Proceedings of Machine Learning Research 120:1–11, 2020 2nd Annual Conference on Learning for Dynamics and Control

Faster saddle-point optimization for solving large-scale Markov decision processes

Joan Bas-Serrano JOANBASSERRANO@GMAIL.COM
Universitat Pompeu Fabra, Barcelona, Spain
Gergely Neu GERGELY.NEU@GMAIL.COM
Universitat Pompeu Fabra, Barcelona, Spain

Editors: A. Bayen, A. Jadbabaie, G. J. Pappas, P. Parrilo, B. Recht, C. Tomlin, M.Zeilinger

Abstract

We consider the problem of computing optimal policies in average-reward Markov decision processes. This classical problem can be formulated as a linear program directly amenable to saddle-point optimization methods, albeit with a number of variables that is linear in the number of states. To address this issue, recent work has considered a linearly relaxed version of the resulting saddle-point problem. Our work aims at achieving a better understanding of this relaxed optimization problem by characterizing the conditions necessary for convergence to the optimal policy, and designing an optimization algorithm enjoying fast convergence rates that are independent of the size of the state space.

1. Introduction

Computing optimal policies in Markov decision processes (MDPs) is one of the most important problems in sequential decision making and control (Puterman, 1994). Arguably, the most classical approach to solve this task is through the method of *dynamic programming*, understood in this context as computing fixed points of certain operators (Bellman, 1957; Howard, 1960; Bertsekas, 2007). The use and influence of dynamic-programming methods like value and policy iteration extend well beyond the world of decision and control theory, as the underlying ideas serve as foundations for most algorithms for *learning* optimal policies in unknown MDPs: the setting of *reinforcement learning* (Szepesvári, 2010; Sutton and Barto, 2018). While being hugely successful, DP-based methods have the downside of being somewhat incompatible with classical machine-learning tools that are rooted in convex optimization. Indeed, most of the popular reductions of dynamic programming to (non-)convex optimization are based on heuristics that are not directly motivated by theory. Examples include the celebrated DQN approach of Mnih et al. (2015) that reduces value-function estimation to minimizing the "squared Bellman error", or the TRPO algorithm of Schulman et al. (2015) that reduces policy updates to minimizing a "regularized surrogate objective". While these methods can be justified to a certain extent, it is technically unknown if solving the resulting optimization problems actually leads to a desirable solution to the original sequential decision-making problem.

In this paper, we explore a family of methods that reduce MDP optimization to a form of convex optimization in a theoretically grounded way. Our starting point is an alternative approach based on linear programming (LP), first proposed roughly at the same time as the

DP methods of Bellman (1957); Howard (1960): the idea of LP-based methods for sequential decision-making goes back to the works of de Ghellinck (1960); Manne (1960); Denardo (1970). While LP-based methods seem to be more obscure in present day than DP methods, they have the clear advantage that they lead to an objective function directly amenable to modern large-scale optimization methods. Recent reinforcement-learning methods inspired by the LP perspective include policy-gradient and actor-critic methods (Sutton et al., 1999; Konda and Tsitsiklis, 1999) and various "entropy-regularized" learning algorithms (Peters et al., 2010; Zimin and Neu, 2013; Neu et al., 2017). While these methods promise to directly tackle the policy-optimization problem through solving the underlying linear program, most of them still require the computation of certain value functions through dynamic programming.

In the present work, we argue for the viability of a method fully based on a form of convex optimization, rooted in the LP approach. Our approach is based on a *bilinear saddle-point* formulation of the linear program, building on a well-known equivalence between the two optimization problems. One particular advantage of this formulation is that it enables a straightforward form of dimensionality reduction of the original problem through a linear parametrization of the optimization variables, which provides a natural framework for studying effects of "function approximation" in the underlying policy optimization problem. Our main contribution lies in characterizing a set of assumptions that allow a reduced-order saddle-point representation of the optimal policy. These include a realizability assumption and a newly identified *coherence assumption* about the subspaces used for approximation. Our main positive result is showing that these conditions are sufficient for constructing an algorithm that outputs an ε-optimal policy with runtime guarantees of $\widetilde{\mathcal{O}}\left(\tau_{\text{mix}}^2 N^3/\varepsilon\right)$, where N is the number of variables in the relaxed optimization problem, and τ_{mix} is a notion of mixing time. Our approach is based on the celebrated Mirror Prox algorithm of Nemirovski (2004) (see also Korpelevich, 1976). We complement our positive results by showing that our newly defined coherence assumption is necessary for the relaxed saddle-point approach to be viable: we construct a simple example violating the assumption, where achieving full optimality on the relaxed problem leads to a suboptimal policy.

