Reward is enough for convex MDPs

Abstract

Maximising a cumulative reward function that is Markov and stationary, *i.e.*, defined over state-action pairs and independent of time, is sufficient to capture many kinds of goals in a Markov Decision Process (MDP) based on the Reinforcement Learning (RL) problem formulation. However, not all goals can be captured in this manner. Specifically, it is easy to see that Convex MDPs in which goals are expressed as convex functions of stationary distributions cannot, in general, be formulated in this manner. In this paper, we reformulate the convex MDP problem as a min-max game between the policy and cost (negative reward) players using Fenchel duality and propose a meta-algorithm for solving it. We show that the average of the policies produced by an RL agent that maximizes the non-stationary reward produced by the cost player converges to an optimal solution to the convex MDP. Finally, we show that the meta-algorithm unifies several disparate branches of reinforcement learning algorithms in the literature, such as apprenticeship learning, variational intrinsic control, constrained MDPs, and pure exploration into a single framework.

1. Introduction

In Reinforcement Learning (RL), an agent learns how to map situations to actions so as to maximize a cumulative numerical reward signal. The learner is not told which actions to take, but instead must discover which actions lead to the most cumulative reward (Sutton and Barto, 2018). Mathematically, the RL problem can be written as finding a policy whose state occupancy has the largest inner product with a reward vector, known as the dual linear problem of RL (Puterman, 1984), *i.e.*, the goal of the agent is to solve

$$\text{RL:}\quad \max_{d_\pi \in \mathcal{K}} \sum_{s,a} r(s,a) d_\pi(s,a), \tag{1}$$

where d_π is the state-action stationary distribution induced by policy π and $\mathcal{K}$ is the set of admissible stationary distributions (Definition 1). A significant body of work is dedicated to solving the RL problem efficiently in challenging domains (Mnih et al., 2015; Silver et al., 2017). However, not all decision making problems of interest take this form. In particular we consider the more general *convex* RL problem,

$$\text{Convex RL:}\quad \min_{d_\pi \in \mathcal{K}} f(d_\pi), \tag{2}$$

where $f : \mathcal{K} \to \mathbb{R}$ is a convex function. Sequential decision making problems that take this form include Apprenticeship Learning (AL), diverse skill discovery, pure exploration, and constrained MDPs, among others; see Table 1. In this paper we prove the following claim:

We can solve Eq. (2) *by using any algorithm that solves Eq.* (1) *as a subroutine.*

In other words, any algorithm that solves the standard RL problem can be used to solve the more general convex RL problem. More specifically, we make the following contributions.

First, we adapt the meta-algorithm of Abernethy and Wang (Abernethy and Wang, 2017) for solving Eq. (2). The key idea is to use Fenchel duality to convert the convex RL problem into a two-player zero-sum game between the agent (henceforth, *policy player*) and an adversary that produces rewards (henceforth, *cost player*) that the agent must maximize (Abernethy and Wang, 2017). From the agent's point of view, the game is bilinear, and so for fixed rewards produced by the adversary the problem reduces to the standard RL problem with non-stationary reward (Fig. 1). Our main result is that the average of the policies produced by the policy player converges to a solution to the convex RL problem (Eq. (2)).

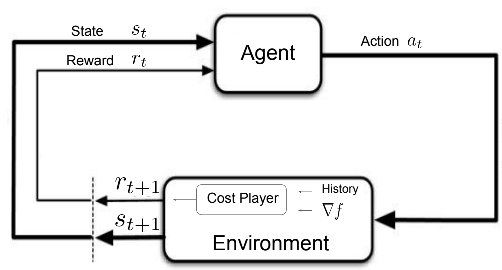


Figure 1: Convex MDP as an RL problem

Second, we explain how to use RL algorithms to implement policy players. The best response, for example, can be implemented as an RL algorithm that solves an RL problem in each iteration. The caveat here is that, for a given sample budget, RL algorithms only find the best response approximately. Instead, we propose a more sample efficient policy player that uses a standard RL algorithm (e.g., (Jaksch et al., 2010; Shani et al., 2020b)), and computes an optimistic policy w.r.t the non-stationary reward at each iteration. In other words, we use algorithms that were developed to achieve low regret in the standard RL setup, to achieve low regret as policy players. Since they achieve low regret w.r.t any

Convex objective f	Cost player	Policy player	Application
$\lambda \cdot d_\pi$	FTL	RL	(Standard) RL with $-\lambda$ as stationary reward function
$\|\|d_\pi - d_E\|\|_2^2$	FTL	Best response	Apprenticeship learning (AL) (Abbeel and Ng, 2004; Zahavy et al., 2020)
$d_\pi \cdot \log(d_\pi)$	FTL	Best response	Pure exploration* (Hazan et al., 2019)
$\|\|d_\pi - d_E\|\|_\infty$	OMD	Best response	AL (Syed et al., 2008; Syed and Schapire, 2008)
$\mathbb{E}_c\left[\lambda c \cdot (d_\pi - d_E(c))\right]^\dagger$	OMD	Best response	Inverse RL in contextual MDPs (Belogolovsky et al., 2021)
$\lambda_1 \cdot d_\pi$, s.t. $\lambda_2 \cdot d_\pi \le c$	OMD	RL	Constrained MDPs (Altman (1999); Szepesvári (2020); Borkar (2005); Tessler et al. (2019), Efroni et al. (2020); Calian et al. (2021); Bhatnagar and Lakshmanan (2012))
$\text{dist}(d_\pi, C)^{\dagger\dagger}$	OMD	Best response	Feasibility of convex-constrained MDPs (Miryoosefi et al., 2019)
$\min_{\lambda_1,\ldots,\lambda_k} d_{\pi^k} \cdot \lambda_k$	OMD	RL	Adversarial Markov Decision Processes (Rosenberg and Mansour, 2019)
$\max_{\lambda\in\Lambda} \ \lambda \cdot (d_\pi - d_E)$	OMD	RL	Online AL (Shani et al., 2021),Wasserstein GAIL (Xiao et al., 2019; Zhang et al., 2020b)
$\text{KL}(d_\pi\|\|d_E)$	FTL	RL	GAIL (Ho and Ermon, 2016), state marginal matching (Lee et al., 2019),
$-\mathbb{E}_z\text{KL}(d_{\pi^z}\|\|\mathbb{E}_k d_{\pi^k})^\ddagger$	FTL	RL	Diverse skill discovery (Gregor et al. (2017); Eysenbach et al. (2019); Hausman et al. (2018), Florensa et al. (2016); Tirumala et al. (2020); Achiam et al. (2018))

Table 1: Instances of Algorithm 1 in various convex MDPs. * as well as other KL divergences. † c is a context variable. †† C is a convex set. ‡ f is concave. See Sections 4 & 5 for more details.

sequence of rewards, they also achieve that w.r.t the rewards that are generated by the cost player, and as a result, they are guaranteed to approximate the policy that minimizes the function f. Inspired by this principle, we also propose a recipe for using (Deep-RL) DRL agents to solve convex MDPs: provide the agent non-stationary rewards from the cost player and the RL agent code base does not require any modifications. We explore this principle in our experiments.

Finally, we show that choosing specific algorithms for the policy and cost players unifies several disparate branches of RL problems, such as apprenticeship learning, variational intrinsic control, constrained MDPs, and pure exploration into a single framework, as we summarize in Table 1.

2. Reinforcement learning preliminaries

In RL an agent interacts with an environment over a number of time steps and seeks to maximize its cumulative reward. We consider two cases, the average reward case and the discounted case. The Markov decision process (MDP) is defined by the tuple (S, A, P, R) for the average reward case and by the tuple $(S, A, P, R, \gamma, d_0)$ for the discounted case. We assume an infinite horizon, finite state-action problem where initially, the agent is sampled according to $s_0 \sim d_0$, then at each time t the agent is in state $s_t \in S$, selects action $a_t \in A$ according to some policy $\pi(s_t, \cdot)$, receives reward $r_t \sim R(s_t, a_t)$ and transitions to new state $s_{t+1} \in S$ according to the probability distribution $P(\cdot, s_t, a_t)$. The two performance metrics we consider are given by

$$J_\pi^{\text{avg}} = \lim_{T\to\infty} \frac{1}{T}\mathbb{E}\sum_{t=1}^{T} r_t, \quad J_\pi^\gamma = (1-\gamma)\mathbb{E}\sum_{t=1}^{\infty}\gamma^t r_t. \quad (3)$$

The goal of the agent is to find a policy that maximizes J_π^{avg} or J_π^γ. Any stationary policy π induces a *state-action occupancy measure* d_π, which relates to how often the agent visits each state-action when following π. Depending on whether the goal is average reward or discounted reward the definition changes slightly. Let $\mathbb{P}_\pi(s_t = \cdot)$ be the probability measure over states at time t under policy π, then

$$d_\pi^{\text{avg}}(s,a) = \lim_{T\to\infty}\frac{1}{T}\mathbb{E}\sum_{t=1}^{T}\mathbb{P}_\pi(s_t = s)\pi(s,a),$$
$$d_\pi^\gamma(s,a) = (1-\gamma)\mathbb{E}\sum_{t=1}^{\infty}\gamma^t\mathbb{P}_\pi(s_t = s)\pi(s,a).$$

With these, we can rewrite the objective in Eq. (1) in terms of the occupancy measure using the following well-known result, which for completeness we prove in Appendix A.

