A Generalized Reduced Linear Program for Markov Decision Processes

Chandrashekar Lakshminarayanan and Shalabh Bhatnagar
Department Computer Science and Automation
Indian Institute of Science, Bangalore-560012, India
{chandrul,shalabh}@csa.iisc.ernet.in

Abstract

Markov decision processes (MDPs) with large number of states are of high practical interest. However, conventional algorithms to solve MDP are computationally infeasible in this scenario. Approximate dynamic programming (ADP) methods tackle this issue by computing approximate solutions. A widely applied ADP method is approximate linear program (ALP) which makes use of linear function approximation and offers theoretical performance guarantees. Nevertheless, the ALP is difficult to solve due to the presence of a large number of constraints and in practice, a reduced linear program (RLP) is solved instead. The RLP has a tractable number of constraints sampled from the original constraints of the ALP. Though the RLP is known to perform well in experiments, theoretical guarantees are available only for a specific RLP obtained under idealized assumptions.

In this paper, we generalize the RLP to define a generalized reduced linear program (GRLP) which has a tractable number of constraints that are obtained as positive linear combinations of the original constraints of the ALP. The main contribution of this paper is the novel theoretical framework developed to obtain error bounds for any given GRLP. Central to our framework are two $\max$-norm contraction operators. Our result theoretically justifies linear approximation of constraints. We discuss the implication of our results in the contexts of ADP and reinforcement learning. We also demonstrate via an example in the domain of controlled queues that the experiments conform to the theory.

Introduction

Markov decision process (MDP) is an important mathematical framework to study optimal sequential decision making problems that arise in science and engineering. Solving an MDP involves computing the optimal *value-function* (J^*), a vector whose dimension is the number of states. MDPs with small number of states can be solved easily by conventional solution methods such as value/ policy iteration or linear programming (LP) (Bertsekas 2013). Dynamic programming is at the heart of all the conventional solution methods for MDPs.

The term *curse-of-dimensionality* (or in short *curse*) denotes the fact that the number of states grows exponentially in the number of state variables. Most practical MDPs suffer from the curse, i.e., have large number of states and J^* is difficult to compute. A practical way to tackle the curse is to compute an approximate value function $\tilde{J}$ instead of J^*. The methods that compute $\tilde{J}$ instead of J^* are known as approximate dynamic programming (ADP) methods whose success depends on the quality of approximation, i.e., on the quantity $||J^* - \tilde{J}||$. Most ADP methods employ linear function approximation (LFA), i.e., let $\tilde{J} = \Phi r^*$, where Φ is a feature matrix and r^* is a learnt weight vector. Dimensionality reduction is achieved by choosing Φ to have far fewer columns in comparison to the number of states and this makes computing $\tilde{J}$ easier.

Approximate linear program (ALP) (de Farias and Roy 2003) employs LFA in the linear programming formulation (Bertsekas 2013) of MDP. The ALP computes an approximate value function and offers sound theoretical guarantees. A serious shortcoming of the ALP is the large number of constraints (of the order of the number of states). A technique studied in literature that tackles the issue of large number of constraints is constraint sampling (de Farias and Roy 2004; Farias and Roy 2006) wherein one solves a reduced linear program (RLP) with a small number of constraints sampled from the constraints of the ALP. (de Farias and Roy 2004) presents performance guarantees for the RLP when the constraints are sampled with respect to the stationary distribution of the optimal policy. Such an idealized assumption on the availability of the optimal policy (which in turn requires knowledge of J^*) is a shortcoming. Nevertheless, the RLP has been shown to perform empirically well (de Farias and Roy 2004; de Farias and Roy 2003; Desai, Farias, and Moallemi 2009) even when the constraints are not sampled using the stationary distribution of the optimal policy.

Motivated by the gap between the limited theoretical guarantees of the RLP as currently available in the literature and its successful practical efficacy, in this paper, we provide a novel theoretical framework to characterize the error due to constraint reduction/approximation. The novelty and salient points of our contribution are listed below:

• We define a generalized reduced linear program (GRLP) which has a tractable number of constraints that are obtained as positive linear combinations of the original constraints of

the ALP.

