MATHEMATICS OF OPERATIONS RESEARCH
Vol. 29, No. 3, August 2004, pp. 462–478
ISSN 0364-765X | EISSN 1526-5471 | 04 | 2903 | 0462

DOI 10.1287/moor.1040.0094
© 2004 INFORMS

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

Daniela Pucci de Farias
Room 3-354, 77 Massachusetts Avenue, Massachusetts Institute of Technology,
Cambridge, Massachussetts 02139, pucci@mit.edu, www.mit.edu/~pucci

Benjamin Van Roy
Terman 315, Stanford University, Stanford, California 93405, bvr@stanford.edu, www.stanford.edu/~bur

In the linear programming approach to approximate dynamic programming, one tries to solve a certain linear program—the ALP—that 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, given an idealized sampling distribution and appropriate constraints on the K variables, m can be chosen independently of M and need grow only as a polynomial in K. We interpret this result in a context involving controlled queueing networks.

Key words: Markov decision processes; policy; value function; basis functions
MSC2000 subject classification: Primary: 90C39, 90C06, 90C08
OR/MS subject classification: Primary: Dynamic programming/optimal control; secondary: Markov, finite state
History: Received August 24, 2001; revised July 31, 2002, and July 28, 2003.

1. 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 (de Farias and Van Roy 2003; Schweitzer and Seidmann 1985; Trick and Zin 1993, 1997). 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. We begin in this section by discussing our working problem formulation, the linear programming approach, constraint sampling, results of our analysis, and related literature.

1.1. Markov decision processes. We consider a Markov decision process (MDP) with a finite state space $\mathcal{S} = \{1, \ldots, |\mathcal{S}|\}$. In each state $x \in \mathcal{S}$, there is a finite set of admissible actions $\mathcal{A}_x$. Further, 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 admissible 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 xyth component is equal to $P_{u(x)}(x, y)$, and g_u is a vector whose xth 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 $T_u J = g_u + \alpha P_u J$. Furthermore, the optimal cost-to-go function $J^* = \min_u J_u$ is the unique solution to Bellman's equation: $J^* = TJ^*$, where the operator T is defined by $TJ = \min_u T_u J$. Note that the minimization here is carried out componentwise. For any vector J, we call a policy u greedy with respect to J if $TJ = T_u J$. Any policy that is greedy with respect to the optimal cost-to-go function J^* is optimal.

1.2. 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 $\phi_k\colon \mathscr{S} \mapsto \Re$, $k = 1, \ldots, K$. The aim is to generate a weight vector $\tilde{r} \in \Re^K$ 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

$$\Phi = \begin{bmatrix} | & & | \\ \phi_1 & \cdots & \phi_K \\ | & & | \end{bmatrix}.$$

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

$$\begin{aligned} (1) \qquad & \text{maximize} \quad c^T \Phi r \\ & \text{subject to} \quad g_a(x) + \alpha \sum_{y \in \mathscr{S}} P_a(x, y)(\Phi r)(y) \geq (\Phi r)(x), \quad \forall\, x \in \mathscr{S},\ a \in \mathscr{A}_x, \end{aligned}$$

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 de Farias and Van Roy (2003), the ALP minimizes $\|J^* - \Phi r\|_{1,c}$, subject to the constraints. For any positive vector ν, we define the weighted L_1 and L_∞ norms

$$\|J\|_{1,\nu} = \sum_x \nu(x)|J(x)|, \qquad \|J\|_{\infty,\nu} = \max_x \nu(x)|J(x)|.$$

Further, de Farias and Van Roy (2003) discuss the role of state-relevance weights c and why, given an appropriate choice of c, minimization of $\|J^* - \Phi r\|_{1,c}$ is desirable.

1.3. 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 $\mathscr{N} \subseteq \Re^K$. The probability measure ψ represents a distribution from which we will sample constraints. In particular, we consider a set $\mathscr{X}$ of m state-action pairs, each independently sampled according to ψ. The set $\mathscr{N}$ 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

$$\begin{aligned} (2) \qquad & \text{maximize} \quad c^T \Phi r \\ & \text{subject to} \quad g_a(x) + \alpha \sum_{y \in \mathscr{S}} P_a(x, y)(\Phi r)(y) \geq (\Phi r)(x), \quad \forall\, (x, a) \in \mathscr{X}, \\ & \qquad\qquad\quad r \in \mathscr{N}. \end{aligned}$$

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

$$\Pr\{|\|J^* - \Phi\hat{r}\|_{1,c} - \|J^* - \Phi\tilde{r}\|_{1,c}| \leq \epsilon\} \geq 1 - \delta,$$

where $\epsilon > 0$ is an error tolerance parameter and $\delta > 0$ parameterizes a level of confidence $1 - \delta$. This paper focusses on understanding the sample size m needed in order to meet such a requirement.

1.4. 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, $1/\epsilon$, and $\log(1/\delta)$, and is independent of the total number of ALP constraints.

Our analysis is carried out in two parts:

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

$$m = O\left(\frac{1}{\epsilon}\left(K \ln \frac{1}{\epsilon} + \ln \frac{1}{\delta}\right)\right),$$

such that, with probability at least $1 - \delta$, there exists a subset $Z \subseteq \mathcal{Z}$ of measure $\psi(Z) \geq 1 - \epsilon$ such that every vector r satisfying

$$\gamma_{z_i}^T r + \kappa_{z_i} \geq 0, \quad \forall i = 1, \ldots, m,$$

also satisfies

$$\gamma_{z_i}^T r + \kappa_{z_i} \geq 0, \quad \forall z \in Z.$$

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.

(ii) *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

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

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

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

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, one can sometimes provide much tighter bounds or even methods for exact solutions of the ALP, and results of this nature can be found in the literature, as discussed in the following literature review. 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.

1.5. 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.