Proposition 1. *For both the average and the discounted case, the agent objective function Eq.* (3) *can be written in terms of the occupancy measure as* $J_\pi = \sum_{s,a} r(s,a)d_\pi(s,a)$.

Given the occupancy measure we can recover the policy using $\pi(s,a) = d_\pi(s,a)/\sum_a d_\pi(s,a)$. Accordingly, in this paper we shall formulate the RL problem using the state-action occupancy measure, in which case both the standard RL problem (Eq. (1)) and the convex RL problem (Eq. (2)) are convex optimization problems. For the purposes of this manuscript we do not make a distinction between the average and discounted settings, other than through the convex polytopes of feasible occupancy measures, which we define next.

Definition 1 (State-action occupancy's polytope (Puterman, 1984))**.** *For the average reward case the set of admissible state-action occupancies is*

$$\mathcal{K}_{\text{avg}} = \{d_\pi \mid d_\pi \ge 0, \ \sum_{s,a} d_\pi(s,a) = 1,$$
$$\sum_a d_\pi(s,a) = \sum_{s',a'} P(s,s',a')d_\pi(s',a') \ \forall s \in S\},$$

and for the discounted case it is given by

$$\mathcal{K}_\gamma = \{d_\pi \mid d_\pi \ge 0, \ \sum_a d_\pi(s,a) = (1-\gamma)d_0(s)$$
$$+ \gamma\sum_{s',a'} P(s,s',a')d_\pi(s',a') \ \forall s \in S\}.$$

3. A Meta Algorithm for Solving Convex MDPs via RL

To solve the convex RL problem (Eq. (2)) we need to discover an occupancy measure d_π (and the associated policy) that minimizes the function f. Since both $f: \mathcal{K} \to \mathbb{R}$ and the set $\mathcal{K}$ are convex this is a convex optimization problem. However, it is a significantly challenging one due to the nature of learning about the environment through stochastic interactions. In this section we show how to reformulate the convex RL problem (Eq. (2)), such that standard RL algorithms can be used to solve it. Doing so will allow us to build on a significant body of work that provably solve the standard RL problem. To do that we will need the following definition.

Definition 2 (Fenchel conjugate)**.** *For a function $f: \mathbb{R}^n \to \mathbb{R} \cup \{-\infty, \infty\}$, its Fenchel conjugate is $f^*: \mathbb{R}^n \to \mathbb{R} \cup \{-\infty, \infty\}$ defined as $f^*(x) := \sup_y x \cdot y - f(y)$.*

Remark 1. *The Fenchel conjugate function f^* is always convex (when it exists) even if f is not. Furthermore, the biconjugate $f^{**} := (f^*)^*$ equals f if and only if f is convex and lower semi-continuous.*

Using this we can rewrite the convex RL problem (Eq. (2)) as

$$\begin{aligned} \boldsymbol{f}^\star = \min_{d_\pi \in \mathcal{K}} f(d_\pi) &= \min_{d_\pi \in \mathcal{K}} \max_\lambda \left(\lambda^\top d_\pi - f^*(\lambda)\right) \\ &= \max_\lambda \min_{d_\pi \in \mathcal{K}} \left(\lambda^\top d_\pi - f^*(\lambda)\right) \end{aligned} \tag{4}$$

where we were able to swap the order of minimization and maximization using the minimax theorem (Von Neumann, 1928). This is a convex-concave saddle-point problem and a zero-sum two-player game (Osborne and Rubinstein, 1994; O'Donoghue et al., 2020a). With this we define the Lagrangian as

$$\mathcal{L}(d_\pi, \lambda) := \lambda^\top d_\pi - f^*(\lambda)$$

For a fixed λ minimizing the Lagrangian is a standard RL problem of the form of Eq. (1), *i.e.*, equivalent to maximizing a reward $r = -\lambda$. Thus, one might hope that by producing an optimal dual variable $\lambda^\star$ we could simply solve $d_\pi^\star = \operatorname{argmin}_{d_\pi \in \mathcal{K}} \mathcal{L}(\cdot, \lambda^\star)$. However the next lemma states that this is not possible in general.

Lemma 1. *There exists an MDP M and convex function f for which there is no stationary reward $r \in \mathbb{R}^{S \times A}$ such that $\arg\max_{d_\pi \in \mathcal{K}} d_\pi \cdot r = \arg\min_{d_\pi \in \mathcal{K}} f(d_\pi)$.*

To see this consider the fact that for any reward r there is a deterministic policy that optimizes the reward (Puterman, 1984), but for some choices of f no deterministic policy is optimal, *e.g.*, when f is the negative entropy function. In other words, even if we have access to an optimal dual-variable we cannot simply use it to recover the stationary distribution that solves the convex RL problem in general (though doing so does yield the optimal objective value $\boldsymbol{f}^\star$).

To overcome this issue we develop an algorithm that generates a *sequence* of rewards $\{r^k\}_{k \in \mathbb{N}}$ and a *sequence* of policies $\{\pi^k\}_{k \in \mathbb{N}}$ such that the average converges to an optimal policy for Eq. (2), *i.e.*, $(1/K)\sum_{k=1}^K d_{\pi^k} \to d_\pi^\star \in \arg\min_{d_\pi \in \mathcal{K}} f(d_\pi)$. To do so, we adapted the meta-algorithm from (Abernethy and Wang, 2017) to solve the minimax problem Eq. (4); this is described in Algorithm 1. It is a meta-algorithm since it depends on the individual algorithms employed by the policy and cost players, denoted Alg_π and Alg_λ. The reinforcement learning algorithm Alg_π takes as input a reward vector and returns a state-action occupancy measure d_π (*e.g.*, it might return the optimal d_π for that reward). We allow the algorithm Alg_λ to be a more general function of the entire history. We discuss concrete examples of Alg_π and Alg_λ in Section 4.

Algorithm 1 Meta Algorithm for convex RL

1: **Input:** convex-concave payoff $\mathcal{L}: \mathcal{K} \times \Lambda \to \mathcal{R}$, algorithms $\text{Alg}_\lambda, \text{Alg}_\pi$, $K \in \mathbb{N}$
2: **for** $k = 1, \ldots, K$ **do**
3: $\quad\lambda^k = \text{Alg}_\lambda(d_\pi^1, \ldots, d_\pi^{k-1}; \mathcal{L})$
4: $\quad d_{\pi^k} = \text{Alg}_\pi(-\lambda^k)$
5: **end for**
6: Return $\bar{d}_\pi^K = \frac{1}{K}\sum_{k=1}^K d_{\pi^k}, \bar{\lambda}^K = \frac{1}{K}\sum_{k=1}^K \lambda^k$

In order to analyze this algorithm and select the algorithms $\text{Alg}_\lambda, \text{Alg}_\pi$ we will need a small detour into online convex optimization (OCO). In OCO, a learner is presented with a sequence of K convex loss functions $\ell_1(\cdot), \ell_2(\cdot), \ldots, \ell_K(\cdot): \mathcal{K} \to \mathbb{R}$ and at each round k must select a point $x_k \in \mathcal{K}$ after which it suffers a loss of $\ell_k(x_k)$. At time period k the learner is assumed to have perfect knowledge of the loss functions $\ell_1, \ldots, \ell_{k-1}$. The learner wants to minimize its *average regret*, defined as

$$\bar{R}_K := \frac{1}{K}\left(\sum_{k=1}^K \ell_k(x_k) - \min_{x \in \mathcal{K}} \sum_{k=1}^K \ell_k(x)\right).$$

In the context of convex reinforcement learning and meta-algorithm 1, the loss functions for the cost player are $\ell_\lambda^k = \mathcal{L}(\cdot, \lambda^k)$, and for the policy player are $\ell_\pi^k = -\mathcal{L}(d_\pi^k, \cdot)$, with associated average regrets $\bar{R}_K^\pi$ and $\bar{R}_K^\lambda$. This brings us to the following theorem.

Theorem 1 (Theorem 5, (Abernethy and Wang, 2017))**.** *Assume that Alg_π and Alg_λ have guaranteed average regret bounded as $\bar{R}_K^\pi \leq \epsilon_K$ and $\bar{R}_K^\lambda \leq \delta_K$, respectively. Then Algorithm 1 outputs $\bar{d}_\pi^K$ and $\bar{\lambda}^K$ satisfying $\min_{d_\pi \in \mathcal{K}} \mathcal{L}(d_\pi, \bar{\lambda}^K) \geq \boldsymbol{f}^\star - \epsilon_K - \delta_K$ and $\max_{\lambda \in \Lambda} \mathcal{L}(\bar{d}_\pi^K, \lambda) \leq \boldsymbol{f}^\star + \epsilon_K + \delta_K$.*

This theorem tells us that so long as the RL algorithm we employ has guaranteed low-regret, and assuming we choose a reasonable low-regret algorithm for deciding the costs, then the meta-algorithm will produce a solution to the convex RL problem (Eq. (2)) to any desired tolerance, this is because $\boldsymbol{f}^\star \leq f(\bar{d}_\pi^K) = \max_\lambda \mathcal{L}(\bar{d}_\pi^K, \lambda) \leq \boldsymbol{f}^\star + \epsilon_K + \delta_K$.