We are not the first to consider saddle-point methods for optimization in Markov decision processes. Wang (2017) proposed variants of Mirror Descent to solve the original saddle-point problem without relaxations and provide runtime guarantees of $\widetilde{\mathcal{O}}\big((\alpha\tau_{\text{mix}})^2\,|\mathcal{X}||\mathcal{A}|/\varepsilon^2\big)$, where $\mathcal{X}$ and $\mathcal{A}$ are the finite state and action spaces, and α is a parameter that characterizes the uniformity of the stationary distributions of every policy. Specifically, their assumption implies[1] that for the stationary distribution d_π any policy π, one has $\frac{\max_x d_\pi(x)}{\min_{x'} d_\pi(x')} \le \alpha$. In most cases of practical interest, this ratio is at least as large as $|\mathcal{X}|$, and can easily be exponentially large in $|\mathcal{X}|$. When specialized to this setting, our bounds replace α^2 by the much more manageable $|\mathcal{X}|$ and also improve the dependence on ε from $1/\varepsilon^2$ to $1/\varepsilon$. One downside of our method is that we need full access to the transition probabilities of the MDP, whereas the algorithm of Wang (2017) only requires a generative model.

The linearly relaxed saddle-point problem we consider was first studied by Lakshminarayanan and Bhatnagar (2015); Lakshminarayanan et al. (2018) and Chen et al. (2018). Our runtime guarantees improve over the ones claimed by Chen et al. (2018) in a similar way as our first set of results improve over those of Wang (2017). Notably, their results still

1. The actual assumption made by Wang (2017) is even more restrictive.

feature a factor of α^2, which generally depends on the size of the original state space rather than the number of features, rendering these guarantees void of meaning in very large state spaces. In contrast, our bounds replace this factor by the number of features N. Furthermore, our characterization highlighting the importance of the coherence assumption discussed above hints at some potential technical issues with the results of Chen, Li, and Wang (2018), who claimed convergence to the optimal policy *without the coherence assumption*.

The rest of the paper is organized as follows. After providing background on MDP optimization in Section 2, we describe the linearly relaxed saddle-point problem in Section 3. We provide our algorithm and state its performance guarantees in Section 4, and conclude with Section 5. Due to space constraints, we relegate all proofs and numerical results to the full version of the paper (Bas-Serrano and Neu, 2019).

Notation. Inner products over vector spaces will be denoted by $\langle \cdot, \cdot \rangle$. We use $\Delta_{\mathcal{S}}$ to denote the set of probability distributions on the finite set $\mathcal{S}$: $\Delta_{\mathcal{S}} = \left\{ p \in \mathbb{R}_+^{\mathcal{S}} : \sum_{s \in \mathcal{S}} p(s) = 1 \right\}$.

2. Preliminaries

Consider an undiscounted Markov decision process $M = (\mathcal{X}, \mathcal{A}, P, r)$, where $\mathcal{X}$ is the finite state space, $\mathcal{A}$ is the finite action space, P is the transition function with $P(x'|x,a)$ denoting the probability of moving to state $x' \in \mathcal{X}$ from state $x \in \mathcal{X}$ when taking action $a \in \mathcal{A}$ and r is the reward function mapping state-action pairs to rewards with $r(x,a)$ denoting the reward of being in state x and taking action a. We assume that $r(x,a) \in [0,1]$ for all x, a. In each round t, the learner observes state $x_t \in \mathcal{X}$, selects action $a_t \in \mathcal{A}$, moves to the next state $x_{t+1} \sim P(\cdot|x_t, a_t)$, and obtains reward $r(x_t, a_t)$.

