Proceedings of Machine Learning Research 129:209–224, 2020 ACML 2020

Constrained Reinforcement Learning via Policy Splitting

Haoxian Chen HC3136@COLUMBIA.EDU
Columbia University, New York, NY 10027
Henry Lam KHL2114@COLUMBIA.EDU
Columbia University, New York, NY 10027
Fengpei Li FL2412@COLUMBIA.EDU
Columbia University, New York, NY 10027
Amirhossein Meisami MEISAMI@ADOBE.COM
Adobe Inc., San Jose, CA 95110

Editors: Sinno Jialin Pan and Masashi Sugiyama

Abstract

We develop a model-free reinforcement learning approach to solve constrained Markov decision processes, where the objective and budget constraints are in the form of infinite-horizon discounted expectations, and the rewards and costs are learned sequentially from data. We propose a two-stage procedure where we first search over deterministic policies, followed by an aggregation with a mixture parameter search, that generates policies with simultaneous guarantees on near-optimality and feasibility. We also numerically illustrate our approach by applying it to an online advertising problem.

Keywords: constrained Reinforcement Learning, online advertisement, policy splitting

1. Introduction

Applications of Reinforcement Learning (RL) in online advertising with recommendation systems have been a topic of major research interests (Cai et al. (2018); Wang et al. (2018); Wu et al. (2018)). However, despite their tremendous success, most RL-methods are not designed to learn optimal policies under constraints, yet they appear ubiquitously when facing budget or safety considerations. A standard framework for studying RL under constraints is the Constrained Markov Decision Process (CMDP), where the objective is to maximize the long-run return, with constraints on one or several types of long-run costs. In this paper, we consider the case where both the objective and the constraint are in the form of an infinite-horizon cumulative discounted expectation, whereas the returns, costs and transitions are revealed from sequential data. The goal is to design an efficient methodology for the constrained problem by assimilating classical optimality properties of CMDP into RL, in order to efficiently use established RL approaches and obtain policies that enjoy both near-optimality and feasibility.

The CMDP in the form described above is motivated from a range of important applications including online advertising. Sponsored search campaigns, for instance, are designed based on predetermined budgets. Therefore, the marketer has to employ effective strategies to accrue the maximum reward while observing certain monetary constraints throughout the campaign. Similarly, in email campaigns, the marketer can only send out a limited

number of emails under different constraints due to user fatigue or limited available discount offers. Thus, it is important to consider information beyond potential revenues, such as the remaining budget or the likely outcomes of different offers. Direct applications of most RL-algorithms do not, in general, consistently produce optimal solutions within these budget constraint. Thus, several lines of work have been devoted to resolve this challenge. In the model-based regime (i.e., parametric-based transition), Geibel (2006) and Lee et al. consider linear programming, Geibel (2006) considers state-space extension, and Feinberg and Rothblum (2012) considers policy iterations. However, model-based algorithms suffer when the state or action space gets large as estimating the transition dynamics of the users can be very challenging or even infeasible. In model-free settings, constrained policy optimization (CPO) (Achiam et al. (2017)) is designed based on trust region policy optimization (TRPO) and its variants (Schulman et al. (2015, 2017)). Through surrogate function approximations, CPO provides safe iterations in each policy update, preventing any constraint violation in the agent's learning process. However, the implementation requires a safe policy to start with and it may be over-conservative to require a safe update in each iteration, especially for areas of advertising where the budget constraint is not as hard a constraint as, say, in auto-driving. Thus, the extra effort and setup in the implementation of CPO might not be as desirable in our setting. Another line of work in tacking constrained MDP uses primal-dual, Lagrangian-based RL methods (Chow et al. (2018); Tessler et al. (2018)), which involves stochastic updates for solving the KKT conditions. In particular, Chow et al. (2018) investigates constraints arising from risk criteria such as conditional-value-at-risk or chance constraints while the reward constrained policy optimization (RCPO) in Tessler et al. (2018) uses an actor-critic updates in the policy space and a stochastic recursion on the Lagrange multiplier updates in the dual space. However, although convergence is guaranteed for primal-dual methods in theory, in practice significant efforts are required to tune the hyper-parameters, especially the learning rates of the dual variable, as the updates become noisy and unstable around convergence and the training process can easily become too slow or overly greedy.

In this paper, we address these issues on the primal-dual formulation and explain the unstable convergence behavior of primal-dual methods around the optimal value. Furthermore, we design a mixing method which aims to alleviate the tuning issues by both exploiting the low-dimensional feature of dual variables (when the number of budget constraints is negligible compared to the cardinality of the state/action space) and investigating a special splitting property of CMDPs (Feinberg and Rothblum (2012)). In particular, for a single budget constraint, the "splitting" property refers to a structure of the optimal randomized policy in CMDP where two possible actions are assigned with a binary distribution to a certain state and the policy stays deterministic elsewhere (Feinberg and Rothblum (2012)). This splitting property contributes to the unstable behaviors of the dual convergence because the RL method is essentially searching for two different optimal policies around the optimal dual value. This splitting property arises from the extreme points of a linear program (LP) formulation of CMDP via the occupation measure (Altman (1999)). It reveals the saddle point structure of the Lagrangian and allows us to confine our policy search in a smaller solution space.

Leveraging the splitting property, our approach bypasses the need to search over large spaces of randomized policies and, by solving a sequence of RL problems without restriction

under the Lagrangian relaxation, finds candidate deterministic policies with direct application of classical RL-methods (e.g. Q-learning, TD-learning or TRPO). To improve on the undesirable properties of primal-dual methods around convergence, we first propose a discretization scheme which exploits the one-dimensional structure of dual variable and allows for parallel computing. Then we propose a novel feasibility mixing procedure which efficiently mixes the candidate policies and find an optimal randomized policy that would achieve both optimality and feasibility. We provide theoretical justifications on our framework, and also conduct experiments on an online advertisement problem to demonstrate its performance.

The remainder of this paper is organized as follows. Section 2 presents our problem setting and notations. Section 3 describes our Lagrangian formulation and its implications. Section 4 presents our main dual Q-learning algorithm that harnesses the splitting property of CMDP in the Lagrangian formulation. Section 5 discusses practical implementation, and Section 6 illustrates our experimental results.

2. Problem Setting

A Constrained Markov Decision Process (CMDP) can be formulated as follows. Let $\mathcal{S}$ be the finite set of states, $\mathcal{A}$ the finite set of actions, and $p(s, a, s')$ the probability measure governing the stochastic transition between states, namely

$$\mathbb{P}(s_{t+1} = s' | s_t = s, a_t = a) = p(s, a, s')$$

with non-negative entries and $\sum_{s'} p(s, a, s') = 1$. Let $r_t = r(s_t, a_t)$ be the corresponding expected reward. Denote Π to be the space of stationary randomized policies π where

$$\mathbb{P}(a_t = a | s_0, a_0, r_1, s_1, a_1, ..., r_t, s_t = s) = \mathbb{P}(a_t = a | s_t = s) = \pi(s, a),$$

and $\sum_a \pi(s, a) = 1, \pi(s, a) \geq 0$ for all a, s. Notice the stationarity comes from the fact that the policy at each state s does not change with t. Moreover, if over any state s, $\pi(s, a)$ is zero for all but one action $a \in \mathcal{A}$, then we say $\pi \in \Pi_0 \subset \Pi$ is a stationary deterministic policy and denote this a by $\pi(s)$. Suppose at each step t, the agent interacting with the environment not only receives random (immediate) reward r_t but also incurs random (immediate) cost denoted by $c_t = c(s_t, a_t)$. Let $s_0 \sim \rho$ be the distribution of the initial state and $\gamma \in [0, 1]$ be the discounted factor. We consider the following CMDP:

