

On Constraint Sampling in the Linear Programming Approach to Approximate Linear Programming

Daniela Pucci de Farias
Department of Mechanical Engineering
Massachusetts Institute of Technology
Cambridge, MA 02139, USA
pucci@mit.edu

Benjamin Van Roy
Department of Management Science and
Engineering
Stanford University
Stanford, CA 94305, USA
bvr@stanford.edu

Abstract—In the linear programming approach to approximate dynamic programming, one tries to solve a certain linear program — the ALP — which has a relatively small number K of variables but an intractable number M of constraints. In this paper, we study a scheme that samples and imposes a subset of $m \ll M$ constraints. A natural question that arises in this context is: How must m scale with respect to K and M in order to ensure that the resulting approximation is almost as good as one given by exact solution of the ALP? We show that, under certain idealized conditions, m can be chosen independently of M and need grow only as a polynomial in K .

I. INTRODUCTION

Due to the “curse of dimensionality,” Markov decision processes typically have a prohibitively large number of states, rendering exact dynamic programming methods intractable and calling for the development of approximation techniques. This paper represents a step in the development of a linear programming approach to approximate dynamic programming [3], [11], [12], [13]. This approach relies on solving a linear program that generally has few variables but an intractable number of constraints. In this paper, we propose and analyze a constraint sampling method for approximating the solution to this linear program.

Markov Decision Processes. We consider a Markov decision process (MDP) with a finite state space $\mathcal{S} = \{1, \dots, |\mathcal{S}|\}$. In each state $x \in \mathcal{S}$, there is a finite set of admissible actions $\mathcal{A}_x$. Given a choice of action $a \in \mathcal{A}_x$, a cost $g_a(x) \geq 0$ is incurred, and the probability that the next state is $y \in \mathcal{S}$ is given by $P_a(x, y)$. A policy u is a mapping from states to actions. Our interest is in finding an optimal policy: one that minimizes expected infinite-horizon discounted costs $J_u(x) = \sum_{t=0}^{\infty} \alpha^t (P_u^t g_u)(x)$, simultaneously for all initial states $x \in \mathcal{S}$. Here, $\alpha \in (0, 1)$ is the discount factor, P_u is a matrix whose xy th component is equal to $P_{u(x)}(x, y)$, and g_u is a vector whose x th component is equal to $g_{u(x)}(x)$.

The cost-to-go function J_u associated with a policy u is the unique solution to $J_u = T_u J_u$, where the operator T_u is defined by $T_u J = g_u + \alpha P_u J$. The optimal cost-to-go function $J^* = \min_u J_u$ is the unique solution to Bellman’s equation: $J^* = T J^*$, where the operator T is defined by $TJ = \min_u T_u J$ and the minimization is carried out component-wise. We call a policy u greedy with respect to a vector J if $TJ = T_u J$. Any policy that is greedy with respect to J^* is optimal.

The Linear Programming Approach. An optimal policy can be obtained through computing J^* , and employing a respective greedy policy. However, in many practical contexts, each state is associated with a vector of state variables, and therefore, the cardinality of the state space grows exponentially with the number of state variables. This makes it infeasible to compute or even to store J^* . One approach to addressing this difficulty involves approximation of J^* .

We consider approximating J^* by a linear combination of preselected basis functions $\phi_k : \mathcal{S} \mapsto \mathbb{R}$, $k = 1, \dots, K$. The aim is to generate a weight vector $\tilde{r} \in \mathbb{R}^K$ such that $J^*(x) \approx \sum_{k=1}^K \phi_k(x) \tilde{r}_k$, and to use a policy that is greedy with respect to the associated approximation. We will use matrix notation to represent our approximation: $\Phi \tilde{r} = \sum_{k=1}^K \phi_k(x) \tilde{r}_k$, where $\Phi = \begin{bmatrix} \phi_1 & \dots & \phi_K \end{bmatrix}$.

In the linear programming approach, weights $\tilde{r}$ are generated by solving a certain linear program — the approximate linear program (ALP):

$$\begin{aligned} \max \quad & c^T \Phi r \\ \text{s.t.} \quad & g_a(x) + \alpha \sum_y P_a(x, y) (\Phi r)(y) \geq (\Phi r)(x), \quad \forall x \in \mathcal{S}, a \in \mathcal{A}_x, \end{aligned} \quad (1)$$

where c is a vector of *state-relevance weights* for which every component is positive, and c^T denotes the transpose of c . As discussed in [3], the ALP minimizes $\|J^* - \Phi r\|_{1,c}$, subject to the constraints. For any positive vector v , we define the weighted L_1 and L_∞ norms $\|J\|_{1,v} = \sum_x v(x) |J(x)|$, $\|J\|_{\infty,v} = \max_x v(x) |J(x)|$. [3] discusses the role of state-relevance weights c and why, given an appropriate choice of c , minimization of $\|J^* - \Phi r\|_{1,c}$ is desirable.