In this paper we focus on the infinite-horizon average-reward scenario where the goal of the learner is to select its actions a_t in a way that maximizes the average reward per time step, $\liminf_{t\to\infty} \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} r_t(x_t, a_t)\right]$. We will work with randomized stationary policies with $\pi(a|x)$ denoting the probability of taking action a in state x. Under technical assumptions discussed shortly, each such policy π generates a unique stationary state distribution $d_\pi \in \Delta_{\mathcal{X}}$ over the state space satisfying $d_\pi(x) = \lim_{t\to\infty} \mathbb{P}\left[x_t = x\right]$ for all x when the trajectory $(x_t)_t$ is generated by following policy π. Similarly, each policy π generates a stationary state-action distribution $\mu_\pi \in \Delta_{\mathcal{X}\times\mathcal{A}}$ satisfying $\mu_\pi(x,a) = \lim_{t\to\infty} \mathbb{P}\left[x_t = x, a_t = a\right] = d_\pi(x)\pi(a|x)$. Given these definitions, the average reward of a policy π can be written as

$$\rho_\pi = \liminf_{t\to\infty} \mathbb{E}_\pi \left[\frac{1}{T}\sum_{t=1}^{T} r_t(x_t, a_t)\right] = \sum_{x,a} \mu(x,a) r(x,a),$$

where the notation $\mathbb{E}_\pi[\cdot]$ indicates that the trajectory $(x_t, a_t)_t$ was generated by following policy π: $a_t \sim \pi(\cdot|x_t)$ and $x_{t+1} \sim P(\cdot|x_t, a_t)$. Under our assumptions, the optimal policy can be shown to be a stationary one; we will denote its average reward as $\rho^* = \max_\pi \rho_\pi$. Thus, one can show that finding the optimal policy is equivalent to solving the following linear program:

$$\begin{aligned} \text{maximize} \quad & \sum_{x,a} \mu(x,a) r(x,a) \\ \text{s.t.} \quad & \mu \in \Delta_{\mathcal{X}\times\mathcal{A}}, \qquad \sum_{a'} \mu(x',a') = \sum_{x,a} P(x'|x,a)\mu(x,a) \quad (\forall x' \in \mathcal{X}). \end{aligned}$$

To simplify our notation, we will represent μ and r by $|\mathcal{X}\times\mathcal{A}|$-dimensional vectors and also define the $|\mathcal{X}\times\mathcal{A}|\times|\mathcal{X}|$-dimensional matrix Q with entries $Q_{(x,a),x'} = P(x'|x,a) - \mathbb{I}_{\{x'=x\}}$. Then, one can easily see[2] that solving the linear program stated above is equivalent to finding the following *saddle point*:

$$\min_{v\in\mathbb{R}^{|\mathcal{X}|}} \max_{\mu\in\Delta} \mathcal{L}(v,\mu) = \min_{v\in\mathbb{R}^{|\mathcal{X}|}} \max_{\mu\in\Delta} \langle \mu, Qv\rangle + \langle \mu, r\rangle \,. \tag{1}$$

Here, we introduced the *Lagrangian function* $\mathcal{L}$ and the shorthand $\Delta = \Delta_{\mathcal{X}\times\mathcal{A}}$. Optimal solutions (v^*,μ^*) to the above saddle-point problem are easily seen to correspond to the stationary distribution μ^* of the optimal policy and the *optimal differential value function* v^* (also known as the optimal bias function, cf. Puterman, 1994). Besides the full saddle-point optimization problem, we will consider a relaxed version based on the introduction feature maps. Details on this variant are provided in Section 3.

We will make two structural assumptions about the underlying Markov decision process. The first of these guarantees the existence of stationary distributions for all policies.

Assumption 1 (Uniform ergodicity) *Every policy π generates an ergodic Markov chain. Specifically, letting P_π be the transition operator of π defined as the matrix with elements $P_\pi(x'|x) = \sum_a \pi(a|x)P(x'|x,a)$, and d, d' be any two distributions over $\mathcal{X}$, the following inequality is satisfied for some $C,\tau > 0$ and for all k:*

$$\left\| \left(d - d'\right) P_\pi^k \right\|_1 \le C e^{-k/\tau} \left\| d - d' \right\|_1 .$$

We say that our MDP is *uniformly ergodic* if it satisfies Assumption 1. Notice that this assumption is significantly weaker than the 1-step mixing assumption often made in the related literature (Even-Dar et al., 2009; Neu et al., 2014), and can be easily seen to hold when all policies induce aperiodic and irreducible Markov chains (cf. Theorem 4.9 in Levin et al., 2017). Clearly, this assumption implies that every policy admits a unique stationary distribution as required in the discussion above. In what follows below, we will often use the notation $\tau_{\text{mix}} = 2C\,(\tau+1)$ and refer to this quantity as the *mixing time* of the MDP[3].