For example, we shall later present algorithms that have regret bounded as $\epsilon_K = \delta_K \leq O(1/\sqrt{K})$, in which case we have

$$f(\bar{d}_\pi^K) - \boldsymbol{f}^\star \leq O(1/\sqrt{K}). \tag{5}$$

Fenchel dual in related work. In (Zhang et al., 2020a), the authors proposed a policy gradient algorithm for convex MDPs in which each step of policy gradient involves solving a new saddle point problem (formulated using the Fenchel dual). This is different from our approach that involves solving a single saddle point problem iteratively, and furthermore we do not need to commit to a specific RL algorithm. Moreover, for the convergence guarantee (Zhang et al., 2020a, Theorem 4.5) to hold, the saddle point problem has to be solved exactly, while in practice it is only solved approximately (Zhang et al., 2020a, Algorithm 1), which hinders its sample efficiency. We discuss a similar scenario in the context of approximating the best response in Section 4. Fenchel duality was also used in off policy evaluation (OPE) in (Nachum et al., 2019; Yang et al., 2020). The difference between these works and ours is that we train a policy to minimize an objective, while in OPE a target policy is fixed and its value is estimated from data that is produced by a behaviour policy.

Non-convex f. Remark 1 implies that the game $\max_\lambda \min_{d_\pi \in \mathcal{K}} \left(\lambda^\top d_\pi - f^*(\lambda)\right)$ is concave-convex for any function f, so we can solve it with Algorithm 1. From weak duality, we get that the output of Algorithm 1, $\bar{d}_\pi, \bar{\lambda}$, is a lower bound on the optimal solution $\boldsymbol{f}^\star$. In addition, we know that $f(d_\pi)$ is always an upper bound on $\boldsymbol{f}^\star$, thus we get an upper bound and a lower bound on the optimal value: $\mathcal{L}(\bar{d}_\pi, \bar{\lambda}) \leq \boldsymbol{f}^\star \leq f(\bar{d}_\pi)$. When the function f is convex, strong duality implies that the duality gap is zero, so solving the game in Eq. (4) implies solving the convex RL problem as Theorem 1 states.

3.1. Extending to Convex MDPs with Convex constraints

We have restricted the presentation so far to unconstrained convex problems, in this section we generalize the above results to the constrained case. The problem we consider is

$$\min_{d_\pi \in \mathcal{K}} f(d_\pi) \quad \text{subject to} \quad g_i(d_\pi) \leq 0, \quad i = 1, \dots m,$$

where f and the constraint functions g_i are convex. Previous work focused on the case that both f and g_i are linear (Altman, 1999; Szepesvári, 2020; Borkar, 2005; Tessler et al., 2019; Efroni et al., 2020; Calian et al., 2021; Bhatnagar and Lakshmanan, 2012). We can use the same Fenchel dual machinery we developed before, but now taking into account the constraints. Consider the Lagrangian

$$\begin{aligned} L(d_\pi, \mu) &= f(d_\pi) + \sum\nolimits_{i=1}^m \mu_i g_i(d_\pi) \\ &= \max_\nu \left(\nu^\top d_\pi - f^*(\nu)\right) \\ &+ \sum\nolimits_{i=1}^m \mu_i \max_{v_i} \left(d_\pi v_i - g_i^*(v_i)\right). \end{aligned}$$

over dual variables $\mu \geq 0$, with new variables v_i and ν. At first glance this does not look convex-concave, however we can introduce new variables $\zeta_i = \mu_i v_i$ to obtain

$$\begin{aligned} L(d_\pi, \mu, \nu, \zeta_1, \dots, \zeta_m) &= \nu^\top d_\pi - f^*(\nu) \\ &+ \sum\nolimits_{i=1}^m \left(d_\pi \zeta_i - \mu_i g_i^*(\zeta_i/\mu_i)\right). \end{aligned}$$

This is convex-concave in $d_\pi, (\nu, \mu, \zeta_1, \dots, \zeta_m)$, since it includes the perspective transform of the functions g_i (Boyd and Vandenberghe, 2004). Again the Lagrangian involves a cost vector, $\nu + \sum_{i=1}^m \zeta_i$, linearly interacting with d_π, and therefore we can use the algorithms we shall develop for the convex MDP case with minor adjustment to solve the more general constrained convex MDP case.

4. Policy and cost players for convex MDPs

In this section we present several algorithms for the policy and cost players that can be used in Algorithm 1. Any combination of these algorithms is valid and will come with different practical and theoretical performance. In the next section we show that several well known methods in the literature correspond to particular choices of cost and policy players and so fall under our framework.

4.1. Cost player

Follow The Leader (FTL) is a classic OCO algorithm that selects λ_k to be the best point in hindsight. In the special case of convex MDPs, as defined in Eq. (4), FTL has a simpler form:

$$\begin{aligned} \lambda^k &= \arg\max_\lambda \sum\nolimits_{j=1}^{k-1} \mathcal{L}(d_\pi^j, \lambda) \qquad (6) \\ &= \arg\max_\lambda -\lambda \cdot \sum\nolimits_{j=1}^{k-1} d_{\pi^k} + f^*(\lambda) = \nabla f(\bar{d}_\pi^{k-1}), \end{aligned}$$

where the last equality follows from the fact that $(\nabla f^*)^{-1} = \nabla f$ (Rockafellar, 1970). Thus, for the FTL cost player, λ^k is the gradient of the function f evaluated on $\bar{d}_\pi^{k-1} = \sum_{j=1}^{k-1} d_{\pi^k}$ (the average of the state occupancies of the last $(k-1)$ policies). The average regret of FTL is guaranteed to be $\bar{R}_K \leq c/\sqrt{K}$ in general (Hazan et al., 2007). In some cases, and specifically when the set $\mathcal{K}$ is a polytope and the function f is strongly convex, FTL can enjoy logarithmic or even constant regret; see (Huang et al., 2016; Hazan et al., 2007) for more details.

Online mirror descent (OMD), an online version of mirror decent (Nemirovskij and Yudin, 1983; Beck and Teboulle, 2003) with the following iterates

$$\begin{aligned} \lambda^k = \arg\max_\lambda &(\nabla_{\lambda'} \mathcal{L}(d_\pi^{k-1}, \lambda')|_{\lambda'=\lambda^{k-1}} \cdot (\lambda - \lambda^{k-1}) \\ &+ \alpha_k B_r(\lambda, \lambda^{k-1})), \end{aligned}$$

where α_k is a learning rate and B_r is a Bregman divergence. For $B_r(x) = 0.5||x||_2^2$, we get online gradient descent (Zinkevich, 2003, OGD) and for $B_r(x) = x \cdot \log(x)$

we get the multiplicative weights (Freund and Schapire, 1997) as special cases. We also note that OMD is equivalent to a linearized version of Follow the Regularized Leader (FTRL) (McMahan, 2011; Hazan, 2016). Finally, the average regret of OMD is $\bar{R}_K \le c/\sqrt{K}$, see (Hazan, 2016).

4.2. Policy player

Best Response. In OCO, the best response is to simply ignore the history and play the best option on the current round, which has guaranteed average regret bound of $\bar{R}_K \le 0$. When applied to Eq. (4), it is possible to find the best response d_π^k using standard RL techniques since

$$d_{\pi^k} = \arg\min_{d_\pi \in \mathcal{K}} \mathcal{L}_k(d_\pi, \lambda^k) = \arg\min_{d_\pi \in \mathcal{K}} d_\pi \cdot \lambda^k - f^*(\lambda^k)$$
$$= \arg\max_{d_\pi \in \mathcal{K}} d_\pi \cdot (-\lambda^k),$$

which is an RL problem for maximizing the reward $(-\lambda^k)$. In principle, any RL algorithm that eventually solves the RL problem can be used to find the best response, which substantiates our claim in the introduction. For example, tabular Q-learning executed for sufficiently long and with a suitable exploration strategy will converge to the optimal policy (Watkins and Dayan, 1992). In the non-tabular case we could parameterize a deep neural network to represent the Q-values (Mnih et al., 2015) and if the network has sufficient capacity then similar guarantees might hold. We make no claims on efficiency or tractability of this approach, just that in principle such an approach would provide the best-response at each iteration and therefore satisfy the required conditions to solve the convex RL problem.

Approximate best response. The caveat in using the best response as a policy player is that in practice, it can only be found approximately by executing an RL algorithm in the environment. In terms of sample complexity, finding an ϵ-optimal solution to an RL problem requires $O(1/\epsilon^2)$ samples, and there are algorithms achieving a matching upper bound (Jaksch et al., 2010; Dann and Brunskill, 2015). This implies that in each iteration of Algorithm 1, the agent interacts with a new MDP and has to learn from scratch how to solve it without using the samples it gathered in previous iterations. The following Lemma analyzes the sample complexity of Algorithm 1 with approximate best response policy player. Other relaxations to the best response for specific algorithms can be found in (Syed and Schapire, 2008; Miryoosefi et al., 2019; Jaggi, 2013; Hazan et al., 2019).

