CV

(c) Environment misspecification OG

(d) Environment misspecification CV

(e) Gridworld environment

Figure 1: **Performance in model-based setting:** Performance metric – *Optimality gap (OG)* and *constraint violation (CV)* – of our proposed algorithm, CBP, and baselines GDA, CRPO in a gridworld environment (Fig. 1e) with access to the true CMDP model. Ideally OG and CV should converge to 0. Figs. 1a and 1b show the performance *sensitivity to hyperparameters*. The dark lines corresponds to the performance with the best hyperparameters. The lighter color lines show performance with other hyperparameter values. Figs. 1c and 1d shows the effect of *environment mis-specification* by varying the discount factor $\gamma = \{0.7, 0.8\}$ and keeping the best hyperparameters obtained from the left two figures (obtained in the original CMDP with $\gamma = 0.9$). GDA and CRPO, exhibit huge variance in performance with different hyperparameters, while CBP is quite *robust to variations in hyperparameters and environment mis-specification*.

# 1 Introduction

In many safety-critical applications, e.g., robotics, finance, autonomous driving, agents must satisfy certain constraints in addition to optimizing long-term returns. This objective can be formalized in a natural way through the framework of constrained Markov decision process (CMDP) (Altman, 1999).

In this paper, we focus on the problem of finding an approximately feasible policy (i.e. a policy which is allowed to violate constraints by a small amount), while (approximately) maximizing the cumulative reward in CMDPs. The literature on this topic can be divided into two types of methods. Methods of the first type, referred to as *primal-only* methods, only update the policy parameters while enforcing the constraints (Achiam et al., 2017). The recent work of Xu et al. (2021) guarantees global convergence for such methods to the optimal feasible policy in both the tabular and function approximation settings . The second approach for planning in CMDPs is to form the Lagrangian, and solve the resulting saddle-point problem using *primal-dual algorithms* (Altman, 1999). Such approaches update both the policy parameters (primal variables), and the Lagrange multipliers (dual variables). The recent work of Ding et al. (2020) proves that this approach converges to the optimal policy in both tabular and function approximation settings as well.

Although there has been substantial progress in designing planning algorithms for CMDPs, *all of the proposed algorithms are sensitive to hyper-parameter tuning*. For example, Figs. 1a and 1b illustrate the effect of varying the hyper-parameters of two provably efficient algorithms – the primal-dual Gradient Descent Ascent (GDA) (Ding et al., 2020) and the primal-only CRPO (Xu et al., 2021)– on a gridworld task. We can see that the magnitude of both optimality gap and constraint violations vary greatly for different hyper-parameters. Hence, in order to obtain reasonably good performance on a new task, the hyper-parameters of these algorithms need to be tuned from scratch, incurring significant computational overhead. Designing robust planning algorithms that require minimal hyper-parameter tuning is the main motivation for this work. We propose Coin-Betting Politex (CBP) as an algorithm that can control both the primal and the dual regret, ensuring better robustness to hyper-parameters (as evidenced by the red lines in the figures).

# 2 Problem Formulation

An infinite-horizon discounted CMDP is defined as a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, r, c, b, \rho, \gamma \rangle$ where $\mathcal{S}$ is the set of states, $\mathcal{A}$ is the action set, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \Delta_{\mathcal{S}}$ is the transition probability function, $\rho \in \Delta_{\mathcal{S}}$ is the initial distribution of states and $\gamma \in [0, 1)$ is the discount factor. The primary reward to be maximized is denoted by $r : \mathcal{S} \times \mathcal{A} \to [0, 1]$. The expected discounted return or *reward value* of a policy $\pi : \mathcal{S} \to \Delta_{\mathcal{A}}$ is defined as $V_r^\pi(\rho) = \mathbb{E}_{s_0, a_0, \ldots} \left[ \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right]$, where $s_0 \sim \rho, a_t \sim \pi(a_t|s_t)$, and $s_{t+1} \sim \mathcal{P}(s_{t+1}|s_t, a_t)$. Similarly, the constrained reward is denoted by $c : \mathcal{S} \times \mathcal{A} \to [0, 1]$ and the *constrained reward* of policy $\pi$ is denoted by $V_c^\pi(\rho)$. For each state-action pair $(s, a)$ and policy $\pi$, the reward action-value function is defined as $Q_r^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, and satisfies the relation: $V_r^\pi(\rho)(s) = \langle \pi(\cdot|s), Q_r^\pi(s, \cdot) \rangle$. We define $Q_c^\pi$ analogously. Given a class of policies $\Pi$, the agent's objective is to return a policy $\pi \in \Pi$ that maximizes $V_r^\pi(\rho)$, while ensuring that $V_c^\pi(\rho) \geq b$. Formally,