• We develop a novel analytical framework in order to relate $\hat{J}$, the solution to the GRLP, and the optimal value function J^*. In particular, we come up with two novel max-norm contraction operators, viz., the least upper bound (LUB) projection operator and the approximate least upper bound projection operator (ALUB).

• We show that $||J^* - \hat{J}|| \leq (c_1 + c_2)$, where $c_1 > 0$, $c_2 > 0$ are constants. While the term c_1 corresponds to the error inherent to the ALP itself, the term c_2 constitutes the additional error introduced due to constraint approximation.

• The results from the GRLP framework solve the problem of theoretically justifying linear approximation of constraints. Unlike the bounds in (de Farias and Roy 2004) that hold only for specific RLP, our bounds hold for any GRLP and consequently for any RLP.

• We also discuss qualitatively the relative importance of our results in the context of ADP and their implication in the reinforcement learning setting.

• We demonstrate via an example in controlled queues that the experiments conform to the theory developed.

The rest of the paper is organized as follows. First, we present the basics of MDP. We then discuss the ALP technique, the basic error bounds as well as the issues and proposed solutions in literature, followed by the long-standing open questions, that we address in this paper. Finally, we present the main results of the paper namely the GRLP and its error analysis. We then present a qualitative discussion of our result followed by the numerical example.

Markov Decision Process (MDP)

In this section, we briefly discuss the basics of Markov Decision Process (MDP) (the reader is referred to (Bertsekas 2013; Puterman 1994) for a detailed treatment).

The MDP Model: An MDP is a 4-tuple $< S, A, P, g >$, where S is the state space, A is the action space, P is the probability transition kernel and g is the reward function. We consider MDPs with large but finite number of states, i.e., $S = \{1, 2, \ldots, n\}$ for some large n, and the action set is given by $A = \{1, 2, \ldots, d\}$. For simplicity, we assume that all actions are feasible in all states. The probability transition kernel P specifies the probability $p_a(s, s')$ of transitioning from state s to state s' under the action a. We denote the reward obtained for performing action $a \in A$ in state $s \in S$ by $g_a(s)$.

Policy: A policy μ specifies the action selection mechanism, and is described by the sequence $\mu = \{u_1, u_2, \ldots, u_n, \ldots\}$, where $u_n \colon S \to A$, $\forall n \geq 0$. A stationary deterministic policy (SDP) is one where $u_n \equiv u$, $\forall n \geq 0$ for some $u \colon S \to A$. By abuse of notation we denote the SDP by u itself instead of μ. In the setting that we consider, one can find an SDP that is optimal (Bertsekas 2013; Puterman 1994). In this paper, we restrict our focus to the class U of SDPs. Under an SDP u, the MDP is a Markov chain with probability transition kernel P_u.

Value Function: Given an SDP u, the infinite horizon discounted reward corresponding to state s under u is denoted by $J_u(s)$ and is defined by $J_u(s) \triangleq \mathbf{E}[\sum_{n=0}^{\infty} \alpha^n g_{a_n}(s_n) | s_0 = s, a_n = u(s_n)\ \forall n \geq 0]$, where $\alpha \in (0, 1)$ is a given discount factor. Here $J_u(s)$ is known as the value of the state s under the SDP u, and the vector quantity $J_u \triangleq (J_u(s), \forall s \in S) \in R^n$ is called the value-function corresponding to the SDP u.

The optimal SDP u^* is obtained as $u^*(s) \triangleq \arg\max_{u \in U} J_u(s)$[1].

The optimal value-function J^* is the one obtained under the optimal policy, i.e., $J^* = J_{u^*}$.