Constraint Sampling. While the ALP may involve only a small number of variables, there is a potentially intractable number of constraints — one per state-action pair. As such, we cannot in general expect to solve the ALP exactly. The focus of this paper is on a tractable approximation to the ALP: the reduced linear program (RLP).

Generation of an RLP relies on three objects: (1) a constraint sample size m , (2) a probability measure ψ over the set of state-action pairs, and (3) a bounding set $\mathcal{N} \subseteq \mathbb{R}^K$. The probability measure ψ represents a distribution from which we will sample constraints. In particular, we consider a set $\mathcal{X}$ of m state-action pairs, each independently sampled according to ψ . The set $\mathcal{N}$ is a parameter that restricts the

magnitude of the RLP solution. This set should be chosen such that it contains $\Phi\tilde{r}$. The RLP is defined by

$$\begin{aligned} \max \quad & c^T \Phi r \\ \text{s.t.} \quad & g_a(x) + \alpha \sum_y P_a(x, y) (\Phi r)(y) \geq (\Phi r)(x), \quad \forall (x, a) \in \mathcal{X} \\ & r \in \mathcal{N}. \end{aligned} \quad (2)$$

Let $\tilde{r}$ be an optimal solution of the ALP and let $\hat{r}$ be an optimal solution of the RLP. In order for the solution of the RLP to be meaningful, we would like $\|J^* - \Phi\hat{r}\|_{1,c}$ to be close to $\|J^* - \Phi\tilde{r}\|_{1,c}$. To formalize this, we consider a requirement that $\Pr \left\{ \left| \|J^* - \Phi\hat{r}\|_{1,c} - \|J^* - \Phi\tilde{r}\|_{1,c} \right| \leq \varepsilon \right\} \geq 1 - \delta$, where $\varepsilon > 0$ is an error tolerance parameter and $\delta > 0$ parameterizes a level of confidence $1 - \delta$. This paper focusses on characterizing a sample size m that is sufficient for meeting such a requirement.

Results of our Analysis. To apply the RLP, given a problem instance, one must select parameters m , ψ , and $\mathcal{N}$. In order for the RLP to be practically solvable, the sample size m must be tractable. Results of our analysis suggest that if ψ and $\mathcal{N}$ are well-chosen, an error tolerance of ε can be accommodated with confidence $1 - \delta$ given a sample size m that grows as a polynomial in K , $1/\varepsilon$, and $\log 1/\delta$, and is independent of the total number of ALP constraints.

Our analysis is carried out in two parts:

1) Sample complexity of near-feasibility. The first part of our analysis applies to constraint sampling in general linear programs – not just the ALP. Suppose that we are given a set of linear constraints

$$\gamma_z^T r + \kappa_z \geq 0, \quad \forall z \in \mathcal{Z},$$

on variables $r \in \mathbb{R}^K$, a probability measure ψ on $\mathcal{Z}$, and a desired error tolerance ε and confidence $1 - \delta$. Let $z_1, z_2, \dots$ be independent identically distributed samples drawn from $\mathcal{Z}$ according to ψ . We will establish that there is a sample size

$$m = O \left(\frac{1}{\varepsilon^2} \left(\ln \frac{1}{\delta} + K \ln \frac{1}{\varepsilon} \right) \right)$$

such that, with probability at least $1 - \delta$, there exists a subset $Z \subseteq \mathcal{Z}$ of measure $\psi(Z) \geq 1 - \varepsilon$ such that every vector r satisfying $\gamma_{z_i}^T r + \kappa_{z_i} \geq 0, \quad \forall i = 1, \dots, m$, also satisfies $\gamma_z^T r + \kappa_z \geq 0, \quad \forall z \in Z$. We refer to the latter criterion as *near-feasibility* — nearly all the constraints are satisfied. The main point of this part of the analysis is that near-feasibility can be obtained with high-confidence through imposing a tractable number m of samples.

2) Sample complexity of a good approximation. We would like the error $\|J^* - \Phi\hat{r}\|_{1,c}$ of an optimal solution $\hat{r}$ to the RLP to be close to the error $\|J^* - \Phi\tilde{r}\|_{1,c}$ of an optimal solution to the ALP. In a generic linear program, near-feasibility is not sufficient to bound such an error metric. However, because of special structure associated with the RLP, given appropriate choices of ψ and $\mathcal{N}$, near-feasibility

leads to such a bound. In particular, given a sample size

$$m = O \left(\frac{A^2 \theta^2}{(1 - \alpha)^2 \varepsilon^2} \left(\ln \frac{1}{\delta} + K \ln \frac{A\theta}{\varepsilon(1 - \alpha)} \right) \right),$$

where $A = \max_x |\mathcal{A}_x|$, with probability at least $1 - \delta$ we have

$$\|J^* - \Phi\hat{r}\|_{1,c} \leq \|J^* - \Phi\tilde{r}\|_{1,c} + \varepsilon.$$

The parameter θ , which is to be defined precisely later, depends on the particular MDP problem instance, the choice of basis functions, and the set $\mathcal{N}$.