Given this assumption and the above definitions, we can establish a number of useful facts about the optimal solutions (v^*,μ^*) to the saddle-point problem (1). We first note that an optimal policy π^* can be extracted from μ^* in the states where $\mu^*(x,\cdot) > 0$ as $\pi^*(a|x) = \frac{\mu^*(x,a)}{\sum_{a'}\mu^*(x,a')}$. Regarding v^*, the following proposition summarizes some of its most important properties:

Proposition 1 *Let (v^*,μ^*) be a solution of the problem (1). Then, v^* satisfies the following properties:*

- *v^* satisfies the Bellman optimality equations $v^*(x) = r(x) - \rho^* + \sum_{x'} P(x'|x,a)v^*(x')$ for all x; for any $c\in\mathbb{R}$, $v^* + c$ is also a solution to (1);*
- *for any x, x', $|v^*(x) - v^*(x')| \le \tau_{mix} = 2C\,(\tau+1)$.*

All of these properties can be proven by standard arguments; we refer the reader to Lemma 1 in Wang (2017) for a proof of the first item and Lemma 3 in Neu et al. (2014) for a proof of the second one.

2. This can be seen, e.g., by introducing the KKT multipliers for the constraints in the linear program.
3. Note that this is just one of many possible definitions of a mixing time, see, e.g., Seneta (2006); Levin et al. (2017).

3. The linearly relaxed saddle-point problem

While one can directly derive optimization algorithms to solve the saddle-point problem (1), such a direct approach would suffer from serious scalability issues due to the sheer number of variables involved in the problem: the size of the objects of interest μ and v are linear in the size of the state space, which results in prohibitive memory and computation costs for most algorithms. To address this issue, we study a *linearly relaxed* version of the full saddle-point problem that reduces the order of the original optimization problem by linearly parametrizing the variables v and μ through two sets of *feature maps*. Formally, we consider the matrices F of size $|\mathcal{X}| \times N$ and W of size $M \times |\mathcal{X} \times \mathcal{A}|$, introduce the new optimization variables $y \in \mathbb{R}^M$ and $u \in \mathbb{R}^N$, and use these to (hopefully) approximate the solutions to (1) as $\mu^* \approx yW$ and $v^* \approx Fu$. For a tractable problem formulation, we will assume that the rows of W are non-negative and sum to one: $W_{m,x} \geq 0$ for all x, m and $\sum_x W_{m,x} = 1$ for all m. We will also assume that all entries of F are bounded by 1 in absolute value. These conditions enable us to optimize y over the probability simplex $\widetilde{\Delta} = \Delta_{[M]}$ and to formulate our relaxed saddle-point problem as

$$\min_{u \in \mathbb{R}^N} \max_{y \in \widetilde{\Delta}} \widetilde{\mathcal{L}}(u, y) = \min_{u \in \mathbb{R}^N} \max_{y \in \widetilde{\Delta}} \langle W^\intercal y, QFu \rangle + \langle W^\intercal y, r \rangle \,. \tag{2}$$

The relaxed optimization problem above has been studied before by Lakshminarayanan and Bhatnagar (2015); Lakshminarayanan et al. (2018), and Chen et al. (2018). Lakshminarayanan and Bhatnagar (2015); Lakshminarayanan et al. (2018) studied the relaxed linear program underlying (2) as a natural extension of the classic relaxed LP analyzed by de Farias and Van Roy (2003), and have focused on understanding the discrepancies between the optimal value function and the relaxed value function attaining the minimum in the above expression. On the other hand, Chen et al. (2018) focused on proposing stochastic optimization algorithms and analyzing the rate of convergence to the optimum, but provide little insight about the quality of the optimal solution of the relaxed problem.