We begin by reviewing work from the first category. Morrison and Kumar (1999) formulate approximate linear programming algorithms for queueing problems with a specific choice of basis functions that renders all but a relatively small number of constraints redundant. Guestrin et al. (2003) exploit the structure arising when *factored* linear architectures are used for approximating the cost-to-go function in *factored MDPs*. In some special cases, this allows for efficient exact solution of the ALP, and in others, this motivates alternative approximate solution methods. Schuurmans and Patrascu (2002) devise a constraint generation scheme, also especially designed for factored MDPs with factored linear architectures. The worst-case computation time of this scheme grows exponentially with the number of state variables, even for special cases treated effectively by the methods of Guestrin et al. (2003). However, the proposed scheme requires a smaller amount of computation time, on average. Grötschel and Holland (1991) present a cutting-plane method tailored for the travelling salesman problem.

As for the second category, Trick and Zin (1993, 1997) study several constraint generation heuristics for the ALP. They apply these heuristics to solve fairly large problems, involving thousands of variables and millions of constraints. This work demonstrates promise for constraint generation methods in the context of the ALP. However, it is not clear how the proposed heuristics can be applied to larger problems involving intractable numbers of constraints.

Clarkson's Las Vegas algorithm (1995) is another general-purpose constraint sampling scheme for linear programs with large numbers of constraints. Las Vegas is a constraint generation algorithm where constraints are iteratively selected by a method designed to identify binding constraints. There are a couple of important differences between Las Vegas and our constraint sampling scheme. First, Las Vegas is an exact algorithm, meaning that it produces the optimal solution, whereas all that can be proved about the RLP is that it produces a good approximation to an optimal solution of the ALP with high probability. Second, Las Vegas's expected run time is polynomial in the number of constraints, whereas the RLP entails run time that is independent of the number of constraints. Given the prohibitive number of constraints in the ALP, Las Vegas will generally not be applicable.

Finally, we note that, after the original version of this paper was submitted for publication, Calafiore and Campi (2003) 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 a single optimal solution to an analog of the RLP, rather than uniform near-feasibility over all feasible solutions to an RLP. If their result is applied to a linear program as discussed in Item (i) of §1.4, their sample complexity bound is $O(K/\epsilon\delta)$. Depending on the desired confidence $1-\delta$, their sample complexity bound of $O(K/\epsilon\delta)$ can be greater than or less than the bound of $O((1/\epsilon)(K\ln 1/\epsilon + \ln 1/\delta))$ that we use. The bounds may be reconciled by the opportunity to "boost confidence" (Haussler et al. 1991). In particular, loosely speaking, in the design of learning algorithms a factor of $\ln 1/\epsilon$ can be traded for a multiple of $1/\delta$. Interestingly, this suggests that the bound of Calafiore and Campi (2003), which ensures near-feasibility of a single optimal solution to the RLP is equivalent—up to a constant factor—to a bound that ensures near-feasibility of all feasible solutions. This observation does not extend, however, to the case of general convex programs.

1.6. Organization of the paper. The remainder of the paper is organized as follows. In §2, we establish a bound on the number of constraints to be sampled so that the RLP generates a near-feasible solution. In §3, we extend the analysis to establish a bound on the number of constraints to be sampled so that the RLP generates a solution that closely approximates an optimal solution to the ALP. To facilitate understanding of this bound, we study in §4 properties of the bound in a context involving controlled queueing networks. Our bounds require sampling a number of constraints that grows polynomially on the number of actions per state, which may present difficulties for problems with large action spaces. We propose an approach for dealing with large action spaces in §5. Section 6 concludes the paper.

2. Sample complexity of near-feasibility. Consider a set of linear constraints

$$\gamma_z^T r + \kappa_z \geq 0, \quad \forall z \in \mathcal{Z}, \tag{3}$$

where $r \in \Re^K$ and $\mathcal{Z}$ is a set of constraint indices. We make the following assumption on the set of constraints:

ASSUMPTION 2.1. *There exists a vector $r \in \Re^K$ that satisfies the system of inequalities* (3).

We are interested in situations where there are relatively few variables and a possibly huge finite or infinite number of constraints; i.e., $K \ll |\mathcal{Z}|$. In such a situation, we expect that almost all the constraints will be irrelevant, either because they are always inactive or

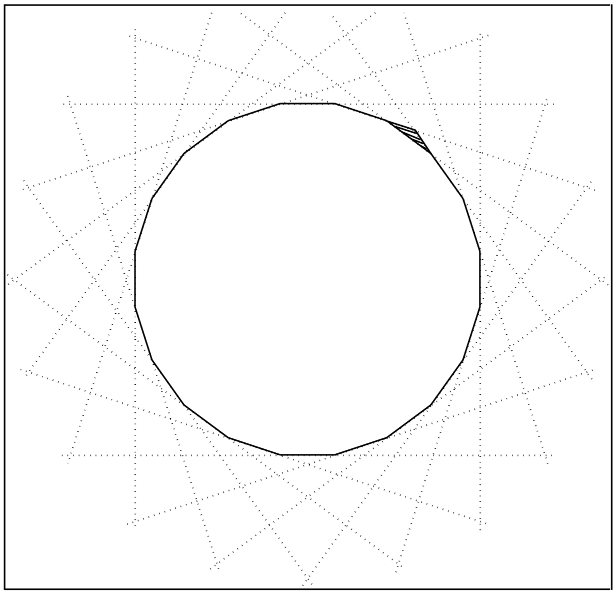

FIGURE 1. Large number of constraints in a low-dimensional feasible space. No constraint can be removed without affecting the feasible region. Shaded area demonstrates the impact of not satisfying one constraint on the feasible region.

because they have a minor impact on the feasible region. Therefore, one might speculate that the feasible region specified by all constraints can be closely 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 on a probability measure ψ over $\mathcal{X}$. The distribution ψ will have a dual role in our approximation scheme: On the 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.

In general, we cannot guarantee that all constraints will be satisfied over the feasible region of any subset of constraints. Figure 1, for instance, illustrates a worst-case scenario in which it is necessary to include all constraints to ensure that all of them are satisfied. Note, however, that the impact of any one of them on the feasible region is minor and might be considered negligible. In this spirit, 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 $\epsilon \in (0,1)$, we want to have $\mathcal{W} \subseteq \mathcal{X}$ satisfying

$$\sup_{\{r \mid \gamma_z^T r + \kappa_z \geq 0, \ \forall z \in \mathcal{W}\}} \psi(\{y \colon \gamma_y^T r + \kappa_y < 0\}) \leq \epsilon. \tag{4}$$

Whenever (4) 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 $\mathcal{W}$ leads to near-feasibility with probability at least $1-\delta$.