Lemma 2 (The sample complexity of approximate best response in convex MDPs)**.** *A cost player with regret $\bar{R}_K^\lambda = O(1/K)$ and an approximate best response policy player that solves the RL problem in iteration k to accuracy $\epsilon_k = 1/k$ requires $O(1/\epsilon^3)$ samples to find an ϵ-optimal solution to the convex RL problem. Similarly, for $\bar{R}_K^\lambda = O(1/\sqrt{K})$, setting $\epsilon_k = 1/\sqrt{k}$ is guaranteed to find an ϵ-optimal solution with $O(1/\epsilon^4)$ samples.*

Non-stationary RL algorithms. We now discuss a different type of policy players; instead of solving an MDP to accuracy ϵ, these algorithms perform a *single* RL update to the policy with respect to $-\lambda_k$. In our setup the reward is known, deterministic but non-stationary, while in the standard RL setup it is unknown, stochastic and stationary. We conjecture that any model-based stochastic RL algorithm can be adapted to the *known* non-stationary reward setup we consider here. In most cases both Bayesian (Osband et al., 2013; O'Donoghue, 2018) and frequentist (Azar et al., 2017; Jaksch et al., 2010) approaches to the stochastic RL problem solve a modified (*e.g.*, to add optimism) Bellman equation at each time period, and so swapping in a known but non-stationary reward is unlikely to present a problem. We shall prove that this is exactly the case for two RL algorithms from the literature, UCRL2 (Jaksch et al., 2010) and MDPO (Shani et al., 2020b). These were designed and analyzed in the standard RL setup, and we shall show that they are easily adapted to the non-stationary but known reward setup that we require. UCRL2 is a model based algorithm that maintains an estimation of the reward and the transition matrix. In addition, it maintains confidence sets around these estimations $\mathcal{P}_k, \mathcal{R}_k$ that shrink as the agent collects more samples. UCRL2 guarantees that in any iteration k, the true reward and dynamics are in the confidence set with high probability $R \in \mathcal{R}_k, P \in \mathcal{P}_k$. In our case the reward at time k is known, so we only consider uncertainty in the dynamics. If we denote by $J_\pi^{P,R}$ the value of policy π in an MDP with dynamics P and reward R then the optimistic policy is $\tilde{\pi}_k = \arg\max_\pi \max_{P' \in \mathcal{P}_k} J_\pi^{P', -\lambda_k}$. Acting according to this policy is guaranteed to attain low regret as the following Lemma states.

Lemma 3 (Non stationary regret of UCRL2)**.** *For an MDP with dynamics P, diameter D (Jaksch et al., 2010, Definition 1), an arbitrary sequence of known rewards $r^1, \ldots, r^K$, such that the optimal average reward at time k, w.r.t P and r_k is $J_k^\star$, then with probability of at least $1-\delta$, the average regret of UCRL2 is at most $\bar{R}_K = \frac{1}{K}\sum_{k=1}^K J_k^\star - J_k^{\tilde{\pi}_k} \le O(DS\sqrt{A\log(K/\delta)/K})$.*

In the supplementary material (Appendix E), we provide a proof sketch that closely follows (Jaksch et al., 2010). We also note that UCRL2 was analyzed in (Rosenberg and Mansour, 2019) in the adversarial setup, that includes our setup as a special case. In a finite horizon MDP with horizon H it was shown that with probability $1 - SA/K$ its regret is bounded by $\bar{R}_K \le O(HS\sqrt{A\log(K)/K})$ (Rosenberg and Mansour, 2019, Corollary 5).

Another optimistic algorithm is Mirror Descent Policy Optimization (Shani et al., 2020b, MDPO), a model free RL algorithm that is very similar to popular DRL algorithms like TRPO (Schulman et al., 2015) and MPO (Abdolmaleki et al., 2018). In (Geist et al., 2019; Shani et al., 2020a; Agarwal et al., 2020), the authors established the global convergence of MDPO and in (Cai et al., 2020; Shani et al., 2020b), the authors showed that MDPO with optimistic exploration enjoys efficient regret. In a finite horizon MDP with horizon H and known non-stationary rewards, the regret of MDPO was bounded as the following Lemma states.

Lemma 4 (Non stationary regret of MDPO (Lemma 4, (Shani et al., 2021))). *For an arbitrary sequence of known rewards $r^1, \dots, r^K$, the average regret of MDPO is at most $\bar{R}_K \le O(H^2 S\sqrt{A/K})$.*

While UCRL2 attains a regret that is better by a factor of H than MDPO, MDPO is much closer to practical DRL algorithms and was shown to perform well as a DRL algorithm (Tomar et al., 2020). We note that MDPO analysis assumes that we can solve the mirror descent sub problem exactly which is only feasible for small tabular problems (Shani et al., 2020a). When function approximation is used, e.g. in the DRL setup, the mirror descent problem is usually solved with gradient descent (Schulman et al., 2015; Abdolmaleki et al., 2018; Tomar et al., 2020), and as a result the error that comes from not solving it exactly has to be considered.

Finally, the two algorithms we considered here achieve regret of $\bar{R}^\pi_K \le O(1/\sqrt{K})$. Thus, according to Eq. (5), combining these policy players with any cost player with regret $\bar{R}^\lambda_K = O(1/\sqrt{K})$ implies that it is enough to run Algorithm 1 for $O(1/\epsilon^2)$ iterations to find an $\epsilon-$optimal solution to the convex RL problem (Eq. (2)). This makes the non-stationary RL algorithm more efficient and more practical compared to the algorithm based on approximate best response. To the best of our knowledge, this is the first result that shows an $O(1/\epsilon^2)$ sample complexity guarantee for the convex RL problem.

5. Examples

In this section we explain how existing algorithms can be seen as instances of the meta algorithm for various choices of the objective function f, the cost and policy player algorithms Alg_λ and Alg_π. We summarizes the relationships in Table 1. In the case of vanilla **RL**, *i.e.*, $f = d_\pi \cdot \lambda$, and if the cost player is playing FTL then we recover the vanilla RL problem.

5.1. Apprenticeship Learning

In AL, there is an MDP without an explicit reward function. Instead, there is an expert that acts according to some policy and provides demonstrations, which are used to estimate its state occupancy d_e. Abbeel and Ng (Abbeel and Ng, 2004) formalized the AL problem as finding a policy π whose state occupancy is close to that of the expert by minimizing the convex function $f = ||d_\pi - d_E||_2^2$.

Consider a slightly more general formulation of the AL problem where the function f measures the distance between the state occupancy of the agent d_π and d_e under a general norm $||\cdot||$, *i.e.*, $f = ||d_\pi - d_e||$. The convex conjugate of f is given by $f^*(y) = y \cdot c$ if $||y||_* \le 1$ and ∞ otherwise, where $||\cdot||_*$ is the dual norm. Plugging f^* in Eq. (4) results in the following max-min game:

$$\max_{d_\pi \in \mathcal{K}} \min_{||\lambda||_* \le 1} \lambda \cdot d_E - \lambda \cdot d_\pi. \tag{7}$$

Eq. (7) implies that the norm, which measures the distance from the expert in the function f, induces the constraint set of the cost variable to be a unit ball in the dual norm. We note that Eq. (7) can also be used without an expert ($d_E = 0$) to find a policy that is robust to the worst case reward (Zahavy et al., 2021).

Alg_λ=OMD, Alg_π=best response/RL. The Multiplicative Weights AL algorithms (Syed and Schapire, 2008, MWAL) was proposed to solve the AL problem with $f = ||d_\pi - d_E||_\infty$, such that the dual norm in Eq. (7) is $||\cdot||_1$. It uses the best response as the policy player and multiplicative weights as the cost player (a special case of OMD). MWAL was also used to solve AL in contextual MDPs (Belogolovsky et al., 2021) and to find feasible solutions to convex-constrained MDPs (Miryoosefi et al., 2019). We note that in practice, the best response can only be solved approximately, as we discussed in Section 4. Instead, in Online AL (Shani et al., 2021), the authors proposed to use MDPO as the policy player which gurantess regret of at most $\bar{R}_K \le c/\sqrt{K}$. They showed that their algorithm is equivalent to Wasserstein GAIL (Xiao et al., 2019; Zhang et al., 2020b) and performs similarly to GAIL.

Alg_λ=FTL, Alg_π=best response. When the policy player plays the best response and the cost player plays FTL, Algorithm 1 is equivalent to the Frank-Wolfe algorithm (Frank and Wolfe, 1956; Abernethy and Wang, 2017) for minimizing the function f (Eq. (2)), pseudo-code for which is included in the appendix (Algorithm 3). The algorithm finds a point $d_{\pi^k} \in \mathcal{K}$ that has the largest correlation (best response) with the negative gradient (FTL).

Abbeel and Ng (Abbeel and Ng, 2004) proposed two algorithms for AL, the projection algorithm and the max margin algorithm. The projection algorithm is essentially a FW algorithm, as was suggested in the supplementary (Abbeel and Ng, 2004) and was later shown formally in (Zahavy et al., 2020). Thus, it is a projection free algorithm in the sense that it avoids projecting d_π into $\mathcal{K}$, despite the perhaps confusing name. Specifically, in this case, the gradient is $\nabla_f = d_\pi - d_E$; thus, finding the best response is equivalent to solving an MDP whose reward is $d_E - d_\pi$. In a similar fashion, FW can be used to solve any other convex MDP (Hazan et al., 2019). Specifically, in (Hazan et al., 2019), the authors considered the problem of pure exploration – finding a policy that visits all the states uniformly – defined as finding a policy with the maximum entropy d_π : $\max_{d_\pi \in \mathcal{K}} H(d_\pi)$, where $H(d_\pi) = -d_\pi \cdot \log(d_\pi)$.