One of our main goals in the present paper is to obtain a better understanding of the effects of approximation on the policies that can be obtained through approximately solving the the relaxed saddle-point problem (2). One peculiar challenge associated with our setting is that it is not enough to ensure that the values of $\widetilde{\mathcal{L}}$ and $\mathcal{L}$ are close at their respective saddle points, but we rather need to understand the performance of the policy extracted from the optimal solution y^*. Precisely, defining the policy extracted from y as

$$\pi_y(a|x) = \frac{(W^\intercal y)(x, a)}{\sum_{a'} (W^\intercal y)(x, a')}$$

for all x, a, and the corresponding stationary distribution as μ_y induced in the original MDP, we are interested in the suboptimality gap $\langle \mu^* - \mu_{y^*}, r \rangle$. In the present paper, we focus on identifying assumptions on the feature maps that allow the computation of true optimal policies with (almost) zero suboptimality gap. Specifically, we will show that the following two assumptions have a decisive role in making this gap small:

Assumption 2 (Realizability) *The optimal solution is realizable by the feature maps: there exists (u^*, y^*) such that $v^* = Fu^*$ and $\mu^* = W^\intercal y^*$. Additionally, $\|u^*\|_\infty \leq U\tau_{mix}$ holds for some $U > 0$.*

Assumption 3 (Coherence) *The image of the set $\widetilde{\Delta}$ under the map $Q^\intercal W^\intercal$ is included the column space of F: for all $y \in \widetilde{\Delta}$ such that $Q^\intercal W^\intercal y \neq 0$, there exists a $u \in \mathbb{R}^N$ such that $\langle Q^\intercal W^\intercal y, Fu\rangle \neq 0$. Additionally, for all $v \in \mathbb{R}^{|\mathcal{X}|}$ with $\|v\|_\infty \leq 1$, there exists a $u \in \mathbb{R}^N$ with $\|u\|_\infty \leq U$ such that $\langle Q^\intercal W^\intercal y, Fu\rangle = \langle Q^\intercal W^\intercal y, v\rangle$.*

The second condition of each assumption is to ensure that the columns of F are well-conditioned and are satisfied if the columns form an orthonormal basis. While the realizability may already seem sufficient for the relaxed problem to be a good enough approximation of the original one, we argue that the second assumption is also necessary for the relaxation scheme to be reliable. Specifically, the following theorem shows that in the absence of the coherence assumption, near-optimal solutions to the relaxed saddle-point problem (2) can still lead to suboptimal policies in the original MDP.

Theorem 1 *For any $\varepsilon > 0$, there exists an MDP with relaxations W, F satisfying Assumption 2 and violating Assumption 3, and a solution $(\widehat{u}, \widehat{y}_\varepsilon)$ simultaneously satisfying*

$$\mathcal{L}(F\widehat{u}, \mu^*) - \mathcal{L}(v^*, W^\intercal \widehat{y}_\varepsilon) = \varepsilon$$

and

$$\left\langle \mu^* - \mu_{\widehat{y}_\varepsilon}, r \right\rangle = 2/3.$$

The proof is provided in appendix of the full version of the paper. The key idea behind the proof is building an MDP with three states and choosing W and F so that they guarantee realizability but for which there exist a y such that $\langle W^\intercal y, r\rangle = \rho^*$ and $\langle W^\intercal y, QFu\rangle = 0$ for all u, despite $W^\intercal y$ being non-stationary.

4. Algorithm and main results

In this section, we provide our main positive results: deriving strong performance guarantees for policies derived from approximate solutions of (2) under Assumptions 2 and 3. Our algorithm attaining these guarantees is based on the Optimistic Mirror Descent framework proposed by Rakhlin and Sridharan (2013a,b), and more specifically on its variant known as Mirror Prox due to Nemirovski (2004) (see also Sections 4.5 and 5.2.3 in Bubeck (2015) for an easily accessible overview of this method).

Our algorithm will compute a sequence of value functions and state-action distributions by starting from $v_0 = 0$ and μ_0 chosen as the uniform distribution over $\mathcal{X} \times \mathcal{A}$, and consecutively computing each update as

$$\widehat{u}_{t+1} = u_t - \eta F^\intercal Q^\intercal W^\intercal y_t, \quad \widehat{y}_{t+1,i} \propto y_{t,i} e^{\eta\left((Wr)_i + (WQFu_t)_i\right)} \tag{3}$$

$$u_{t+1} = u_t - \eta F^\intercal Q^\intercal W^\intercal \widetilde{y}_{t+1}, \quad y_{t+1,i} \propto y_{t,i} e^{\eta\left((Wr)_i + (WQF\widehat{u}_{t+1})_i\right)}, \tag{4}$$