A major weakness of our error bound is that it relies on an idealized choice of ψ . In particular, the choice we will put forth assumes knowledge of an optimal policy. Alas, we typically do not know an optimal policy — that is what we are after in the first place. Nevertheless, the result provides guidance on what makes a desirable choice of distribution. The spirit here is analogous to one present in the importance sampling literature. In that context, the goal is to reduce variance in Monte Carlo simulation through intelligent choice of a sampling distribution and appropriate distortion of the function being integrated. Characterizations of idealized sampling distributions guide the design of heuristics that are ultimately implemented.

The set $\mathcal{N}$ also plays a critical role in the bound. It influences the value of θ , and an appropriate choice is necessary in order for this term to scale gracefully with problem size. Ideally, given a class of problems, there should be a mechanism for generating $\mathcal{N}$ such that θ grows no faster than a low-order polynomial function of the number of basis functions and the number of state variables. As we will later discuss through an example involving controlled queueing networks, we expect that it will be possible to design effective mechanisms for selecting $\mathcal{N}$ for practical classes of problems.

It is worth mentioning that our sample complexity bounds are loose. Our emphasis is on showing that the number of required samples can be independent of the total number of constraints and can scale gracefully with respect to the number of variables. Furthermore, our emphasis is on a general result that holds for a broad class of MDPs, and therefore we do not exploit special regularities associated with particular choices of basis functions or specific problems. In the presence of such special structure, it may be possible to provide much tighter bounds or even methods for exact solution of the ALP. The significance of our results is that they suggest viability of the linear programming approach to approximate dynamic programming even in the absence of such favorable special structure.

Literature Review. We classify approaches to solving the ALP and, more generally, linear programs with large numbers of constraints into two categories. The first focusses on exploiting problem-specific structure, whereas the second devises general methods for solving problems with large

numbers of constraints. Our work falls into the second category. Morrison and Kumar [9], Guestrin et al. [6], Schuurmans and Patrascu [10] and Grötschel and Holland [5] formulate approximate linear programming algorithms for problems such as control of queueing networks, factored MDP's and the travelling salesman. The second category includes the work of Trick and Zin [12], [13], Clarkson [8] and Calafiore and Campi [2]. [12], [13] present several constraint generation heuristics for the ALP for low-dimensional problems. Clarkson's Las Vegas algorithm [8] is another general-purpose constraint sampling scheme for linear programs with large numbers of constraints, based on iteratively selecting constraints until all binding constraints are identified. Finally, we note that, about a year after this paper was submitted for publication, Calafiore and Campi [2] independently developed a very similar constraint sampling scheme and sample complexity bound. Their work focusses on the number of samples needed for near-feasibility of an optimal solution to an optimization problem with sampled constraints. It is not intended to address relations between near-feasibility and approximation error. Their work is an important additional contribution to the topic of constraint sampling, as there are several notable differences from what is presented in this paper. First, their work treats general convex programs, rather than linear programs. Second, their analysis is quite different and focusses on near-feasibility of an optimal solution, rather than all feasible solutions. Interestingly, depending on the desired confidence $1 - \delta$, their bound can be stronger or weaker than the one stated later in this paper. The bounds may be reconciled — at least in part — by the possibility of “confidence pumping” [7].

II. SAMPLE COMPLEXITY OF NEAR-FEASIBILITY

We consider a set of linear constraints $\gamma_z^T r + \kappa_z \geq 0$, $\forall z \in \mathcal{Z}$, where $r \in \mathbb{R}^K$ and $\mathcal{Z}$ is a set of constraint indices. To keep the exposition simple, we assume that $\mathcal{Z}$ is finite or countable. This assumption is sufficient for our purposes, since we consider MDP's with finite state and action spaces. We assume that there are relatively few variables and a possibly huge number of constraints, i.e., $K \ll |\mathcal{Z}|$, as is the case in the ALP. In such a situation, we expect that many of the constraints will be irrelevant, either because they are always inactive or because they have a minor impact on the feasible region. Therefore one might speculate that the feasible region specified by all constraints can be well-approximated by a sampled subset of these constraints. In the sequel, we show that this is indeed the case, at least with respect to a certain criterion for a good approximation. We also show that the number of constraints necessary to guarantee a good approximation does not depend on the total number of constraints, but rather on the number of variables.

Our constraint sampling scheme relies heavily on a probability measure ψ on $\mathcal{Z}$. The distribution ψ will have a dual

role in our approximation scheme: on one hand, constraints will be sampled according to ψ ; on the other hand, the same distribution will be involved in the criterion for assessing the quality of a particular set of sampled constraints.

We consider a subset of constraints to be good if we can guarantee that, by satisfying this subset, the set of constraints that are not satisfied has small measure. In other words, given a tolerance parameter $0 < \varepsilon < 1$, we want to have $\mathcal{W} \subset \mathcal{Z}$ satisfying

$$\sup_{r: \gamma_z^T r + \kappa_z \geq 0, \forall z \in \mathcal{W}} \psi(\{y: \gamma_y^T r + \kappa_y < 0\}) \leq \varepsilon. \quad (3)$$

Whenever (3) holds for a subset $\mathcal{W}$, we say that $\mathcal{W}$ leads to near-feasibility.

