Reward is enough for convex MDPs

Tom Zahavy, Brendan O'Donoghue, Guillaume Desjardins and Satinder Singh
{tomzahavy,bodonoghue,gdesjardins,baveja}@deepmind.com
DeepMind, London

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). However, not all goals can be captured in this manner. Specifically, we study Convex MDPs in which goals are expressed as convex functions of stationary distributions and show that they cannot be formulated in this manner. Convex MDPs include the Reinforcement Learning (RL) problem formulation, in which the goal is to maximize long term cumulative reward and is linear in the state occupancy, and extends it into a larger framework that includes many supervised and unsupervised RL problems, such as apprenticeship learning, discovery of diverse skills, constrained MDPs, and pure exploration. We reformulate the convex MDP problem as a min-max game between the policy and cost (negative reward) players, using Fenchel duality, propose a meta-algorithm for solving it and show that it unifies many existing algorithms in the literature. 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 and analyze its sample complexity. Furthermore, we show that for a specific cost player, the non-stationary reward is in fact the gradient of the objective with respect to the state occupancy of previous policies.

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 [58]. 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 [49], *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 [39, 56]. 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*.

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

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 [3] 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 [3, 6]. 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)).

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., [34, 54]), 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 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.

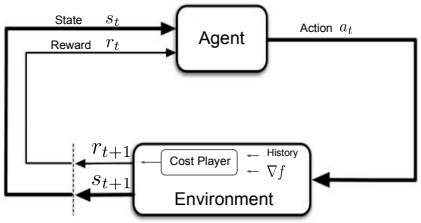

Figure 1: Convex MDP as an RL problem

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.

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) [1, 69]
$d_\pi \cdot \log(d_\pi)$	FTL	Best response	Pure exploration* [29]
$\|\|d_\pi - d_E\|\|_\infty$	OMD	Best response	AL [60, 59]
$\mathbb{E}_c\left[\lambda c \cdot (d_\pi - d_E(c))\right]^\dagger$	OMD	Best response	Inverse RL in contextual MDPs [10]
$\lambda_1 \cdot d_\pi,\ \text{s.t. } \lambda_2 \cdot d_\pi \leq c$	OMD	RL	Constrained MDPs [7, 61, 12, 62, 17, 15, 11]
$\text{dist}(d_\pi, C)^{\dagger\dagger}$	OMD	Best response	Feasibility of convex-constrained MDPs [38]
$\min_{\lambda_1,\ldots,\lambda_k} d_{\pi^k} \cdot \lambda_k$	OMD	RL	Adversarial Markov Decision Processes [51]
$\max_{\lambda \in \Lambda}\ \lambda \cdot (d_\pi - d_E)$	OMD	RL	Online AL [55],Wasserstein GAIL [67, 72]
$\text{KL}(d_\pi \|\| d_E)$	FTL	RL	GAIL [30], state marginal matching [36],
$-\mathbb{E}_z \text{KL}(d_{\pi^z} \|\| \mathbb{E}_k d_{\pi^k})^\ddagger$	FTL	RL	Diverse skill discovery [25, 19, 26, 20, 63, 4]

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 & 6 for more details.

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 state of 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

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

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 B.

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 an occupancy measure it is possible to recover the policy by setting $\pi(s,a) = d_\pi(s,a)/\sum_a d_\pi(s,a)$ in non transient states (where $\sum_a d_\pi(s,a) > 0$) and $\pi(s,a) = 1/|A|$ otherwise. 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 [49])**.** *For the average reward case the set of admissible state-action occupancies is*

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

and for the discounted case it is given by

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

We note that being a polytope implies that $\mathcal{K}$ is a convex and compact set.