for all $t > 0$, where we used the notation "$\propto$" to signify that $\widehat{y}_{t+1}$ and y_{t+1} are normalized multiplicatively after each update so that $\sum_j y_{t+1,j} = 1$ is satisfied. Also introducing the notations $\overline{y}_T = \frac{1}{T}\sum_{t=1}^T y_t$ and $\overline{u}_T = \frac{1}{T}\sum_{t=1}^T \widehat{u}_t$, the algorithm outputs the policy extracted from the distribution $\overline{y}_T$: $\pi_T = \pi_{\overline{y}_T}$. Letting $d_T = d_{\pi_T}$ be the stationary distribution associated with π_T, the corresponding average reward can be written as $\rho_T = \sum_{x,a} d_T(x)\pi_T(a|x)r(x,a)$. The following theorem presents our main result regarding the suboptimality of the resulting policy in terms of its average reward.

Theorem 2 *Suppose that Assumptions 1, 2 and 3 hold and $\eta \leq 1/4N$. Then, the average reward ρ_T output by the algorithm satisfies*

$$\rho^* - \rho_T \leq \frac{11\tau_{mix}^2 U^2 N + 7\log M}{\eta T}.$$

In particular, setting $\eta = 1/4N$, the bound becomes $\rho^* - \rho_T = \mathcal{O}\left(\frac{\tau_{mix}^2 N^2 U^2}{T}\right)$.

On a high level, the proof of this theorem builds on some well-known results regarding the performance of Mirror Prox. One key component is using the classical bound on the duality gap that can be written as $\left\langle \mu^* - W^\top \bar{y}_T, r \right\rangle \leq \frac{D_\Phi(z^*, z_0)}{\eta T} + \left\langle Q^\top W^\top \bar{y}_T, v^* \right\rangle$. The remaining challenge is then to connect the quantity on the left-hand side to the suboptimality gap of the extracted policy π_T. This is achieved by a novel technique that asserts the relationship $\langle W^\top \bar{y}_T, r\rangle - \rho_T \leq \tau_{\text{mix}} \left\| Q^\top W^\top \bar{y}_T \right\|_1$. The rest of the work is in bounding the remaining terms by exploiting further properties of Mirror Prox. The detailed proof can be found in appendix of the full paper.

In the special case where F and W are the identity maps, the relaxed saddle-point problem becomes the original problem (1), and our Assumptions 2 and 3 are clearly satisfied with $U = 1$. In this case, our algorithm satisfies the following bound:

Corollary 3 *Suppose that Assumption 1 holds, W and F are the identity maps, and $\eta \leq 1/4$. Then, the average reward ρ_T of the policy output by our algorithm satisfies*

$$\rho^* - \rho_T \leq \frac{11\tau_{mix}^2 |\mathcal{X}| + 7\log\left(|\mathcal{X}||\mathcal{A}|\right)}{\eta T}.$$

In particular, setting $\eta = 1/4$, the bound becomes $\rho^* - \rho_T = \widetilde{\mathcal{O}}\left(\frac{\tau_{mix}^2 |\mathcal{X}|}{T}\right)$.

A brief inspection of Equations (3)-(4) suggests that each update of our algorithm can be computed in $\mathcal{O}(MN)$ time, the most expensive operation being computing the matrix-vector products $WQFu$ and $y^\intercal WQF$. While this suggests that the algorithm may have runtime and memory complexity independent of the size of the state space, we note that exact computation of the matrix WQF can still take $\mathcal{O}\left(|\mathcal{X}|^2|\mathcal{A}|\right)$ time in the worst case. This can be improved to $\mathcal{O}(K)$ when assuming that only K entries of the transition matrix P are nonzero, which can be of order $|\mathcal{X}||\mathcal{A}|$ in many interesting problems where the support of $P(\cdot|x,a)$ is of size $\mathcal{O}(1)$ for all x, a. We stress however that the matrix WQF only needs to be computed *once* as an initialization step of our algorithm. In contrast, a general algorithm like value iteration needs at least $\Theta(K) = \Theta\left(|\mathcal{X}||\mathcal{A}|\right)$ time for computing *each update*, showing a clear computational advantage of our method.