The next theorem establishes a bound on the number m of (possibly repeated) sampled constraints necessary to ensure that the set $\mathcal{W}$ formed by them leads to near-feasibility with a prespecified probability $1 - \delta$.

Theorem 2.1: Let $0 < \delta < 1$, $0 < \varepsilon < 1$ be given constants and let $\mathcal{W}$ be the set formed by m i.i.d. random variables on $\mathcal{Z}$ with distribution ψ . Then if

$$m \geq \frac{32}{\varepsilon^2} \left[\ln 8 + \ln \frac{1}{\delta} + K \left(\ln \frac{16e}{\varepsilon} + \ln \ln \frac{16e}{\varepsilon} \right) \right], \quad (4)$$

we have

$$\sup_{r: \gamma_z^T r + \kappa_z \geq 0, \forall z \in \mathcal{W}} \psi(\{y: \gamma_y^T r + \kappa_y < 0\}) \leq \varepsilon, \quad (5)$$

with probability at least $1 - \delta$.

The proof of Theorem 2.1 is based on a simple application of the theory of uniform convergence of empirical probabilities [4], [14], and is presented in our full-length paper. The intuition is as follows. Since we are only interested in whether a constraint is violated or not, we may assume, without loss of generality, that $\|[\gamma_y^T \kappa_y]^T\| = 1$, for some norm $\|\cdot\|$. In this light, constraints on r can be interpreted as keeping $z_y^T \begin{bmatrix} r \\ 1 \end{bmatrix} \leq 0$ for a set of vectors z_y that are in the unit hypersphere in $\mathbb{R}^{k+1}$. Since the hypersphere is bounded, constraints should predominantly be redundant after a certain number have been sampled. The number required for this should depend linearly on k , independently of the size of the state-action space.

Theorem 2.1 shows that near-feasibility can be attained efficiently, even for problems with high-dimensional state spaces. However, near-feasibility is not enough to ensure that the RLP will offer a good approximation. In the next section, we introduce additional assumptions and the approximation error bounds that they enable.

III. SAMPLE COMPLEXITY OF A GOOD APPROXIMATION

In this section, we investigate the impact of using the RLP instead of the ALP on the error in the approximation of the cost-to-go function. We show in Theorem 3.1 that, by sampling a tractable number of constraints, the approximation

error yielded by the RLP is comparable to the error yielded by the ALP. The proof of Theorem 3.1 relies on special structure of the ALP — it is easy to see that such a result cannot hold for general linear programs.

Let us introduce certain constants and functions involved in our error bound. We first define a family of probability distributions on the state space $\mathcal{S}$, given by

$$\mu_u^T = (1 - \alpha)c^T(I - \alpha P_u)^{-1}, \quad (6)$$

for each policy u . Note that, if c is a probability distribution, $\mu_u(x)/(1 - \alpha)$ is the expected discounted number of visits to state x under policy u , if the initial state is distributed according to c . Furthermore, $\lim_{\alpha \uparrow 1} \mu_u(x)$ is a stationary distribution associated with policy u . We interpret μ_u as a measure of the relative importance of states under policy u .

We will make use of a *Lyapunov function* $V : \mathcal{S} \mapsto \mathbb{R}^+$, which is defined as follows.

Definition 3.1 (Lyapunov function): A function $V : \mathcal{S} \mapsto \mathbb{R}^+$ is called a *Lyapunov function* if there is a scalar $\beta_V < 1$ and an optimal policy u^* such that $\alpha P_{u^*} V \leq \beta_V V$.

Lemma 3.1: Let V be a Lyapunov function for an optimal policy u^* . Then T_{u^*} is a contraction with respect to $\|\cdot\|_{\infty, 1/V}$.

For any Lyapunov function V , we also define another family of probability distributions on the state space $\mathcal{S}$, given by $\mu_{u,V}(x) = \frac{\mu_u(x)V(x)}{\mu_u^T V}$. We also define a distribution over state-action pairs $\psi_{u,V}(x, a) = \frac{\mu_{u,V}(x)}{|\mathcal{A}_x|}$, $\forall a \in \mathcal{A}_x$. Finally, we define constants $A = \max_x |\mathcal{A}_x|$ and $\theta = \mu_{u^*}^T V \sup_{r \in \mathcal{N}} \|J^* - \Phi r\|_{\infty, 1/V}$.

We now present the main result of the paper — a bound on the approximation error introduced by constraint sampling.