$$\begin{aligned} \max_{\pi \in \Pi} \quad & \mathbb{E}_{s_0 \sim \rho, \pi} \left[\sum_{t=1}^{\infty} \gamma^{t-1} r_t \right] \\ \text{s.t.} \quad & \mathbb{E}_{s_0 \sim \rho, \pi} \left[\sum_{t=1}^{\infty} \gamma^{t-1} c_t \right] \leq B, \end{aligned} \tag{1}$$

where $\mathbb{E}_{s_0 \sim \rho, \pi}$ denotes the expectation under policy π and initial distribution $s_0 \sim \rho$. We confine our policy search in Π because it is well-known (see, e.g., Altman (1999)) that the optimal policy $\pi^\star$ for CMDP lies in the space Π. Also, we do not assume the distributions of $r(\cdot, \cdot)$ $c(\cdot, \cdot)$, or $p(\cdot, \cdot, \cdot)$ are known.

3. Lagrangian with Reduced Policy Space

A common way to solve CMDP (1) is to formulate it as the following LP (Altman (1999)):

$$\begin{aligned} \max_{\mathbf{x}\geq 0} \quad & \sum_{s,a} x_{sa} r(s,a) \\ \text{s.t.} \quad & \sum_{s,a} x_{sa} c(s,a) \leq B, \\ & \sum_{a} x_{sa} - \gamma \sum_{s',a} x_{s'a} p(s',a,s) = \rho(s) \ \forall s, \end{aligned} \tag{2}$$

where $x_{sa} = \sum_{t=1}^{\infty} \gamma^{t-1} \mathbb{P}(s_t = s, a_t = a | \pi, s_0 \sim \rho)$ is referred to as the *occupation measure* of policy π under initial distribution ρ. It can be interpreted as the total discounted expected number of times state-action pair (s,a) is visited under policy π, so that $\mathbb{E}_\pi[\sum_{t=1}^{\infty} \gamma^{t-1} r_t]$ can be seen to be expressible as $\sum_{s,a} x_{sa} r(s,a)$ and similarly $\mathbb{E}_\pi[\sum_{t=1}^{\infty} \gamma^{t-1} c_t]$ as $\sum_{s,a} x_{sa} c(s,a)$, and the second constraint in (2) follows from a first-step Markovian analysis. Moreover, it is shown in Altman (1999) that an optimal randomized policy $\pi^\star$ can be computed from an optimal solution $\mathbf{x}^\star$ of (2) by letting

$$\pi^\star(s,a) = \frac{x^\star_{sa}}{\sum_a x^\star_{sa}}. \tag{3}$$

However, formulating the above optimization problem requires the knowledge of $r(s,a), c(s,a)$ and $p(s,a,s')$ of the MDP which in our setting can only be learned implicitly. Also, the number of state-action pair may get too large to use tabular methods. On the other hand, the more efficient, large-scale approximate RL methods such as TD-learning, Q-learning or TRPO (Sutton and Barto (2018); Watkins and Dayan (1992)) cannot directly help us with the search of an optimal randomized policy. To address this issue, we first consider the dual optimization problem (Bertsimas and Tsitsiklis (1997)) of (2):

$$\begin{aligned} \min_{\lambda \geq 0, \mathbf{v}} \quad & \sum_{s} v_s \rho(s) + \lambda B \\ \text{s.t.} \quad & v_s \geq r(s,a) - \lambda c(s,a) + \gamma \sum_{s'} p(s,a,s') v_{s'} \ \forall s. \end{aligned} \tag{4}$$

For fixed $\lambda \geq 0$, the minimization in (4) is exactly the LP formulation for solving the value function of an unconstrained MDP with adjusted reward $r_t^\lambda = r_t - \lambda c_t$ instead of r_t at each step t (plus the constant term λB), and the constraint follows from the Bellman optimality equation (Puterman (2014)). This allows us to convert (1) into the form (5) (shown below). Advantageously, for any fixed λ, because of its unconstrained nature, the inner maximization problem in (5) now suffices to search for policy π in the deterministic policy space Π_0 instead of the randomized policy space Π. Hence we can apply many suitable approximation algorithms in RL to search for the optimal deterministic policy (Sutton and Barto (2018)). We have the following theorem (Notice the reduction of policy space into Π_0 as a key transition in this dual):

Theorem 1 *Problem (1) can be reformulated as*

$$\min_{\lambda \geq 0} \quad \max_{\pi \in \Pi_0} \quad \mathcal{R}(\pi, \rho) - \lambda \big(\mathcal{C}(\pi, \rho) - B \big) \tag{5}$$

where $\mathcal{R}(\pi,\rho) \triangleq \mathbb{E}_{s_0\sim\rho,\pi}[\sum_{t=1}^{\infty}\gamma^{t-1}r_t]$ *and* $\mathcal{C}(\pi,\rho) \triangleq \mathbb{E}_{s_0\sim\rho,\pi}[\sum_{t=1}^{\infty}\gamma^{t-1}c_t]$.

Proof Based on our discussion and the LP duality, we only have to show that for any fixed $\lambda \geq 0$,

$$\begin{aligned} \min_{\mathbf{v}} \quad & \sum_s v_s\rho(s) \\ \text{subject to} \quad & v_s \geq r(s,a) - \lambda c(s,a) + \gamma\sum_{s'} p(s,a,s')v_{s'} \ \forall s \end{aligned} \tag{6}$$

is equivalent to

$$\max_{\pi\in\Pi_0} \ \mathcal{R}(\pi,\rho) - \lambda\mathcal{C}(\pi,\rho). \tag{7}$$

In particular, for fixed $\lambda \geq 0$, problem (6) obtains the optimal expected total discounted reward $\sum_s v_s\rho(s)$ with adjusted reward $r_t^\lambda = r_t - \lambda c_t$ guaranteed by the Bellman optimality constraint as well as the condition that $\rho(s) > 0, \forall s$ (Puterman (2014)). On the other hand, given the discounted adjusted reward r_t^λ, we know from classical MDP results that for any unconstrained infinite-horizon discounted MDP there exists a stationary and deterministic optimal policy $\pi^\star \in \Pi_0$ for any initial state distribution satisfying $\rho(s) > 0, \forall s$. Moreover, the optimal expected total discounted reward is $\max_{\pi\in\Pi_0}\mathcal{R}(\pi,\rho) - \lambda\mathcal{C}(\pi,\rho)$. ■

Theorem 1 suggests that the search for optimal policies can first proceed with a deterministic policy search fixing some set of λ. Then, we optimize with respect to λ in (5) to find an optimal $\lambda^\star$ which closes the duality gap between (2) and (4) with optimal policies that maximize the penalized expected reward $r_t - \lambda^* c_t$ plus the term $\lambda^\star B$.

4. Policy Mixing and Dual Q-Learning

The two steps discussed above recover the optimal value of the primal (2). However, to recover the optimal, possibly randomized policy, we need to look more closely at the dual problem (5). To begin, it is known that if an LP has an optimal solution, then it also has an optimal basic feasible solution (Bertsimas and Tsitsiklis (1997)), meaning that we can find optimal solution $\mathbf{x}^\star$ with at most $s+1$ non-zeros entries. This leads to the following proposition.