$$\max_{\pi \in \Pi} V_r^\pi(\rho) \quad \text{s.t.} \quad V_c^\pi(\rho) \geq b. \tag{1}$$

Throughout, we will assume the existence of a feasible policy, and denote the optimal feasible policy by $\pi^*$. In this work, we will consider an easier problem with a relaxed feasibility requirement. In particular, given a target error $\varepsilon$, our aim is to return a policy $\hat{\pi}$ such that, $V_r^{\hat{\pi}}(\rho) \geq V_r^{\pi^*}(\rho) - \varepsilon$, s.t. $V_c^{\hat{\pi}}(\rho) \geq b - \varepsilon$. In the next section, we specify a generic primal-dual framework to solve the problem in Eq. (1) and introduce our algorithm.

## 3 Methodology

### 3.1 Primal-Dual framework

In order to specify the primal-dual framework, we use the Lagrangian formulation, and express the constrained optimization problem in Eq. (1) as the following saddle-point problem:

$$\max_{\pi \in \Pi} \min_{\lambda \geq 0} V_r^{\pi}(\rho) + \lambda [V_c^{\pi}(\rho) - b]. \tag{2}$$

Here, $\lambda \in \mathbb{R}$ is the Lagrange multiplier for the constraint and $\lambda^*$ refers to its optimal value, meaning that $(\pi^*, \lambda^*)$ is the solution to the above saddle-point problem.

We will solve the above primal-dual saddle-point problem iteratively, by alternatively updating the policy (primal variable) and the Lagrange multiplier (dual variable). If $T$ is the total number of iterations of such an algorithm, we define $\pi_t$ and $\lambda_t$ to be the primal and dual iterates for $t \in [T]$. We define $\hat{Q}_r^t := \hat{Q}_r^{\pi_t}$ and $\hat{Q}_c^t := \hat{Q}_c^{\pi_t}$ as the *estimated* action-value functions corresponding to the policy $\pi_t$. In this section, we assume that we have uniform control over the approximation errors in the action-value functions for every state-action pair implying that $\|Q_r^t - \hat{Q}_r^t\|_\infty \leq \tilde{\varepsilon}$ and $\|Q_c^t - \hat{Q}_c^t\|_\infty \leq \tilde{\varepsilon}$.

Given a generic primal-dual algorithm, our task is to characterize its performance in terms of its cumulative reward and constraint violation. For a sequence of policies $\pi_t$ and Lagrange multipliers $\lambda_t$ generated by an algorithm, we define the **average optimality gap (OG)** and **average constraint violation (CV)** as follows:

$$OG := \frac{1}{T} \sum_{t=0}^{T-1} [V_r^{\pi^*}(\rho) - V_r^t(\rho)] \quad ; \quad CV := \frac{1}{T} \sum_{t=0}^{T-1} [b - V_c^t(\rho)]_+. \tag{3}$$

Here, $[x]_+ = \max\{x, 0\}$. We define the *primal regret* and *dual regret* with respect to the optimal policy $\pi^*$ as follows. If $\hat{V}_c^t(\rho)(s) = \langle \pi_t(\cdot|s), \hat{Q}_c^t(s, \cdot) \rangle$ is the cost value function at state $s$, then:

$$\mathcal{R}^p(\pi^*, T) := \sum_{t=0}^{T-1} \sum_{s=0}^{\mathcal{S}-1} \langle \pi^*(\cdot|s) - \pi_t(\cdot|s), \hat{Q}_r^t + \lambda_t \hat{Q}_c^t] \rangle \quad ; \quad \mathcal{R}^d(\lambda, T) := \sum_{t=0}^{T-1} \langle \lambda_t - \lambda, \hat{V}_c^t(\rho) - b \rangle. \tag{4}$$

The above quantities correspond to the regret for online linear optimization algorithms that can independently update the primal and dual variables. In the unconstrained setting, reducing the policy optimization problem to that of online linear optimization has been previously explored through the *Politex* algorithm (Abbasi-Yadkori et al., 2019). We build upon Politex to reduce the problem from Eq. (3) to that of online linear optimization in Eq. (4). Theorem 3.1 characterizes the performance of a generic algorithm in terms of its primal and dual regret (proof omitted due to lack of space).

