Reward is Enough for Convex MDPs

Tom Zahavy
DeepMind, London
tomzahavy@deepmind.com

Brendan O'Donoghue
DeepMind, London
bodonoghue@deepmind.com

Guillaume Desjardins
DeepMind, London
gdesjardins@deepmind.com

Satinder Singh
DeepMind, London
baveja@deepmind.com

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. In this paper we study convex MDPs in which goals are expressed as convex functions of the stationary distribution and show that they cannot be formulated using stationary reward functions. Convex MDPs generalize the standard reinforcement learning (RL) problem formulation to a larger framework that includes many supervised and unsupervised RL problems, such as apprenticeship learning, constrained MDPs, and so-called 'pure exploration'. Our approach is to reformulate the convex MDP problem as a min-max game involving policy and cost (negative reward) 'players', using Fenchel duality. We propose a meta-algorithm for solving this problem and show that it unifies many existing algorithms in the literature.

1 Introduction

In reinforcement learning (RL), an agent learns how to map situations to actions so as to maximize a cumulative scalar reward signal. The learner is not told which actions to take, but instead must discover which actions lead to the most reward [64]. Mathematically, the RL problem can be written as finding a policy whose state occupancy has the largest inner product with a reward vector [55], *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 (see Definition 1). A significant body of work is dedicated to solving the RL problem efficiently in challenging domains [45, 62]. However, not all decision making problems of interest take this form. In particular we consider the more general *convex* MDP problem,

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

where $f : \mathcal{K} \to \mathbb{R}$ is a convex function. Sequential decision making problems that take this form include Apprenticeship Learning (AL), 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 MDP problem. More specifically, we make the following contributions.

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

Firstly, 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 MDP 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).

Secondly, we propose a sample efficient policy player that uses a standard RL algorithm (*eg*, [35, 60]), and computes an optimistic policy with respect to 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 in the min-max game we formulate to solve the convex MDP. Our main result is that the average of the policies produced by the policy player converges to a solution to the convex MDP problem (Eq. (2)). Inspired by this principle, we also propose a recipe for using deep-RL (DRL) agents to solve convex MDPs heuristically: provide the agent non-stationary rewards from the cost player. We explore this principle in our experiments.

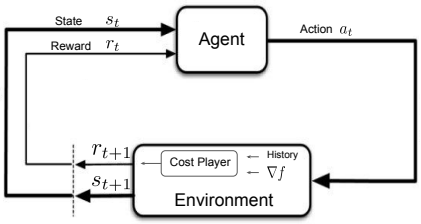

Figure 1: Convex MDP as an RL problem

Finally, we show that choosing specific algorithms for the policy and cost players unifies several disparate branches of RL problems, such as apprenticeship learning, 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, 75]
$d_\pi \cdot \log(d_\pi)$	FTL	Best response	Pure exploration* [30]
$\|\|d_\pi - d_E\|\|_\infty$	OMD	Best response	AL [66, 65]
$\mathbb{E}_c\left[\lambda c \cdot (d_\pi - d_E(c))\right]^\dagger$	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, 67, 12, 68, 18, 16, 11]
$\text{dist}(d_\pi, C)^{\dagger\dagger}$	OMD	Best response	Feasibility of convex-constrained MDPs [44]
$\min_{\lambda_1,\ldots,\lambda_k} d_\pi^k \cdot \lambda_k$	OMD	RL	Adversarial Markov Decision Processes [57]
$\max_{\lambda \in \Lambda}\ \lambda \cdot (d_\pi - d_E)$	OMD	RL	Online AL [61],Wasserstein GAIL [73, 78]
$\text{KL}(d_\pi \|\| d_E)$	FTL	RL	GAIL [31], state marginal matching [41],
$-\mathbb{E}_z \text{KL}(d_\pi^z \|\| \mathbb{E}_k d_\pi^k)^\ddagger$	FTL	RL	Diverse skill discovery [26, 20, 27, 21, 69, 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

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

for the average reward case and discounted case respectively. 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 measures how often the agent visits each state-action when following π. Let $\mathbb{P}_\pi(s_t = \cdot)$ be the probability measure over states at time t under policy π, then

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

for the average reward case and the discounted case respectively. With these, we can rewrite the RL objective in Eq. (3) 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 $\pi(s,a) = d_\pi(s,a)/\sum_a d_\pi(s,a)$ if $\sum_a d_\pi(s,a) > 0$, and $\pi(s,a) = 1/|A|$ otherwise. Accordingly, in this paper we shall formulate the RL problem using the state-action occupancy measure, and both the standard RL problem (Eq. (1)) and the convex MDP problem (Eq. (2)) are convex optimization problems in variable d_π. 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 [55])**.** *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.