THEOREM 2.1. *For any* $\delta \in (0,1)$ *and* $\epsilon \in (0,1)$, *and*

$$m \geq \frac{4}{\epsilon}\left(K \ln \frac{12}{\epsilon} + \ln \frac{2}{\delta}\right), \tag{5}$$

a set $\mathcal{W}$ of m i.i.d. random variables drawn from $\mathcal{Z}$ according to distribution ψ, satisfies

$$\sup_{\{r:\, \gamma_z^T r + \kappa_z \geq 0,\ \forall z \in \mathcal{W}\}} \psi(\{y\colon \gamma_y^T r + \kappa_y < 0\}) \leq \epsilon \tag{6}$$

with probability at least $1-\delta$.

This theorem implies that even without any special knowledge about the constraints, we can ensure near-feasibility, with high probability, through imposing a tractable subset of constraints. The result follows immediately from Corollary 8.4.2 in Anthony and Biggs (1992) and the fact that the collection of sets $\{\{(\gamma,\kappa)|\gamma^T r + \kappa \geq 0\}|r \in \Re^K\}$ has VC-dimension K, as established in Dudley (1978).

Theorem 2.1 may be perceived as a puzzling result: The number of sampled constraints necessary for a good approximation of a set of constraints indexed by $z \in \mathcal{Z}$ depends only on the number of variables involved in these constraints and not on the set $\mathcal{Z}$. Some geometric intuition can be derived as follows. The constraints are fully characterized by vectors $[\gamma_z^T \kappa_z]$ of dimension equal to the number of variables plus one. Because near-feasibility involves only consideration of whether constraints are violated, and not the magnitude of violations, we may assume without loss of generality that $\|[\gamma_z^T \kappa_z]\| = 1$, for an arbitrary norm. Hence, constraints can be thought of as vectors in a low-dimensional unit sphere. After a large number of constraints are sampled, they are likely to form a *cover* for the original set of constraints—i.e., any other constraint is close to one of the already-sampled ones, so that the sampled constraints *cover* the set of constraints. The number of sampled constraints necessary in order to have a cover for the original set of constraints is bounded above by the number of sampled vectors necessary to form a cover to the unit sphere, which naturally depends only on the dimension of the sphere or, alternatively, on the number of variables involved in the constraints.

3. 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. Indeed, it is easy to see that such a result cannot hold for general linear programs. For instance, consider a linear program with two variables, which are to be selected from the feasible region illustrated in Figure 2. If we remove all but a small random sample of the constraints, the new solution to the linear program is likely to be far from the solution to the original linear program. In fact, one can construct examples where the solution to a linear program is changed by an arbitrary amount by relaxing just one constraint.

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

$$\mu_u^T = (1-\alpha)c^T(I - \alpha P_u)^{-1}, \tag{7}$$

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* $V\colon \mathcal{S} \mapsto \Re^+$. Given a function V, we define a scalar

$$\beta_V = \max_{x\in\mathcal{S}} \frac{\alpha(P_{u^*}V)(x)}{V(x)}.$$

If $\beta_V < 1$, the Lyapunov function V would satisfy a "downward drift" condition. However, we will make no such requirement—β_V could potentially be larger than 1.

The following lemma captures a useful property of T_{u^*}.

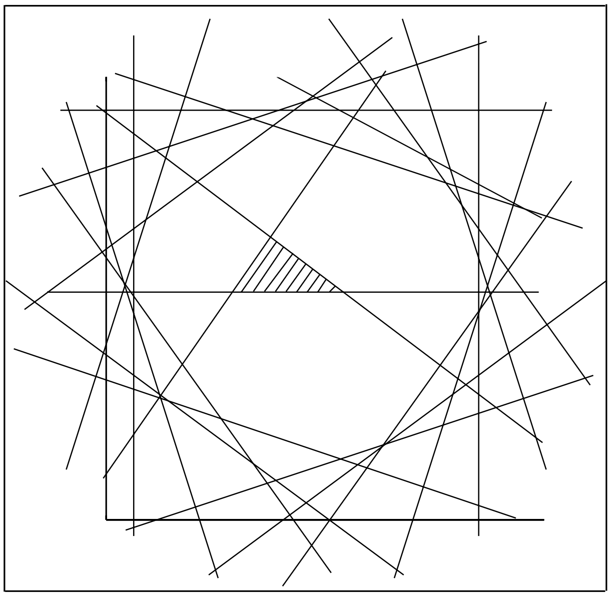

FIGURE 2. A feasible region defined by a large number of redundant constraints. Removing all but a random sample of constraints is likely to bring about a significant change in the solution of the associated linear program.

LEMMA 3.1. *Let V be a Lyapunov function for an optimal policy u^*. Then*

$$\|T_{u^*}J - T_{u^*}\overline{J}\|_{\infty,1/V} \le \beta_V \|J - \overline{J}\|_{\infty,1/V}.$$

PROOF. Let J and $\overline{J}$ be two arbitrary vectors in $\Re^{|\mathcal{S}|}$. Then

$$T_{u^*}J - T_{u^*}\overline{J} = \alpha P_{u^*}(J - \overline{J}) \le \|J - \overline{J}\|_{\infty,1/V}\, \alpha P_{u^*} V \le \|J - \overline{J}\|_{\infty,1/V}\, \beta_V V. \quad \square$$

For each Lyapunov function V, we define a probability distribution on the state space $\mathcal{S}$, given by

$$\mu_{u,V}(x) = \frac{\mu_u(x)V(x)}{\mu_u^T V}. \tag{8}$$

We also define a distribution over state-action pairs

$$\psi_{u,V}(x,a) = \frac{\mu_{u,V}(x)}{|\mathcal{A}_x|}, \quad \forall\, a \in \mathcal{A}_x.$$

Finally, we define constants

$$A = \max_x |\mathcal{A}_x|$$

and

$$\theta = \frac{1+\beta_V}{2} \frac{\mu_{u^*}^T V}{c^T J^*} \sup_{r \in \mathcal{N}} \|J^* - \Phi r\|_{\infty,1/V}. \tag{9}$$

