Linear Programming for Large-Scale Markov Decision Problems

Yasin Abbasi-Yadkori YASIN.ABBASIYADKORI@QUT.EDU.AU
Queensland University of Technology, Brisbane, QLD, Australia 4000

Peter L. Bartlett BARTLETT@EECS.BERKELEY.EDU
University of California, Berkeley, CA 94720
and Queensland University of Technology, Brisbane, QLD, Australia 4000

Alan Malek MALEK@EECS.BERKELEY.EDU
University of California, Berkeley, CA 94720

Abstract

We consider the problem of controlling a Markov decision process (MDP) with a large state space, so as to minimize average cost. Since it is intractable to compete with the optimal policy for large scale problems, we pursue the more modest goal of competing with a low-dimensional family of policies. We use the dual linear programming formulation of the MDP average cost problem, in which the variable is a stationary distribution over state-action pairs, and we consider a neighborhood of a low-dimensional subset of the set of stationary distributions (defined in terms of state-action features) as the comparison class. We propose a technique based on stochastic convex optimization and give bounds that show that the performance of our algorithm approaches the best achievable by any policy in the comparison class. Most importantly, this result depends on the size of the comparison class, but not on the size of the state space. Preliminary experiments show the effectiveness of the proposed algorithm in a queuing application.

1. Introduction

We study the average loss Markov decision process problem. The problem is well-understood when the state and action spaces are small (Bertsekas, 2007). Dynamic programming (DP) algorithms, such as value iteration (Bellman, 1957) and policy iteration (Howard, 1960), are standard techniques to compute the optimal policy. In large state space problems, exact DP is not feasible as the computational complexity scales quadratically with the number of states.

A popular approach to large-scale problems is to restrict the search to the linear span of a small number of features. The objective is to compete with the best solution within this comparison class. Two popular methods are Approximate Dynamic Programming (ADP) and Approximate Linear Programming (ALP). This paper focuses on ALP. For a survey on theoretical results for ADP see (Bertsekas and Tsitsiklis, 1996; Sutton and Barto, 1998), (Bertsekas, 2007, Vol. 2, Chapter 6), and more recent papers (Sutton et al., 2009b;a; Maei et al., 2009; 2010).

Our aim is to develop methods that find policies with performance guaranteed to be close to the best in the comparison class but with computational complexity that does not grow with the size of the state space. All prior work on ALP either scales badly or requires access to samples from a distribution that depends on the optimal policy.

This paper proposes a new algorithm to solve the Approximate Linear Programming problem that is computationally efficient and does not require knowledge of the optimal policy. In particular, we introduce new proof techniques and tools for average cost MDP problems and use these techniques to derive a reduction to stochastic convex optimization with accompanying error bounds.

1.1. Notation

Let X and A be positive integers. Let $\mathcal{X} = \{1, 2, \ldots, X\}$ and $\mathcal{A} = \{1, 2, \ldots, A\}$ be state and action spaces, respectively. Let Δ_S denote probability distributions over set S. A policy π is a map from the state space to $\Delta_{\mathcal{A}}$: $\pi : \mathcal{X} \to \Delta_{\mathcal{A}}$. We use $\pi(a|x)$ to denote the probability of choosing action a in state x under policy π. A transition probability

Proceedings of the 31^{st} International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32.

kernel (or transition kernel) $P : \mathcal{X} \times \mathcal{A} \to \Delta_{\mathcal{X}}$ maps from the direct product of the state and action spaces to $\Delta_{\mathcal{X}}$. Let P^π denote the probability transition kernel under policy π. A loss function is a bounded real-valued function over state and action spaces, $\ell : \mathcal{X} \times \mathcal{A} \to [0, 1]$. Let $M_{i,:}$ and $M_{:,j}$ denote ith row and jth column of matrix M respectively. Let $\|v\|_{1,c} = \sum_i c_i |v_i|$ and $\|v\|_{\infty,c} = \max_i c_i |v_i|$ for a positive vector c. We use $\mathbf{1}$ and $\mathbf{0}$ to denote vectors with all elements equal to one and zero, respectively. We use $\wedge$ and $\vee$ to denote the minimum and the maximum, respectively. For vectors v and w, $v \leq w$ means element-wise inequality, i.e. $v_i \leq w_i$ for all i.

1.2. Linear Programming Approach to Markov Decision Problems

Under certain assumptions, there exist a scalar λ_* and a vector $h_* \in \mathbb{R}^X$ that satisfy the Bellman optimality equations for average loss problems,

$$\lambda_* + h_*(x) = \min_{a \in \mathcal{A}} \left[\ell(x, a) + \sum_{x' \in \mathcal{X}} P_{(x,a),x'} h_*(x') \right] .$$

The scalar λ_* is called the optimal average loss, while the vector h_* is called a differential value function. The action that minimizes the right-hand side of the above equation is the optimal action in state x and is denoted by $a_*(x)$. The optimal policy is defined by $\pi_*(a_*(x)|x) = 1$. Given ℓ and P, the objective of the *planner* is to compute the optimal action in all states, or equivalently, to find the optimal policy.

The MDP problem can also be stated in the LP formulation (Manne, 1960),

$$\begin{aligned} &\max_{\lambda, h} \lambda \,, \\ &\text{s.t.} \quad B(\lambda \mathbf{1} + h) \leq \ell + Ph \,, \end{aligned} \tag{1}$$

where $B \in \{0, 1\}^{XA \times X}$ is a binary matrix such that the ith column has A ones in rows $1+(i-1)A$ to iA. Let v_π be the stationary distribution under policy π and let $\mu_\pi(x, a) = v_\pi(x)\pi(a|x)$. We can write

$$\begin{aligned} \pi_* &= \operatorname*{argmin}_\pi \sum_{x \in \mathcal{X}} v_\pi(x) \sum_{a \in \mathcal{A}} \pi(a|x) \ell(x, a) \\ &= \operatorname*{argmin}_\pi \sum_{(x,a) \in \mathcal{X} \times \mathcal{A}} \mu_\pi(x, a) \ell(x, a) \\ &= \operatorname*{argmin}_\pi \mu_\pi^\top \ell \,. \end{aligned}$$

In fact, the dual of LP (1) has the form of

$$\begin{aligned} &\min_{\mu \in \mathbb{R}^{XA}} \mu^\top \ell \,, \\ &\text{s.t.} \quad \mu^\top \mathbf{1} = 1, \, \mu \geq \mathbf{0}, \, \mu^\top (P - B) = \mathbf{0} \,. \end{aligned} \tag{2}$$

The objective function, $\mu^\top \ell$, is the average loss under stationary distribution μ. The first two constraints ensure that μ is a probability distribution over state-action space, while the last constraint ensures that μ is a stationary distribution. Given a solution μ, we can obtain a policy via $\pi(a|x) = \mu(x, a) / \sum_{a' \in \mathcal{A}} \mu(x, a')$.