**Theorem 3.1.** *Assume that $\|Q_r^t - \hat{Q}_r^t\|_\infty \leq \tilde{\varepsilon}$ and $\|Q_c^t - \hat{Q}_c^t\|_\infty \leq \tilde{\varepsilon}$, for a generic algorithm producing a sequence of polices $\{\pi_0, \pi_1, \ldots, \pi_{T-1}\}$ and dual variables $\{\lambda_0, \lambda_1, \ldots, \lambda_{T-1}\}$ such that for all t. $\lambda_t$ is constrained to lie in the $[0, U]$, where $U > \lambda^*$. Here $g(U) = \frac{\tilde{\varepsilon}}{1-\gamma}(1 + U) + U\tilde{\varepsilon}$. Then, the average optimality gap (OG) and constraint violation (CV) are bounded by:*

$$\textit{OG} \leq \frac{\mathcal{R}^p(\pi^*, T) + (1-\gamma)\mathcal{R}^d(0, T)}{(1-\gamma)T} + g(U) \quad ; \quad \textit{CV} \leq \frac{\mathcal{R}^p(\pi^*, T) + (1-\gamma)\mathcal{R}^d(U, T)}{(U - \lambda^*)(1-\gamma)T} + g(U).$$

Importantly, the above result does not depend on the class of policies $\Pi$, nor does it require any assumption about the underlying CMDP. In order to bound the average reward optimality gap and the average constraint violation, we need to (i) project the dual variables onto the $[0, U]$ interval and ensure that $U > \lambda^*$, (ii) update the primal and dual variables to control the respective regret in Eq. (4), and (iii) control the approximation error $\tilde{\varepsilon}$. In the next section, we use this recipe to design algorithms with provable guarantees.

### 3.2 Coin-Betting Politex Algorithm

Orabona and Pal (2016) propose *coin-betting* algorithms that reduce the online linear optimization problems in Eq. (4) to online betting, and leverage this idea to design parameter-free algorithms. First, we instantiate the Algorithm 2 in Orabona and Pal (2016) for updating the policy, which is completely parameter-free, to our problem. The policy update requires the computation of additional variables $w_t$ for each $(s,a)$ pair and iteration $t$, as follows:

$$\begin{aligned} w_{t+1}(s,a) &= \frac{\sum_{i=0}^{t} \tilde{A}_l^i(s,a)}{(t+1)+T/2}\left(1+\sum_{i=0}^{t} \tilde{A}_l^i(s,a)\,w_i(s,a)\right) \\ \pi_{t+1}(a|s) &= \begin{cases} \pi_0(a|s), & \text{if } \sum_a \pi_0(a|s)\,[w_{t+1}(s,a)]_+ = 0 \\ \frac{\pi_0(a|s)\,[w_{t+1}(s,a)]_+}{\sum_a \pi_0(a|s)\,[w_{t+1}(s,a)]_+}, & \text{otherwise,} \end{cases} \end{aligned} \tag{5}$$

where, $\tilde{A}_l^t(s,a) = \hat{A}_l^t(s,a)\,\mathcal{I}\{w_t(s,a) > 0\} + [\hat{A}_l^t(s,a)]_+\,\mathcal{I}\{w_t(s,a) \leq 0\}$. Here, $\hat{A}_l^t(s,a)$ can be interpreted as the normalized advantage function, $\hat{A}_l^t(s,a) = \frac{1-\gamma}{1+U}\left[\hat{Q}_l^t(s,a) - \left\langle \hat{Q}_l^t(s,\cdot), \pi_t(\cdot|s)\right\rangle\right]$. We normalize the action-value function to ensure boundedness within $[0,1]$ range. Note that the dual variables are projected in $[0,U]$ range, where $U \geq \lambda^*$.