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

$$f^{\text{OPT}} = \min_{d_\pi \in \mathcal{K}} f(d_\pi) = \min_{d_\pi \in \mathcal{K}} \max_{\lambda \in \Lambda} \left(\lambda^\top d_\pi - f^*(\lambda)\right) = \max_{\lambda \in \Lambda} \min_{d_\pi \in \mathcal{K}} \left(\lambda^\top d_\pi - f^*(\lambda)\right) \tag{4}$$

where Λ is the closure of (sub-)gradient space $\{\partial f(d_\pi) | d_\pi \in \mathcal{K}\}$, which is a convex set [3, Theorem 4]. As both sets are convex, this is a convex-concave saddle-point problem and a zero-sum two-player

game [48, 43], and we were able to swap the order of minimization and maximization using the minimax theorem [65].

In addition, we assume that $R_{\max} = \max_{\lambda \in \Lambda} \max_i \lambda_i < \infty$, which holds when the set Λ is compact. One way to guarantee that Λ is compact is to consider functions f with lipschitz continuous gradients (which implies bounded gradients since the set $\mathcal{K}$ is compact). For simplicity, we further assume that $R_{\max} = 1$, as is usually done in RL. This will assure that the non stationary rewards produced by the cost player are bounded by 1.

With this we define the Lagrangian as

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

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 [49], 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 f^{OPT}).

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 [3] 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

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

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 2, [3])**.** *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* $\min_{d_\pi \in \mathcal{K}} \mathcal{L}(d_\pi, \bar{\lambda}^K) \geq f^{OPT} - \epsilon_K - \delta_K$ *and* $\max_{\lambda \in \Lambda} \mathcal{L}(\bar{d}_\pi^K, \lambda) \leq f^{OPT} + \epsilon_K + \delta_K$.

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 $f^{\text{OPT}} \leq f(\bar{d}_\pi^K) = \max_\lambda \mathcal{L}(\bar{d}_\pi^K, \lambda) \leq f^{\text{OPT}} + \epsilon_K + \delta_K$. 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

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

Non-convex f. Remark 1 implies that the game $\max_{\lambda \in \Lambda} \min_{d_\pi \in \mathcal{K}} \left(\lambda^\top d_\pi - f^*(\lambda)\right)$ is concave-convex for any function f, so we can solve it with Algorithm 1 (even for a non-convex f). From weak duality, we get that the output of Algorithm 1, $\bar{d}_\pi, \bar{\lambda}$, is a lower bound on the optimal solution f^{OPT}. In addition, we know that $f(d_\pi)$ is always an upper bound on f^{OPT}, thus we get an upper bound and a lower bound on the optimal value: $\mathcal{L}(\bar{d}_\pi, \bar{\lambda}) \leq f^{\text{OPT}} \leq f(\bar{d}_\pi)$. 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.

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:

$$\lambda^k = \arg\max_{\lambda \in \Lambda} \sum\nolimits_{j=1}^{k-1} \mathcal{L}(d_\pi^j, \lambda) = \arg\max_{\lambda \in \Lambda} \lambda \cdot \sum\nolimits_{j=1}^{k-1} d_{\pi^k} - f^*(\lambda) = \nabla f(\bar{d}_\pi^{k-1}), \tag{6}$$

where the last equality follows from the fact that $(\nabla f^*)^{-1} = \nabla f$ [50]. 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 [28]. 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 [31, 28] for more details.

Online mirror descent (OMD), an online version of mirror decent [41, 9] with the following iterates

$$\lambda^k = \arg\max_{\lambda \in \Lambda} \left(\nabla_{\lambda'} \mathcal{L}(d_\pi^{k-1}, \lambda') \Big|_{\lambda' = \lambda^{k-1}} \cdot (\lambda - \lambda^{k-1}) + \alpha_k B_r(\lambda, \lambda^{k-1}) \right),$$

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 [73, OGD] and for $B_r(x) = x \cdot \log(x)$ we get the multiplicative weights [22] as special cases. We also note that OMD is equivalent to a linearized version of Follow the Regularized Leader (FTRL) [37, 27]. Finally, the average regret of OMD is $\bar{R}_K \leq c/\sqrt{K}$, see, for example [27].

4.2 Policy players

4.2.1 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 $\bar{R}_K \leq 0$. 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 [66]. In the non-tabular case we could parameterize a deep neural network to represent the Q-values [39] 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.