Fully corrective FW. The FW algorithm has many variants (see (Jaggi, 2013) for a survey) and some of them can enjoy faster rates of convergence in special cases. Specifically, when the constraint set is a polytope, which is the case in convex MDPs (Definition 1), some variants achieve a linear rate of convergence (Jaggi and Lacoste-Julien, 2015; Zahavy et al., 2020). One such variant is the Fully corrective FW, which replaces the learning rate update (line 4 of Algorithm 3), with a minimization problem over the previous state-occupancy's. This step is guaranteed to be at least as good as the learning rate update since:

$$f\big((1-\alpha_k)\bar{d}_\pi^k + \alpha_k d_\pi^{k+1}\big) \geq \min_{x \in \text{Co}(\bar{d}_\pi^k, d_\pi^{k+1})} f(x) \qquad (8)$$
$$\geq \min_{x \in \text{Co}(d_\pi^1, d_\pi^2, \ldots, d_\pi^{k+1})} f(x).$$

Interestingly, the second algorithm of Abbeel and Ng (Abbeel and Ng, 2004), the max margin algorithm, is exactly equivalent to this fully corrective FW variant. This implies that the max-margin algorithm enjoys a much better convergence rate than the 'projection' variant, as was observed empirically in (Abbeel and Ng, 2004).

5.2. GAIL and DIAYN: Alg_λ=FTL, Alg_π=RL

We now show how mutual-information based diversity objectives (Gregor et al., 2017; Eysenbach et al., 2019) can be derived through the lens of convex MDPs. To do so, we reformulate the objective of DIAYN (Eysenbach et al., 2019) as a convex RL problem with the following objective function (see Appendix D for details):

$$\mathbb{E}_z \text{KL}(d_{\pi^z} || \mathbb{E}_k d_{\pi^k}). \qquad (9)$$

Intuitively, Eq. (9) implies that the policies $\pi^1, \ldots, \pi^z$ are diverse when they visit different states, measured using the KL distance between their respective state occupancies $d_\pi^1, \ldots, d_{\pi^z}$. It is easy to see that Eq. (9) is convex because the KL-divergence is jointly convex in both arguments (Boyd and Vandenberghe, 2004, Example 3.19). Recall that according to Eq. (6) the FTL cost player is $\nabla_f(\bar{d}_\pi^{k-1})$. Thus, we now compute the gradient of Eq. (9) w.r.t d_{π^z} and compare it to the intrinsic reward in DIAYN. The gradient is a vector of size $|s|$, given by $\nabla_{d_{\pi^z}} \text{KL}(d_{\pi^z} || \sum_k p(k) d_{\pi^k}) =$

$$\mathop{\mathbb{E}}_{z \sim p(z)} \Big[\log \frac{d_{\pi^z}}{\sum_k d_{\pi^k} p(k)} + 1 - \frac{d_{\pi^z} p(z)}{\sum_k d_{\pi^k} p(k)} \Big] \qquad (10)$$
$$= \mathop{\mathbb{E}}_{z \sim p(z)} \Big[\underbrace{\log(p(z|s)) - \log(p(z))}_{\text{Mutual Information}} \underbrace{+1 - p(z|s)}_{\text{Gradient correction}} \Big],$$

where the equality follows from writing the posterior as a function of the per-skill state occupancy $d_{\pi^z} = p(s \mid z)$, and using Bayes rules, $p(z|s) = \frac{d_{\pi^z}(s)p(z)}{\sum_k d_{\pi^k}(s)p(k)}$. Replacing the posterior $p(z|s)$ with a learnt discriminator $q_\phi(z|s)$ recovers the mutual-information rewards of DIAYN, with additional terms $1 - p(z \mid s)$ which we refer to as "gradient correction" terms. Inspecting the common scenario of a uniform prior over the latent variables, $p(z) = 1/|Z|$, we get that the expectation of the gradient correction term $\sum_z p(z)(1 - p(z|s)) = 1 - 1/|z|$ in each state. From the perspective of the policy player, adding a constant to the reward does not change the best response policy, nor the optimistic policy. Therefore, the gradient correction term does not have an effect on the optimization under a uniform prior, and we retrieved DIAYN as a convex MDP algorithm. These algorithms differ however for more general priors $p(z)$, which we explore empirically in Section 6. Finally, note that the reward in DIAYN is the first term in Eq. (10) without a negative sign. This implies that DIAYN performs convex *maximization*, which is a hard problem in general.

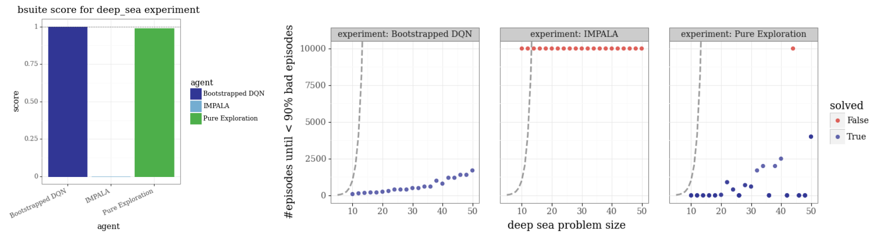
GAIL. We further show how Eq. (9) extends to GAIL (Ho and Ermon, 2016) via a simple construction. Consider a binary latent space of size 2 corresponding to the agent and the expert and a uniform prior over the latents. By removing the constant terms in Eq. (10), one retrieves the GAIL (Ho and Ermon, 2016) algorithm. The cost $\log(p(z|s))$ is the probability of the discriminator to identify the agent, and the policy player is MDPO (which is similar to TRPO in GAIL). This implies that GAIL and DIAYN have the same objective function, where in one setup it is maximized and in the other minimized.

6. Experiments

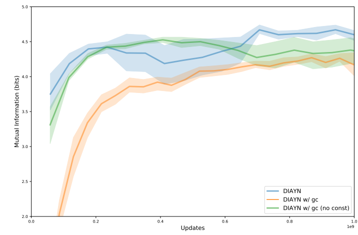
Above, we presented a principled approach to using standard RL algorithms to solve convex MDPs. We also suggested that DRL agents can use this principle and solve convex MDPs by optimizing the reward from the cost player. We now demonstrate this by performing experiments with Impala (Espeholt et al., 2018), a distributed actor-critic DRL algorithm. Our main message is that in domains where Impala can solve RL problems (*e.g.*, problems without hard exploration), it can also solve convex RL problems.

DIAYN. In our first experiment, we focus on the convex RL formulation of DIAYN as we defined in Eq. (9). We compare the intrinsic reward that results from an FTL cost player in Eq. (10) and the original mutual-information based reward in DIAYN by performing ablative analysis on the gradient correction terms in Eq. (10). In both cases, we also include the standard action entropy regularizer. Since the two intrinsic rewards were shown to be equivalent under a uniform prior, we consider a fixed but non-uniform prior.[1] The environment is a simple 9×9 gridworld, where the agent can move along the four cardinal directions. We maximize undiscounted rewards over episodes of length 32. Given trajectories generated by the distributed actors, a central learner computes the gradients and updates the parameters for the policy, critic and the (variational) reverse predictor. Fig. 2b plots the average (per timestep) mutual information $I(z, s)$, between code z and states $s \sim d_{\pi^z}$, which is equivalent to the objective in Eq. (9), see Appendix D for the derivation. Performance is averaged over 10 seeds, with the shaded area representing the standard error on the mean. Inspecting Fig. 2b we can see that DIAYN reaches around 4.5 bits. We can also see that using the full gradient correction term in Eq. (10) ("DIAYN w/ gc") degrades performance both in terms of convergence and final performance. On the other hand, removing the constant from the gradient correction ("DIAYN w/ gc (no const)"), which does not affect the optimal policy, recovers the performance of DIAYN.

[1] $p(z)$ is a Categorical distribution over $n = 2^5$ outcomes, with $p(z = i) = u_i / \sum_{j=1}^n u_j$, $u_i \sim U(0, 1)$.



(a) Entropy constrained RL



(b) DIAYN, non uniform prior.

Entropy constrained RL. Here we focus on an MDP with a convex constraint, where the goal is to maximize the extrinsic reward provided by the environment with the constraint that the entropy of the state-action occupancy measure must be bounded below. In other words, the agent must solve $\max_{d_\pi \in \mathcal{K}} \sum_{s,a} r(s,a) d_\pi(s,a)$ subject to $H(d_\pi) \geq C$, where H denotes entropy and $C > 0$ is a constant. The policy that maximizes the entropy over the MDP acts to visit each state as close to uniformly often as is feasible. So, a solution to this convex MDP is a policy that, loosely speaking, maximizes the extrinsic reward under the constraint that it explores the state space sufficiently. The presence of the constraint means that this is not a standard RL problem in the form of Eq. (1). However, the agent can solve this problem using the techniques developed in this paper, in particular those discussed in Section 3.1.

We evaluated the approach on the bsuite environment 'Deep Sea', which is a hard exploration problem where the agent must take the exact right sequence of actions to discover the sole positive reward in the environment; more details can be found in (Osband et al., 2019). In this domain, the features are one-hot state features, and we estimate d_π by counting the state visitations. For these experiments we chose C to be half the maximum possible entropy for the environment, which we can compute at the start of the experiment and hold fixed thereafter. We equipped the agent with the (non-stationary) Impala algorithm, and the cost-player used FTL. We present the results in Figure 2a where we compare the basic Impala agent, the entropy-constrained Impala agent and bootstrapped DQN (Osband et al., 2016). As made clear in (O'Donoghue et al., 2020b) algorithms that do not properly account for uncertainty cannot in general solve hard exploration problems. This explains why vanilla Impala, considered a strong baseline, has such poor performance on this problem. Bootstrapped DQN accounts for uncertainty via an ensemble, and consequently has good performance. Surprisingly, the entropy regularized Impala agent performs approximately as well as bootstrapped DQN, despite not handling uncertainty. This suggests that the entropy constrained approach, solved using Algorithm 1, can be a reasonably good heuristic in hard exploration problems.