1.3. Approximations for Large State Spaces

The LP formulations (1) and (2) are not practical for large scale problems as the number of variables and constraints grows linearly with the number of states. Schweitzer and Seidmann (1985) propose approximate linear programming (ALP) formulations. These methods were later improved by de Farias and Van Roy (2003a;b); Hauskrecht and Kveton (2003); Guestrin et al. (2004); Petrik and Zilberstein (2009); Desai et al. (2012). As noted by Desai et al. (2012), the prior work on ALP either requires access to samples from a distribution that depends on the optimal policy or assumes the ability to solve an LP with as many constraints as states. (See Appendix A for a more detailed discussion.) Our objective is to design algorithms for very large MDPs that do not require knowledge of the optimal policy.

In contrast to the aforementioned methods, which solve the primal ALPs (with value functions as variables), we work with the dual form (2) (with stationary distributions as variables). Analogous to ALPs, we control the complexity by limiting our search to a linear subspace defined by a small number of *features*. Let d be the number of features and Φ be a $(XA) \times d$ matrix with features as column vectors. By adding the constraint $\mu = \Phi\theta$, we get

$$\begin{aligned} &\min_\theta \theta^\top \Phi^\top \ell \,, \\ &\text{s.t.} \quad \theta^\top \Phi^\top \mathbf{1} = 1, \, \Phi\theta \geq \mathbf{0}, \, \theta^\top \Phi^\top (P - B) = \mathbf{0} \,. \end{aligned}$$

If a stationary distribution μ_0 is known, it can be added to the linear span to get the ALP

$$\begin{aligned} &\min_\theta (\mu_0 + \Phi\theta)^\top \ell \,, \\ &\text{s.t.} \quad (\mu_0 + \Phi\theta)^\top \mathbf{1} = 1, \, \mu_0 + \Phi\theta \geq \mathbf{0}, \\ &\qquad (\mu_0 + \Phi\theta)^\top (P - B) = \mathbf{0} \,. \end{aligned} \tag{3}$$

Although $\mu_0 + \Phi\theta$ might not be a stationary distribution, it still defines a policy[1]

$$\pi_\theta(a|x) = \frac{[\mu_0(x, a) + \Phi_{(x,a),:}\theta]_+}{\sum_{a'} [\mu_0(x, a') + \Phi_{(x,a'),:}\theta]_+} \,, \tag{4}$$

We denote the stationary distribution of this policy μ_θ, which is only equal to $\mu_0 + \Phi\theta$ if θ is in the feasible set.

[1] We use the notation $[v]_- = v \wedge 0$ and $[v]_+ = v \vee 0$.

1.4. Problem definition

With the above notation, we can now be explicit about the problem we are solving.

Definition 1 (Efficient Large-Scale Dual ALP)**.** *For an MDP specified by ℓ and P, a feature matrix Φ and a stationary distribution μ_0, the efficient large-scale dual ALP problem is to produce parameters $\widehat{\theta}$ such*

$$\mu_{\widehat{\theta}}^\top \ell \leq \min\left\{\mu_\theta^\top \ell : \theta \text{ feasible for (3)}\right\} + O(\epsilon) \qquad (5)$$

in time polynomial in d and $1/\epsilon$. The model of computation allows access to arbitrary entries of Φ, ℓ, P, μ_0, $P^\top\Phi$, and $\mathbf{1}^\top\Phi$ in unit time.

Importantly, the computational complexity cannot scale with X and we do not assume any knowledge of the optimal policy. In fact, as we shall see, we solve a harder problem, which we define as follows.

Definition 2 (Expanded Efficient Large-Scale Dual ALP)**.** *Let $V : \Re^d \to \Re_+$ be some "violation function" that represents how far $\mu_0 + \Phi\theta$ is from a valid stationary distribution, satisfying $V(\theta) = 0$ if θ is a feasible point for the ALP (3). The expanded efficient large-scale dual ALP problem is to produce parameters $\widehat{\theta}$ such that*

$$\mu_{\widehat{\theta}}^\top \ell \leq \min\left\{\mu_\theta^\top \ell + \frac{1}{\epsilon}V(\theta) : \theta \in \Re^d\right\} + O(\epsilon), \qquad (6)$$

in time polynomial in d and $1/\epsilon$, under the same model of computation as in Definition 1.

Note that the expanded problem is strictly more general as guarantee (6) implies guarantee (5). Also, many feature vectors Φ may not admit any feasible points. In this case, the dual ALP problem is trivial, but the expanded problem is still meaningful.

Having access to arbitrary entries of the quantities in Definition 1 arises naturally in many situations. In many cases, entries of $P^\top\Phi$ are easy to compute. For example, suppose that for any state x' there is a small number of state-action pairs (x, a) such that $P(x'|x, a) > 0$. Consider Tetris; although the number of board configurations is large, each state has a small number of possible neighbors. Dynamics specified by graphical models with small connectivity also satisfy this constraint. Computing entries of $P^\top\Phi$ is also feasible given reasonable features. If a feature φ_i is a stationary distribution, then $P^\top\varphi_i = B^\top\varphi_i$. Otherwise, it is our prerogative to design sparse feature vectors, hence making the multiplication easy. We shall see an example of this setting later.