The convex MDP problem is defined for the tuple (S, A, P, f) in the average cost case and $(S, A, P, f, \gamma, d_0)$ in the discounted case. This tuple is defining a state-action occupancy's polytope $\mathcal{K}$ (Definition 1), and the problem is to find a policy π whose state occupancy d_π is in this polytope and minimizes the function f (Eq. (2)).

3 A Meta-Algorithm for Solving Convex MDPs via RL

To solve the convex MDP problem (Eq. (2)) we need to find an occupancy measure d_π (and 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 challenging one due to the nature of learning about the environment through stochastic interactions. In this section we show how to reformulate the convex MDP problem (Eq. (2)) so that standard RL algorithms can be used to solve it, allowing us to harness decades of work on solving vanilla RL problems. 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 denoted* $f^* : \mathbb{R}^n \to \mathbb{R} \cup \{-\infty, \infty\}$ *and 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 MDP problem (Eq. (2)) as

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

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 [54, 49], and we were able to swap the order of minimization and maximization using the minimax theorem [71].

With this we define the Lagrangian as $\mathcal{L}(d_\pi, \lambda) \coloneqq \lambda \cdot d_\pi - f^*(\lambda)$. For a fixed $\lambda \in \Lambda$, 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 $d_\pi^\star = \operatorname{argmin}_{d_\pi \in \mathcal{K}} \mathcal{L}(\cdot, \lambda^\star)$ for the optimal occupancy measure. 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 note that for any reward r there is a deterministic policy that optimizes the reward [55], but for some choices of f no deterministic policy is optimal, *eg*, when f is the negative entropy function. This result tells us that even if we have access to an optimal dual-variable we cannot simply use it to recover the stationary distribution that solves the convex MDP problem in general.

To overcome this issue we develop an algorithm that generates a *sequence* of policies $\{\pi^k\}_{k \in \mathbb{N}}$ such that the average converges to an optimal policy for Eq. (2), *i.e.*, $(1/K)\sum_{k=1}^K d_\pi^k \to d_\pi^\star \in \arg\min_{d_\pi \in \mathcal{K}} f(d_\pi)$. The algorithm we develop is described in Algorithm 1 and is adapted from the meta-algorithm described in Abernethy and Wang [3]. It is referred to as a *meta-algorithm* since it relies on supplied sub-routine algorithms 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. The cost algorithm Alg_λ can 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 MDPs

1: **Input:** convex-concave payoff $\mathcal{L} : \mathcal{K} \times \Lambda \to \mathcal{R}$, algorithms Alg_λ, Alg_π, $K \in \mathbb{N}$
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 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 $\ell_1, \ell_2, \ldots, \ell_K : \mathcal{K} \to \mathbb{R}$ 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 $\bar{R}_K^\pi \le \epsilon_K$ and $\bar{R}_K^\lambda \le \delta_K$, respectively. Then Algorithm 1 outputs $\bar{d}_\pi^K$ and $\bar{\lambda}^K$ satisfying $\min_{d_\pi \in \mathcal{K}} \mathcal{L}(d_\pi, \bar{\lambda}^K) \ge f^{OPT} - \epsilon_K - \delta_K$ and $\max_{\lambda \in \Lambda} \mathcal{L}(\bar{d}_\pi^K, \lambda) \le f^{OPT} + \epsilon_K + \delta_K$.*

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 MDP problem (Eq. (2)) to any desired tolerance, this is because $f^{\text{OPT}} \le f(\bar{d}_\pi^K) = \max_\lambda \mathcal{L}(\bar{d}_\pi^K, \lambda) \le f^{\text{OPT}} + \epsilon_K + \delta_K$. For example, we shall later present algorithms that have regret bounded as $\epsilon_K = \delta_K \le O(1/\sqrt{K})$, in which case we have

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

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

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.

In addition, in this section we assume that

$$\lambda_{\max} = \max_{\lambda \in \Lambda} \max_{s,a} |\lambda(s,a)| < \infty,$$

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 $\lambda_{\max} \leq 1$. By making this assumption we assure that the non stationary rewards produced by the cost player are bounded by 1 as is usually done in RL.