4.2.2 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 [34, 16]. 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 an approximate best response policy player. The result is w.r.t the mixed policy $\bar{\pi}^K$, that is produced by running Algorithm 1 for K iterations and then selecting policy π^k with probability $1/K$ where π^k is the policy in iteration k. The state occupancy of $\bar{\pi}^K$ is $\bar{d}_\pi^K$ as defined in Algorithm 1. Other relaxations to the best response for specific algorithms can be found in [59, 38, 32, 29].

Lemma 2 (The sample complexity of approximate best response in convex MDPs)**.** *For a convex function f, running Algorithm 1 with an oracle 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 $\epsilon_k = 1/k$ returns a mixed policy $\bar{\pi}^K$ that satisfies $f(\bar{d}_\pi^K) - f^{OPT} \leq \epsilon$ with probability $1-\delta$ after seeing $O(H^2S^2A\log(2HK/\epsilon\delta)/\epsilon^3\delta^3)$ samples.*

Similarly, for $\bar{R}_K^\lambda = O(1/\sqrt{K})$, setting $\epsilon_k = 1/\sqrt{k}$ requires $O(H^2S^2A\log(2HK/\epsilon\delta)/\epsilon^4\delta^4)$ samples.

4.2.3 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 [45, 42] and frequentist [8, 34] 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.

To support this conjecture we shall prove that this is exactly the case for two RL algorithms from the literature, UCRL2 [34] and MDPO [54]. 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 [34, Definition 1], an arbitrary sequence of known rewards $r^1, \ldots, r^K \in [0,1]$, 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 $\bar{R}_K = \frac{1}{K}\sum_{k=1}^K J_k^\star - J_k^{\tilde{\pi}_k} \leq O(DS\sqrt{A\log(K/\delta)/K})$.*

In the supplementary material (Appendix F), we provide a proof sketch that closely follows [34]. We also note that UCRL2 was analyzed in [51] 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 $\bar{R}_K \leq O(HS\sqrt{A\log(K)/K})$ [51, Corollary 5].

Another optimistic algorithm is Mirror Descent Policy Optimization [54, MDPO], a model free RL algorithm that is very similar to popular DRL algorithms like TRPO [52] and MPO [2]. In [23, 53, 5], the authors established the global convergence of MDPO and in [14, 54], 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, [55]))**.** *For an arbitrary sequence of known rewards $r^1, \dots, r^K$, the average regret of MDPO is at most $\bar{R}_K \leq O(H^2 S\sqrt{A/K})$.*

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 [64]. We note that MDPO analysis assumes that we can solve the mirror descent sub problem exactly which is only feasible for small tabular problems [53]. When function approximation is used, e.g. in the DRL setup, the mirror descent problem is usually solved with gradient descent [52, 2, 64], and as a result the error that comes from not solving it exactly has to be considered.

Next, we give a PAC(ϵ, δ) sample complexity result for the mixed policy $\bar{\pi}^K$, that is produced by running Algorithm 1 with UCRL2 as a policy player. The result is given for UCRL2 but can easily be adapted to MDPO.

Lemma 5 (The sample complexity of non-stationary RL algorithms in convex MDPs)**.** *For a convex function f, running Algorithm 1 with an oracle cost player with regret $\bar{R}_K^\lambda \leq c_0/\sqrt{K}$ and UCRL2 as a policy player returns a mixed policy $\bar{\pi}^K$ that satisfies $f(\bar{d}_\pi^K) - f^{OPT} \leq \epsilon$ with probability $1-\delta$ after $K = O\left(\frac{D^2 S^2 A}{\delta^2 \epsilon^2} \log(\frac{2DSA}{\delta\epsilon})\right)$ steps.*

Discussion. Comparing the results in Lemma 5 with Lemma 2 suggests that using an RL algorithm with non stationary reward as a policy player requires O$(1/\epsilon^2)$ samples to find an $\epsilon-$optimal policy, while using an approximate best response requires O$(1/\epsilon^3)$. In first glance, this results also improves the previously best known result of Hazan et al. [29] for approximate FW that requires O$(1/\epsilon^3)$ samples. Furthermore, the sample complexity of the standard RL problem is also known to be O$(1/\epsilon^2)$, see, for example, [16], for matching upper and lower bounds. Thus, our results also suggest that convex RL is as hard as the standard RL problem in terms of ϵ. However, there are more details that have to be considered as we now discuss.