1.5. Our Contributions

In this paper, we introduce an algorithm that solves the expanded efficient large-scale dual ALP problem under a (standard) assumption that any policy converges quickly to its stationary distribution. Our algorithm take as input a constant S and an error tolerance ϵ, and has access to the various quantities listed in Definition 1. Define $\Theta = \{\theta : \theta^\top\Phi^\top\mathbf{1} = 1 - \mu_0^\top\mathbf{1},\ \|\theta\| \leq S\}$. If no stationary distribution is known, we can simply choose $\mu_0 = \mathbf{0}$. The algorithm is based on stochastic convex optimization. We prove that for any $\delta \in (0, 1)$, after $O(1/\epsilon^4)$ steps of gradient descent, the algorithm finds a vector $\widehat{\theta} \in \Theta$ such that, with probability at least $1 - \delta$,

$$\begin{aligned}\mu_{\widehat{\theta}}^\top \ell \leq& \mu_\theta^\top \ell + O\left(\frac{1}{\epsilon}\left\|[\mu_0 + \Phi\theta]_-\right\|_1\right)\\ &+ O\left(\frac{1}{\epsilon}\left\|(P - B)^\top(\mu_0 + \Phi\theta)\right\|_1\right) + O(\epsilon\log(1/\delta))\end{aligned}$$

holds for all $\theta \in \Theta$; i.e., we solve the expanded problem for $V(\theta)$ bounded by a constant times the L_1 error of the violation. The second and third terms are zero for feasible points (points in the intersection of the feasible set of LP (2) and the span of the features). For points outside the feasible set, these terms measure the extent of constraint violations for the vector $\mu_0 + \Phi\theta$, which indicates how well stationary distributions can be represented by the chosen features.

2. A Reduction to Stochastic Convex Optimization

In this section, we describe our algorithm as a reduction from Markov decision problems to stochastic convex optimization. The main idea is to convert the ALP (3) into an unconstrained optimization over Θ by adding a function of the constraint violations to the objective, then run stochastic gradient descent with unbiased estimated of the gradient.

For a positive constant H, form the following convex cost function by adding a multiple of the total constraint violations to the objective of the LP (3):

$$\begin{aligned}c(\theta) &= \ell^\top(\mu_0 + \Phi\theta) + H\left\|[\mu_0 + \Phi\theta]_-\right\|_1\\ &\quad + H\left\|(P - B)^\top(\mu_0 + \Phi\theta)\right\|_1\\ &= \ell^\top(\mu_0 + \Phi\theta) + H\left\|[\mu_0 + \Phi\theta]_-\right\|_1\\ &\quad + H\left\|(P - B)^\top\Phi\theta\right\|_1\\ &= \ell^\top(\mu_0 + \Phi\theta) + H\sum_{(x,a)}\left|[\mu_0(x, a) + \Phi_{(x,a),:}\theta]_-\right|\\ &\quad + H\sum_{x'}\left|(P - B)_{:,x'}^\top\Phi\theta\right| .\end{aligned} \qquad (7)$$

We justify using this surrogate loss as follows. Suppose we find a near optimal vector $\widehat{\theta}$ such that $c(\widehat{\theta}) \leq \min_{\theta\in\Theta} c(\theta) + O(\epsilon)$. We will prove

1. that $\left\|[\mu_0 + \Phi\widehat{\theta}]_-\right\|_1$ and $\left\|(P-B)^\top(\mu_0 + \Phi\widehat{\theta})\right\|_1$ are small and $\mu_0 + \Phi\widehat{\theta}$ is close to $\mu_{\widehat{\theta}}$ (by Lemma 2 in Section 2.1), and
2. that $\ell^\top(\mu_0 + \Phi\widehat{\theta}) \leq \min_{\theta\in\Theta} c(\theta) + O(\epsilon)$.

As we will show, these two facts imply that with high probability, for any $\theta \in \Theta$,

$$\mu_{\widehat{\theta}}^\top \ell \leq \mu_\theta^\top \ell + O\left(\frac{1}{\epsilon}\left\|[\mu_0 + \Phi\theta]_-\right\|_1\right) + O\left(\frac{1}{\epsilon}\left\|(P-B)^\top(\mu_0+\Phi\theta)\right\|_1\right) + O(\epsilon)\,,$$

which is to say that minimization of $c(\theta)$ solves the extended efficient large-scale ALP problem.

Unfortunately, calculating the gradients of $c(\theta)$ is $O(XA)$. Instead, we construct unbiased estimators and use stochastic gradient descent. Let T be the number of iterations of our algorithm. Let q_1 and q_2 be distributions over the state-action and state space, respectively (we will later discuss how to choose them). Let $((x_t, a_t))_{t=1\ldots T}$ be i.i.d. samples from q_1 and $(x'_t)_{t=1\ldots T}$ be i.i.d. samples from q_2. At round t, the algorithm estimates subgradient $\nabla c(\theta)$ by

$$g_t(\theta) = \ell^\top\Phi - H\frac{\Phi_{(x_t,a_t),:}}{q_1(x_t,a_t)}\mathbb{I}_{\{\mu_0(x_t,a_t)+\Phi_{(x_t,a_t),:}\theta<0\}} + H\frac{(P-B)^\top_{:,x'_t}\Phi}{q_2(x'_t)}s((P-B)^\top_{:,x'_t}\Phi\theta). \tag{8}$$

This estimate is fed to the projected subgradient method, which in turn generates a vector θ_t. After T rounds, we average vectors $(\theta_t)_{t=1\ldots T}$ and obtain the final solution $\widehat{\theta}_T = \sum_{t=1}^T \theta_t/T$. Vector $\mu_0 + \Phi\widehat{\theta}_T$ defines a policy, which in turn defines a stationary distribution $\mu_{\widehat{\theta}_T}$.[2] The algorithm is shown in Figure 1.

Input: Constant $S > 0$, number of rounds T, constant H.
Let Π_Θ be the Euclidean projection onto Θ.
Initialize $\theta_1 = 0$.
for $t := 1, 2, \ldots, T$ **do**
  Sample $(x_t, a_t) \sim q_1$ and $x'_t \sim q_2$.
  Compute subgradient estimate g_t (8).
  Update $\theta_{t+1} = \Pi_\Theta(\theta_t - \eta_t g_t)$.
end for
$\widehat{\theta}_T = \frac{1}{T}\sum_{t=1}^T \theta_t$.
Return policy $\pi_{\widehat{\theta}_T}$.