Proposition 1 *If* $\rho(s) > 0\ \forall s$*, then there is an optimal policy* $\pi^\star$ *for the primal problem* (1) *with* $\pi^\star(s)$ *following a deterministic action for all but possibly one state.*

Proof Given that we can find optimal solution $\mathbf{x}^\star$ for problem (2) with at most $s+1$ non-zero entries, if we further assume that $\rho(s) > 0$ for all state s, then the second constraint of (2) would force any feasible solution $\mathbf{x}$ to satisfy $\sum_a x_{sa} > 0$ for any s. This condition implies that for any s, we can find at least one a such that $x^\star_{sa} > 0$. Since $\mathbf{x}^\star$ has at most $s+1$ non-zeros entries, we can have at most one positive entry among all entries of $x^\star_{sa}$. It then follows from (3) that the optimal policy $\pi^\star$ for (1) is deterministic at all states except possibly one, where the optimal policy splits into two possible actions. ■

Following Proposition 1, we can characterize an important property regarding the optimal policy for (5). In particular, we consider the dual function

$$\mathcal{D}(\lambda) \triangleq \max_{\pi\in\Pi_0} \mathcal{R}(\pi,\rho) - \lambda\big(\mathcal{C}(\pi,\rho) - B\big). \tag{8}$$

Theorem 2 *Assume $\rho(s) > 0\ \forall s$ and the optimal policy $\pi^\star$ for problem (1) is unique. Then the maximization in (8), at the optimal λ^* that solves (5), admits either a deterministic optimal policy $\pi^\star$, or a pair of optimal deterministic policies π_1, π_2 with actions different in one state s and $\pi^\star = (1-t)\pi_1 + t\pi_2$ for some $0 < t < 1$.*

Proof Let $\pi^\star$ be the optimal, possibly randomized policy for the primal (1). By the LP duality (Bertsimas and Tsitsiklis (1997)), we know the optimal values for (1) and (5) are equal and we must have, for some $\lambda^\star \in \underset{\lambda\geq 0}{\operatorname{argmin}}\ \mathcal{D}(\lambda) \geq 0$, that

$$\mathcal{R}(\pi^\star, \rho) = \min_{\lambda\geq 0} \mathcal{D}(\lambda) = \mathcal{D}(\lambda^\star). \tag{9}$$

If there exists $\lambda^\star = 0$ where (9) holds, then

$$\min_{\lambda\geq 0} \mathcal{D}(\lambda) = \mathcal{D}(0) = \max_{\pi\in\Pi_0} \mathcal{R}(\pi, \rho). \tag{10}$$

Combining (9) and (10), we have $\mathcal{R}(\pi^\star, \rho) = \max_{\pi\in\Pi_0} \mathcal{R}(\pi, \rho)$ and by the uniqueness we have $\pi^\star = \underset{\pi\in\Pi_0}{\operatorname{argmax}} \mathcal{R}(\pi, \rho)$. The primal feasibility of (1) guarantees $\mathcal{C}(\pi^\star, \rho) \leq B$. In fact, notice in this case, the optimal policy for the unconstrained MDP in (1) is actually feasible, and thus CMDP (1) reduces to an unconstrained MDP.

On the other hand, if we have $\underset{\lambda\geq 0}{\operatorname{argmin}} \mathcal{D}(\lambda) > 0$, then we observe that $\mathcal{D}(\lambda) = \max_{\pi\in\Pi_0} \mathcal{R}(\pi, \rho) - \lambda\big(\mathcal{C}(\pi, \rho) - B\big)$ is the maximum of a finite number (i.e. the number of deterministic policies is finite) of linear functions in λ. Thus, $\mathcal{D}(\lambda)$ is piece-wise linear and convex in λ. Since $\lambda^\star > 0$ is the global minimum of $\mathcal{D}(\lambda)$ and $\mathcal{D}(\lambda)$ is piece-wise linear, we must have $\mathcal{D}^+(\lambda^\star) = \lim_{t\to 0} \frac{\mathcal{D}(\lambda^\star + t) - \mathcal{D}(\lambda^\star)}{t} \geq 0$ and $\mathcal{D}^-(\lambda^\star) = \lim_{t\to 0} \frac{\mathcal{D}(\lambda^\star) - \mathcal{D}(\lambda^\star - t)}{t} \leq 0$.

Now if $\lambda^\star = \underset{\lambda\geq 0}{\operatorname{argmin}} \mathcal{D}(\lambda) > 0$ is not unique, then by convexity we can find an interval of λ with the same optimal $\mathcal{D}(\lambda)$, implying the optimal deterministic policy under this λ is both feasible (zero slope means $\mathcal{C}(\pi, \rho) = B$) and optimal. Thus, suppose $\lambda^\star = \underset{\lambda\geq 0}{\operatorname{argmin}} \mathcal{D}(\lambda) > 0$ is unique, then we have $\mathcal{D}^-(\lambda^\star) < 0 < \mathcal{D}^+(\lambda^\star)$, and there exists some $\epsilon > 0$ and policies π_1, π_2 such that

$$\mathcal{D}(\lambda) = \mathcal{D}(\lambda^\star) + \mathcal{D}^+(\lambda^\star)(\lambda - \lambda^\star) = \mathcal{R}(\pi_1, \rho) - \lambda\big(\mathcal{C}(\pi_1, \rho) - B\big) \tag{11}$$

for $\lambda^\star \leq \lambda \leq \lambda^\star + \epsilon$ and

$$\mathcal{D}(\lambda) = \mathcal{D}(\lambda^\star) + \mathcal{D}^-(\lambda^\star)(\lambda - \lambda^\star) = \mathcal{R}(\pi_2, \rho) - \lambda\big(\mathcal{C}(\pi_2, \rho) - B\big) \tag{12}$$

for $\lambda^\star - \epsilon \leq \lambda \leq \lambda^\star$. In particular, at $\lambda^\star$, we have

$$\mathcal{R}(\pi_1, \rho) - \lambda^\star\big(\mathcal{C}(\pi_1, \rho) - B\big) = \mathcal{R}(\pi_2, \rho) - \lambda^\star\big(\mathcal{C}(\pi_2, \rho) - B\big) \tag{13}$$

which implies

$$\pi_1 = \pi_2 = \underset{\pi\in\Pi_0}{\operatorname{argmax}} \mathcal{R}(\pi, \rho) - \lambda^\star \mathcal{C}(\pi, \rho). \tag{14}$$

We know from Bellman (2013) that for a finite unconstrained MDP problem, there exists a unique optimal value function such that $v^\star(s) \geq v^\pi(s)$ for all state s. Thus, (14) and the fact that $\rho(s) > 0\ \forall s$ implies that we must have

$$v^\star(s) = v^{\pi_1}(s) = v^{\pi_2}(s) \quad \forall s \tag{15}$$

where $\mathbf{v}^\star$ is the optimal value function for the MDP with adjusted reward $r_t^{\lambda^\star} = r_t - \lambda^\star c_t$ and $\mathbf{v}^{\pi_i}$ is the value of policy π_i under this adjusted reward. This implies $\mathbf{v}^\star, \mathbf{v}^{\pi_1}$ and $\mathbf{v}^{\pi_2}$ must satisfy all three forms of the Bellman equations:

$$\begin{aligned} v(s) =& \max_a r^{\lambda^\star}(s,a) + \gamma \sum_{s'} p(s,a,s')v(s'), \\ =& r^{\lambda^\star}(s,\pi_1(s)) + \gamma \sum_{s'} p(s,\pi_1(s),s')v(s') = r^{\lambda^\star}(s,\pi_2(s)) + \gamma \sum_{s'} p(s,\pi_2(s),s')v(s'), \end{aligned} \tag{16}$$

for all s. Now, for any $0 \leq t \leq 1$, let π_t be the randomized policy $\pi_t = (1-t)\pi_1 + t\pi_2$. Then the value of policy π_t uniquely satisfies the following Bellman equation:

$$\begin{aligned} v^{\pi_t}(s) &= (1-t) r^{\lambda^\star}(s,\pi_1(s)) + t \cdot r^{\lambda^\star}(s,\pi_2(s)) \\ &+ \gamma \sum_{s'} \Big((1-t)p(s,\pi_1(s),s') + tp(s,\pi_2(s),s') \Big) v^{\pi_t}(s') \end{aligned} \tag{17}$$