Theorem 3.1: Let u^* be an optimal policy and $\mathcal{X}$ be a (random) set of m state-action pairs sampled independently according to the distribution $\psi_{u^*, V}(x, a)$, for some Lyapunov function V , where

$$m \geq \frac{512A^2\theta^2}{(1 - \alpha)^2\varepsilon^2} \left[\ln \frac{8}{\delta} + K \ln \left(\frac{64A\theta e}{(1 - \alpha)\varepsilon} \ln \frac{64A\theta e}{(1 - \alpha)\varepsilon} \right) \right]. \quad (7)$$

Let $\tilde{r}$ be an optimal solution of the ALP that is in $\mathcal{N}$, and let $\hat{r}$ be an optimal solution of the corresponding RLP. If $\tilde{r} \in \mathcal{N}$ then, with probability at least $1 - \delta$, we have

$$\|J^* - \Phi \hat{r}\|_{1,c} \leq \|J^* - \Phi \tilde{r}\|_{1,c} + \varepsilon. \quad (8)$$

Proof: From Theorem 2.1, given a sample size m , we have, with probability no less than $1 - \delta$,

$$\begin{aligned} \frac{(1 - \alpha)\varepsilon}{4A\theta} &\geq \psi_{u^*, V}(\{(x, a) : (T_a \Phi \hat{r})(x) < (\Phi \hat{r})(x)\}) \\ &= \sum_{x \in \mathcal{S}} \frac{\mu_{u^*, V}(x)}{|\mathcal{A}_x|} \sum_{a \in \mathcal{A}_x} 1_{(T_a \Phi \hat{r})(x) < (\Phi \hat{r})(x)} \\ &\geq \frac{1}{A} \sum_{x \in \mathcal{S}} \mu_{u^*, V}(x) 1_{(T_{u^*} \Phi \hat{r})(x) < (\Phi \hat{r})(x)}. \end{aligned} \quad (9)$$

For any vector J , we denote the positive and negative parts by $J^+ = \max(J, 0)$ and $J^- = \max(-J, 0)$, where the

maximization is carried out componentwise. Note that

$$\begin{aligned} \|J^* - \Phi \hat{r}\|_{1,c} &= c^T |(I - \alpha P_{u^*})^{-1}(g_{u^*} - (I - \alpha P_{u^*})\Phi \hat{r})| \\ &\leq c^T (I - \alpha P_{u^*})^{-1} |g_{u^*} - (I - \alpha P_{u^*})\Phi \hat{r}| \\ &= c^T (I - \alpha P_{u^*})^{-1} [(g_{u^*} - (I - \alpha P_{u^*})\Phi \hat{r})^+ + \\ &\quad + (g_{u^*} - (I - \alpha P_{u^*})\Phi \hat{r})^-] \\ &= c^T (I - \alpha P_{u^*})^{-1} [(g_{u^*} - (I - \alpha P_{u^*})\Phi \hat{r})^+ - \\ &\quad - (g_{u^*} - (I - \alpha P_{u^*})\Phi \hat{r})^- + \\ &\quad + 2(g_{u^*} - (I - \alpha P_{u^*})\Phi \hat{r})^-] \\ &= c^T (I - \alpha P_{u^*})^{-1} [g_{u^*} - (I - \alpha P_{u^*})\Phi \hat{r} + \\ &\quad + 2(T_{u^*} \Phi \hat{r} - \Phi \hat{r})^-] \\ &= c^T (J^* - \Phi \hat{r}) + \frac{2\mu_{u^*}^T}{1 - \alpha} (T_{u^*} \Phi \hat{r} - \Phi \hat{r})^-. \end{aligned} \quad (10)$$

Now let $\tilde{r}$ be any optimal solution of the ALP¹. Clearly, $\tilde{r}$ is feasible for the RLP. Since $\hat{r}$ is the optimal solution of the same problem, we have $c^T \Phi \hat{r} \geq c^T \Phi \tilde{r}$ and

$$\begin{aligned} c^T (J^* - \Phi \hat{r}) &\leq c^T (J^* - \Phi \tilde{r}) \\ &= \|J^* - \Phi \tilde{r}\|_{1,c}, \end{aligned} \quad (11)$$

therefore we just need to show that the second term in (10) is small to guarantee that the performance of the RLP is not much worse than that of the ALP.

Now

$$\begin{aligned} &\frac{2\mu_{u^*}^T}{1 - \alpha} (T_{u^*} \Phi \hat{r} - \Phi \hat{r})^- \\ &= \frac{2}{1 - \alpha} \sum_{x \in \mathcal{S}} \mu_{u^*}(x) ((T_{u^*} \Phi \hat{r})(x) - (\Phi \hat{r})(x)) 1_{(T_{u^*} \Phi \hat{r})(x) < (\Phi \hat{r})(x)} \\ &= \frac{2}{1 - \alpha} \sum_{x \in \mathcal{S}} \frac{(T_{u^*} \Phi \hat{r})(x) - (\Phi \hat{r})(x)}{V(x)} \mu_{u^*}(x) V(x) 1_{(T_{u^*} \Phi \hat{r})(x) < (\Phi \hat{r})(x)} \\ &\leq \frac{2\mu_{u^*}^T V}{1 - \alpha} \|T_{u^*} \Phi \hat{r} - \Phi \hat{r}\|_{\infty, 1/V} \sum_{x \in \mathcal{S}} \mu_{u^*, V}(x) 1_{(T_{u^*} \Phi \hat{r})(x) < (\Phi \hat{r})(x)} \\ &\leq \frac{\varepsilon}{2\theta} \mu_{u^*}^T V \|T_{u^*} \Phi \hat{r} - \Phi \hat{r}\|_{\infty, 1/V} \\ &\leq \frac{\varepsilon}{2\theta} \mu_{u^*}^T V (\|T_{u^*} \Phi \hat{r} - J^*\|_{\infty, 1/V} + \|J^* - \Phi \hat{r}\|_{\infty, 1/V}) \\ &\leq \frac{\varepsilon}{2\theta} \mu_{u^*}^T V (1 + \beta_V) \|J^* - \Phi \hat{r}\|_{\infty, 1/V} \\ &\leq \varepsilon, \end{aligned}$$