Figure 1. The Stochastic Subgradient Method for Markov Decision Processes

2.1. Analysis

In this section, we state and prove our main result, Theorem 1. We begin with a discussion of the assumptions we make then follow with the main theorem. We break the proof into two main ingredients. First, we demonstrate that a good approximation to the surrogate loss gives a feature vector that is almost a stationary distribution; this is Lemma 2. Second, we justify the use of unbiased gradients in Theorem 3 and Lemma 5. The section concludes with the proof of Theorem 1.

We make a mixing assumption on the MDP so that any policy quickly converges to its stationary distribution.

Assumption A1 ***(Fast Mixing)*** Let M^π be a $X \times (XA)$ matrix that encodes policy π, $M^\pi_{(i,(i-1)A+1)\text{-}(i,iA)} = \pi(\cdot|x_i)$. Other entries of this matrix are zero. For any policy π, there exists a constant $\tau(\pi) > 0$ such that for all distributions d and d' over the state-action space, $\|dPM^\pi - d'PM^\pi\|_1 \leq e^{-1/\tau(\pi)}\|d - d'\|_1$.

Further, we assume columns of the feature matrix Φ are positive and sum to one. Define

$$C_1 = \max_{(x,a)\in\mathcal{X}\times\mathcal{A}} \frac{\left\|\Phi_{(x,a),:}\right\|}{q_1(x,a)}\,,$$
$$C_2 = \max_{x\in\mathcal{X}} \frac{\left\|(P-B)^\top_{:,x}\Phi\right\|}{q_2(x)}\,.$$

These constants appear in our performance bounds. So we would like to choose distributions q_1 and q_2 such that C_1 and C_2 are small. For example, if there is $C' > 0$ such that for any (x,a) and i, $\Phi_{(x,a),i} \leq C'/(XA)$ and each column of P has only N non-zero elements, then we can simply choose q_1 and q_2 to be uniform distributions. Then it is easy to see that

$$\frac{\left\|\Phi_{(x,a),:}\right\|}{q_1(x,a)} \leq C'\,, \qquad \frac{\left\|(P-B)^\top_{:,x}\Phi\right\|}{q_2(x)} \leq C'(N+A)\,.$$

As another example, if $\Phi_{:,i}$ are exponential distributions and feature values at neighboring states are close to each other, then we can choose q_1 and q_2 to be appropriate exponential distributions so that $\left\|\Phi_{(x,a),:}\right\|/q_1(x,a)$ and $\left\|(P-B)^\top_{:,x}\Phi\right\|/q_2(x)$ are always bounded. Another example is when there exists a constant $C'' > 0$ such

[2] Recall that μ_θ is the stationary distribution of policy

$$\pi_\theta(a|x) = \frac{[\mu_0(x,a) + \Phi_{(x,a),:}\theta]_+}{\sum_{a'}[\mu_0(x,a') + \Phi_{(x,a'),:}\theta]_+}\,.$$

With an abuse of notation, we use μ_θ to denote policy π_θ as well.

that,[3] for any x, $\left\|P_{:,x}^\top \Phi\right\| / \left\|B_{:,x}^\top \Phi\right\| < C''$ and we have access to an efficient algorithm that computes $Z_1 = \sum_{(x,a)} \left\|\Phi_{(x,a),:}\right\|$ and $Z_2 = \sum_x \left\|B_{:,x}^\top \Phi\right\|$ and can sample from $q_1(x,a) = \left\|\Phi_{(x,a),:}\right\| / Z_1$ and $q_2(x) = \left\|B_{:,x}^\top \Phi\right\| / Z_2$. In what follows, we assume that appropriate distributions q_1 and q_2 are known.

We now state the main theorem.

Theorem 1. *Consider an expanded efficient large-scale dual ALP problem, with violation function $V = O(V_1 + V_2)$, defined by*

$$V_1(\theta) = \|[\mu_0 + \Phi\theta]_-\|_1$$
$$V_2(\theta) = \left\|(P - B)^\top (\mu_0 + \Phi\theta)\right\|_1 .$$

Assume $\tau := \sup\{\tau(\mu_\theta) : \theta \in \Theta\} < \infty$ is finite. Suppose we apply the stochastic subgradient method (shown in Figure 1) to the problem. Let $\epsilon \in (0,1)$. Let $T = 1/\epsilon^4$ be the number of rounds and $H = 1/\epsilon$ be the constraints multiplier in the subgradient estimate (8). Let $\widehat{\theta}_T$ be the output of the stochastic subgradient method after T rounds and let the learning rate be $\eta_t = S/(G'\sqrt{T})$, where $G' = \sqrt{d} + H(C_1 + C_2)$. Then, for any $\delta \in (0,1)$, with probability at least $1 - \delta$,

$$\mu_{\widehat{\theta}_T}^\top \ell \le \min_{\theta \in \Theta} \left(\mu_\theta^\top \ell + O\left(\frac{1}{\epsilon} (V_1(\theta) + V_2(\theta)) \right) + O(\epsilon) \right) , \tag{9}$$

where the constants hidden in the big-O notation are polynomials in S, d, C_1, C_2, and $\log(1/\delta)$.

The functions V_1 and V_2 are bounded by small constants for any set of normalized features: for any $\theta \in \Theta$,

$$\begin{aligned} V_1(\theta) &\le \|\mu_0\|_1 + \|\Phi\theta\|_1 \\ &\le 1 + \sum_{(x,a)} \left|\Phi_{(x,a),:}\theta\right| \le 1 + S\sqrt{d} , \end{aligned}$$

where the last step follows from the fact that columns of Φ are probability distributions. Further,

$$\begin{aligned} V_2(\theta) &\le \sum_{x'} \left|P_{:,x'}^\top (\mu_0 + \Phi\theta)\right| + \sum_{x'} \left|B_{:,x'}^\top (\mu_0 + \Phi\theta)\right| \\ &\le \sum_{x'} P_{:,x'}^\top \left|\mu_0 + \Phi\theta\right| + \sum_{x'} B_{:,x'}^\top \left|\mu_0 + \Phi\theta\right| \\ &= 2\mathbf{1}^\top \left|\mu_0 + \Phi\theta\right| \\ &\le 2\mathbf{1}^\top \left(|\mu_0| + |\Phi\theta|\right) \\ &\le 2(1 + S\sqrt{d}) . \end{aligned}$$

Thus V_1 and V_2 can be small given a carefully designed set of features.