Similarly, we instantiate Algorithm 2 in Orabona and Tommasi (2017) for updating the dual variable $\lambda$:

$$\lambda_{t+1} = \lambda_0 + \frac{\sum_{i=0}^{t} g_i}{L_t \max(\sum_{i=0}^{t} |g_i| + L_t, \alpha_\lambda L_t)}\left(L_t + \sum_{i=0}^{t} [(\lambda_i - \lambda_0) g_i]_+\right), \tag{6}$$

where $g_t := b - \hat{V}_c^t(\rho)$, $\alpha_\lambda$ is a tunable hyper-parameter, $L_t = \max(L_{t-1}, |g_t|)$ and $L_0$ is initialized to $0$. Algorithm 1 summarizes CBP in the linear function approximation setting. In particular, for a given *coreset* $\mathcal{C}$, we estimate the action-value functions for a subset of $(s,a) \in \mathcal{C}$.

**Algorithm 1:** Coin-Betting Politex (CBP)

```
Input: $\alpha_\lambda > 0$ (parameter), $\pi_0$ (policy initialization), $\lambda_0$ (dual variable initialization), $m$ (Number of trajectories), $T$ (Number of iterations), Feature map $\Phi$, Coreset $\mathcal{C}$
for $t = 0, \ldots, T-1$ do
    Using $m$ trajectories for every $(s,a) \in \mathcal{C}$, compute weight vectors $\theta_r^{\pi_t}$ and $\theta_c^{\pi_t}$ of $Q_r$ and $Q_c$ respectively using LSTDQ (Lagoudakis and Parr, 2003).
    Compute $\hat{Q}_r^t(s,a) = \langle \theta_r^{\pi_t}, \phi(s,a) \rangle$, $\hat{Q}_c^t(s,a) = \langle \theta_c^{\pi_t}, \phi(s,a) \rangle$ and $\hat{Q}_l^t(s,a) = \hat{Q}_r^t(s,a) + \lambda_t\,\hat{Q}_c^t(s,a)$.
    Use stored vectors $\{\theta_r^{\pi_i}, \theta_c^{\pi_i}\}_{i=0}^t$ and compute $\pi_{t+1}(a|s)$ using Eq. (5).
    Compute $\hat{V}_c^{\pi_t}(\rho)$ and update $\lambda_{t+1}$ using Eq. (6).
end
return mixture policy $\bar{\pi}_T := \frac{\sum_{t=0}^{T-1} \pi_t}{T}$.
```

Setting $U = \frac{2}{([\max_\pi V_c^\pi(\rho)] - b)(1-\gamma)}$, and using the primal and dual regret expressions for coin-betting (Orabona and Tommasi, 2017), we can show that a variant of CBP results in an $O\left(\frac{1}{(1-\gamma)^3\sqrt{T}} + \frac{\varepsilon\sqrt{d}}{1-\gamma}\right)$ bound on the reward sub-optimality and a constraint violation of $O\left(\frac{1}{(1-\gamma)^2\sqrt{T}} + \varepsilon\sqrt{d}\right)$. The proof is omitted due to lack of space.