with probability greater than or equal to $1 - \delta$, where second inequality follows from (9) and the fourth inequality follows from Lemma 3.1. The error bound (8) then follows from (10) and (11). $\square$

Three aspects of Theorem 3.1 deserve further consideration. The first of them is the dependence of the number of sampled constraints (7) on θ . Two parameters of the RLP

¹Note that all optimal solutions of the ALP yield the same approximation error $\|J^* - \Phi r\|_{1,c}$, hence the error bound (8) is independent of the choice of $\tilde{r}$.

influence the behavior of θ : the Lyapunov function V and the bounding set $\mathcal{N}$. Graceful scaling of the sample complexity bound depends on the ability to make appropriate choices for these parameters. In Section IV, we demonstrate how, for a wide class of queueing network problems, V and $\mathcal{N}$ can be chosen so as to ensure that the number of sampled constraints grows quadratically in the system dimension.

The number of sampled constraints also grows polynomially with the maximum number of actions available per state A , which makes the proposed approach inapplicable to problems with a large number of actions per state. In our full-length paper, we show that complexity in the action space can be exchanged for complexity in the state space, so that such problems can be recast in a format that is amenable to our approach.

Finally, a major weakness of Theorem 3.1 is that it relies on sampling constraints according to the distribution $\psi_{u^*,V}$. In general, $\psi_{u^*,V}$ is not known, and constraints must be sampled according to an alternative distribution $\bar{\psi}$. Suppose that $\bar{\psi}(x,a) = \bar{\mu}(x)/|\mathcal{A}_x|$ for some state distribution $\bar{\mu}$. If $\bar{\mu}$ is “similar” to $\mu_{u^*,V}$, one might hope that the error bound (8) holds with a number of samples m close to the number suggested in the theorem. We discuss two possible motivations for this:

1) It is conceivable that sampling constraints according to $\bar{\psi}$ leads to a small value of

$$\mu_{u^*,V}(\{x : (\Phi\hat{r})(x) \geq (T_{u^*}\Phi\hat{r})(x)\}) \leq (1-\alpha)\varepsilon/2,$$

with high probability, even though $\mu_{u^*,V}$ is not identical to $\bar{\mu}$. This would lead to a graceful sample complexity bound, along the lines of (7). Establishing such a guarantee is closely related to the problem of UCEP in which the training and testing distributions differ. This is a well-known but not so well-understood problem in machine learning (see, e.g., [14]).

2) If $\mu_{u^*}^T(T_{u^*}\Phi r - \Phi r)^- \leq C\tilde{\mu}^T(T_{u^*}\Phi r - \Phi r)^-$, for some scalar C and all r , where $\tilde{\mu}(x) = \frac{\bar{\mu}(x)/V(x)}{\sum_{y \in S} \bar{\mu}(y)/V(y)}$, then the error bound (8) holds with probability $1 - \delta$ given $\frac{512A^2\theta^2C^2}{(1-\alpha)^2\varepsilon^2} \left[\ln 8 + \ln \frac{1}{\delta} + K \left(\ln \frac{64A\theta C\varepsilon}{(1-\alpha)\varepsilon} + \ln \ln \frac{64A\theta C\varepsilon}{(1-\alpha)\varepsilon} \right) \right]$ samples. It is conceivable that this will be true for a reasonably small value of C in relevant contexts.

How to choose $\bar{\mu}$ is an open question, and most likely to be addressed adequately having in mind the particular application at hand. As a simple heuristic, noting that $\mu_{u^*}(x) \rightarrow c(x)$ as $\alpha \rightarrow 0$, one might choose $\bar{\mu}(x) = c(x)V(x)$. This is the approach taken in the examples presented in [3].

IV. EXAMPLE: CONTROLLED QUEUEING NETWORKS

In order for the error bound (8) to be useful, the parameter θ should scale gracefully with problem size. We anticipate that for many relevant classes of MDPs, natural choices of V and $\mathcal{N}$ will ensure that this term scales gracefully. In this section, we illustrate this point through an example involving controlled queueing networks. The key result is Theorem 4.1,

which offers a sample complexity bound specialized to our formulation of controlled queueing networks and choices of Φ , $\mathcal{N}$, and V . This bound exhibits a graceful dependence on the number of queues in the network.