[3] This condition requires that columns of Φ are close to their *one step look-ahead*.

The optimal choice for ϵ is $\epsilon = \sqrt{V_1(\theta_*) + V_2(\theta_*)}$, where θ_* is the minimizer of the RHS of (9). Thus, the optimized error bound scales like $O(\sqrt{V_1(\theta_*) + V_2(\theta_*)})$. Unfortunately, θ_* is not known in advance. To partially alleviate the problem, once we obtain $\widehat{\theta}_T$, we can estimate $V_1(\widehat{\theta}_T)$ and $V_2(\widehat{\theta}_T)$ and use input $\epsilon = \sqrt{V_1(\widehat{\theta}_T) + V_2(\widehat{\theta}_T)}$ in a second run of the algorithm.

The next lemma, providing the first ingredient of the proof, shows how the amount of constraint violation of a vector θ shifts the resulting stationary distribution μ_θ.

Lemma 2. *Let $u \in \mathbb{R}^{XA}$ be a vector. Assume*

$$\sum_{x,a} u(x,a) = 1, \ \|[u]_-\|_1 \le \epsilon', \ \left\|u^\top (P - B)\right\|_1 \le \epsilon'' .$$

The vector $[u]_+ / \|[u]_+\|_1$ defines a policy, which in turn defines a stationary distribution μ_u. We have that

$$\|\mu_u - u\|_1 \le (\tau(\mu_u) \log(1/(2\epsilon' + \epsilon'')) + 2)(2\epsilon' + \epsilon'') .$$

Proof. Define $h = [u]_+ / \|[u]_+\|_1$. We first show that h is almost a stationary distribution, in the sense that

$$\left\|h^\top (P - B)\right\|_1 \le 2\epsilon' + \epsilon'' . \tag{10}$$

To see this, notice that the first assumption is equivalent to $\|[u]_+\|_1 - \|[u]_-\|_1 = 1$, so $\left\|h^\top (P - B)\right\|_1$ is equal to

$$\begin{aligned} &\left\| \frac{[u]_+^\top}{\|[u]_+\|_1} (P - B) \right\|_1 \\ &= \frac{\left\|(u - [u]_-)^\top (P - B)\right\|_1}{1 + \|[u]_-\|_1} \\ &\le \left\|u^\top (P - B)\right\|_1 + \left\|[u]_-^\top (P - B)\right\|_1 \\ &\le \epsilon'' + \|[u]_-\|_1 \left\|(P - B)^\top\right\|_1 \\ &\le \epsilon'' + 2\epsilon', \end{aligned}$$

because the linear maps defined by P and B have operator norms (corresponding to the 1-norm) bounded by 1. Next, notice that

$$\begin{aligned} \|h - u\|_1 &\le \|h - [u]_+\|_1 + \|[u]_+ - u\|_1 \\ &= \|[u]_-\|_1 + \|[u]_-\|_1 \le 2\epsilon'. \end{aligned}$$

Next we bound $\|\mu_h - h\|_1$. Let $\nu_0 = h$ be the initial state-action distribution. We will show that as we run policy h (equivalently, policy μ_h), the state-action distribution converges to μ_h and this vector is close to h. From (10), we have $\nu_0^\top P = h^\top B + v_0$, where v_0 is such that $\|v_0\|_1 \le 2\epsilon' + \epsilon''$. Let M^h be the $X \times (XA)$ matrix that encodes policy h, via $M^h_{(i,(i-1)A+1)\text{-}(i,iA)} = h(\cdot | x = i)$. Other entries of this matrix are zero. Define the state-action distribution after running policy h for one step as

$$\begin{aligned} \nu_1^\top &:= h^\top P M^h = (h^\top B + v_0) M^h \\ &= h^\top B M^h + v_0 M^h = h^\top + v_0 M^h . \end{aligned}$$

Let $v_1 = v_0 M^h P = v_0 P^h$ and notice that $\|v_1\|_1 = \left\|P^{h\top} v_0^\top\right\|_1 \le \|v_0\|_1 \le 2\epsilon' + \epsilon''$. Thus,

$$\nu_2^\top = \nu_1^\top P M^h = h^\top + (v_0 + v_1) M^h .$$

By repeating this argument for k rounds, we get that

$$\nu_k^\top = h^\top + (v_0 + v_1 + \cdots + v_{k-1}) M^h .$$

Since the operator norm of M^h is no more than 1, $\left\|(v_0 + v_1 + \cdots + v_{k-1}) M^h\right\|_1 \le \sum_{i=0}^{k-1} \|v_i\|_1 \le k(2\epsilon' + \epsilon'')$. Thus, $\|\nu_k - h\|_1 \le k(2\epsilon' + \epsilon'')$. Now, since ν_k is the state-action distribution after k rounds of policy μ_h, by the mixing assumption, $\|\nu_k - \mu_h\|_1 \le 2e^{-k/\tau(h)}$. By the choice of $k = \tau(h)\log(1/(2\epsilon' + \epsilon''))$, we get that $\|\mu_h - h\|_1 \le (\tau(h)\log(1/(2\epsilon' + \epsilon'')) + 2)(2\epsilon' + \epsilon'')$. □

The second ingredient is the validity of using estimates of the subgradients. We assume access to estimates of the subgradient of a convex cost function. Error bounds can be obtained from results in the stochastic convex optimization literature; the following theorem, a high-probability version of Lemma 3.1 of Flaxman et al. (2005) for stochastic convex optimization, is sufficient. The proof can be found in Appendix B.