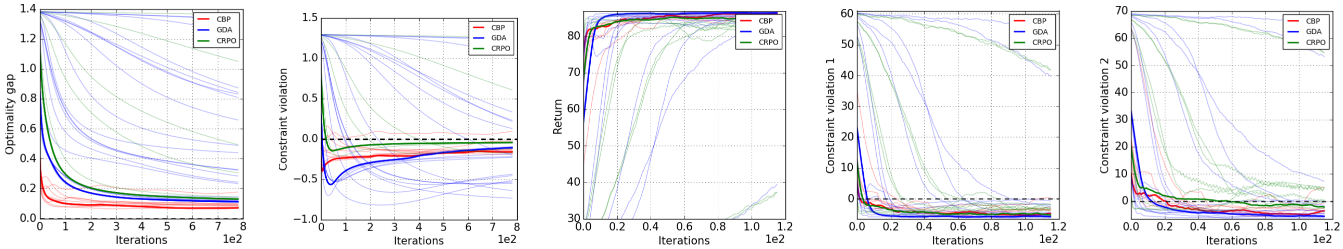
## 4 Experiments

### 4.1 Model-based setting

We show experiments on a simple tabular environment to compare our proposed algorithm, CBP, with the primal-dual approach, exemplified by GDA (Ding et al., 2020), and the primal-only approach exemplified by CRPO (Xu et al., 2021). We consider the $5X5$ gridworld introduced in (Sutton and Barto, 2018) and depicted in Fig. 1e. All four cardinal actions in the special states $A$ and $B$ receive a non-zero reward and cost values and land in states $A'$ and $B'$ respectively. The remaining states receive $0$ reward and cost for all actions. Here $\gamma = 0.9$. In this section, we assume access to the true CMDP model. Figs. 1a and 1b show the performance metrics, OG and CV, for the three algorithms. CBP is *robust* to hyperparameter settings. The *best hyperparameters* have been chosen to satisfy $-\eta \leq CV \leq 0$, and lead to the smallest OG value. We set $\eta = 0.25$. Next, we verify the robustness of the algorithms with respect to environment mis-specification. We use the best value of the hyperparameters obtained for discount factor $\gamma = 0.9$ and use them for different CMDP corresponding to two other values of $\gamma = \{0.7, 0.8\}$ in Figs. 1c and 1d. This experiment suggests that CBP is robust to environment mis-specification and consistently converges faster than the baselines.

### 4.2 Model-free setting

In this section, we assume the model-free setting with no access to the true CMDP model. We use linear function approximation (LFA) to learn estimates of action-values using the LSTDQ algorithm (Lagoudakis and Parr, 2003). Tile coding (Sutton and Barto, 2018) is used to obtain features. We conduct experiments on two environments: Gridworld (Fig. 1e) and Cartpole (OpenAI gym). Along with the usual reward, we add two penalty constraints in the Cartpole environment, when (a) entering specific areas on x-axis and (b) having the angle of the pole larger than a threshold. Fig. 2a shows the OG and CV in the Gridworld environment with LFA. Similarly, Figs. 2c to 2e show the return and two constraint violations for constrained Cartpole environment. For the Cartpole experiment, we kept the CV threshold $\eta = 6$. In both the environments, CBP is robust to variations in the values of the hyperparameters.



(a) Gridworld OG (b) Gridworld CV (c) Cartpole return (d) Cartpole CV 1 (e) Cartpole CV2

Figure 2: **Model-free with linear function approximation:** We compare performance of CBP with baselines GDA and CRPO. Figs. 2a and 2b shows the performance in the gridworld environment with function approximation. Figs. 2c to 2e shows the return, CV1 and CV2 in constrained Cartpole environment. Here, dark lines correspond to the best hyper-parameters for each of the three algorithms. Lighter lines show the variation in performance with different hyper-parameters.CBP is robust to hyperparameter settings in both domains.

## 5 Conclusion

We proposed a general primal-dual framework to solve CMDPs with function approximation. We instantiated this framework using coin-betting algorithms from online linear optimization, and proposed the CBP algorithm. We proved that CBP is theoretically sound and has a good regret bounds. In addition, we showed empirically that CBP is robust to hyper-parameter tuning and environment mis-specification. We believe that developing robust, parameter-free algorithms is important for reproducibility in RL, and hope that our work will encourage future research in this area.

### References

Yasin Abbasi-Yadkori, Peter Bartlett, Kush Bhatia, Nevena Lazic, Csaba Szepesvari, and Gellért Weisz. Politex: Regret bounds for policy iteration using expert prediction. In *International Conference on Machine Learning*, 2019.

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *International conference on machine learning*, pages 22–31. PMLR, 2017.

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

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

Michail G Lagoudakis and Ronald Parr. Least-squares policy iteration. *Journal of Machine Learning Research*, 4:1107–1149, 2003.

Francesco Orabona and David Pal. Coin betting and parameter-free online learning. *Advances in Neural Information Processing Systems*, 29:577–585, 2016.

Francesco Orabona and Tatiana Tommasi. Training deep networks without learning rates through coin betting. *Advances in Neural Information Processing Systems*, 30, 2017.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction. Second Edition*. MIT Press, 2018.

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.