First, while our focus is on the dependence in ϵ, our bound Lemma 5 is not tight in δ, i.e., it scales with $1/\delta^2$, whereas it should be possible to achieve a $\log(1/\delta)$ scaling. The reason for that is the regret is only guaranteed to be bounded with high probability (Lemma 4 & Lemma 3) and when it is not, we have to assume the worst case in the analysis. When we use these high probability regret bounds for an online to batch conversion, they result in a polynomial scaling with δ. We believe that it should be possible to improve our algorithm in future work to scale better with δ, e.g., via the online to batch conversion in [35, Appendix F.].

Second, Lemma 5 and Lemma 2 assume that a cost player with some regret is given, and analyzes the sample complexity of the different policy players given such a cost player. However, we did not consider how to implement such a cost player. The main challenge is that the agent does not have access to the true state occupancy and has to estimate it from samples. Concretely, if we sample trajectories from the current policy to estimate the state occupancy in each round (as in [29]), it will require O$(1/\epsilon^3)$ trajectories overall. The issue with this approach is that the trajectories from previous iterations are not used in the current iterations. Instead, a more sample efficient approach will be to use all the samples to learn a model (which is already done in UCRL2) and then, given a policy, to compute the state occupancy using the model. We hope that in future work, such an approach will lead to a O$(1/\epsilon^2)$ sample complexity, closing the gap from standard RL.

5 Convex constraints

We have restricted the presentation so far to unconstrained convex problems, in this section we extend 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 [7, 61, 12, 62, 17, 15, 11]. We can use the same Fenchel dual machinery we developed before, but now taking into account the constraints. Consider the Lagrangian

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

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

$$L(d_\pi, \mu, \nu, \zeta_1, \ldots, \zeta_m) = \nu^\top d_\pi - f^*(\nu) + \sum_{i=1}^m \left(d_\pi \zeta_i - \mu_i g_i^*(\zeta_i/\mu_i)\right). \tag{7}$$

This is convex in d_π and concave in $(\nu, \mu, \zeta_1, \ldots, \zeta_m)$, since it includes the perspective transform of the functions g_i [13]. The Lagrangian involves a cost vector, $\nu + \sum_{i=1}^m \zeta_i$, linearly interacting with d_π, and therefore we can use the same policy players as before to minimize this cost. For the cost player, it is possible to use OMD on Eq. (7) jointly for the variables ν, μ and ζ. It is more challenging to use FTL for the joint variables as the maximizer for the μ variable (the Lagrange multiplier) will be infinity when the constrained is violated. Instead, we would recommend splitting the cost player into two independent players, one for ν and ζ and another for μ.

Analyzing the regret of this 3-player game is out of the scope of this paper. Instead, we quickly describe how the players might look like. For the μ player we would recommend using gradient descent (or OMD), similar to linearly constrained MDPs [7]. For the ν, ζ player, it is possible to use FTL as follows. Since ν and ζ are independent, ν is equivalent to λ (in the unconstrained problem) and becomes gradient of f as before. For ζ, the situation is slightly more complicated as it appears jointly in $g^*(\frac{\zeta}{\mu})$ (the Fenchel conjugate of the constraint) via the fraction $(\frac{\zeta}{\mu})$. In this case, the FTL player for ζ will be

$$\begin{aligned}\zeta_i^k &= \arg\max_{\zeta_i \in \Lambda} \sum_{j=1}^{k-1} \mathcal{L}(d_\pi^j, \nu, \mu^j, \zeta) = \arg\max_{\zeta_i \in \Lambda} \bar{d}_\pi^{k-1} \zeta_i - \frac{1}{K} \sum_{j=1}^{k-1} \mu_i^k g_i^*(\zeta_i/\mu_i^k) \\ &= \frac{1}{K} \sum_{j=1}^{k-1} \mu_i^k \nabla g(\bar{d}_\pi^{k-1}),\end{aligned}$$

where the last equality follows from the fact that for $f(x) = \sum_k a_k h(x/a_k)$, $f^*(x) = \sum_k a_k h^*(x)$.