Theorem 3. *Let Z be a positive constant and $\mathcal{Z}$ be a bounded convex subset of $\mathbb{R}^d$ such that for any $z \in \mathcal{Z}$, $\|z\| \le Z$. Let $(f_t)_{t=1,2,\ldots,T}$ be a sequence of real-valued convex cost functions defined over $\mathcal{Z}$. Let $z_1, z_2, \ldots, z_T \in \mathcal{Z}$ be defined by $z_1 = 0$ and $z_{t+1} = \Pi_{\mathcal{Z}}(z_t - \eta f_t')$, where $\Pi_{\mathcal{Z}}$ is the Euclidean projection onto $\mathcal{Z}$, $\eta > 0$ is a learning rate, and $f_1', \ldots, f_T'$ are unbiased subgradient estimates such that $\mathbb{E}[f_t'|z_t] = \nabla f(z_t)$ and $\|f_t'\| \le F$ for some $F > 0$. Then, for $\eta = Z/(F\sqrt{T})$, for any $\delta \in (0,1)$, with probability at least $1 - \delta$,*

$$\sum_{t=1}^T f_t(z_t) - \min_{z\in\mathcal{Z}} \sum_{t=1}^T f_t(z) \le ZF\sqrt{T} + \sqrt{(1 + 4Z^2T)\left(2\log\frac{1}{\delta} + d\log\left(1 + \frac{Z^2T}{d}\right)\right)} . \tag{11}$$

Remark 4. *Let B_T denote the RHS of (11). If all cost functions are equal to f, then by convexity of f and an application of Jensen's inequality, we obtain that $f(\sum_{t=1}^T z_t/T) - \min_{z\in\mathcal{Z}} f(z) \le B_T/T$.*

As the next lemma shows, Theorem 3 can be applied in our problem to optimize the cost function c. The proof can be found in Appendix B.

Lemma 5. *Under the same conditions as in Theorem 1, we have that for any $\delta \in (0,1)$, with probability at least $1 - \delta$,*

$$c(\widehat{\theta}_T) - \min_{\theta\in\Theta} c(\theta) \le \frac{SG'}{\sqrt{T}} + \sqrt{\frac{1 + 4S^2T}{T^2}\left(2\log\frac{1}{\delta} + d\log\left(1 + \frac{S^2T}{d}\right)\right)} . \tag{12}$$

With both ingredients in place, we can prove our main result.

Proof of Theorem 1. Let b_T be the RHS of (12). Lemma 5 implies that with high probability for any $\theta \in \Theta$,

$$\ell^\top(\mu_0 + \Phi\widehat{\theta}_T) + H\,V_1(\widehat{\theta}_T) + H\,V_2(\widehat{\theta}_T) \le \ell^\top(\mu_0 + \Phi\theta) + H\,V_1(\theta) + H\,V_2(\theta) + b_T . \tag{13}$$

From (13), we get that

$$V_1(\widehat{\theta}_T) \le \frac{1}{H}\left(2(1 + S\sqrt{d}) + H\,V_1(\theta) + H\,V_2(\theta) + b_T\right) \stackrel{\text{def}}{=} \epsilon' , \tag{14}$$

$$V_2(\widehat{\theta}_T) \le \frac{1}{H}\left(2(1 + S\sqrt{d}) + H\,V_1(\theta) + H\,V_2(\theta) + b_T\right) \stackrel{\text{def}}{=} \epsilon'' . \tag{15}$$

Inequalities (14) and (15) and Lemma 2 give the following bound:

$$\left|\mu_{\widehat{\theta}_T}^\top \ell - (\mu_0 + \Phi\widehat{\theta}_T)^\top \ell\right| \le (\tau(\mu_{\widehat{\theta}_T})\log(1/(2\epsilon' + \epsilon'')) + 2)(2\epsilon' + \epsilon'') . \tag{16}$$

From (13) we also have

$$\ell^\top(\mu_0 + \Phi\widehat{\theta}_T) \le \ell^\top(\mu_0 + \Phi\theta) + H\,V_1(\theta) + H\,V_2(\theta) + b_T ,$$

which, together with (16) and Lemma 2, gives the final result:

$$\begin{aligned}
\mu_{\widehat{\theta}_T}^\top \ell &\le \ell^\top(\mu_0 + \Phi\theta) + H\,V_1(\theta) + H\,V_2(\theta) + b_T \\
&\quad + (\tau(\mu_{\widehat{\theta}_T})\log(1/(2\epsilon' + \epsilon'')) + 2)(2\epsilon' + \epsilon'') \\
&\le \mu_\theta^\top \ell + H\,V_1(\theta) + H\,V_2(\theta) + b_T \\
&\quad + (\tau(\mu_{\widehat{\theta}_T})\log(1/(2\epsilon' + \epsilon'')) + 2)(2\epsilon' + \epsilon'') \\
&\quad + (\tau(\mu_\theta)\log(1/(2V_1(\theta) + V_2(\theta)))) \\
&\qquad \times (2V_1(\theta) + V_2(\theta)) .
\end{aligned}$$

Recall that $b_T = O(H/\sqrt{T})$. Because $H = 1/\epsilon$ and $T = 1/\epsilon^4$, we get that with high probability, for any $\theta \in \Theta$, $\mu_{\widehat{\theta}_T}^\top \ell \le \mu_\theta^\top \ell + O\left(\frac{1}{\epsilon}(V_1(\theta) + V_2(\theta))\right) + O(\epsilon)$.

□

Let's compare Theorem 1 with results of de Farias and Van Roy (2006). Their approach is to relate the original MDP to a perturbed version[4] and then analyze the corresponding ALP. (See Appendix A for more details.) Let Ψ be a feature matrix that is used to estimate value functions. Recall that λ_* is the average loss of the optimal policy and λ_w is the average loss of the greedy policy with respect to value function Ψw. Let h^*_γ be the differential value function when the restart probability in the perturbed MDP is $1-\gamma$. For vector v and positive vector u, define the weighted maximum norm $\|v\|_{\infty,u} = \max_x u(x)\,|v(x)|$. de Farias and Van Roy (2006) prove that for appropriate constants $C, C' > 0$ and weight vector u,

$$\lambda_{w_*} - \lambda_* \leq \frac{C}{1-\gamma} \min_w \left\| h^*_\gamma - \Psi w \right\|_{\infty,u} + C'(1-\gamma)\ . \tag{17}$$

This bound has similarities to bound (9): tightness of both bounds depends on the quality of feature vectors in representing the relevant quantities (stationary distributions in (9) and value functions in (17)). Once again, we emphasize that the algorithm proposed by de Farias and Van Roy (2006) is computationally expensive and requires access to a distribution that depends on optimal policy.