5. Discussion

Our most important contributions concern the relaxed saddle-point problem (2), most notably including our discussion on the necessity and sufficience of the coherence assumption (Assumption 3). As we've mentioned earlier, several relaxation schemes similar to ours have been studied in the literature. In fact, relaxing the linear program underlying (1) through

the introduction of the feature map F for approximating the value function v^* is one of the oldest ideas in approximate dynamic programming, originally introduced by Schweitzer and Seidman (1985). The effects of this approximation were studied by de Farias and Van Roy (2003) in the context of discounted Markov decision processes. A relaxation scheme involving both the feature maps F and W was considered by Lakshminarayanan and Bhatnagar (2015); Lakshminarayanan et al. (2018). Both sets of authors carefully observed that introducing relaxations may make the linear program unbounded, and proposed algorithmic steps and structural assumptions of F and W to fight this issue. The results of these works are incomparable to ours since they focus on controlling the errors in approximating the optimal value function v^* rather than controlling the suboptimality of the policies output by the algorithm. Interestingly, the widely popular REPS algorithm of Peters et al. (2010) is also originally derived from the relaxed linear program analyzed by de Farias and Van Roy (2003), even if this connection has not been pointed out by the authors.

The work of Chen et al. (2018) is very close to ours in spirit. Chen et al. consider a variation of the relaxed saddle-point problem (2) with W being block-diagonal with $F^\intercal$ in each of its blocks, and claim convergence results for their algorithm to the optimal policy under only a realizability assumption. Unfortunately, their choice of W does not necessarily ensure that the coherence assumption holds, which raises concerns regarding the generality of their guarantees. Indeed, the results of Chen et al. require an additional assumption that implies that $\frac{\max_x d_\pi(x)}{\min_{x'} d_\pi(x')}$ remains bounded by a constant for any policy π, which is extremely difficult to ensure in problems of practical interest. In fact, this ratio is already exponentially large in $|\mathcal{X}|$ in very simple problems like the one we consider in our experiments. Additionally, the analysis of Chen et al. is based on the potentially erroneous claim that under the realizability assumption, the representation (u^*, y^*) of the original optimal solution $(v^*, \mu^*) = (Fu^*, W^\intercal y^*)$ always remains an optimal solution to the relaxed saddle-point problem. It is currently unclear if this claim is indeed true, or to what extent their condition regarding the boundedness of stationary distribution can be relaxed.

In any case, we believe that our coherence assumption is more fundamental than the previously considered conditions, and it enables a much more transparent analysis of optimization algorithms addressing the relaxed saddle-point problem (2). Beyond this particular positive result, our work also cleans the slate for further theoretical work on approximate optimization in Markov decision processes. Indeed, the form of our coherence assumption naturally leads to the question: can we compute good approximate solutions to the original problem when our assumptions are only satisified approximately? Similar questions are not without precedent in the reinforcement-learning literature. Translated to our notation, classical results concerning the performance of (least-squares) temporal difference learning algorithms imply that the approximation errors are controlled by the projection error of $QFu^* + r$ to the column space of F (Tsitsiklis and Van Roy, 1997; Bradtke and Barto, 1996; Lazaric et al., 2010). When using more general function classes to approximate v^*, Munos and Szepesvári (2008) show that the approximation errors are controlled by the *inherent Bellman error* of the function class, which captures an approximation property related to our coherence condition. Whether or not we can generalize our techniques to construct provably efficient algorithms under such milder assumptions remains an exciting open problem that we leave open for future research.

References

Joan Bas-Serrano and Gergely Neu. Faster saddle-point optimization for solving large-scale Markov decision processes. *arXiv preprint arXiv:1909.10904*, 2019.

R. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, New Jersey, 1957.

D. P. Bertsekas. *Dynamic Programming and Optimal Control*, volume 2. Athena Scientific, Belmont, MA, 3 edition, 2007.

S. J. Bradtke and A. G. Barto. Linear least-squares algorithms for temporal difference learning. *Machine Learning*, 22:33–57, 1996.

Sébastien Bubeck. Convex optimization: Algorithms and complexity. *Foundations and Trends in Machine Learning*, 8(3-4):231–357, 2015.

Yichen Chen, Lihong Li, and Mengdi Wang. Scalable bilinear π learning using state and action features. In *International Conference on Machine Learning*, pages 833–842, 2018.