It follows from (16) that $\mathbf{v}^\star$ satisfies (17) and is thus the value function (i.e. fixed point) of policy π_t. Thus any policy $\pi_t, 0 \leq t \leq 1$ is optimal for the MDP with adjusted reward $r_t^{\lambda^\star} = r_t - \lambda^\star c_t$ and achieves primal optimality in the sense that

$$\mathcal{R}(\pi^\star, \rho) = \mathcal{D}(\lambda^\star) = \mathcal{R}(\pi_t, \rho) - \lambda^\star \big(\mathcal{C}(\pi_t, \rho) - B \big). \tag{18}$$

Now, it follows from (11) and (12) that $\mathcal{D}^+(\lambda^\star) = B - \mathcal{C}(\pi_1, \rho) > 0$ and $\mathcal{D}^-(\lambda^\star) = B - \mathcal{C}(\pi_2, \rho) < 0$. Furthermore, $\mathcal{C}(\pi_t, \rho)$ can be shown to be a continous function of t. Thus, we must have $\mathcal{C}(\pi_t, \rho) = B$ for some $0 < t < 1$. Then such π_t satisfies not only primal feasibility but also primal optimality due to (18):

$$\mathcal{R}(\pi^\star, \rho) = \mathcal{R}(\pi_t, \rho) - \lambda^\star \big(\mathcal{C}(\pi_t, \rho) - B \big) = \mathcal{R}(\pi_t, \rho). \tag{19}$$

The claim that π_1 and π_2 differ by one state now follows from (1) and the uniqueness assumption. The other cases where one or both of $\mathcal{D}^+(\lambda^\star)$ and $\mathcal{D}^-(\lambda^\star)$ are 0 lead to either $t = 0$ or 1, which further lead to deterministic policy. The analysis is similar so we omit it. ■

Theorem 2 postulates that the maximization of the Lagrangian or penalized objective $\mathcal{R}(\pi, \rho) - \lambda^* \big(\mathcal{C}(\pi, \rho) - B \big)$ generally leads to multiple (deterministic) optimal solutions, even if the primal problem (1) has a unique optimal policy. Note that the maximization of $\mathcal{R}(\pi, \rho) - \lambda^* \big(\mathcal{C}(\pi, \rho) - B \big)$ is an unconstrained MDP, which allows us to use any classical RL methods to learn its optimal policy. The key is that in order to retrieve the primal optimal policy, we need to identify *two* optimal policies for this penalized objective, and mix them together with a search for the optimal mixture parameter t.

Before presenting practical algorithms for implementation, we first propose a straightforward theoretical procedure in Algorithm 1 that would demonstrate the asymptotic optimality of our method. For demonstration, we would simply use Q-learning on the penalized problem along with subsequent TD-learning for dual updates. However, we note that Algorithm 1 can be replaced by any type of Actor-Critic updates as in Tessler et al. (2018). Notation-wise, we use π^λ to denote the optimal deterministic policy for penalized reward $r_t^\lambda = r_t - \lambda c_t$. Given the simple dual Q-learning method described in Algorithm 1, we have the following Theorem 3. Notice the N chosen large is fixed and does not grow with iterations.

Algorithm 1 Dual Q-learning on Candidates for Mixture

Input: Dual range $0 \leq \lambda_{min} < \lambda_{max}$, discretization parameter n, maximum episode E_1 and E_2, maximum trajectory M_1 and M_2, learning rate α_e, ϵ_{greedy} for the greedy policy and discretized $\lambda_{min} = \lambda_1 < ... < \lambda_n = \lambda_{max}$.

for $i = 1$ **to** n **do**

Initialize : $e \leftarrow 0$, $\hat{Q}_e^i$, the Q-function array for storage (e.g. to 0), an estimate of $Q^i(s,a) = \mathbb{E}_{\pi^{\lambda_i}}[\sum_{t=0}^{\infty} \gamma^t (r_t - \lambda_i c_t) | s_0 = s, a_0 = a]$ and $\{\hat{v}_{cost}\}_e^i$ cost value function array for storage, an estimate of $\mathbb{E}_{\pi^{\lambda_i}}[\sum_{t=0}^{\infty} \gamma^t c_t | s_0 = s]$.

repeat

$e \leftarrow e + 1$, initialize $t \leftarrow 0$ and sample $s_0 \sim \rho$

while s_t is not terminal **and** $t \leq M_1$ **do**

Take action a_t at s_t derived from $\hat{Q}_{e-1}^i$ using ϵ_{greedy}-greedy policy and observe r_{t+1}, s_{t+1}, then let $\hat{Q}_{e-1}^i(s_t, a_t) \leftarrow \hat{Q}_{e-1}^i(s_t, a_t) + \alpha_e\big(r_{t+1} - \lambda_i c_{t+1} + \gamma \max_{a'} \hat{Q}_{e-1}^i(s_{t+1}, a') - \hat{Q}_{e-1}^i(s_t, a_t)\big)$ and update $t \leftarrow t + 1$

end while

Update $\hat{Q}_e^i \leftarrow \hat{Q}_{e-1}^i$.

until $e \geq E_1$ **or** changes in $\hat{Q}^i$ are small

$e \leftarrow 0$.

repeat

$e \leftarrow e + 1$, initialize $t \leftarrow 0$ and sample $s_0 \sim \rho$

while s_t is not terminal **and** $t \leq M_2$ **do**

$\{\hat{v}_{cost}\}_{e-1}^i(s_t) \leftarrow \{\hat{v}_{cost}\}_{e-1}^i(s_t) + \alpha_e\big(c_{t+1} + \gamma\{\hat{v}_{cost}\}_{e-1}^i(s_{t+1}) - \{\hat{v}_{cost}\}_{e-1}^i(s_t)\big)$

Update $t \leftarrow t + 1$

end while

Update $\{\hat{v}_{cost}\}_e^i \leftarrow \{\hat{v}_{cost}\}_{e-1}^i$.

until $e \geq E_2$ **or** changes in $\hat{V}_{cost}^i$ are small

Compute $\hat{\mathcal{D}}(\lambda_i) = \sum_s (\max_a \hat{Q}^i(s,a))\rho(s) + \lambda_i B$. Find $\pi^{\lambda_i}(s) = \text{argmax}_a \hat{Q}^i(s,a)$

end for

Output: $\pi_1 = \pi^{\lambda_i}$ and $\pi_2 = \pi^{\lambda_{i'}}$ where $\lambda_i = \text{argmin}\{\hat{\mathcal{D}}(\lambda_j) | \sum_s \hat{v}_{cost}^j(s)\rho(s) \leq B\}$ and $\lambda_{i'} = \text{argmin}\{\hat{\mathcal{D}}(\lambda_j) | \sum_s \hat{v}_{cost}^j \rho(s) \geq B, \pi^{\lambda_j} \neq \pi_1\}$.