We begin by describing the class of problems we will address. Consider a queueing network with d queues, each with a finite buffer of size B . The state space is given by $S = \{0, \dots, B\}^d$, with each component x_i of each state $x \in S$ representing the number of jobs in queue i . The cost per stage is the average queue length: $g(x) = (1/d) \sum_{i=1}^d x_i$. Rewards are discounted by a factor of $\alpha \in (0, \bar{\alpha})$ per time step, for some $\bar{\alpha} \in (0, 1)$. At each time step, an action $a \in \mathcal{A}_x$ is selected. Transition probabilities $P_a(x, y)$ govern how jobs arrive, move from queue to queue, or leave the network. We assume that the number of exogenous arrivals at each time step is less than or equal to $\bar{\lambda}d$. The class of problems we consider is constrained by $\bar{\alpha}$ and $\bar{\lambda}$. For each $\bar{\alpha} \in (0, 1)$ and $\bar{\lambda} \in (0, \infty)$ we denote the associated class of problems by $\mathcal{Q}(\bar{\alpha}, \bar{\lambda})$.

Consider a problem instance $Q \in \mathcal{Q}(\bar{\alpha}, \bar{\lambda})$ and its associated parameters $B, d, \alpha, \mathcal{A}(\cdot)$, and $P(\cdot, \cdot)$. Let u_Q^* and J_Q^* denote an optimal policy and the optimal cost-to-go function for a problem instance Q . We have the following upper bound on J_Q^* .

$$\text{Lemma 4.1: } J_Q^*(x) \leq \frac{1}{d(1-\alpha)} \sum_{i=1}^d x_i + \frac{\lambda}{(1-\alpha)^2}.$$

We consider approximating J_Q^* via fitting a linear combination of basis functions $\phi_k(x) = x_k, k = 1, \dots, d$ and $\phi_{d+1}(x) = 1$ using an ALP:

$$\begin{aligned} \max \quad & \sum_{x \in S} c_Q(x) \left(\sum_{k=1}^{d+1} r_k x_k + r_{d+1} \right) \\ \text{s.t.} \quad & \frac{1}{d} \sum_{i=1}^d x_i + \alpha \sum_{y \in \mathcal{S}} P_a(x, y) \left(\sum_{k=1}^d r_k y_k + r_{d+1} \right) \\ & \geq \sum_{k=1}^d r_k x_k + r_{d+1}, \quad \forall x \in \mathcal{S}, a \in \mathcal{A}_x, \end{aligned} \quad (12)$$

with state-relevance weights

$$c_Q(x) = (1-\xi)^d \xi^{-\sum_{i=1}^d x_i},$$

for some $\xi \in (0, 1)$.

The number of constraints imposed by the ALP (12) grows exponentially with the number of queues d . For even a moderate number of queues (e.g., ten), the number of constraints is unmanageable. Constraint sampling offers an approach to alleviating this computational burden. To formulate an RLP, given a problem instance Q , we must define a constraint set $\mathcal{N}_Q$ and a sampling distribution ψ . We take $\mathcal{N}_Q$ to be the set of vectors $r \in \mathbb{R}^{d+1}$ that satisfy the following linear constraints:

$$\begin{aligned} r_{d+1} &\leq \frac{\lambda}{(1-\alpha)^2}, \\ r_k + \frac{r_{d+1}}{B} &\leq \frac{1}{(1-\alpha)d} + \frac{\lambda}{B(1-\alpha)^2} \quad \forall k = 1, \dots, d, \\ \left(\frac{\xi(1-\xi^{B+1})}{1-\xi} - \xi^{B+1}(B+1) \right) \sum_{k=1}^d r_k + r_{d+1} &\geq 0. \end{aligned}$$

Note that the resulting RLP is a linear program. A desirable quality of $\mathcal{N}_Q$ is that it contains optimal solutions to the ALP (12), as asserted by the following lemma.

Lemma 4.2: For all $\bar{\alpha} \in (0, 1)$ and $\bar{\lambda} \in (0, \infty)$ and all $Q \in \mathcal{Q}(\bar{\alpha}, \bar{\lambda})$, each optimal solution $\tilde{r}_Q$ to the ALP (12) is in $\mathcal{N}_Q$.

Another desirable quality of $\mathcal{N}_Q$ is that it is a bounded set, if the buffer size B is large enough.

Lemma 4.3: Suppose that $B > 2d\xi/(1-\xi)$. Then we have

$$\sup_{r \in \mathcal{N}_Q} \|r\|_\infty \leq \frac{2[1 + \xi/(1-\xi)] + \lambda(1-\xi)/\xi + 2\lambda}{(1-\alpha)^2}.$$

As the sampling distribution, we use $\psi_{u_Q^*, V_Q}$, where

$$V_Q(x) = \frac{1}{d(1-\alpha)} \sum_{i=1}^d x_i + \frac{2\lambda}{(1-\alpha)^2}. \quad (13)$$

Lemma 4.4: For any $\bar{\alpha} \in (0, 1)$ and $\bar{\lambda} \in (0, \infty)$ and any $Q \in \mathcal{Q}(\bar{\alpha}, \bar{\lambda})$, V_Q is a Lyapunov function.