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

THEOREM 3.1. *Let ϵ and δ be scalars in $(0,1)$. 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*

$$m \ge \frac{16A\theta}{(1-\alpha)\epsilon}\left(K \ln \frac{48A\theta}{(1-\alpha)\epsilon} + \ln \frac{2}{\delta}\right). \tag{10}$$

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

$$\|J^* - \Phi\hat{r}\|_{1,c} \leq \|J^* - \Phi\tilde{r}\|_{1,c} + \epsilon\|J^*\|_{1,c}. \tag{11}$$

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

$$\begin{aligned} \frac{(1-\alpha)\epsilon}{4A\theta} &\geq \psi_{u^*,V}(\{(x,a)\colon (T_a\Phi\hat{r})(x) < (\Phi\hat{r})(x)\}) \\ &= \sum_{x\in S}\frac{\mu_{u^*,V}(x)}{|\mathcal{A}_x|}\sum_{a\in\mathcal{A}_x} 1_{(T_a\Phi\hat{r})(x)<(\Phi\hat{r})(x)} \\ &\geq \frac{1}{A}\sum_{x\in S}\mu_{u^*,V}(x)1_{(T_{u^*}\Phi\hat{r})(x)<(\Phi\hat{r})(x)}. \end{aligned} \tag{12}$$

For any vector J, we denote the positive and negative parts by

$$J^+ = \max(J,0), \qquad J^- = \max(-J,0),$$

where the maximization is carried out componentwise. Note that

$$\begin{aligned} \|J^* - \Phi\hat{r}\|_{1,c} &= c^T|(I-\alpha P_{u^*})^{-1}(g_{u^*} - (I-\alpha P_{u^*})\Phi\hat{r})| \\ &\leq c^T(I-\alpha P_{u^*})^{-1}|g_{u^*} - (I-\alpha P_{u^*})\Phi\hat{r}| \\ &= c^T(I-\alpha P_{u^*})^{-1}[(g_{u^*} - (I-\alpha P_{u^*})\Phi\hat{r})^+ + (g_{u^*} - (I-\alpha P_{u^*})\Phi\hat{r})^-] \\ &= c^T(I-\alpha P_{u^*})^{-1}[(g_{u^*} - (I-\alpha P_{u^*})\Phi\hat{r})^+ - (g_{u^*} - (I-\alpha P_{u^*})\Phi\hat{r})^- \\ &\qquad + 2(g_{u^*} - (I-\alpha P_{u^*})\Phi\hat{r})^-] \\ &= c^T(I-\alpha P_{u^*})^{-1}[g_{u^*} - (I-\alpha P_{u^*})\Phi\hat{r} + 2(T_{u^*}\Phi\hat{r} - \Phi\hat{r})^-] \\ &= c^T(J^* - \Phi\hat{r}) + 2c^T(I-\alpha P_{u^*})^{-1}(T_{u^*}\Phi\hat{r} - \Phi\hat{r})^-. \end{aligned} \tag{13}$$

The inequality comes from the fact that $c > 0$ and

$$(I-\alpha P_{u^*})^{-1} = \sum_{n=0}^{\infty}\alpha^n P_{u^*}^n \geq 0,$$

where the inequality is componentwise, so that

$$\begin{aligned} |(I-\alpha P_{u^*})^{-1}(g_{u^*} - (I-\alpha P_{u^*})\Phi\hat{r})| &\leq |(I-\alpha P_{u^*})^{-1}|\,|(g_{u^*} - (I-\alpha P_{u^*})\Phi\hat{r})| \\ &= (I-\alpha P_{u^*})^{-1}|(g_{u^*} - (I-\alpha P_{u^*})\Phi\hat{r})|. \end{aligned}$$

Now let $\tilde{r}$ be any optimal solution of the ALP. (Note that all optimal solutions of the ALP yield the same approximation error $\|J^* - \Phi r\|_{1,c}$; hence, the error bound (11) is independent of the choice of $\tilde{r}$.) 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

$$\begin{aligned} c^T(J^* - \Phi\hat{r}) &\leq c^T(J^* - \Phi\tilde{r}) \\ &= \|J^* - \Phi\tilde{r}\|_{1,c}; \end{aligned} \tag{14}$$

therefore we just need to show that the second term in (13) is small to guarantee that the performance of the RLP is not much worse than that of the ALP.

Now

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

with probability greater than or equal to $1-\delta$, where the second inequality follows from (12) and the fourth inequality follows from Lemma 3.1. The error bound (11) then follows from (13) and (14). □

Three aspects of Theorem 3.1 deserve further consideration. The first of them is the dependence of the number of sampled constraints (10) on θ. Two parameters of the RLP 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 §4, we demonstrate how, for a broad 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 §5, we show how 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 (11) holds with a number of samples m close to the number suggested in the theorem. We discuss two possible motivations for this:

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

$$\mu_{u^*,V}(\{x\colon (\Phi\hat{r})(x)\ge(T_{u^*}\Phi\hat{r})(x)\})\le(1-\alpha)\epsilon/2,$$

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 (10). Establishing such a guarantee is related to the problem of computational learning when the training and testing distributions differ.

(ii) If

$$\mu_{u^*}^T(T_{u^*}\Phi r-\Phi r)^-\le C\tilde{\mu}^T(T_{u^*}\Phi r-\Phi r)^-,$$

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 (11) holds with probability $1-\delta$ given

$$m \geq \frac{16A\theta C}{(1-\alpha)\epsilon}\left(K \ln \frac{48A\theta C}{(1-\alpha)\epsilon} + \ln\frac{2}{\delta}\right)$$

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 $\mu_{u^*}(x) \to c(x)$ as $\alpha \to 0$, one might choose $\bar{\mu}(x) = c(x)V(x)/c^T V$.

4. Example: Controlled queueing networks.

In order for the error bound (11) to be useful, the parameter

$$\theta = \frac{1+\beta_V}{2}\frac{\mu_{u^*}^T V}{c^T J^*}\sup_{r\in\mathcal{N}} \|J^* - \Phi r\|_{\infty,1/V}$$

should scale gracefully with problem size. We anticipate that for many relevant classes of MDPs, natural choices of V and $\mathcal{N}$ will ensure this. In this section, we illustrate this point through an example involving controlled queueing networks. The key result is Theorem 4.1, which establishes that—given certain reasonable choices of Φ, $\mathcal{N}$, and V—θ grows at most linearly with the number of queues.