Theorem 3 *Assume $\rho(s) > 0$ $\forall s$, the optimal policy $\pi^\star$ for problem (1) is unique and there exists some $\lambda^\star \in$ argmin $\mathcal{D}(\lambda)$ such that $\lambda_{min} < \lambda^\star < \lambda_{max}$. Fix $n \geq 0$, assume for each Q^i-learning problem and TD-learning problem for $1 \leq i \leq n$, every state and every*

state-action pair are visited infinitely often. Furthermore, sequence α_e *satisfies*

$$\sum_e \alpha_e = \infty \quad \textit{and} \quad \sum_e \alpha_e^2 < \infty. \tag{20}$$

Then there exists N *large enough and* ϵ_g *small enough such that if we fix* $n = N$ *and* $\epsilon_{greedy} \leq \epsilon_g$*, we will recover a pair of deterministic policies* π_1, π_2 *such that* $\pi^\star = (1-t)\pi_1 + t\pi_2$ *for some* $0 \leq t \leq 1$ *with probability 1 as the number of episode* $E_1, E_2 \to \infty$*.*

Proof Following Theorem 2, first consider the case where $\lambda^\star > 0$ is unique and $\mathcal{D}^-(\lambda^\star) < 0 < \mathcal{D}^+(\lambda^\star)$. Then, as discussed in Theorem 2, (11) and (12), there exist some $\epsilon > 0$ and policies π_1', π_2' which differ by one state such that $\pi^\star = (1-t)\pi_1' + t\pi_2'$ for some $0 < t < 1$,

$$\mathcal{D}(\lambda) = \mathcal{D}(\lambda^\star) + \mathcal{D}^+(\lambda^\star)(\lambda - \lambda^\star) = \mathcal{R}(\pi_1', \rho) - \lambda\big(\mathcal{C}(\pi_1', \rho) - B\big) \tag{21}$$

for $\lambda^\star \leq \lambda \leq \lambda^\star + \epsilon$ and some deterministic π_1' while

$$\mathcal{D}(\lambda) = \mathcal{D}(\lambda^\star) + \mathcal{D}^-(\lambda^\star)(\lambda - \lambda^\star) = \mathcal{R}(\pi_2', \rho) - \lambda\big(\mathcal{C}(\pi_2', \rho) - B\big) \tag{22}$$

for $\lambda^\star - \epsilon \leq \lambda \leq \lambda^\star$ and some deterministic π_2'. It is clear from the definition of $\mathcal{D}(\lambda)$ and our assumption on the uniqueness of $\pi^\star$ that $\pi_1' = \pi^\lambda$ for $\lambda^\star < \lambda < \lambda^\star + \epsilon$ and $\pi_2' = \pi^\lambda$ for $\lambda^\star - \epsilon < \lambda < \lambda^\star$. Then, for $n = N$ large enough, where $(\lambda_{max} - \lambda_{min})/N \leq \epsilon$, we must have some $\lambda^\star - \epsilon \leq \lambda_i \leq \lambda^\star \leq \lambda_{i+1} \leq \lambda^\star + \epsilon$ for some $1 \leq i \leq n$ and due to the strict convexity of $\mathcal{D}(\lambda)$ around $[\lambda^\star - \epsilon, \lambda^\star + \epsilon]$, we must have $\mathcal{D}(\lambda_i) < \mathcal{D}(\lambda_{i-1}) < ... < \mathcal{D}(\lambda_1)$ and $\mathcal{D}(\lambda_{i+1}) < \mathcal{D}(\lambda_{i+2}) < ... < \mathcal{D}(\lambda_n)$. Now, by the assumption on the Q-learning procedure (infinitely often visit for state-action pair under ϵ-greedy policy, the Robbins-Monro (Robbins and Monro (1985)) type condition (20)), it follows that the Q^i-learning for every $1 \leq i \leq n$ converges to the optimal Q^i value (or ϵ_{greedy}-optimal assuming *optimistic*, large initialization for Q values (Even-Dar and Mansour (2002))) and we can recover the optimal value (λ-adjusted) function $\max_a Q^i(s,a)$ with probability 1 as $E \to \infty$ (Watkins and Dayan (1992); Sutton and Barto (2018); Tsitsiklis (1994)). Thus, as $E \to \infty$, we will have $\hat{\mathcal{D}}(\lambda_i) < \hat{\mathcal{D}}(\lambda_{i-1}) < ... < \hat{\mathcal{D}}(\lambda_1)$ and $\hat{\mathcal{D}}(\lambda_{i+1}) < \hat{\mathcal{D}}(\lambda_{i+2}) < ... < \hat{\mathcal{D}}(\lambda_n)$. On the other hand, the assumption also guarantees that the TD learning on $\hat{v}_{cost}^j$ will converge to $v_{cost}^{\lambda_j}$ (or $v_{cost}^{\pi_{\epsilon_{greedy}}^{\lambda_j}}$, where $\pi_{\epsilon_{greedy}}^{\lambda_j}$ is the ϵ_{greedy} greedy policy from the optimal π^{λ_j}). If we pick $\epsilon_{greedy} > 0$ small enough, we can make $\sum_s |v_{cost}^{\pi^{\lambda_j}}(s) - v_{cost}^{\pi_{\epsilon_{greedy}}^{\lambda_j}}(s)|\rho(s)$ arbitrarily small. However, we know from the piece-wise linearity and convexity of $\mathcal{D}(\lambda)$ that, for all $\lambda_j \geq \lambda^\star$, the gradient $B - \mathcal{C}(\pi^\lambda, \rho) > 0$ which implies $\sum_s v_{cost}^{\pi^{\lambda_j}}(s)\rho(s) = C(\pi^{\lambda_j}, \rho) < B$, and we can find ϵ_{greedy} small enough such that $\sum_s v_{cost}^{\pi_{\epsilon_{greedy}}^{\lambda_j}}(s)\rho(s) < B$ and thus (in both cases) $\sum_s \hat{v}_{cost}^j(s)\rho(s) < B$ with $\lambda_{i+1} = \text{argmin}\{\hat{\mathcal{D}}(\lambda_j) | \sum_s \hat{v}_{cost}^j(s)\rho(s) \leq B\}$ implying $\pi_1 = \pi_1'$ as $E_1, E_2 \to \infty$. Similarly we can show $\pi_2 = \pi_2'$. For other cases where $\lambda_\star = 0$ and one or both of $\mathcal{D}^+(\lambda^\star)$ and $\mathcal{D}^-(\lambda^\star)$ are 0, it can be shown that the unique deterministic policy $\pi^\star$ can be recovered. ■

Theorem 3 guarantees that with suitable algorithmic parameter choices, Algorithm 1 can retrieve two candidate optimal policies such that their mixture gives rise to the optimal randomized policy for the constrained problem (1). Next we will discuss in more detail the implementation issues, including how to search for the mixture parameter.