7. Summary

In this work we reformulated the convex RL problem as a convex-concave game between the agent and another player that is producing costs (negative rewards). We proposed a meta algorithm for solving this game, and discussed various options for each player. For the policy player, we discussed the best response and showed that it is equivalent to the FW algorithm used in related work (Abbeel and Ng, 2004; Zahavy et al., 2020; Hazan et al., 2019). We then considered the scenario that an $\epsilon-$optimal best response is computed by executing an RL algorithm and suggested that it is not sample efficient. Instead, we proposed using standard RL algorithms, with a non-stationary reward, as policy players. We proved a regret bound for a UCRL2 player and proposed that any efficient RL algorithm can be used instead. For the cost player, we have shown that choosing FTL for the cost player, results in non-stationary reward that is equivalent to the gradient of the convex objective evaluated on $\bar{d}_\pi^{k-1}$. Using this equivalence, we demonstrated that many intrinsic rewards in the literature can be understood as gradients of objectives in convex MDPs. We experimented with a vanilla actor-critic agent and showed that in domains where the baseline agent can solve an RL problem, it can also solve convex RL problems simply by using the non-stationary reward from the cost player. Finally, we demonstrated that many attributes of intelligence, such as learning to mimic an expert, learning diverse policies, and learning to maximize reward while satisfying constraints, can be well understood as convex RL problems and solved via the maximization of a non-stationary reward. We hope that our formulation of the convex RL problem will help to define and solve more aspects of intelligence in future work, for example, in unsupervised RL.

References

P. Abbeel and A. Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1. ACM, 2004.

A. Abdolmaleki, J. T. Springenberg, Y. Tassa, R. Munos, N. Heess, and M. Riedmiller. Maximum a posteriori policy optimisation. *arXiv preprint arXiv:1806.06920*, 2018.

J. D. Abernethy and J.-K. Wang. On frank-wolfe and equilibrium computation. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/7371364b3d72ac9a3ed8638e6f0be2c9-Paper.pdf.

J. Achiam, H. Edwards, D. Amodei, and P. Abbeel. Variational option discovery algorithms. *arXiv preprint arXiv:1807.10299*, 2018.

A. Agarwal, S. M. Kakade, J. D. Lee, and G. Mahajan. Optimality and approximation with policy gradient methods in markov decision processes. In *Conference on Learning Theory*, pages 64–66. PMLR, 2020.

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

M. G. Azar, I. Osband, and R. Munos. Minimax regret bounds for reinforcement learning. In *International Conference on Machine Learning*, pages 263–272, 2017.

A. Beck and M. Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31:167–175, 2003.

S. Belogolovsky, P. Korsunsky, S. Mannor, C. Tessler, and T. Zahavy. Inverse reinforcement learning in contextual mdps. *Machine Learning*, 2021.

S. 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.

V. S. Borkar. An actor-critic algorithm for constrained markov decision processes. *Systems & control letters*, 54 (3):207–213, 2005.

S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Q. Cai, Z. Yang, C. Jin, and Z. Wang. Provably efficient exploration in policy optimization. In *International Conference on Machine Learning*, pages 1283–1294. PMLR, 2020.

D. A. Calian, D. J. Mankowitz, T. Zahavy, Z. Xu, J. Oh, N. Levine, and T. Mann. Balancing constraints and rewards with meta-gradient d4{pg}. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=TQt98Ya7UMP.

C. Dann and E. Brunskill. Sample complexity of episodic fixed-horizon reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 2818–2826, 2015.

Y. Efroni, S. Mannor, and M. Pirotta. Exploration-exploitation in constrained mdps. *arXiv preprint arXiv:2003.02189*, 2020.

L. Espeholt, H. Soyer, R. Munos, K. Simonyan, V. Mnih, T. Ward, Y. Doron, V. Firoiu, T. Harley, I. Dunning, S. Legg, and K. Kavukcuoglu. IMPALA: Scalable distributed deep-RL with importance weighted actor-learner architectures. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

B. Eysenbach, A. Gupta, J. Ibarz, and S. Levine. Diversity is all you need: Learning skills without a reward function. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SJx63jRqFm.

C. Florensa, Y. Duan, and P. Abbeel. Stochastic neural networks for hierarchical reinforcement learning. In *International Conference on Learning Representations*, 2016.

M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110, 1956.

Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

M. Geist, B. Scherrer, and O. Pietquin. A theory of regularized markov decision processes. In *International Conference on Machine Learning*, pages 2160–2169. PMLR, 2019.

K. Gregor, D. J. Rezende, and D. Wierstra. Variational intrinsic control. *International Conference on Learning Representations, Workshop Track*, 2017. URL https://openreview.net/forum?id=Skc-Fo4Yg.

K. Hausman, J. T. Springenberg, Z. Wang, N. Heess, and M. Riedmiller. Learning an embedding space for transferable robot skills. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rk07ZXZRb.

E. Hazan. Introduction to online convex optimization. *Foundations and Trends in Optimization*, 2(3-4):157–325, 2016.

E. Hazan, A. Agarwal, and S. Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.

E. Hazan, S. Kakade, K. Singh, and A. Van Soest. Provably efficient maximum entropy exploration. In *International Conference on Machine Learning*, pages 2681–2691. PMLR, 2019.

J. Ho and S. Ermon. Generative adversarial imitation learning. *arXiv preprint arXiv:1606.03476*, 2016.

R. Huang, T. Lattimore, A. György, and C. Szepesvári. Following the leader and fast rates in linear prediction: Curved constraint sets and other regularities. In *Advances in Neural Information Processing Systems*, pages 4970–4978, 2016.

M. Jaggi. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *Proceedings of the 30th international conference on Machine learning*. ACM, 2013.

M. Jaggi and S. Lacoste-Julien. On the global linear convergence of frank-wolfe optimization variants. *Advances in Neural Information Processing Systems*, 28, 2015.

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

L. Lee, B. Eysenbach, E. Parisotto, E. Xing, S. Levine, and R. Salakhutdinov. Efficient exploration via state marginal matching. *arXiv preprint arXiv:1906.05274*, 2019.

B. McMahan. Follow-the-regularized-leader and mirror descent: Equivalence theorems and l1 regularization. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 525–533. JMLR Workshop and Conference Proceedings, 2011.

S. Miryoosefi, K. Brantley, H. Daumé III, M. Dudík, and R. Schapire. Reinforcement learning with convex constraints. *arXiv preprint arXiv:1906.09323*, 2019.

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

O. Nachum, Y. Chow, B. Dai, and L. Li. Dualdice: Behavior-agnostic estimation of discounted stationary distribution corrections. *arXiv preprint arXiv:1906.04733*, 2019.

A. S. Nemirovskij and D. B. Yudin. Problem complexity and method efficiency in optimization. In *Wiley-Interscience*, 1983.

B. O'Donoghue. Variational Bayesian reinforcement learning with regret bounds. *arXiv preprint arXiv:1807.09647*, 2018.

B. O'Donoghue, T. Lattimore, and I. Osband. Stochastic matrix games with bandit feedback. *arXiv preprint arXiv:2006.05145*, 2020a.

B. O'Donoghue, I. Osband, and C. Ionescu. Making sense of reinforcement learning and probabilistic inference. In *International Conference on Learning Representations*, 2020b.

I. Osband, D. Russo, and B. Van Roy. (More) efficient reinforcement learning via posterior sampling. In *Advances in Neural Information Processing Systems*, pages 3003–3011, 2013.

I. Osband, C. Blundell, A. Pritzel, and B. V. Roy. Deep exploration via bootstrapped dqn. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 4033–4041, 2016.

I. Osband, Y. Doron, M. Hessel, J. Aslanides, E. Sezener, A. Saraiva, K. McKinney, T. Lattimore, C. Szepesvari, S. Singh, et al. Behaviour suite for reinforcement learning. In *International Conference on Learning Representations*, 2019.

M. J. Osborne and A. Rubinstein. *A course in game theory*. MIT press, 1994.

M. L. Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 1984.

R. T. Rockafellar. *Convex analysis*. Princeton university press, 1970.

A. Rosenberg and Y. Mansour. Online convex optimization in adversarial markov decision processes. In *International Conference on Machine Learning*, pages 5478–5486. PMLR, 2019.

J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015.

L. Shani, Y. Efroni, and S. Mannor. Adaptive trust region policy optimization: Global convergence and faster rates for regularized mdps. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5668–5675, 2020a.

L. Shani, Y. Efroni, A. Rosenberg, and S. Mannor. Optimistic policy optimization with bandit feedback. In *International Conference on Machine Learning*, pages 8604–8613. PMLR, 2020b.

L. Shani, T. Zahavy, and S. Mannor. Online apprenticeship learning. *arXiv preprint arXiv:2102.06924*, 2021.

D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

A. L. Strehl and M. L. Littman. An analysis of model-based interval estimation for markov decision processes. *Journal of Computer and System Sciences*, 74(8):1309–1331, 2008.

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

U. Syed and R. E. Schapire. A game-theoretic approach to apprenticeship learning. In *Advances in neural information processing systems*, pages 1449–1456, 2008.

U. Syed, M. Bowling, and R. E. Schapire. Apprenticeship learning using linear programming. In *Proceedings of the 25th international conference on Machine learning*, pages 1032–1039. ACM, 2008.