4.1. Problem formulation.

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 $B \geq 2d\xi/(1-\xi)$, for some parameter $\xi \in (0,1)$. The state space is given by $S = \{0,\ldots,B\}^d$, 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 α per time step. 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 λd, for some scalar $\lambda \in (0,\infty)$.

Each class of problems we consider—denoted by $\mathcal{Q}(\xi,\alpha,\lambda)$—is constrained by parameters $\xi \in (0,1)$, $\alpha \in (0,1)$, and $\lambda \in (0,\infty)$. Each problem instance $Q \in \mathcal{Q}(\xi,\alpha,\lambda)$ is identified by a quadruple:

- number of queues $d_Q \geq 1$;
- buffer size $B_Q \geq d_Q\xi/(1-\xi)$;
- action sets $\mathcal{A}_\cdot^Q$;
- transition probabilities $P_\cdot^Q(\cdot,\cdot)$.

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^*.

Lemma 4.1. *For any $\xi \in (0,1)$, $\alpha \in (0,1)$, $\lambda \in (0,\infty)$, and $Q \in \mathcal{Q}(\xi,\alpha,\lambda)$, we have*

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

Proof. The first inequality follows from the fact that $g(x) \leq J_Q^*(x)$. Recall that the expected number of exogenous arrivals in any time step is less than or equal to λd_Q.

Therefore, $(P^Q_{u^*_Q})^t g \leq g + \lambda t$. It follows that

$$\begin{aligned} J^*_Q(x) &= \sum_{t=0}^{\infty} \alpha^t ((P^Q_{u^*_Q})^t g)(x) \\ &\leq \sum_{t=0}^{\infty} \alpha^t (g(x) + \lambda t) = \frac{1}{d_Q(1-\alpha)} \sum_{i=1}^{d_Q} x_i + \frac{\alpha\lambda}{(1-\alpha)^2}. \quad \square \end{aligned}$$

4.2. The ALP and the RLP. We consider approximating J^*_Q via fitting a linear combination of basis functions $\phi^Q_k(x) = x_k$, $k = 1, \ldots, d_Q$ and $\phi^Q_{d_Q+1}(x) = 1$ using an ALP:

$$\begin{aligned} &\text{maximize} \quad \sum_{x \in S} c_Q(x) \left(\sum_{k=1}^{d_Q+1} r_k x_k + r_{d+1} \right) \\ &\text{subject to} \quad \frac{1}{d_Q} \sum_{i=1}^{d_Q} x_i + \alpha \sum_{y \in \mathcal{S}} P^Q_a(x, y) \left(\sum_{k=1}^{d_Q} r_k y_k + r_{d_Q+1} \right) \\ &\qquad\qquad \geq \sum_{k=1}^{d_Q} r_k x_k + r_{d_Q+1}, \quad \forall\, x \in \mathcal{S}_Q, \ a \in \mathcal{A}^Q_x, \end{aligned} \tag{15}$$

where $\mathcal{S}_Q = \{0, \ldots, B_Q\}^{d_Q}$ and the state-relevance weights are given by

$$c_Q(x) = \frac{\xi^{-\sum_{i=1}^{d_Q} x_i}}{\sum_{y \in \mathcal{S}_Q} \xi^{-\sum_{i=1}^{d_Q} y_i}}.$$

The number of constraints imposed by the ALP (15) grows exponentially with the number of queues d_Q. For even a moderate number of queues (e.g., 10), the number of constraints becomes 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_Q. We begin by defining and studying a constraint set. Let $\mathcal{N}_Q$ to be the set of vectors $r \in \Re^{d+1}$ that satisfies the following linear constraints:

$$r_{d_Q+1} \leq \frac{\lambda}{(1-\alpha)^2}; \tag{16}$$

$$B_Q r_k + r_{d_Q+1} \leq \frac{B_Q}{(1-\alpha)d_Q} + \frac{\lambda}{(1-\alpha)^2} \quad \forall\, k = 1, \ldots, d_Q; \tag{17}$$

$$\left(\frac{\xi}{1-\xi} - \frac{\xi^{B_Q+1}(B_Q+1)}{1-\xi^{B_Q+1}} \right) \sum_{k=1}^{d_Q} r_k + r_{d_Q+1} \geq 0. \tag{18}$$

Note that the resulting RLP is a linear program with $m + d_Q + 2$ constraints, where m is the number of sampled ALP constraints.

A desirable quality of $\mathcal{N}_Q$ is that it contains optimal solutions of the ALP (15), as asserted by the following lemma.

Lemma 4.2. *For each* $\xi \in (0, 1)$, $\alpha \in (0, 1)$, $\lambda \in (0, \infty)$, *and each* $Q \in \mathcal{Q}(\xi, \alpha, \lambda)$, $\mathcal{N}_Q$ *contains every optimal solution of the ALP* (15).

Proof. Any feasible solution of the ALP is bounded above by J^*_Q; therefore by Lemma 4.1, we have

$$(\Phi \tilde{r}_Q)(x) \leq \frac{1}{d_Q(1-\alpha)} \sum_{i=1}^{d_Q} x_i + \frac{\alpha\lambda}{(1-\alpha)^2} \tag{19}$$

for all optimal solutions $\tilde{r}_Q$ and all $x \in \mathcal{S}_Q$. By considering the case of $x = 0$, we see that (19) implies (16). Further, by considering the case where $x_k = B$ and $x_i = 0$ for all $i \neq k$, we see that (19) implies (17). Because one-stage costs $g(x)$ are nonnegative, $r = 0$ is a feasible solution to the ALP. It follows that $c_Q^T \Phi^Q \tilde{r}_Q \geq 0$. With our particular choice of c_Q and Φ^Q, this implies (18). □

Another desirable quality of $\mathcal{N}_Q$ is that it is uniformly bounded over $\mathscr{Q}(\xi, \alpha, \lambda)$.