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:

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

where $\bar{d}_\pi^{k-1} = \sum_{j=1}^{k-1} d_\pi^j$ and the last equality follows from the fact that $(\nabla f^*)^{-1} = \nabla f$ [56]. The average regret of FTL is guaranteed to be $\bar{R}_K \leq c/\sqrt{K}$ under some assumptions [29]. 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 [32, 29] for more details.

Online Mirror Descent (OMD) uses the following update [47, 9]:

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

where α_k is a learning rate and B_r is a Bregman divergence [14]. For $B_r(x) = 0.5||x||_2^2$, we get online gradient descent [79] and for $B_r(x) = x \cdot \log(x)$ we get multiplicative weights [23] as special cases. We also note that OMD is equivalent to a linearized version of Follow the Regularized Leader (FTRL) [43, 28]. The average regret of OMD is $\bar{R}_K \leq c/\sqrt{K}$ under some assumptions, see, for example [28].

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 $\bar{R}_K \leq 0$ (this requires knowledge of the *current* loss function, which is usually not applicable but is in this case). 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 [72]. In the non-tabular case we could parameterize a deep neural network to represent the Q-values [45] 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 MDP 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. This leads to defining an approximate best response via the Probably Approximately Correct (PAC) framework. We say that an policy player is a PAC(ϵ, δ), if it finds an ϵ-optimal policy to an RL problem with probability of at least $1-\delta$. In addition, we say that a policy π' is ϵ-optimal if its state occupancy d'_π is such that

$$\max_{d_\pi \in \mathcal{K}} d_\pi \cdot (-\lambda^k) - d'_\pi \cdot (-\lambda^k) \leq \epsilon.$$

For example, the algorithm in [40] can find ϵ-optimal policy after seeing $O\left(\frac{SA}{(1-\gamma)^3\epsilon^2}\log(\frac{1}{\delta})\right)$ samples in the discounted RL problem, and the algorithm in [36] requires $O\left(\frac{t_{\text{mix}}^2 SA}{\epsilon^2}\log(\frac{1}{\delta})\right)$ samples to find ϵ-optimal policy for the average reward RL problem, where t_{mix} is the mixing time (see [42, 76], for example, for a formal definition). The following Lemma analyzes the sample complexity of Algorithm 1 with an approximate best response policy player for the average reward RL problem [36]. The result can be easily extended to the discounted case using the algorithm in [40]. Other relaxations to the best response for specific algorithms can be found in [65, 44, 33, 30].

Lemma 2 (The sample complexity of approximate best response in convex MDPs with average occupancy measure)**.** *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 average reward RL problem in iteration k to accuracy $\epsilon_k = 1/k$ returns an occupancy measure $\bar{d}_\pi^K$ that satisfies $f(\bar{d}_\pi^K) - f^{OPT} \leq \epsilon$ with probability $1-\delta$ after seeing $O(t_{mix}^2 SA\log(2K/\epsilon\delta)/\epsilon^3\delta^3)$ samples. Similarly, for $\bar{R}_K^\lambda = O(1/\sqrt{K})$, setting $\epsilon_k = 1/\sqrt{k}$ requires $O(t_{mix}^2 SA\log(2K/\epsilon\delta)/\epsilon^4\delta^4)$ 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 cost $-\lambda_k$. In our setup the reward is known and deterministic but non-stationary, while in the standard RL setup it is unknown, stochastic, and stationary. We conjecture that any stochastic RL algorithm can be adapted to the *known* non-stationary reward setup we consider here. In most cases both Bayesian [51, 48] and frequentist [8, 35] approaches to the stochastic RL problem solve a modified (*eg*, by adding optimism) Bellman equation at each time period and 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 UCRL2 [35]. UCRL2 is an RL algorithm that was designed and analyzed in the standard RL setup, and we shall show that it is easily adapted to the non-stationary but known reward setup that we require. To make this claim more general, we will also discuss a similar result for the MDPO algorithm [61] that was given in a slightly different setup.

UCRL2 is a model based algorithm that maintains an estimate of the reward and the transition function as well as confidence sets about those estimates. In our case the reward at time k is known, so we only need to consider uncertainty in the dynamics. UCRL2 guarantees that in any iteration k, the true transition function is in a confidence set with high probability, *i.e.*, $P \in \mathcal{P}_k$ for confidence set $\mathcal{P}_k$. 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. In the following results for UCRL2 we will use the constant D, which denotes the diameter of the MDP, see [35, Definition 1] for more details. In the supplementary material (Appendix E), we provide a proof sketch that closely follows [35].