The Bellman Equation and Operator: Given an MDP, our aim is to find the optimal value function J^* and the optimal policy u^*. The optimal policy and value function obey the Bellman equation (BE) as under: $\forall s \in S$,

$$J^*(s) = \max_{a \in A} \big(g_a(s) + \alpha \sum_{s'} p_a(s, s') J^*(s')\big), \tag{1a}$$

$$u^*(s) = \arg\max_{a \in A} \big(g_a(s) + \alpha \sum_{s'} p_a(s, s') J^*(s')\big). \tag{1b}$$

Typically J^* is computed first and u^* is obtained by substituting J^* in (1b).

The Bellman operator $T \colon \mathbf{R}^n \to \mathbf{R}^n$ is defined using the model parameters of the MDP as follows:

$$(TJ)(s) = \max_{a \in A} \big(g_a(s) + \alpha \sum_{s'} p_a(s, s') J(s')\big), J \in \mathbf{R}^n.$$

Basis Solution Methods: When the number of states of the MDP is small, J^* and u^* can be computed exactly using conventional methods such as value/policy iteration and linear programming (LP) (Bertsekas 2013).

Curse-of-Dimensionality is a term used to denote the fact that the number of states grows exponentially in the number of state variables. Most MDPs occurring in practice suffer from the curse, i.e., have large number of states and it is difficult to compute $J^* \in \mathbf{R}^n$ exactly in such scenarios.

Approximate Dynamic Programming (Bertsekas 2013) (ADP) methods compute an approximate value function $\tilde{J}$ instead of J^*. In order to make the computations easier, ADP methods employ function approximation (FA) where in $\tilde{J}$ is chosen from a parameterized family of functions. The problem then boils down to finding the optimal parameter which is usually of lower dimension and is easily computable.

Linear Function Approximation (LFA) (de Farias and Roy 2003; Nedić and Bertsekas 2003; Konidaris, Osentoski, and Thomas 2011; Mahadevan and Liu 2010; Mahadevan and Maggioni 2007) is a widely used FA scheme where the approximate value function $\tilde{J} = \Phi r^*$, with $\Phi = [\phi_1 | \ldots | \phi_k]$ being an $n \times k$ feature matrix and r^*, is the parameter to be learnt.

Approximate Linear Programming

We now present the linear programming formulation of the MDP which forms the basis for ALP. The LP formulation is

[1] Such u^* exists and is well defined in the case of infinite horizon discounted reward MDP, for more details see (Puterman 1994).

obtained by unfurling the max operator in the BE in (1) into a set of linear inequalities as follows:

$$\min_{J\in\mathbf{R}^n} c^\top J$$
$$\text{s.t } J(s) \geq g_a(s) + \alpha \sum_{s'} p_a(s,s')J(s'),\ \forall s \in S, a \in A, \quad (2)$$

where $c \in \mathbf{R}^n_+$ is a probability distribution and denotes the relative importance of the various states. One can show that J^* is the solution to (2) (Bertsekas 2013). The LP formulation in (2) can be represented in short[2] as,

$$\min_{J\in\mathbf{R}^n} c^\top J$$
$$\text{s.t } J \geq TJ. \quad (3)$$

The approximate linear program (ALP) is obtained by making use of LFA in the LP, i.e., by letting $J = \Phi r$ in (3) and is given as

$$\min_{r\in\mathbf{R}^k} c^\top \Phi r$$
$$\text{s.t } \Phi r \geq T\Phi r. \quad (4)$$

Unless specified otherwise we use $\tilde{r}_c$ to denote the solution to the ALP and $\tilde{J}_c = \Phi\tilde{r}_c$ to denote the corresponding approximate value function. The following is a preliminary error bound for the ALP from (de Farias and Roy 2003):

Theorem 1 *Let* $\mathbf{1}$*, i.e., the vector with all-components equal to* 1*, be in the span of the columns of* Φ *and* c *be a probability distribution. Then, if* $\tilde{J}_c = \Phi\tilde{r}_c$ *is an optimal solution to the ALP in* (4)*, then* $||J^* - \tilde{J}_c||_{1,c} \leq \frac{2}{1-\alpha}\min_r ||J^* - \Phi r||_\infty$*, where* $||x||_{1,c} = \sum_{i=1}^n c(i)|x(i)|$.