5. Discussion and Implementation

Theorem 3 not only gives us theoretical guarantees on recovering the candidates for optimal mixtures, but also partially explains why the behavior of a direct primal dual method becomes unstable around convergence. In particular, the splitting of action forces the primal update to search for different optimal polices around the $\lambda^\star$ and makes the convergence especially difficult. To overcome such a difficulty, we use the mixing of policies which is to be explained later in this section. The discretization of dual variable λ is designed for this purpose as well. Notice this special discretization also allows for efficient parallel computing on different λ. On the other hand, the conditions can be restrictive in practice and the implementation for Algorithm 1 becomes inefficient as the accuracy parameters increase. In particular, there are several main issues concerning the implementation of Algorithm 1:

1. How to find the a reasonable set of $\lambda_{min}, \lambda_{max}$?
2. What if Algorithm 1 cannot converge to the correct pair of policies (e.g. π_1 and π_2 differ by more than one state)?
3. Given two candidate policies π_1, π_2, and the results from Theorem 2 that $\pi^\star = (1-t)\pi_1 + t\pi_2$ for some $0 \leq t \leq 1$, how do we find t?

The first point is not a major concern. As mentioned, the dual variable λ is one-dimensional and we can use many efficient RL methods such as Q-learning. In fact, we can use RCPO efficiently before we run into convergence issues, at which point we can already observe a good range of dual value λ for which the optimal $\lambda^\star$ is likely to be contained in. To address the second and third issues, we note that in both minimizing $\mathcal{D}(\lambda)$ and mixing $\pi_t = (1-t)\pi_1 + t\pi_2$, it is critical to efficiently estimate $\mathcal{C}(\pi, \rho)$ for a given policy π.

Cost Evaluation. Suppose we have found $\pi^\lambda \in \operatorname{argmax}_{\pi \in \Pi_0} \mathcal{R}(\pi, \rho) - \lambda \mathcal{C}(\pi, \rho)$. Then an estimate of $\mathcal{C}(\pi^\lambda, \rho)$ can help evaluate a sub-gradient (Boyd and Vandenberghe (2004)) of the piece-wise linear dual function $\mathcal{D}(\lambda)$, which is given by $B - \mathcal{C}(\pi^\lambda, \rho)$. This in turn helps decide a search direction for $\lambda^\star$ based on first-order optimization methods. On the other hand, when mixing the policies $\pi_t = (1-t)\pi_1 + t\pi_2$, we know from duality that

$$\mathcal{R}(\pi^\star) = \mathcal{D}(\lambda^\star) = \mathcal{R}(\pi_t, \rho) - \lambda^\star\big(\mathcal{C}(\pi_t, \rho) - B\big). \tag{23}$$

Thus, if we can find t such that $\mathcal{C}(\pi_t, \rho) = B$, it then follows from (23) that policy π_t satisfies primal feasibility and optimality simultaneously and is the solution of (1).

There are many ways to estimate $\mathcal{C}(\pi, \rho)$, e.g., TD-learning $\sum_s v_s \rho(s)$, or Monte Carlo by Sutton and Barto (2018). Thus, from now on we assume an efficient oracle $Eval_C(\pi, \rho)$ which takes as input policy π and initial distribution ρ and outputs an estimate of $\mathcal{C}(\pi, \rho)$.

Dual Variable Range. Given the oracle $Eval_C(\pi, \rho)$, we can construct algorithms that effectively select a reasonable pair of λ_{min} and λ_{max}. In particular, given a $\lambda \geq 0$, if we have found π^λ by Q-learning on function $\mathcal{D}(\lambda)$, then by the convexity of $\mathcal{D}(\lambda)$, we know if $\mathcal{C}(\pi^\lambda, \rho) > B$, it indicates $\lambda \leq \lambda^\star$ whereas if $C(\pi^\lambda, \rho) < B$, it indicates $\lambda \geq \lambda^\star$. Thus, we can make use of the oracle $Eval_C(\pi, \rho)$ to estimate $\mathcal{C}(\pi, \rho)$. However, the estimate would inevitably be corrupted by noise so we want to ensure an empirically over-budget policy π (i.e. $\mathcal{C}(\pi, \rho) > B$) is indeed over-budgeted, by setting a "safety margin" θ to account for

Algorithm 2 Dual Variable Range Selection

Input: A threshold $0 < \theta < 1$ (e.g. $\theta = 1/2$), step size λ_{step} and a tolerance for budget constraint τ.
Initialization: λ, λ_{min},λ_{max} (e.g. 0)
Find π^λ by Q-learning
if $B - \tau \leq Eval_C(\pi^\lambda, \rho) \leq B + \tau$, **then**
  Break search and accept π^λ as optimal policy.
end if
if $Eval_C(\pi^\lambda, \rho) < (1-\theta)B$ **then**
  Set $\lambda_{max} = \lambda$, Break Search and restart algorithm with $\lambda \leftarrow \lambda - \lambda_{step}$. (Also Break if $\lambda_{max} = 0$, suggesting the MDP is unconstrained.)
end if
if $Eval_C(\pi^\lambda, \rho) > (1+\theta)B$ **then**
  Set $\lambda_{min} = \lambda$. Break Search and restart algorithm with $\lambda \leftarrow \lambda + \lambda_{step}$.
end if

statistical significance. For example, if $Eval_C(\pi^\lambda, \rho) > (1+\theta)B$, then with high probability we have $\mathcal{C}(\pi^\lambda, \rho) > B$ and we can set $\lambda_{min} = \lambda$. On the other hand, if during the search we have found a policy π^λ that is close to feasibility (i.e. $\mathcal{C}(\pi^\lambda, \rho) \approx B$), then we make use of weak duality (Bertsimas and Tsitsiklis (1997)):

$$\mathcal{R}(\pi^\lambda, \rho) \approx \mathcal{R}(\pi^\lambda, \rho) - \lambda\big(\mathcal{C}(\pi^\lambda, \rho) - B\big) = \mathcal{D}(\lambda) \geq \mathcal{R}(\pi^\star, \rho),$$

and accept π^λ as a near-optimal, near-feasible solution. Of course such cases will not occur in general. Based on these discussion, we propose one possible Algorithm 2.

Feasibility Mixing. As we have discussed in (23), we need to build an oracle that given two policies π_1, π_2 with $\mathcal{C}(\pi_1, \rho) \leq B$ and $\mathcal{C}(\pi_2, \rho) \geq B$, we can find $\pi_t = (1-t)\pi_1 + t\pi_2$ satisfying $\mathcal{C}(\pi_t, \rho) = B$. Here we make use of oracle $Eval_C$ again to present an approximate algorithm that combines linear interpolation and bisection to quickly search for a feasible policy. Specifically, for the interpolation part, we notice that, for $L \leq B \leq U$, $(1-t)L + tU = B$ where $t = \frac{B-L}{U-L}$. In practice, we may use a direct bisection. Feasibility mixing is especially practical because we might only obtain approximately optimal candidate policies π_1', π_2' (i.e. they might not be the optimal pair of polices) under two dual variables λ_1' and λ_2' (i.e. they might be different from the desired λ_1 and λ_2 in Theorem 2) from Algorithm 1 that in turn might only be approximately optimal for λ_i' (meaning that $\mathcal{R}(\pi_i'.\rho) - \lambda_i'(C(\pi_i', \rho) - B) \leq \mathcal{D}(\lambda_i)$). However, based on the piecewise-linearity and the convexity of $\mathcal{D}(\lambda)$, as long as feasibility mixing is performed, it is straight-forward to show that the reward function of the mixing policy π_t satisfies $\mathcal{D}(\lambda^\star) - \mathcal{R}(\pi_t, \rho) = \mathcal{O}(\epsilon_1 \cdot \epsilon_2 \cdot \epsilon_3)$ where $\epsilon_1 = \max_{1 \leq i \leq 2} |\lambda_i - \lambda^\star|$, $\epsilon_2 = \max_{1 \leq i \leq 2} |\mathcal{D}(\lambda_i) - \mathcal{D}(\lambda^\star)|$ and $\epsilon_3 = \max_{1 \leq i \leq 2} |\mathcal{R}(\pi_i'.\rho) - \lambda_i'(C(\pi_i', \rho) - B)|$.