Lemma 3 (Non stationary regret of UCRL2)**.** *For an MDP with dynamics P, diameter D, an arbitrary sequence of known and bounded rewards $\left\{r^i : \max_{s,a}|r^i(s,a)| \leq 1\right\}_{i=1}^K$, such that the optimal average reward at episode k, with respect to P and r_k is $J_k^\star$, then with probability at least $1-\delta$, the average regret of UCRL2 is at most $\bar{R}_K = \frac{1}{K}\sum_{k=1}^K J_k^\star - J_k^{\tilde{\pi}_k} \leq O(DS\sqrt{A\log(K/\delta)/K})$.*

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.

Lemma 4 (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 an occupancy measure $\bar{d}_\pi^K$ that satisfies $f(\bar{d}_\pi^K) - f^{OPT} \leq \epsilon$ with probability $1-\delta$ after $K = O\left(\frac{D^2S^2A}{\delta^2\epsilon^2}\log(\frac{2DSA}{\delta\epsilon})\right)$ steps.*

MDPO. Another optimistic algorithm is Mirror Descent Policy Optimization [60, MDPO]. MDPO is a model free RL algorithm that is very similar to popular DRL algorithms like TRPO [58] and MPO [2]. In [24, 59, 5], the authors established the global convergence of MDPO and in [15, 60], the authors showed that MDPO with optimistic exploration enjoys low regret.

The analysis for MDPO is given in a finite horizon MDP with horizon H, which is not the focus of our paper. Nevertheless, to support our conjecture that any stochastic RL algorithm can be adapted to the *known* non-stationary reward setup, we quickly discuss the regret of MDPO in this setup. We also note that MDPO is closer to practical DRL algorithms [70]. In a finite horizon MDP with horizon H and known, non-stationary and bounded rewards, the regret of MDPO is bounded by $\bar{R}_K \leq O(H^2S\sqrt{A/K})$ [61, Lemma 4] with high probability.

To compare this result with UCRL2, we refer to a result from [57], which analyzed UCRL2 in the adversarial setup, that includes our setup as a special case. In a finite horizon MDP with horizon H it was shown that setting $\delta = SA/K$ with probability $1-\delta$ its regret is bounded by $\bar{R}_K \leq O(HS\sqrt{A\log(K)/K})$ [57, Corollary 5], which is better by a factor of H than MDPO.

Discussion. Comparing the results in Lemma 4 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. [30] for approximate Frank-Wolfe (FW) that requires O$(1/\epsilon^3)$ samples. However, there are more details that have to be considered as we now discuss.

Firstly, Lemma 4 and Lemma 2 assume access to an oracle cost player with some regret and do not consider how to implement such a cost player. The main challenge is that the cost player does not have access to the true state occupancy and must estimate it from samples. Concretely, if we sample trajectories from the current policy to estimate the state occupancy in each round (as in [30]), we will require O$(1/\epsilon^3)$ trajectories overall. The issue with this naive approach is that the trajectories in each iteration have to be sampled from the current policy, and therefore, we cannot use samples from previous policies to estimate the state occupancy of the current policy. Instead, a more sample efficient approach will be to use all the samples to learn a model (which is already done in UCRL2). Then, given the model and the policy, we can compute an approximation of the state occupancy. We conjecture that such an approach will lead to a O$(1/\epsilon^2)$ sample complexity, closing the gap with standard RL.

Secondly, while our focus is on the dependence in ϵ, our bound Lemma 4 is not tight in δ, *i.e.*, it scales with $1/\delta^2$ where it should be possible to achieve a $\log(1/\delta)$ scaling. Again we conjecture an improvement in the bound is possible [38, Appendix F.].

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 where both f and g_i are linear [7, 67, 12, 68, 18, 16, 11]. We can use the same Fenchel dual machinery we developed before, but now taking into account the constraints. Consider the Lagrangian

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

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

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

This is convex (indeed affine) 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 best-response and FTL for the cost-player variables as the maximum value of the Lagrangian is unbounded for some values of d_π. Another option is to treat the problem as a *three*-player game. In this case the policy player controls d_π as before, one cost player chooses $(\nu, \zeta_1, \ldots, \zeta_m)$ and can use the algorithms we have previously discussed, and the other cost player chooses μ with some restrictions on their choice of algorithm. Analyzing the regret in that case is outside the scope of this paper.