Remark 6. *In our algorithm, we estimate the subgradient by sampling constraints of the LP. A natural question to ask is if we can first sample constraints then exactly solve the resulting LP. Analysis for such an algorithm is presented in Appendix C. However the analysis requires stronger conditions on the choice of feature vectors.*

3. Experiments

In this section, we apply our algorithm to the four-dimensional discrete-time queueing network illustrated in Figure 3. This network has a relatively long history; see, e.g. (Rybko and Stolyar, 1992) and more recently (de Farias and Van Roy, 2003a) (c.f. Section 6.2). There are four queues, $\mu_1, \ldots, \mu_4$, each with state $0, \ldots, B$. Since the cardinality of the state space is $X = (1+B)^4$, even a modest B results in huge state spaces. For time t, let $X_t \in \mathcal{X}$ be the state and let $s_{i,t} \in \{0,1\}$, $i = 1,2,3,4$ denote the actions. The value $s_{i,t} = 1$ indicates that queue i is being served. Server 1 only serves queue 1 or 4, server 2 only serves queue 2 or 3, and neither server can idle. Thus, $s_{1,t} + s_{4,t} = 1$ and $s_{2,t} + s_{3,t} = 1$. The dynamics are defined by the rate parameters $a_1, a_3, d_1, d_2, d_3, d_4 \in (0,1)$ as follows. At each time t, the following random variables are sampled independently: $A_{1,t} \sim$ Bernoulli(a_1), $A_{3,t} \sim$ Bernoulli(a_3), and $D_{i,t} \sim$ Bernoulli$(d_i s_{i,t})$ for $i = 1,2,3,4$. Using $e_1, \ldots, e_4$ to denote the standard basis vectors, the dynamics are:

$$\begin{aligned} X'_{t+1} =& X_t + A_{1,t}e_1 + A_{3,t}e_3 \\ &+ D_{1,t}(e_2 - e_1) - D_{2,t}e_2 \\ &+ D_{3,t}(e_4 - e_3) - D_{4,t}e_4, \end{aligned}$$

and $X_{t+1} = \max(\mathbf{0}, \min(B, X'_{t+1}))$ (i.e. all four states are thresholded from below by 0 and above by B). The loss function is the total queue size: $\ell(X_t) = ||X_t||_1$. We compared our method against two common heuristics. In the first, denoted LONGER, each server operates on the queue that is longer with ties broken uniformly at random (e.g. if queue 1 and 4 had the same size, they are equally likely to be served). In the second, denoted LBFS (last buffer first served), the downstream queues always have priority (server 1 will serve queue 4 unless it has length 0, and server 2 will serve queue 2 unless it has length 0). These heuristics are common and have been used an benchmarks for queueing networks (e.g. (de Farias and Van Roy, 2003a)).

We used $a_1 = a_3 = .08$, $d_1 = d_2 = .12$, and $d_3 = d_4 = .28$, and buffer sizes $B_1 = B_4 = 38$, $B_2 = B_3 = 25$ as the parameters of the network. The asymmetric size was chosen because server 1 is the bottleneck and tends to have longer queues. The first two features are the stationary distributions corresponding to the two heuristics LONGER and LBFS. We also included two types of features that do not correspond to stationary distribution. For every interval $(0,5], (6,10], \ldots, (45,50]$ and action A, we added a feature ψ with $\varphi(x,a) = 1$ if $\ell(x,a)$ is in the interval and $a = A$. To define the second type, consider the three intervals $I_1 = [0,10]$, $I_2 = [11,20]$, and $I_3 = [21,25]$. For every 4-tuple of intervals $(J_1, J_2, J_3, J_4) \in \{I_1, I_2, I_3\}^4$ and action A, we created a feature ψ with $\psi(x,a) = 1$ only if $x_i \in J_i$ and $a = A$. Every feature was normalized to sum to 1. In total, we had 372 features which is about a 10^4 reduction in dimension from the original problem.

To obtain a lower variance estimate of our gradient, we sampled $g_t(\theta)$ 1000 times and averaged (which is equivalent to sampling 1000 i.i.d. constraints from both q_1 and q_2). Rather than the fixed learning rate η considered in Section 2, our learning rate began at 10^{-4} and halved every 2000 iterations. The results of the simulations are plotted in Figure 3, where $\widehat{\theta}_t$ denotes the running average of θ_t. The left plot is of the LP objective, $\ell^\top(\mu_0 + \Phi\widehat{\theta}_t)$. The middle plot is of the sum of the constraint violations, $\left\|[\mu_0 + \Phi\widehat{\theta}_t]_-\right\|_1 + \left\|(P-B)^\top\Phi\widehat{\theta}_t\right\|_1$. Thus, $c(\widehat{\theta}_t)$ is a scaled sum of the first two plots. Finally, the right plot is of the average losses, $\ell^\top\mu_{\widehat{\theta}_t}$ and the two horizontal lines correspond to the loss of the two heuristics, LONGER and LBFS. The right plot demonstrates that, as predicted by our theory, minimizing the surrogate loss $c(\theta)$ does lead to lower average losses.

[4] In a perturbed MDP, the state process restarts with a certain probability to a *restart distribution*. Such perturbed MDPs are closely related to discounted MDPs.

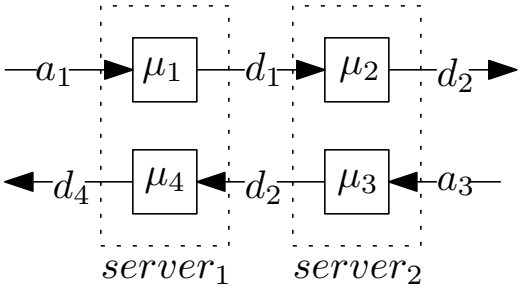

Figure 2. The 4D queueing network. Customers arrive at queue μ_1 or μ_3 then are referred to queue μ_2 or μ_4, respectively. Server 1 can either process queue 1 or 4, and server 2 can only process queue 2 or 3.