6 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.

6.1 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 [1] 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 $||y||_* \leq 1$ and ∞ otherwise, where $||\cdot||_*$ is the dual norm. Plugging f^* in Eq. (4) results in the following max-min game:

$$\max_{d_\pi \in \mathcal{K}} \min_{||\lambda||_* \leq 1} \lambda \cdot d_E - \lambda \cdot d_\pi. \tag{8}$$

Eq. (8) 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. (8) can also be used without an expert ($d_E = 0$) to find a policy that is robust to the worst case reward [70].

Alg_λ=OMD, Alg_π=best response/RL. The Multiplicative Weights AL algorithms [59, MWAL] was proposed to solve the AL problem with $f = ||d_\pi - d_E||_\infty$, such that the dual norm in Eq. (8) 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 [10] and to find feasible solutions to convex-constrained MDPs [38]. We note that in practice, the best response can only be solved approximately, as we discussed in Section 4. Instead, in Online AL [55], the authors proposed to use MDPO as the policy player which gurantess regret of at most $\bar{R}_K \leq c/\sqrt{K}$. They showed that their algorithm is equivalent to Wasserstein GAIL [67, 72] and performs similarly to GAIL.

Alg$_\lambda$=FTL, Alg$_\pi$=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 [21, 3] 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 [1] 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 [1] and was later shown formally in [69]. 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 [29]. Specifically, in [29], 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 [32] 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 [33, 69]. 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:

$$f\big((1-\alpha_k)\bar{d}_\pi^k + \alpha_k d_\pi^{k+1}\big) \geq \min_{x \in \mathrm{Co}(\bar{d}_\pi^k, d_\pi^{k+1})} f(x) \geq \min_{x \in \mathrm{Co}(d_\pi^1, d_\pi^2, \ldots, d_\pi^{k+1})} f(x). \tag{9}$$

Interestingly, the second algorithm of Abbeel and Ng [1], 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 [1].

6.2 GAIL and DIAYN: Alg$_\lambda$=FTL, Alg$_\pi$=RL

We now show how mutual-information based diversity objectives [25, 19] can be derived through the lens of convex MDPs. To do so, we reformulate the objective of DIAYN [19] as a convex RL problem with the following objective function (see Appendix E for details):

$$\mathbb{E}_z \mathrm{KL}(d_{\pi^z} || \mathbb{E}_k d_{\pi^k}). \tag{10}$$

Intuitively, Eq. (10) 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. (10) is convex because the KL-divergence is jointly convex in both arguments [13, 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. (10) 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 $\nabla_{d_{\pi^z}} \mathrm{KL}(d_{\pi^z} || \sum_k p(k) d_{\pi^k}) =$

$$\mathop{\mathbb{E}}_{z \sim p(z)} \Big[\log \frac{d_{\pi^z}}{\sum_k d_{\pi^k} p(k)} + 1 - \frac{d_{\pi^z} p(z)}{\sum_k d_{\pi^k} p(k)} \Big] = \mathop{\mathbb{E}}_{z \sim p(z)} \Big[\underbrace{\log(p(z|s)) - \log(p(z))}_{\text{Mutual Information}} \underbrace{+1 - p(z|s)}_{\text{Gradient correction}} \Big], \tag{11}$$

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 Appendix G. Finally, note that the reward in DIAYN is the first term in Eq. (11) without a negative sign. This implies that DIAYN performs convex *maximization*, which is a hard problem in general.

GAIL. We further show how Eq. (10) extends to GAIL [30] 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. (11), one retrieves the GAIL [30] 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.