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 and the cost and policy player algorithms Alg_λ and Alg_π. We summarizes the relationships in Table 1.

6.1 Apprenticeship Learning

In apprenticeship learning (AL), we have an MDP without an explicit reward function. Instead, an expert provides demonstrations which are used to estimate the expert state occupancy measure 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 $f(d_\pi) = ||d_\pi - d_E||$. The convex conjugate of f is given by $f^*(y) = y \cdot d_E$ if $||y||_* \leq 1$ and ∞ otherwise, where $||\cdot||_*$ denotes the dual norm. Plugging f^* into Eq. (4) results in the following game:

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

Inspecting Eq. (8), we can see that the norm in the function f that is used to measure the distance from the expert induces a constraint set for the cost variable, which is a unit ball in the dual norm.

Alg_λ=OMD, Alg_π=Best Response/RL. The Multiplicative Weights AL algorithms [65, MWAL] was proposed to solve the AL problem with $f(d_\pi) = ||d_\pi - d_E||_\infty$. It uses the best response as the policy player and multiplicative weights as the cost player (a special case of OMD). MWAL has also been used to solve AL in contextual MDPs [10] and to find feasible solutions to convex-constrained MDPs [44]. We note that in practice the best response can only be solved approximately, as we discussed in Section 4. Instead, in online AL [61] the authors proposed to use MDPO as the policy player, which guarantees a regret bound of $\bar{R}_K \leq c/\sqrt{K}$. They showed that their algorithm is equivalent to Wasserstein GAIL [73, 78] and in practice tends to perform 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 [22, 3] for minimizing f (Eq. (2)). Pseudo-code for this is included in the appendix (Algorithm 3). The algorithm finds a point $d_\pi^k \in \mathcal{K}$ that has the largest inner-product (best response) with the negative gradient (*i.e.*, 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 [75]. Thus, it is a projection free algorithm in the sense that it avoids projecting d_π into $\mathcal{K}$, despite the name. In their case the gradient is given by $\nabla_f(d_\pi) = d_\pi - d_E$. 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 convex MDPs more generally [30]. Specifically, in [30], the authors considered the problem of pure exploration, which they defined as finding a policy that maximizes entropy.

Fully Corrective FW. The FW algorithm has many variants (see [33] for a survey) some of which enjoy faster rates of convergence in special cases. Concretely, when the constraint set is a polytope, which is the case for convex MDPs (Definition 1), some variants achieve a linear rate of convergence [34, 75]. 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 occupancy measure at the previous time-step.

This is guaranteed to be at least as good as the learning rate update. 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 better theoretical convergence rate than the 'projection' variant, as was observed empirically in [1].

6.2 GAIL and DIAYN: Alg_λ=FTL, Alg_π=RL

We now discuss the objectives of two popular algorithms, GAIL [31] and DIAYN [20], which perform AL and diverse skill discovery respectively. Our analysis suggests that GAIL and DIAYN share the same objective function. In GAIL, this objective function is minimized, which is a convex MDP, however, in DIAYN it is maximized, which is therefore not a convex MDP. We start the discussion with DIAYN and follow with a simple construction showing the equivalence to GAIL.

DIAYN. Discriminative approaches [26, 20] 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 [20] is the mutual information:

$$I(S; Z) = \mathbb{E}_{z\sim p; s\sim d_\pi^z}[\log p(z|s) - \log p(z)], \tag{9}$$

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. 9 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 $|Z|$ outcomes, yielding $|Z|$ policies π^z, and q_ϕ is a classifier over these $|Z|$ skills with parameters ϕ.

We now show that a similar intrinsic reward 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. (9) yields:

$$\begin{aligned}\mathbb{E}_{z\sim p(z), s\sim d_\pi^z}[\log p(z|s) - p(z)] &= \sum_z p(z) \sum_s d_\pi^z(s) \left[\log\left(\frac{d_\pi^z(s)p(z)}{\sum_k d_\pi^k(s)p(k)}\right) - \log p(z)\right] \\ &= \sum_z p(z)\text{KL}(d_\pi^z || \sum_k p(k)d_\pi^k) = \mathbb{E}_z \text{KL}(d_\pi^z || \mathbb{E}_k d_\pi^k),\end{aligned} \tag{10}$$

where KL denotes the Kullback–Leibler divergence [39].