LEMMA 4.3. *For each $\xi \in (0,1)$, $\alpha \in (0,1)$, $\lambda \in (0,\infty)$, there exists a scalar $C_{\xi,\alpha,\lambda}$ such that*

$$\sup_{r \in \mathcal{N}_Q} \|r\|_\infty \leq C_{\xi,\alpha,\lambda}$$

for all $Q \in \mathscr{Q}(\xi, \alpha, \lambda)$.

PROOF. Take an arbitrary $r \in \mathcal{N}_Q$. Constraint (16) provides an upper bound on r_{d+1}. We now derive a lower bound on r_{d_Q+1}. For shorthand, let

$$\kappa = c_Q^T \phi_1^Q = \frac{\xi}{1-\xi} - \frac{\xi^{B_Q+1}(B_Q+1)}{1-\xi^{B_Q+1}}.$$

We then have

$$\begin{aligned}
r_{d+1} &\geq -\kappa \sum_{k=1}^{d_Q} r_k \\
&\geq -\kappa \sum_{k=1}^{d_Q} \left(-\frac{r_{d_Q+1}}{B_Q} + \frac{1}{(1-\alpha)d_Q} + \frac{\lambda}{B_Q(1-\alpha)^2} \right) \\
&= \frac{\kappa d_Q r_{d_Q+1}}{B_Q} - \frac{\kappa}{1-\alpha} - \frac{d_Q \kappa \lambda}{B_Q(1-\alpha)^2} \\
&\geq \frac{\kappa d_Q r_{d_Q+1}}{B_Q} - \frac{\kappa}{1-\alpha} - \frac{\lambda}{(1-\alpha)^2},
\end{aligned}$$

where the first inequality follows from (18), the second one follows from (17), and the final inequality follows from the fact that $B_Q > 2d_Q\kappa$. Gathering the terms involving r_{d_Q+1}, we obtain

$$r_{d+1} \geq -\frac{\kappa/(1-\alpha) + \lambda/(1-\alpha)^2}{1-\kappa d_Q/B_Q} \geq -2\left(\frac{\kappa}{1-\alpha} + \frac{\lambda}{(1-\alpha)^2}\right),$$

where the final inequality follows from the fact that $B_Q > 2d_Q\kappa$.

We now derive upper and lower bounds on r_k, $k = 1, \ldots, d$. For the upper bounds, we have

$$\begin{aligned}
r_k &\leq \frac{1}{(1-\alpha)d_Q} + \frac{\lambda}{B_Q(1-\alpha)^2} - \frac{r_{d+1}}{B_Q} \\
&\leq \frac{1}{(1-\alpha)d_Q} + \frac{\lambda}{B_Q(1-\alpha)^2} + \frac{2\xi/(1-\xi)}{B_Q(1-\alpha)} + \frac{\lambda}{B(1-\alpha)^2} \\
&\leq \frac{2}{(1-\alpha)d_Q} + \frac{2\lambda}{B_Q(1-\alpha)^2}.
\end{aligned} \tag{20}$$

The first inequality follows from $B_Q \geq 2\kappa d$ and (17), and the second inequality follows from (16). Finally, for the lower bounds, we have

$$\begin{aligned}
r_k &\geq -\sum_{k'=1,\, k' \neq k}^{d_Q} r_{k'} - \frac{r_{d+1}}{1-\kappa d_Q/B_Q} \\
&\geq -\frac{2}{1-\alpha} - \frac{2\lambda d_Q}{B_Q(1-\alpha)^2} - \frac{2\lambda}{(1-\alpha)^2} \\
&\geq -\frac{2}{1-\alpha} - \frac{\lambda(1-\xi)}{\xi(1-\alpha)^2} - \frac{2\lambda}{(1-\alpha)^2},
\end{aligned}$$

where the first inequality follows from (18) and the second inequality follows from (16) and (20). The result follows. □

We now turn to select and study our sampling distribution. We will use the distribution $\psi_Q = \psi_{u_Q^*, V_Q}$, where

$$V_Q(x) = \frac{1}{d_Q(1-\alpha)} \sum_{i=1}^{d_Q} x_i + \frac{2\lambda}{(1-\alpha)^2}. \tag{21}$$

The following lemma establishes that $\beta_{V_Q} < 1$.

LEMMA 4.4. *For each $\xi \in (0,1)$, $\alpha \in (0,1)$, $\lambda \in (0,\infty)$, and each $Q \in \mathcal{Q}(\xi,\alpha,\lambda)$, we have $\beta_{V_Q} < 1$.*

PROOF. Recall that the expected number of exogenous arrivals in any time period is less than or equal to λd_Q. We therefore have

$$\begin{aligned}
(\alpha P_{u_Q^*}^Q V_Q)(x) &\le \alpha\left[\frac{1}{d_Q(1-\alpha)}\left(\sum_{i=1}^{d} x_i + \lambda d_Q\right) + \frac{2\lambda}{(1-\alpha)^2}\right] \\
&= \alpha \frac{1}{d_Q(1-\alpha)} \sum_{i=1}^{d_Q} x_i + \frac{\alpha(3-\alpha)}{2} \frac{2\lambda}{(1-\alpha)^2} \\
&< \frac{\alpha(3-\alpha)}{2}\left(\frac{1}{d_Q(1-\alpha)} \sum_{i=1}^{d_Q} x_i + \frac{2\lambda}{(1-\alpha)^2}\right) \\
&= \frac{\alpha(3-\alpha)}{2} V_Q(x),
\end{aligned}$$

where the strict inequality holds because $\alpha < \alpha(3-\alpha)/2$ for all $\alpha < 1$. Since $\alpha(3-\alpha)/2 < 1$ for all $\alpha < 1$, the result follows. □

4.3. A bound on θ. Our bound on sample complexity for the RLP, as given by Equation (10), is affected by a parameter θ. In our context of controlled queueing networks, we have a parameter θ_Q for each problem instance $Q \in \mathcal{Q}(\xi,\alpha,\lambda)$:

$$\theta_Q = \frac{1+\beta_{V_Q}}{2} \frac{\mu_{u_Q^*}^T V_Q}{c_Q^T J_Q^*} \sup_{r \in \mathcal{N}_Q} \|J_Q^* - \Phi_Q r\|_{\infty, 1/V_Q}.$$

Building on ideas developed in the previous subsections, for $Q \in \mathcal{Q}(\xi,\alpha,\lambda)$, we can bound θ_Q by a linear function of the number of queues.