6. Numerical Experiments

6.1. Environment Description and Setup

We evaluate the proposed algorithms on a real world dataset collected from [anonymized for review purpose] during a sponsored search campaign portfolio which spans over six months

Algorithm 3 Feasibility Mixing

Input: policies π_1, π_2 with $Eval_C(\pi_1, \rho) \leq B$, $Eval_C(\pi_2, \rho) \geq B$, a tolerance for the budget τ
Initialize: $t \leftarrow \frac{B - Eval_C(\pi_1, \rho)}{Eval_C(\pi_2, \rho) - Eval_C(\pi_1, \rho)}$, (or $i \leftarrow 1$, $t_i \leftarrow 1/2$ for direct bisection)
Set policy $\pi_t = (1-t)\pi_1 + t\pi_2$
if $B - \tau \leq Eval_C(\pi_t, \rho) \leq B + \tau$, **then**
  Break search and accept π_t as optimal policy.
end if
if $Eval_C(\pi_t, \rho) < B - \tau$ **then**
  Update $\pi_1 \leftarrow \pi_t$ and $t \leftarrow \frac{B - Eval_C(\pi_1, \rho)}{Eval_C(\pi_2, \rho) - Eval_C(\pi_1, \rho)}$, (or $i \leftarrow i + 1$ $t \leftarrow t + 1/2^i$)
end if
if $Eval_C(\pi_t, \rho) > B + \tau$ **then**
  Update $\pi_2 \leftarrow \pi_t$ and $t \leftarrow \frac{B - Eval_C(\pi_1, \rho)}{Eval_C(\pi_2, \rho) - Eval_C(\pi_1, \rho)}$, (or $i \leftarrow i + 1$ $t \leftarrow t - 1/2^i$)
end if
Output: t (or π_t).

and contains over a million distinct user search trajectories. The dataset provides ad click records of anonymous users before conversion with their corresponding timestamps. The ad click records are associated with a matching of the user's query with a keyword group. This particular dataset has ten different keyword groups each containing hundreds of keywords. Similar to other advertiser-specific data, we do not directly observe the events in which the users did not click on the ad. Similarly, the data does not record the searches for which the ad was not shown to the user for any reason such as low bid values, budget constraint, etc. On the other hand, a smaller version of the experiment allows a clear validation of our key theorem on policy splitting, because the optimal policy and its two splitting policies in a CMDP is difficult to recover in complicated, large MDPs. However, we note that our algorithm allows for larger experiments in a model-free algorithm setting.

For the experiment setup, we first retrieve the cost information for our sampled dataset with CPC (cost per click) metric averaged at the keyword group level for the similar time period as the collected data. The average cost for the ten keyword groups in our experiment is estimated to be $[0.2, 0.4, 0.25, 0.5, 0.3, 0.6, 0.5, 0.3, 0.3, 0.4]$ in dollars. Additionally, the reward for converting a user is estimated to be worth \$10 for this campaign. Then, we follow the framework in Archak et al. (2012) to establish a CMDP. In particular, user state represents the matching of the user's last query with any of the keyword groups that translates to ten states in our experiment. Then, our action space is binary and includes "advertise" and "do not advertise" actions and transition probabilities between states are directly estimated from the data. In order to overcome the issue of estimating transition probabilities for "do not advertise", we follow the remedy suggested by Archak et al. (2012). That is, we assume the transitions between states are independent of the ad presented to the user if the time period between two consecutive searches is longer than one day. Moreover, we bundle all possible advertisement keywords in 10 keyword groups. Finally, we add 4 states, which contain a beginning state, a conversion state, a non-conversion state and eventually the final state to incorporate the situation where users may convert temporarily but eventually become disinterested in the ad push (see Figure 1). Consequently, we have

14 states in our environment in total with a transition probability matrix in $\mathbb{R}^{2\times14\times14}$. We run Algorithm 1 with hyper-parameters $\lambda_{min} = 0$, $\lambda_{max} = 2$, $M_1 = 10^5$, $E_1 = 3.5 \times 10^5$, $M_2 = 10^4$, $E_2 = 2 \times 10^5$, $\alpha_e = \frac{9}{9+0.2e}$, $\epsilon_{greedy} = 0.2$, $B = 0.45$, $\gamma = 0.6$, $\tau = 10^{-4}$ and early stopping criterion requires $\|\cdot\|_\infty$ norm within 10^{-4}. The metrics here for reward and cost are averaged accumulative rewards and averaged accumulative costs defined in (1), In order to show the advantage of our method, we pick RCPO as a baseline. For the sake of fairness, all experiments are implemented in `Python 3.7` and executed on a standard `1.7 GHz Dual-Core Intel Core i7`.

Figure 1: MDP on advertisement (red node denotes a conversion/non-conversion state).

6.2. Algorithm Performances

Figure 2(a) demonstrates the averaged accumulative costs of the two candidate policies (Policy 1 and Policy 2) selected by Algorithm 1. Moreover, for each λ, $\mathcal{D}(\lambda)$ can be computed efficiently with RL-methods and its convexity is shown in Figure 2(b). After identifying two candidate policies from Algorithm 1, we run Algorithm 3 which mixes the policies to satisfy the budget constraint. As shown in Figure 2(c), we start with Policies 1 and 2 corresponding to $t = 0$ and 1 and use a simple bisection to search for the target value of t. Figure 2(d) shows the searching process stabilizes after a few iterations and the corresponding long-run budget for different mixture policies gradually converges to the target budget value. As we expect, in this case the optimal policy comes from the mixture, one policy going over budget and the other under.

To show the robustness of the procedure, we perform a large number of experiments to see the effectiveness of Algorithm 1 in recovering the correct pair of optimal policies. Figure 3 (a)(b) shows that, in this example, the correct pair of policies can be recovered in 78% of the experimental repetitions. More importantly, we plot the distribution of the reward-budget pairs of the resulting mixture policy across all experiments and show that, among the occasions Algorithm 1 does not pick the correct pair, the resulting mixture is still approximately optimal and feasible, within a controllable error margin, showing the stability of the procedure. In addition, we compare the performances between our method and RCPO. As shown in Figure 3(c), the learning curve on rewards of RCPO is between the learning curves of two candidate policies. However, as shown in Table 1 and 3(d), our mixing method can find a randomized policy that has a higher average accumulative reward in lesser time. As discussed, RCPO converges fast initially, yet the convergence slows down and exhibits a zigzag motion when it is quite close to the optimal λ. Advantageously, our mixing method bypass this problem around convergence.

Methods	Accumulative Rewards	Accumulative Costs	Clock Time (s)
RCPO	1.229	**0.405**	924.961
Policy Mixing (τ = 1e-4)	1.271	**0.449**	839.708
Policy Mixing (τ = 1e-3)	**1.277**	**0.449**	702.927
Policy Mixing (τ = 1e-2)	1.276	**0.449**	**558.763**

Table 1: Performance comparison summary (Bold means either the best or valid).