Intuitively, finding a set of policies $\pi^1, \ldots, \pi^z$ that minimize Eq. (10) will result in finding policies that visit similar states, measured using the KL distance between their respective state occupancies $d_\pi^1, \ldots, d_\pi^z$. This is a convex MDP because the KL-divergence is jointly convex in both arguments [13, Example 3.19]. We will soon show that this is the objective of GAIL. On the other hand, a set of policies that maximize Eq. (10) is diverse, as the policies visit different states, measured using the KL distance between their respective state occupancies $d_\pi^1, \ldots, d_\pi^z$.

We follow on with deriving the FTL player for the convex MDP in Eq. (10). We will then show that this FTL player is producing an intrinsic reward that is equivalent to the intrinsic reward used in GAIL and DIAYN (despite the fact that DIAYN is not a convex MDP). According to Eq. (6), the FTL cost player will produce a cost λ^k at iteration k given by

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

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 the reward of DIAYN. These algorithms differ however for more general priors $p(z)$, which we explore empirically in Appendix F.

GAIL. We further show how Eq. (10) extends to GAIL [31] via a simple construction. Consider a binary latent space of size $|Z| = 2$, where $z = 1$ corresponds to the policy of the agent and $z = 2$ corresponds to the policy of the expert which is fixed. In addition, consider a uniform prior over the latent variables, *i.e.*, $p(z = 1) = \frac{1}{2}$. By removing the constant terms in Eq. (11), one retrieves the GAIL [31] 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).

7 Discussion

In this work we reformulated the convex MDP 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.

We observed that many algorithms in the literature can be interpreted as instances of the meta-algorithm by selecting different pairs of subroutines employed by the policy and cost players. The Frank-Wolfe algorithm, which combines best response with FTL, was originally proposed for AL [1, 75] but can be used for any convex MDP problem as was suggested in [30]. Zhang et al. [77], unified the problems of RL, AL, constrained MDPs with linear constraints and maximum entropy exploration under the framework of convex MDPs. 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. Finally, we would like to point out a recent work by Geist et al. [25], which was published concurrently to ours, and studies the convex MDP 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. [77] 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 since we solve a single saddle point problem iteratively, and furthermore we have much more flexibility about which algorithms the policy player can use. Moreover, for the convergence guarantee [77, Theorem 4.5] to hold, the saddle point problem has to be solved exactly, while in practice it is only solved approximately [77, Algorithm 1], which hinders its sample efficiency. Fenchel duality has also been used in off policy evaluation (OPE) in [46, 74]. 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 produced by a behaviour policy.

In order to solve a practical convex MDP problem in a given domain it would be prudent to use an RL algorithm that is known to be high performing for the vanilla RL problem as the policy player. From the theoretical point of view this could be MDPO or UCRL2, which we have shown come with strong guarantees. From the practical point of view using a high performing DRL algorithm, which may be specific to the domain, will usually yield the best results. For the cost player using FTL, *i.e.*, using the gradient of the objective function, is typically the best choice.

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 authors 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] L. M. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR computational mathematics and mathematical physics*, 7(3):200–217, 1967.

[15] 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.

[16] 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.

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

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

[19] 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.

[20] 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.

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

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

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

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

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

[30] 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.

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

[32] 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.

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

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

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

[36] Y. Jin and A. Sidford. Efficiently solving mdps with stochastic mirror descent. In *International Conference on Machine Learning*, pages 4890–4900. PMLR, 2020.

[37] Y. Jin and A. Sidford. Towards tight bounds on the sample complexity of average-reward mdps. *arXiv preprint arXiv:2106.07046*, 2021.

[38] 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.

[39] S. Kullback. *Information theory and statistics*. Courier Corporation, 1997.

[40] T. Lattimore and M. Hutter. Pac bounds for discounted mdps. In *International Conference on Algorithmic Learning Theory*, pages 320–334. Springer, 2012.

[41] 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.

[42] D. Levin, Y. Peres, and E. Wilmer. *Markov Chains and Mixing Times*. American Mathematical Society, 2017.

[43] 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.

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

[45] 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.

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

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

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

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

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

[51] 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.

[52] 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.

[53] 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.

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

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

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

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

[58] 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.

[59] 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.

[60] 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.

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

[62] 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.

[63] 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.

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

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

[66] 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.

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

[68] 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.

[69] 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.

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

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

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

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

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

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

[76] T. Zahavy, A. Cohen, H. Kaplan, and Y. Mansour. Average reward reinforcement learning with unknown mixing times. *The Conference on Uncertainty in Artificial Intelligence (UAI)*, 2020.

[77] 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.

[78] 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.

[79] 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.