7 Discussion

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) and proposed a meta algorithm for solving it. In the previous section we observed that many algorithms in the convex RL literature, can be explained as instances of Algorithm 1 by selecting different pairs of algorithms for the policy and cost players. We would like to emphasize, that this is not suggesting that a different combination of the two players *should* be used in different occasions, but rather, that we can observe in hindsight that different algorithms were preferred in different problems over the years.

In principle, any combination of the two players can be used to solve any convex RL problem. The FW algorithm, which combines best response with FTL, was originally proposed for AL [1, 69] but can be used for for any convex RL problem as was suggested in [29]. Zhang et al. [71], unified supervised RL objectives like reward maximization, AL and linear constraints with unsupervised RL objectives like maximum entropy exploration under the framework of convex RL. We extended the framework to allow convex constraints (Section 5) and explained the objective of GAIL as a convex MDP (Section 6.2). We also discussed non convex objectives (Section 3) and analyzed unsupervised skill discovery via the maximization of mutual information (Section 6.2) as a special case. While the theoretical guarantees in this case are weaker (Section 3), diversity maximizing algorithms perform well in practice and we hope that thinking on them under the convex MDP framework will be useful. Finally, we would like to point out a recent work by Geist et al. [24], which was published concurrently to ours, and studies the convex RL problem from the viewpoint of mean field games.

There are also algorithms for convex MDPs that cannot be explained as instances of Algorithm 1. In particular, Zhang et al. [71] 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 [71, Theorem 4.5] to hold, the saddle point problem has to be solved exactly, while in practice it is only solved approximately [71, Algorithm 1], which hinders its sample efficiency, similar to the approximate best response (Section 4). We would also like to mention that Fenchel duality was also used in off policy evaluation (OPE) in [40, 68]. 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.

In order to solve a practical convex RL problem in a given domain, our advice would be to use an RL algorithm that is known to be high performing in the standard RL problem in this domain as a policy player. From the theoretical point of view, this should be MDPO or UCRL2, for which we have proved a regret bound. From the practical point of view, we would recommend using a high performing DRL algorithm, which may be specific to the domain. This design choice is inspired by its simplicity, from our sample complexity analysis and from the fact that it has been adapted by practitioners in various problems (e.g. GAIL, DIAYN). For the cost player, we would recommend FTL, i.e., to compute the gradient of the objective.

8 Acknowledgements

We would like to thank Yasin Abbasi-Yadkorie, Vlad Mnih, Jacob Abernethy, Lior Shani and Doina Precup for their comments and discussion on this work. Work done at DeepMind, the author(s) received no specific funding for this work.

References

[1] 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.

[2] 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.

[3] 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.

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

[5] 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.

[6] S. Agrawal and N. R. Devanur. Fast algorithms for online stochastic convex programming. In *Proceedings of the twenty-sixth annual ACM-SIAM symposium on Discrete algorithms*, pages 1405–1424. SIAM, 2014.

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

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

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

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

[11] 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.

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

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

[14] 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.

[15] 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.

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

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

[18] 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.

[19] 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.

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

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

[22] 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.

[23] 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.

[24] M. Geist, J. Pérolat, M. Laurière, R. Elie, S. Perrin, O. Bachem, R. Munos, and O. Pietquin. Concave utility reinforcement learning: the mean-field game viewpoint. *arXiv preprint arXiv:2106.03787*, 2021.

[25] 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.

[26] 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.

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

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

[29] 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.

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

[31] 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.

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

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

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

[35] E. Kaufmann, P. Ménard, O. D. Domingues, A. Jonsson, E. Leurent, and M. Valko. Adaptive reward-free exploration. In *Algorithmic Learning Theory*, pages 865–891. PMLR, 2021.

[36] 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.

[37] 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.

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

[39] 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.

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

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

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

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

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

[45] 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.

[46] 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.

[47] 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.

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

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

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

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

[52] 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.

[53] 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, 2020.

[54] 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, 2020.

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

[56] 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.

[57] 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.

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

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

[60] 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.

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

[62] 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.

[63] 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.

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

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

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

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

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

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

[70] 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.

[71] 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*, 2020.

[72] 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, 2020.

[73] 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.