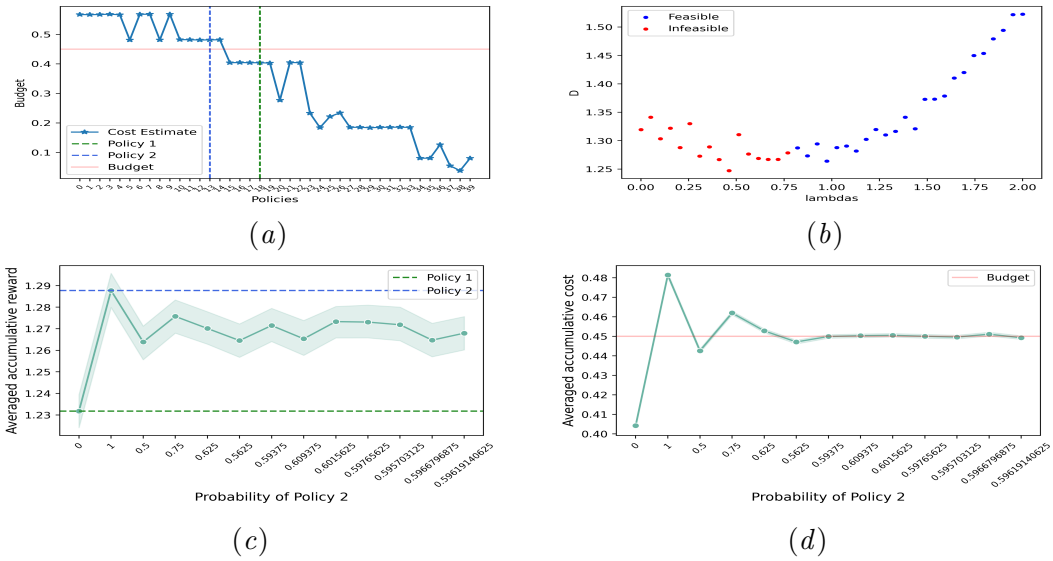

Figure 2: (a) Budget estimates of policies with different λ; (b) Convexity of $D(\lambda)$; (c) Accumulative adjusted rewards during policy mixing; (d) Accumulative costs during policy mixing.

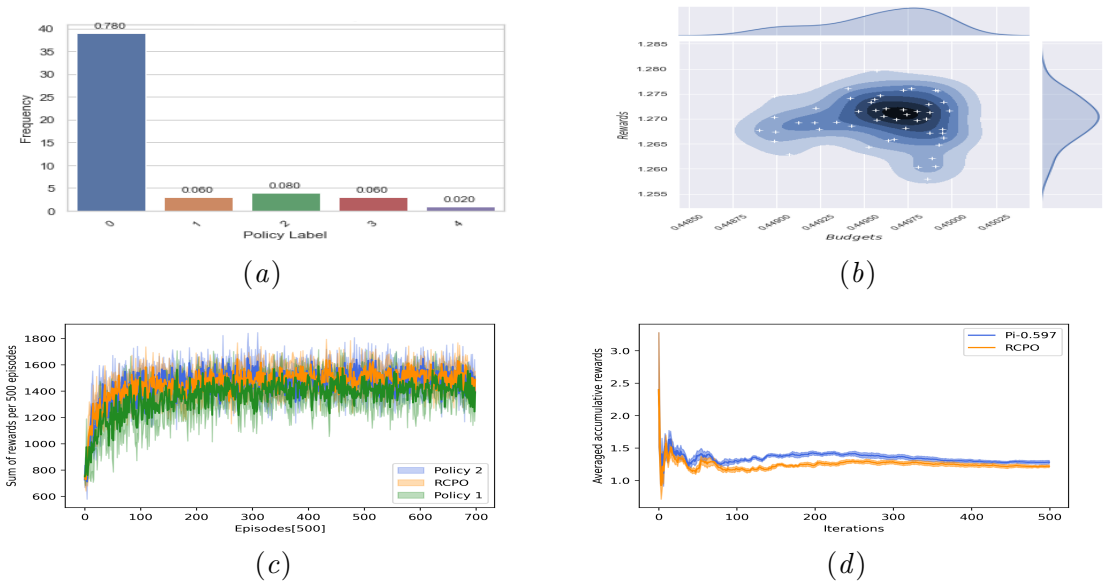

Figure 3: (a) Occurrences of policy pairs (Label 0 denotes valid policy pairs with only one state with different actions); (b) Joint distribution of averaged reward and cost, where each dot represents each experiment and the heat map is estimated from kernel density estimation; (c) Learning curves of policies 1, 2 and RCPO. A tick on x-axis denotes 500 episodes and y-axis denotes the total rewards for every 500 episodes; (d) MC evaluation of averaged accumulative rewards.

7. Conclusion

We focus on solving CMDPs which, although arise frequently in practice, are not amenable to efficient solution techniques offered by most established RL-methods on unconstrained problems. Through incorporating the "splitting" property of CMDP in a Lagrangian formulation, our approach investigates the potential issues around convergence for current primal-dual RL-methods and offers a suitable alternative. The approach aims to identify two candidate optimal policies which through mixing would result in an optimal randomized policy of the CMDPs. We illustrate our performances through an online advertising problem with budget calibrated by real-world data.

References

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 22–31. JMLR. org, 2017.

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

Nikolay Archak, Vahab Mirrokni, and S Muthukrishnan. Budget optimization for online campaigns with positive carryover effects. In *International Workshop on Internet and Network Economics*, pages 86–99. Springer, 2012.

Richard Bellman. *Dynamic programming*. Courier Corporation, 2013.

Dimitris Bertsimas and John N Tsitsiklis. *Introduction to linear optimization*, volume 6. Athena Scientific Belmont, MA, 1997.

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Qingpeng Cai, Aris Filos-Ratsikas, Pingzhong Tang, and Yiwei Zhang. Reinforcement mechanism design for e-commerce. In *Proceedings of the 2018 World Wide Web Conference*, pages 1339–1348, 2018.

Yinlam Chow, Mohammad Ghavamzadeh, Lucas Janson, and Marco Pavone. Risk-constrained reinforcement learning with percentile risk criteria. *Journal of Machine Learning Research*, 2018.

Eyal Even-Dar and Yishay Mansour. Convergence of optimistic and incremental q-learning. In *Advances in neural information processing systems*, pages 1499–1506, 2002.

Eugene A Feinberg and Uriel G Rothblum. Splitting randomized stationary policies in total-reward markov decision processes. *Mathematics of Operations Research*, 37(1):129–153, 2012.

Peter Geibel. Reinforcement learning for mdps with constraints. In *European Conference on Machine Learning*, pages 646–653. Springer, 2006.

Jongmin Lee, Youngsoo Jang, Pascal Poupart, and Kee-Eung Kim. Constrained bayesian reinforcement learning via approximate linear programming.

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

Herbert Robbins and Sutton Monro. A stochastic approximation method. In *Herbert Robbins Selected Papers*, pages 102–109. Springer, 1985.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Chen Tessler, Daniel J Mankowitz, and Shie Mannor. Reward constrained policy optimization. *arXiv preprint arXiv:1805.11074*, 2018.

John N Tsitsiklis. Asynchronous stochastic approximation and q-learning. *Machine learning*, 16(3):185–202, 1994.

Xiting Wang, Yiru Chen, Jie Yang, Le Wu, Zhengtao Wu, and Xing Xie. A reinforcement learning framework for explainable recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 587–596. IEEE, 2018.

Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

Di Wu, Xiujun Chen, Xun Yang, Hao Wang, Qing Tan, Xiaoxun Zhang, Jian Xu, and Kun Gai. Budget constrained bidding by model-free reinforcement learning in display advertising. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 1443–1451, 2018.