C. Szepesvári. Constrained mdps and the reward hypothesis, 2020. URL https://readingsml.blogspot.com/2020/03/constrained-mdps-and-reward-hypothesis.html.

C. Tessler, D. J. Mankowitz, and S. Mannor. Reward constrained policy optimization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SkfrvsA9FX.

D. Tirumala, A. Galashov, H. Noh, L. Hasenclever, R. Pascanu, J. Schwarz, G. Desjardins, W. M. Czarnecki, A. Ahuja, Y. W. Teh, et al. Behavior priors for efficient reinforcement learning. *arXiv preprint arXiv:2010.14274*, 2020.

M. Tomar, L. Shani, Y. Efroni, and M. Ghavamzadeh. Mirror descent policy optimization. *arXiv preprint arXiv:2005.09814*, 2020.

J. Von Neumann. Zur theorie der gesellschaftsspiele. *Mathematische annalen*, 100(1):295–320, 1928.

C. J. Watkins and P. Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

H. Xiao, M. Herman, J. Wagner, S. Ziesche, J. Etesami, and T. H. Linh. Wasserstein adversarial imitation learning. *arXiv preprint arXiv:1906.08113*, 2019.

M. Yang, O. Nachum, B. Dai, L. Li, and D. Schuurmans. Off-policy evaluation via the regularized lagrangian. *arXiv preprint arXiv:2007.03438*, 2020.

T. Zahavy, A. Cohen, H. Kaplan, and Y. Mansour. Apprenticeship learning via frank-wolfe. *AAAI, 2020*, 2020.

T. Zahavy, A. Barreto, D. J. Mankowitz, S. Hou, B. O'Donoghue, I. Kemaev, and S. Singh. Discovering a set of policies for the worst case reward. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=PUkhWz65dy5.

J. Zhang, A. Koppel, A. S. Bedi, C. Szepesvari, and M. Wang. Variational policy gradient method for reinforcement learning with general utilities. *arXiv preprint arXiv:2007.02151*, 2020a.

M. Zhang, Y. Wang, X. Ma, L. Xia, J. Yang, Z. Li, and X. Li. Wasserstein distance guided adversarial imitation learning with reward shape exploration. In *2020 IEEE 9th Data Driven Control and Learning Systems Conference (DDCLS)*, pages 1165–1170. IEEE, 2020b.

M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th international conference on machine learning (icml-03)*, pages 928–936, 2003.

A. Proposition 1

Proposition 1. *For both the average and the discounted case, the agent objective function Eq.* (3) *can be written in terms of the occupancy measure as* $J_\pi = \sum_{s,a} r(s,a) d_\pi(s,a)$.

Proof. Beginning with the discounted case, the average cost is given by

$$\begin{aligned}
J_\pi^\gamma &= (1-\gamma)\mathbb{E}\sum_{t=1}^{\infty}\gamma^t r_t \\
&= (1-\gamma)\sum_{t=1}^{\infty}\sum_s \mathbb{P}_\pi(s_t = s)\sum_a \pi(s,a)\gamma^t r(s,a) \\
&= (1-\gamma)\sum_{s,a}\left(\sum_{t=1}^{\infty}\gamma^t \mathbb{P}_\pi(s_t = s)\pi(s,a)\right) r(s,a) \\
&= \sum_{s,a} d_\pi^\gamma(s,a) r(s,a).
\end{aligned}$$

Similarly, for the average reward case

$$\begin{aligned}
J_\pi^{\text{avg}} &= \lim_{T\to\infty}\frac{1}{T}\mathbb{E}\sum_{t=1}^{T} r_t \\
&= \lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\sum_s \mathbb{P}_\pi(s_t = s)\sum_a \pi(s,a) r(s,a) \\
&= \sum_{s,a}\left(\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\mathbb{P}_\pi(s_t = s)\pi(s,a)\right) r(s,a) \\
&= \sum_{s,a} d_\pi^{\text{avg}}(s,a) r(s,a).
\end{aligned}$$

■

B. FW algorithms

B.1. Pseudo code

Algorithm 3 Frank-Wolfe algorithm

Input: a convex and smooth function f
Initialize: Pick a random element $d_\pi^1 \in \mathcal{K}$.
for $i = 1, \ldots, T$ **do**
  $d_\pi^{k+1} = \arg\max_{\pi \in \Pi} d_\pi \cdot -\nabla f(\bar{d}_\pi^k)$
  $\bar{d}_\pi^{k+1} = (1 - \alpha_i)\bar{d}_\pi^k + \alpha_i d_\pi^{k+1}$
end for

B.2. Linear convergence

Theorem 2 (Linear Convergence (Jaggi and Lacoste-Julien, 2015)). *Suppose that f has L-Lipschitz gradient and is μ-strongly convex. Let $D = \{d_\pi, \forall \pi \in \Pi\}$ be the set of all the state occupancy's of deterministic policies in the MDP and let $\mathcal{K} = Co(D)$ be its Convex Hull. Such that $\mathcal{K}$ a polytope with vertices D, and let $M = diam(\mathcal{K})$. Also, denote the Pyramidal Width of D, $\delta = PWidth(D)$ as in (Jaggi and Lacoste-Julien, 2015, Equation 9 1).*

Then the suboptimality h_t of the iterates of all the fully corrective FW algorithm decreases geometrically at each step, that is

$$f(\bar{d}_\pi^{k+1}) \leq (1 - \rho) f(\bar{d}_\pi^k) \text{ , where } \rho = \frac{\mu \delta^2}{4LM^2}$$

C. Proof of Lemma 2

Lemma (The sample complexity of approximate best response in convex MDPs). *A cost player with regret $\bar{R}_K^\lambda = O(1/K)$ and an approximate best response policy player that solves the RL problem in iteration k to accuracy $\epsilon_k = 1/k$ requires $O(1/\epsilon^3)$ samples to find an $\epsilon-$optimal solution to the convex RL problem. Similarly, for $\bar{R}_K^\lambda = O(1/\sqrt{K})$, setting $\epsilon_k = 1/\sqrt{k}$ is guaranteed to find a solution with $O(1/\epsilon^4)$ samples.*

Recall that the regret of the best response is $\bar{R}_K \leq 0$. If we solve the best response approximately at iteration k up to accuracy ϵ_k, then the overall regret of the policy player is $\bar{R}_K^\pi = \frac{1}{K}\sum \epsilon_k$. The overall regret of the game is the sum of the regret of the policy player and the cost player, and the regret of the game is asymptotically

$$\bar{R}_K = \bar{R}_K^\pi + \bar{R}_K^\lambda = O\left(\max(\bar{R}_K^\pi, \bar{R}_K^\lambda)\right) \tag{11}$$

We consider two cases for the cost player. In the first, it will have constant regret, and therefore average regret of $\bar{R}_K^\lambda = O(1/K)$, which is possible to achieve under some assumptions (Huang et al., 2016). In the second scenario, we will consider average regret of $\bar{R}_K^\lambda = O(1/\sqrt{K})$, which is feasible for any of the cost players we considered in this paper.

Consider the general case of $\epsilon_k = 1/k^p$. Note that for the average regret $\frac{1}{K}\sum_{k=1}^K 1/k^p$ to go to zero as K grows, the sum $\frac{1}{K}\sum_{k=1}^K 1/k^p$ must be smaller than K, so p must be positive. In addition, for larger values of p, ϵ_k is smaller. Thus the regret is smaller, but at the same time, it requires more samples to solve each RL problem. Inspecting the maximum in Eq. (11), we observe that it does not make sense to choose a value for p for which $\frac{1}{K}\sum_{k=1}^K 1/k^p < \bar{R}_K^\lambda$, since it will not improve the overall regret and will require more samples, than, for example, setting p such that $\frac{1}{K}\sum_{k=1}^K 1/k^p = \bar{R}_K^\lambda$.

Thus, in the case that the cost player has constant regret, $\bar{R}_K^\lambda = O(1/K)$, we set $p \in (0, 1]$, and in the case that the cost player has regret of $\bar{R}_K^\lambda = O(1/\sqrt{K})$, we set $p \in (0, 0.5]$.

We now continue and further inspect the regret. We have that $\frac{1}{K}\sum_{k=1}^K \epsilon_k = \frac{1}{K}\sum_{k=1}^K 1/k^p = O(k^{-p})$ for $p \in (0, 1)$, and $\log(K)/K$ for $p = 1$. Neglecting logarithmic terms, we continued with $O(k^{-p})$ for both cases. In other words, it is sufficient to run the meta algorithm for $K = 1/\epsilon^p$ iterations to guarantee an error of at most ϵ for the convex RL problem.

To solve an MDP to accuracy ϵ_k, it is sufficient to run an RL algorithm for $O(1/\epsilon_k^2)$ iterations. This is a lower bound and an upper bound, see, for example (Dann and Brunskill, 2015) for an upper bound of $CO(H^2 S^2 A \log(1/\delta)$ and a lower bound of $O(H^2 SA \log(1/\delta)$ in the finite horizon or (Jaksch et al., 2010) for the average case. Thus, to solve an MDP to accuracy $\epsilon_k = 1/k^p$ it requires k^{2p} iterations, and the overall sample complexity is therefore $\sum_{k=1}^{1/\epsilon^p} k^{2p} = O(1/\epsilon^{\frac{2p+1}{p}})$.