THEOREM 4.1. *For each $\xi \in (0,1)$, $\alpha \in (0,1)$, $\lambda \in (0,\infty)$, there exists a scalar $C_{\xi,\alpha,\lambda}$ such that $\theta_Q \le C_{\xi,\alpha,\lambda} d_Q$.*

PROOF. First, since $\beta_{V_Q} < 1$, we have $(1+\beta_{V_Q})/2 < 1$. Next, we bound the term $\mu_{u_Q^*}^T V_Q / c_Q^T J_Q^*$. We have

$$\begin{aligned}
\frac{\mu_{u_Q^*}^T V_Q}{c_Q^T J_Q^*} &= \frac{1}{c_Q^T J_Q^*}(1-\alpha) c_Q^T (I - \alpha P_{u_Q^*}^Q)^{-1}\left(\frac{1}{d_Q(1-\alpha)} \sum_{i=1}^{d} x_i + \frac{2\lambda}{(1-\alpha)^2}\right) \\
&= 1 + \frac{2\lambda}{c_Q^T J_Q^* (1-\alpha)}.
\end{aligned}$$

It then follows from Lemma 4.1 and the fact that $\xi > 0$ that $\mu_{u_Q^*}^T V_Q / c_Q^T J_Q^*$ is bounded above and below by positive scalars that do not depend on the problem instance Q.

We now turn attention to the term $\sup_{r\in\mathcal{N}_Q}\|J_Q^*-\Phi_Q r\|_{\infty,1/V_Q}$. From Lemma 4.1 and the definition of V_Q (21), we have

$$\|J_Q^*\|_{\infty,1/V_Q}\le 1. \tag{22}$$

We also have, from Lemma 4.3, that

$$|(\Phi\tilde{r})(x)|\le \overline{C}_{\xi,\alpha,\lambda}\left(\sum_{i=1}^{d_Q}x_i+1\right)$$

for some $\overline{C}_{\xi,\alpha,\lambda}$. Therefore,

$$\begin{aligned}\|\Phi\tilde{r}\|_{\infty,1/V_Q}&\le \overline{C}_{\xi,\alpha,\lambda}\max_{x\in\mathcal{S}_Q}\frac{\sum_{i=1}^{d_Q}x_i+1}{1/[(1-\alpha)d_Q]\sum_{i=1}^{d_Q}x_i+2\lambda/(1-\alpha)^2}\\&\le \overline{C}_{\xi,\alpha,\lambda}\left((1-\alpha)d_Q+\frac{(1-\alpha)^2}{2\lambda}\right).\end{aligned}$$

The result then follows from the triangle inequality and the fact that $d_Q\ge 1$. □

Combining this theorem with the sample complexity bound of Theorem 3.1, we see that for any $Q\in\mathcal{Q}(\xi,\alpha,\lambda)$, a number of samples

$$m=O\left(\frac{A_Qd_Q}{(1-\alpha)\epsilon}\left(d_Q\ln\frac{A_Qd_Q}{(1-\alpha)\epsilon}+\ln\frac{1}{\delta}\right)\right),$$

where $A_Q=\max_{x\in\mathcal{S}_Q}|\mathcal{A}_x^Q|$, suffices to guarantee that

$$\|J_Q^*-\Phi^Q\hat{r}\|_{1,c_Q}\le\|J_Q^*-\Phi^Q\tilde{r}\|_{1,c_Q}+\epsilon\|J_Q^*\|_{1,c_Q}$$

with probability $1-\delta$. Hence, the number of samples grows at most quadratically in the number of queues.

5. Dealing with large action spaces.

Cost-to-go function approximation aims to alleviate problems arising when one deals with large state spaces. Some applications also involve large action spaces, with a possibly exponential number of available actions per state. Large action spaces may impose additional difficulties to exact or approximate dynamic programming algorithms; in the specific case of approximate linear programming, the number of constraints involved in the reduced LP becomes intractable as the cardinality of the action sets $\mathcal{A}_x$ increases. In particular, our bound (10) on the number of sampled constraints grows polynomially in A, the cardinality of the largest action set.

Complexity in the action space can be exchanged for complexity in the state space by transforming each action under consideration into a sequence of actions taking values in smaller sets (Bertsekas and Tsitsiklis 1996). For instance, if actions are described by a collection of action variables, one could assign values to the action variables sequentially, instead of simultaneously. More generally, given an alphabet with N symbols—assume for simplicity the symbols are $0,1,\ldots,N-1$—and a finite set of actions $\mathcal{A}_x$ of cardinality less than or equal to A, actions in this set can be mapped to words of length of at most $\lceil\log_N A\rceil$. Hence, we can change the decision on an action $a\in\mathcal{A}_x$ into a decision on a sequence $\hat{a}$ of size $\lceil\log_N A\rceil$.

We define a new MDP as follows. It is not difficult to verify that it solves the same problem as the original MDP.

- States $\bar{x}$ are given by a tuple $(x,\hat{a},i)$, interpreted as follows: $x\in\mathcal{S}$ represents the state in the original MDP; $\hat{a}\in\{0,1,\ldots,N-1\}^{\lceil\log_N A\rceil}$ represents an encoding of an action in $\mathcal{A}_x$ being taken; $i\in\{1,2,\ldots,\lceil\log_N A\rceil\}$ represents which entry in vector $\hat{a}$ we will decide upon next.

- There are N actions associated with each state $(x, \hat{a}, i)$, corresponding to setting $\hat{a}_i$ to $0, 1, \ldots N-1$.
- Taking action "set $\hat{a}_i$ to v" causes a deterministic transition from $\hat{a}$ to $\hat{a}^+$, where $\hat{a}_j^+ = \hat{a}_j$ for $j \neq i$ and $\hat{x}_i^+ = v$. The system transitions from state $(x, \hat{a}, i)$ to state $(x, \hat{a}^+, i+1)$, and no cost is incurred if $i < \lceil \log_N A \rceil$. It transitions from state $(x, \hat{a}, \lceil \log_N A \rceil)$ to state $(y, \hat{a}^+, 1)$ with probability $P_a(x, y)$, where a is the action in $\mathscr{A}_x$ corresponding to the encoding $\hat{a}^+$. A cost $g_a(x)$ is incurred in this transition.
- The discount factor is given by $\alpha^{1/\lceil \log_N A \rceil}$.