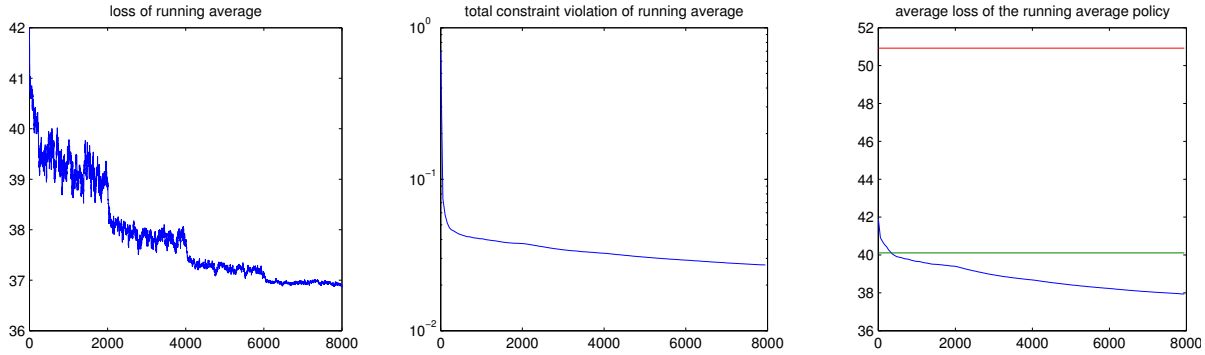

Figure 3. The left plot is of the linear objective of the running average, i.e. $\ell^\top \Phi \widehat{\theta}_t$. The center plot is the sum of the two constraint violations of $\widehat{\theta}_t$, and the right plot is $\ell^\top \tilde{\mu}_{\widehat{\theta}_t}$ (the average loss of the derived policy). The two horizontal lines correspond to the loss of the two heuristics, LONGER and LBFS.

All previous algorithms (including (de Farias and Van Roy, 2003a)) work with value functions, while our algorithm works with stationary distributions. Due to this difference, we cannot use the same feature vectors to make a direct comparison. The solution that we find in this different approximating set is comparable to the solution of de Farias and Van Roy (2003a).

4. Conclusions

In this paper, we defined and solved the extended large-scale efficient ALP problem. We proved that, under certain assumptions about the dynamics, the stochastic subgradient method produces a policy with average loss competitive to all $\theta \in \Theta$, not not just all θ producing a stationary distribution. We demonstrated this algorithm on the Rybko-Stoylar four-dimensional queueing network and recovered a policy better than two common heuristics and comparable to previous results on ALPs (de Farias and Van Roy, 2003a). A future direction is to find other interesting regularity conditions under which we can handle large-scale MDP problems. We also plan to conduct more experiments with challenging large-scale problems.

5. Acknowledgements

We gratefully acknowledge the support of the NSF through grant CCF-1115788 and of the ARC through an Australian Research Council Australian Laureate Fellowship (FL110100281).

References

Y. Abbasi-Yadkori. *Online Learning for Linearly Parametrized Control Problems*. PhD thesis, University of Alberta, 2012.

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

D. P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 2007.

D. P. Bertsekas and J. Tsitsiklis. *Neuro-Dynamic Programming*. Athena scientific optimization and computation series. Athena Scientific, 1996.

G. Calafiore and M. C. Campi. Uncertain convex programs: randomized solutions and confidence levels. *Mathematical Programming*, 102(1):25–46, 2005.

M. C. Campi and S. Garatti. The exact feasibility of randomized solutions of uncertain convex programs. *SIAM Journal on Optimization*, 19(3):1211–1230, 2008.

D. P. de Farias and B. Van Roy. The linear programming approach to approximate dynamic programming. *Operations Research*, 51, 2003a.

D. P. de Farias and B. Van Roy. Approximate linear programming for average-cost dynamic programming. In *NIPS*, 2003b.

D. P. de Farias and B. Van Roy. On constraint sampling in the linear programming approach to approximate dynamic programming. *Mathematics of Operations Research*, 29, 2004.

D. P. de Farias and B. Van Roy. A cost-shaping linear program for average-cost approximate dynamic programming with performance guarantees. *Mathematics of Operations Research*, 31, 2006.

V. H. de la Peña, T. L. Lai, and Q-M. Shao. *Self-normalized processes: Limit theory and Statistical Applications*. Springer, 2009.

V. V. Desai, V. F. Farias, and C. C. Moallemi. Approximate dynamic programming via a smoothed linear program. *Operations Research*, 60(3):655–674, 2012.

A. D. Flaxman, A. T. Kalai, and H. B. McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. In *SODA*, 2005.

C. Guestrin, M. Hauskrecht, and B. Kveton. Solving factored mdps with continuous and discrete variables. In *UAI*, 2004.

M. Hauskrecht and B. Kveton. Linear program approximations to factored continuous-state markov decision processes. In *NIPS*, 2003.

R. A. Howard. *Dynamic Programming and Markov Processes*. MIT, 1960.

H. R. Maei, Cs. Szepesvári, S. Bhatnagar, D. Precup, D. Silver, and R. S. Sutton. Convergent temporal-difference learning with arbitrary smooth function approximation. In *NIPS*, 2009.

H. R. Maei, Cs. Szepesvári, S. Bhatnagar, and R. S. Sutton. Toward off-policy learning control with function approximation. In *ICML*, 2010.

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

M. Petrik and S. Zilberstein. Constraint relaxation in approximate linear programs. In *ICML*, 2009.

A. N. Rybko and A. L. Stolyar. Ergodicity of stochastic processes describing the operation of open queueing networks. *Problemy Peredachi Informatsii*, 28(3):3–26, 1992.

P. Schweitzer and A. Seidmann. Generalized polynomial approximations in Markovian decision processes. *Journal of Mathematical Analysis and Applications*, 110: 568–582, 1985.

R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. Bradford Book. MIT Press, 1998.

R. S. Sutton, H. R. Maei, D. Precup, S. Bhatnagar, D. Silver, Cs. Szepesvári, and E. Wiewiora. Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *ICML*, 2009a.

R. S. Sutton, Cs. Szepesvári, and H. R. Maei. A convergent O(n) algorithm for off-policy temporal-difference learning with linear function approximation. In *NIPS*, 2009b.

V. N. Vapnik and A. Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16 (2):264–280, 1971.

M. H. Veatch. Approximate linear programming for average cost mdps. *Mathematics of Operations Research*, 38 (3), 2013.

T. Wang, D. Lizotte, M. Bowling, and D. Schuurmans. Dual representations for dynamic programming. *Journal of Machine Learning Research*, pages 1–29, 2008.

M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *ICML*, 2003.