The function $1/\epsilon^{\frac{2p+1}{p}}$ is monotonically increasing in p, so it attains minimum for the highest value of p which is 0.5 or 1, depending on the cost player. We conclude that the optimal sample complexity with approximate best response is $O(1/\epsilon^3)$ for the cost player that has constant regret and $O(1/\epsilon^4)$ for a cost player with average regret of $\bar{R}_K^\lambda = O(1/\sqrt{K})$.

D. DIAYN

Discriminative approaches rely on the intuition that skills are diverse when they are entropic and easily discriminated by observing the states that they visit. Given a probability space $(\Omega, \mathcal{F}, \mathcal{P})$, state random variables $S : \Omega \to \mathcal{S}$ and latent skills $Z : \Omega \to \mathcal{Z}$ with prior p, the key term of interest being maximized in DIAYN (Eysenbach et al., 2019) is the mutual information:

$$I(S; Z) = \mathbb{E}_{z\sim p; s\sim d_{\pi^z}} [\log p(z|s) - \log p(z)], \tag{12}$$

where d_{π^z} is the stationary distribution induced by the policy $\pi(a \mid s, z)$. For each skill z, this corresponds to a standard RL problem with (conditional) policy $\pi(a \mid s, z)$ and reward function $r(s|z) = \log p(z|s) - \log p(z)$. The first term encourages the policy to visit states for which the underlying skill has high-probability under the posterior $p(z \mid s)$, while the second term ensures a high entropy distribution over skills. In practice, the full DIAYN objective further regularizes the learnt policy by including entropy terms $-\log \pi(a \mid s, z)$. For large state spaces, $p(z|s)$ is typically intractable and Eq. 12 is replaced with a variational lower-bound, where the true posterior is replaced with a learned discriminator $q_\phi(z|s)$. Here, we focus on the simple setting where z is a categorical distribution over n outcomes, yielding n policies π^z, and q is a classifier over these n skills.

We now show that a similar objective can be derived using the framework of convex MDPs. We start by writing the true posterior as a function of the per-skill state occupancy $d_{\pi^z} = p(s \mid z)$, and using Bayes rules, $p(z|s) = \frac{d_{\pi^z}(s)p(z)}{\sum_k d_{\pi^k}(s)p(k)}$. Combing this with Eq. (12) yields:

$$\begin{aligned}
\mathbb{E}_{z\sim p(z), s\sim d(\pi^z)}[\log p(z|s) - p(z)] &= \sum_z p(z) \sum_s d_{\pi^z}(s) \left[\log\left(\frac{d_{\pi^z}(s)p(z)}{\sum_k d_{\pi^k}(s)p(k)}\right) - \log p(z)\right] \\
&= \sum_z p(z) \mathrm{KL}(d_{\pi^z} || \sum_k p(k) d_{\pi^k}) = \mathbb{E}_z \mathrm{KL}(d_{\pi^z} || \mathbb{E}_k d_{\pi^k}).
\end{aligned} \tag{13}$$

E. Proof sketch for Lemma 3

We denote by r_k^* the optimal average reward at time k in an MDP with dynamics P and reward $r_k = -\lambda_k$. We want to show that

$$R_k = \sum_k r_k^* - r_k(s_k, a_k) \le c/\sqrt{K},$$

that is, that the total reward that the agent collects has low regret compared to the sum of optimal average rewards.

To show that, we make two minor adaptations to the UCRL2 algorithm and then verify that its original analysis also applies to this non-stationary setup. The first modification is that the nonstatioanry version of UCRL2 uses the known reward r_k at time k (which in our case is the output of the cost player) instead of estimating the unknown, stochastic, stationary, extrinsic reward. Since the current reward r_k is known and deterministic, there is no uncertainty about it, and we only have to deal with uncertainty w.r.t to the dynamics. The second modification is that we compute a new optimistic policy (using extended value iteration) in each iteration. This optimistic policy is computed with the current reward r_k, and the current uncertainty set about the dynamics $\mathcal{P}_k$. This also means that all of our episodes are of length 1.

After making these two clarifications, we follow the proof of UCRL2 and make changes when appropriate. We note that the analysis, basically, does not require any modifications, but we repeat the relevant parts for completeness. We begin with the definition of the regret at episode k, which is now just the regret at time k :

$$\Delta_k = \sum_{s,a} v_k(s, a)(r_k^* - r_k(s, a)),$$

where $v_k(s, a)$ in our case is an indicator on the state action pair s_k, a_k, and $R_k = \sum_k \Delta_k$.

The instantaneous regret Δ_k measures the difference between the optimal average reward r_k^*, w.r.t reward r_k, and the reward $r_k(s, a)$ that the agent collected at time k by visiting state s and taking action k from the reward that is produced by the cost player.

Section 4.1 in the UCRL2 paper is the first step in the analysis. It bounds possible fluctuations in the random reward. This step is not required in our case since our reward at time k is the output of the cost player, which is known in all the states and deterministic.

Section 4.2 considers the regret that is caused by failing confidence regions, that is, the event that the true dynamics and true reward are not in the confidence region. In our case there is only confidence region for the dynamics (since the reward is known), which we denote by $\mathcal{P}_k$. Summing the expected regret from episodes in which $P \notin \mathcal{P}_k$ results in a $\sqrt{K}$ term in the regret,

$$\Delta_k \leq \sum_{s,a} v_k(s,a)(r^*_k - r_k(s,a)) + \sqrt{K},$$

where from now on, we continue with the event that $P \in \mathcal{P}_k$.

Next, we denote the optimistic policy and optimistic MDP as the solution of the following problem $\tilde{\pi}_k, \tilde{P}_k = \arg\max_{\pi\in\Pi, P'\in\mathcal{P}_k} J_\pi^{P',r_k}$. In addition, we denote by $\tilde{r}_k$ the optimstic average reward, that is, the average reward of the policy $\tilde{\pi}_k$ in the MDP with the optimstic dynamics $\tilde{P}_k$ and reward r_k. We also note that $\tilde{\pi}_k$ is the optimal average reward policy in this MDP by its definition.

We now continue with the case that $P \in \mathcal{P}_k$. The next step is to bound the difference between the optimal average reward r^*_k and the optimistic average reward $\tilde{r}_k$. We note that both $\tilde{r}_k$ and r^*_k are average rewards that correspond to r_k. The difference between them is that r^*_k is the optimal average reward in an MDP with the true dynamics P and $\tilde{r}_k$ is the optimal average reward in an MDP with the optimistic dynamics $\tilde{P}_k$. Thus, the fact that the reward is known, in our case, does not change the fact that that the optimstic reward is a function of the dynamics uncertainty set $\mathcal{P}_k$.

To compute the optimstic policy and dynamics, UCRL2 uses the extended value iteration procedure of (Strehl and Littman, 2008) to efficiently compute the following iterations:

$$\begin{aligned} &u_0(s) = 0 \\ &u_{i+1}(s) = \max_{a\in A}\left\{ r_k(s,a) + \max_{P\in\tilde{P}_k} \sum_{s'\in S} P(s'|s,a)u_i(s') \right\}, \end{aligned} \tag{14}$$

Using Theorem 7 from (Jaksch et al., 2010) we have that running extended value iteration to find the optimistic policy in the optimistic MDP for t_k iterations guarantees that $\tilde{r}_k \geq r^*_k - 1/\sqrt{t_k}$. Thus, we have that:

$$\Delta_k \leq \sum_{s,a} v_k(s,a)(r^*_k - r_k(s,a)) + \sqrt{K} \leq \sum_{s,a} v_k(s,a)(\tilde{r}_k - r_k(s,a)) + 1/\sqrt{t_k} + \sqrt{K}$$

Using Eq. (14), we write the last iteration of the extended value iteration procedure as:

$$u_{i+1}(s) = r_k(s_k, \tilde{\pi}_k(s)) + \sum_{s'\in S} \tilde{P}_k(s'|s,(\tilde{\pi}_k(s)))u_i(s') \tag{15}$$

Theorem 7 from (Jaksch et al., 2010) guarantees that after running extended value iteration for t_k we have that

$$\|u_{i+1}(s) - u_i(s) - \tilde{r}_k\| \leq 1/\sqrt{t_k}. \tag{16}$$

Plugging Eq. (15) in Eq. (16) we have that:

$$\|r_k(s_k, \tilde{\pi}_k(s)) - \tilde{r}_k + \sum_{s'\in S} \tilde{P}_k(s'|s,(\tilde{\pi}_k))u_i(s') - u_i(s)\| \leq 1/\sqrt{t_k}, \tag{17}$$

and therefore

$$\tilde{r}_k - r_k(s_k, a_k) = \tilde{r}_k - r_k(s_k, \tilde{\pi}_k(s)) \leq v_k(\tilde{P}_k - I)u_i + 1/\sqrt{t_k}.$$

In the next step in the proof, the vector u_i is replaced with w_k, which is later upper bounded by the diameter of the MDP D. To conclude, we have that

$$\Delta_k \leq \sum_{s,a} v_k(s,a)(\tilde{r}_k - r_k(s,a)) + 1/\sqrt{t_k} + \sqrt{K} \leq v_k(\tilde{P}_k - I)w_k + 2/\sqrt{t_k} + \sqrt{K}.$$

From this point on, the proof follows by bounding the term $v_k(\tilde{P}_k - I)w_k$, which is only related to the dynamics, and combines all of the previous results into the final result, thus, it is possible to follow the original proof without any modification. Since the leading terms in the original proof come from uncertainty about the dynamics, we obtain the same bound as in the original paper.