D. P. de Farias and B. Van Roy. The linear programming approach to approximate dynamic programming. *Operations Research*, 51(6):850–865, 2003.

Guy de Ghellinck. Les problèmes de décisions séquentielles. *Cahiers du Centre d'Études de Recherche Opérationnelle*, 2:161–179, 1960.

Eric V Denardo. On linear programming in a markov decision problem. *Management Science*, 16(5):281–288, 1970.

E. Even-Dar, S. M. Kakade, and Y. Mansour. Online Markov decision processes. *Mathematics of Operations Research*, 34(3):726–736, 2009.

R. A. Howard. *Dynamic Programming and Markov Processes*. The MIT Press, Cambridge, MA, 1960.

V. R. Konda and J. N. Tsitsiklis. Actor-critic algorithms. In Solla et al. (1999), pages 1008–1014.

GM Korpelevich. The extragradient method for finding saddle points and other problems. *Matecon*, 12:747–756, 1976.

Chandrashekar Lakshminarayanan and Shalabh Bhatnagar. A generalized reduced linear program for Markov decision processes. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pages 2722–2728. AAAI Press, 2015.

Chandrashekar Lakshminarayanan, Shalabh Bhatnagar, and Csaba Szepesvári. A linearly relaxed approximate linear program for Markov decision processes. *IEEE Transactions on Automatic control*, 2018.

Alessandro Lazaric, Mohammad Ghavamzadeh, and Rémi Munos. Finite-sample analysis of LSTD. In J. Fürnkranz and T. Joachims, editors, *Proceedings of the 27th International Conference on Machine Learning (ICML 2010)*, pages 615–622. Omnipress, 2010.

David A Levin, Yuval Peres, and Elizabeth L Wilmer. Markov chains and mixing times. 2nd edition. 2017.

Alan S Manne. Linear programming and sequential decisions. *Management Science*, 6(3): 259–267, 1960.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, and Georg Ostrovski. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

Rémi Munos and Csaba Szepesvári. Finite-time bounds for fitted value iteration. *Journal of Machine Learning Research*, 9(May):815–857, 2008.

Arkadi Nemirovski. Prox-method with rate of convergence $O(1/t)$ for variational inequalities with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 15(1):229–251, 2004.

Gergely Neu, András György, Csaba Szepesvári, and András Antos. Online Markov decision processes under bandit feedback. *IEEE Transactions on Automatic Control*, 59:676–691, 2014.

Gergely Neu, Anders Jonsson, and Vicenç Gómez. A unified view of entropy-regularized markov decision processes. *arXiv preprint arXiv:1705.07798*, 2017.

Jan Peters, Katharina Mülling, and Yasemin Altun. Relative entropy policy search. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI-10)*, pages 1607–1612, Menlo Park, CA, USA, 2010. AAAI Press.

Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley-Interscience, April 1994.

Alexander Rakhlin and Karthik Sridharan. Online learning with predictable sequences. In *Conference on Learning Theory*, pages 993–1019, 2013a.

Alexander Rakhlin and Karthik Sridharan. Optimization, learning, and games with predictable sequences. In *Advances in Neural Information Processing Systems*, pages 3066–3074, 2013b.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 1889–1897, 2015.

P.J. Schweitzer and A. Seidman. Generalized polynomial approximations in Markovian decision processes. *J. of Math. Anal. and Appl.*, 110:568–582, 1985.

Eugene Seneta. *Non-negative matrices and Markov chains*. Springer Science & Business Media, 2006.

S.A. Solla, T.K. Leen, and K.R. Müller, editors. *Advances in Neural Information Processing Systems 12*, Cambridge, MA, USA, 1999. MIT Press.

R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In Solla et al. (1999), pages 1057–1063.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction. 2nd edition.* 2018.

Cs. Szepesvári. *Algorithms for Reinforcement Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2010.

J. N. Tsitsiklis and B. Van Roy. An analysis of temporal difference learning with function approximation. *IEEE Transactions on Automatic Control*, 42:674–690, 1997.

Mengdi Wang. Primal-dual π learning: Sample complexity and sublinear run time for ergodic Markov decision problems. *arXiv preprint arXiv:1710.06100*, 2017.

A. Zimin and G. Neu. Online learning in episodic Markovian decision processes by relative entropy policy search. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 1583–1591, 2013.