The new MDP involves a higher-dimensional state space and smaller action spaces, and is hopefully amenable to treatment by approximate dynamic programming methods. In particular, dealing with the new MDP instead of the original one affects the constraint sampling complexity bounds provided for approximate linear programming. The following quantities involved in the bound are affected.

- The number of actions per state.

In the new MDP, the number of actions per state is reduced from A to N. In principle, N is arbitrary and can be made as low as two, but as we show next, it affects other factors in constraint sampling complexity bound; hence, we have to keep these effects in mind for a suitable choice.

- The term $1/(1-\alpha)$.

In the new MDP, the discount factor is increased from α to $\alpha^{1/\lceil \log_N A \rceil}$. Note that

$$\begin{aligned} 1-\alpha &= 1 - \left(\alpha^{1/\lceil \log_N A \rceil}\right)^{\lceil \log_N A \rceil} \\ &= \left(1 - \alpha^{1/\lceil \log_N A \rceil}\right) \sum_{i=0}^{\lceil \log_N A \rceil - 1} \alpha^i \\ &\leq \left(1 - \alpha^{1/\lceil \log_N A \rceil}\right) \lceil \log_N A \rceil, \end{aligned}$$

so that $1/(1-\alpha^{1/\lceil \log_N A \rceil}) \leq \lceil \log_N A \rceil / (1-\alpha)$.

- The number of basis functions K.

In the new MDP, we have a higher-dimensional state space; hence, we may need a larger number of basis functions in order to achieve an acceptable approximation to the optimal cost-to-go function. The actual increase on the number of basis functions will depend on the structure of the problem at hand.

The bound on the number of constraints being sampled is polynomial in the three terms above. Hence, implementation of the RLP for the modified version of an MDP will require a number of constraints polynomial in N and in $\lceil \log_N A \rceil$, to be contrasted with the number of constraints necessary for the original MDP, which is polynomial in A. However, the original complexity of the action space is transformed into extra complexity in the state space, which may incur extra difficulties in the selection and/or increase in the number of basis functions. (Recall that the number of constraints is also polynomial in the number of basis functions.) Nevertheless, there is a potential advantage of using the new MDP, as it provides an opportunity for structures associated with the action space to be exploited in the same way as structures associated with the state space are.

6. Closing remarks.

In this paper, we have analyzed a constraint sampling algorithm as an approximation method for dealing with the large number of constraints involved in the ALP. We have shown how near-feasibility can be ensured by sampling a tractable number of constraints. We have established a bound on the number of samples required, under idealized conditions, to ensure small error in approximating the optimal solution of the ALP. Through an example involving controlled queueing networks, we demonstrated that this bound can scale gracefully with problem size.

There are several important directions in which the present results should be extended:

(i) The sampling scheme we have studied is idealized in that it makes use of the stationary distribution of an optimal policy, which is generally unknown. We anticipate that in specific contexts of practical relevance, it will be possible to derive similar sample complexity bounds based on samples drawn from a known distribution. However, for the moment this remains an open issue.

(ii) In §4, we offered an example of how the constraint set $\mathcal{N}$ might be chosen in a specific context to guarantee a graceful sample complexity bound. This represents a start, but further work is required to better understand how the constraints set should be chosen in broader contexts.

(iii) The error bounds we have developed revolve around the norm $\|\cdot\|_{1,c}$. This is motivated by ideas from our companion paper de Farias and Van Roy (2003), which argued that minimization of this norm is aligned with minimization of average cost associated with a greedy policy that is based on the resulting approximation. However, the analysis in that paper required that the approximation is a lower bound to the optimal cost-to-go function. This is guaranteed for solutions of the ALP but not the RLP. Further work is required to understand the impact of this issue.

Acknowledgments. 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. The authors thank the anonymous reviewers and Mike Veatch for helpful comments.

References

Anthony, D., N. Biggs. 1992. *Computational Learning Theory*. Cambridge University Press, Cambridge, UK, 95.

Bertsekas, D., J. N. Tsitsiklis. 1996. *Neuro-Dynamic Programming*. Athena Scientific, Belmont, MA.

Calafiore, G., M. Campi. 2003. Uncertain convex programs: Randomized solutions and confidence levels. Preprint.

Clarkson, K. L. 1995. Las Vegas algorithms for linear and integer programming when the dimension is small. *J. Assoc. Comput. Machinery* **42**(2) 488–499.

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

Dudley, R. M. 1978. Central limit theorems for empirical measures. *Ann. Probab.* **6**(6) 899–929.

Grötschel, M., O. Holland. 1991. Solution of large-scale symmetric travelling salesman problems. *Math. Programming* **51** 141–202.

Guestrin, C., D. Koller, R. Parr. 2003. Efficient solution algorithms for factored MDPs. *J. Artificial Intelligence Res.* **19** 399–468.

Haussler, D., M. Kearns, N. Littlestone, M. K. Warmuth. 1991. Equivalence of models for polynomial learnability. *Inform. Comput.* **95**(2) 129–161.

Morrison, J. R., P. R. Kumar. 1999. New linear program performance bounds for queueing networks. *J. Optim. Theory Appl.* **100**(3) 575–597.

Schuurmans, D., R. Patrascu. 2002. Direct value-approximation for factored MDPs. *Proc. 2001 Neural Inform. Processing Systems (NIPS) Conf., Adv. Neural Inform. Processing Systems*, Vol. 14. MIT Press, Cambridge, MA.

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

Trick, M., S. Zin. 1993. A linear programming approach to solving stochastic dynamic programs. Unpublished manuscript.

Trick, M., S. Zin. 1997. Spline approximations to value functions: A linear programming approach. *Macroeconomic Dynam.* **1** 255–277.