Our next lemma establishes that, given our choices of N_Q and V_Q , $\sup_{r \in \mathcal{N}_Q} \|J_Q^* - \Phi r\|_{\infty, 1/V_Q}$ scales linearly with the number of queues.

Lemma 4.5: Consider a set of problems $\mathcal{Q}(\bar{\alpha}, \bar{\lambda})$ for some $\bar{\alpha} \in (0, 1)$ and $\bar{\lambda} \in (0, \infty)$. Let d be the number of queues for a problem instance $Q \in \mathcal{Q}(\bar{\alpha}, \bar{\lambda})$. Suppose that $B > d\xi/(1-\xi)$ and $2\bar{\lambda}d > 1 - \bar{\alpha}$. Then,

$$\sup_{r \in \mathcal{N}_Q} \|J_Q^* - \Phi r\|_{\infty, 1/V_Q} \leq 1 + \left[2 + \frac{2\xi}{1-\xi} + \bar{\lambda} \frac{1-\xi}{\xi} + 2\bar{\lambda} \right] \frac{d}{1-\bar{\alpha}}.$$

Given the preceding lemmas, it is easy to establish the following result, which specializes Theorem 3.1 to our queueing context. We will make use of the constant

$$\bar{\theta} = \left[2 + \frac{2\xi}{1-\xi} + \bar{\lambda} \frac{1-\xi}{\xi} + 2\bar{\lambda} \right] \left(1 + \frac{2\bar{\lambda}}{(1-\bar{\alpha})c^T J_Q^*} \right).$$

Theorem 4.1: Consider a set of problems $\mathcal{Q}(\bar{\alpha}, \bar{\lambda})$ for some $\bar{\alpha} \in (0, 1)$ and $\bar{\lambda} \in (0, \infty)$. Let d be the number of queues for a problem instance $Q \in \mathcal{Q}(\bar{\alpha}, \bar{\lambda})$. Let $\mathcal{X}$ be a (random) set of m state-action pairs sampled independently according to the distribution $\psi_{u_Q^*, V_Q}(x, a)$, where

$$m \geq \frac{512A^2\bar{\theta}^2 d^2}{(1-\alpha)^2 \varepsilon^2} \left[\ln \frac{8}{\delta} + (d+1) \ln \left(\frac{64A\bar{\theta}de}{(1-\alpha)\varepsilon} \ln \frac{64A\bar{\theta}de}{(1-\alpha)\varepsilon} \right) \right].$$

Let $\tilde{r}$ be an optimal solution of the ALP (12) and $\hat{r}$ be an optimal solution of the corresponding RLP. Then, with probability at least $1 - \delta$, we have

$$\|J_Q^* - \Phi \hat{r}\|_{1,c} \leq \|J_Q^* - \Phi \tilde{r}\|_{1,c} + \varepsilon c^T J_Q^*.$$

ACKNOWLEDGEMENTS

This research was supported by NSF CAREER Grant ECS-9985229, by the ONR under Grant MURI N00014-00-1-0637, and by an IBM Research Fellowship.

V. REFERENCES

- [1] D. Bertsekas and J.N. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, 1996.
- [2] G. Calafiore and M. Campi. Personal communication, 2003.
- [3] D.P. de Farias and B. Van Roy. The linear programming approach to approximate dynamic programming. To appear in *Operations Research*, 2001.
- [4] R.M. Dudley. *Uniform Central Limit Theorems*. Cambridge University Press, Cambridge, 1998.
- [5] M. Grötschel and O. Holland. Solution of large-scale symmetric travelling salesman problems. *Mathematical Programming*, 51:141–202, 1991.
- [6] C. Guestrin, D. Koller, and R. Parr. Efficient solution algorithms for factored MDPs. To appear in *Journal of Artificial Intelligence Research*, 2002.
- [7] D. Haussler, M. Kearns, N. Littlestone, and M. K. Warmuth. Equivalence of models for polynomial learnability. *Information and Computation*, 95(2):129–161, 1991.
- [8] K. L. Clarkson. Las vegas algorithms for linear and integer programming when the dimension is small. *Journal of the ACM*, 42(2):488–499, 1995.
- [9] J.R. Morrison and P.R. Kumar. New linear program performance bounds for queueing networks. *Journal of Optimization Theory and Applications*, 100(3):575–597, 1999.
- [10] D. Schuurmans and R. Patrascu. Direct value-approximation for factored MDPs. In *Advances in Neural Information Processing Systems*, volume 14, 2001.
- [11] P. Schweitzer and A. Seidmann. Generalized polynomial approximations in Markovian decision processes. *Journal of Mathematical Analysis and Applications*, 110:568–582, 1985.
- [12] M. Trick and S. Zin. A linear programming approach to solving dynamic programs. Unpublished manuscript, 1993.
- [13] M. Trick and S. Zin. Spline approximations to value functions: A linear programming approach. *Macroeconomic Dynamics*, 1, 1997.
- [14] M. Vidyasagar. *A Theory of Learning and Generalization*. Springer, London, 1997.