For a more detailed treatment of the ALP and sophisticated bounds, the reader is referred to (de Farias and Roy 2003). Note that the ALP is a linear program in k ($<< n$) variables as opposed to the LP in (3) which has n variables. Nevertheless, the ALP has nd constraints (same as the LP) which is an issue when n is large and calls for constraint approximation/reduction techniques.

Related Work

Constraint sampling and The RLP: The most important work in the direction of constraint reduction is constraint sampling (de Farias and Roy 2004) wherein a reduced linear program (RLP) is solved instead of the ALP. While the objective of the RLP is same as that of the ALP, the RLP has only $m << nd$ constraints. These m constraints are *sampled* from the original nd constraints of the ALP according to a special sampling distribution $\psi_{u^*,V}$, where u^* is the optimal policy and V is a Lyapunov function (see (de Farias and Roy 2004) for a detailed presentation). If $\tilde{r}$ and $\tilde{r}_{RLP}$ are the solutions to the ALP and the RLP respectively, from (de Farias and Roy 2004) we know that $||J^* - \Phi\tilde{r}_{RLP}||_{1,c} \leq ||J^* - \Phi\tilde{r}||_{1,c} + \epsilon||J^*||_{1,c}$. A major gap in the theoretical analysis is that the error bounds are known for only a specific RLP formulated using idealized assumptions, i.e., under knowledge of u^*.

Other works: Most works in literature make use of the underlying structure of the problem to cleverly reduce the number of constraints of the ALP. A good example is (Guestrin et al. 2003), wherein the structure in factored linear functions is exploited. The use of basis function also helps constraint reduction in (Morrison and Kumar 1997). In (Borkar, Pinto, and Prabhu 2009), the constraints are approximated indirectly by approximating the square of the Lagrange multipliers. In (Petrik and Zilberstein 2009) the transitional error is reduced ignoring the representational and sampling errors. Empirical successes include repeated application of constraint sampling to solve Tetris (Farias and Roy 2006).

Long-Standing Open Questions: The fact that RLP works well empirically goads us to build a more elaborate theory for constraint reduction. In particular, one would like to answer the following questions related to constraint reduction in ALP that have so far remained open.

• As a natural generalization of the RLP, what happens if we define a generalized reduced linear program (GRLP) whose constraints are positive linear combinations of the original constraints of the ALP?

• Unlike (de Farias and Roy 2004) which provides error bounds for a specific RLP formulated using an idealized sampling distribution, is it possible to provide error bounds for any GRLP (and hence any RLP)? In this paper, we address both of the questions above.

Generalized Reduced Linear Program

We define the generalized reduced linear program (GRLP) as below:

$$\min_{r\in\chi} c^\top \Phi r,$$
$$\text{s.t } W^\top \Phi r \geq W^\top T\Phi r, \quad (5)$$

where $W \in \mathbf{R}^{nd\times m}_+$ is an $nd\times m$ matrix with all positive entries and $\chi \subset \mathbf{R}^k$ is any bounded set such that $\hat{J}_c \in \chi$. Thus the i^{th} ($1 \leq i \leq m$) constraint of the GRLP is a positive linear combination of the original constraints of the ALP, see Assumption 1. Constraint reduction is achieved by choosing $m << nd$. Unless specified otherwise we use $\hat{r}_c$ to denote the solution to the GRLP in (5) and $\hat{J}_c = \Phi\hat{r}_c$ to denote the corresponding approximate value function. We assume the following throughout the rest of the paper:

Assumption 1 $W \in \mathbf{R}^{nd\times m}_+$ *is a full rank* $nd \times m$ *matrix with all non-negative entries. The first column of the feature matrix* Φ *(i.e.,*ϕ_1*) is* $\mathbf{1}$[3] $\in \mathbf{R}^n$ *and that* $c = (c(i), i = 1,\ldots,n) \in \mathbf{R}^n$ *is a probability distribution, i.e.,* $c(i) \geq 0$ *and* $\sum_{i=1}^n c(i) = 1$*. It is straightforward to see that a RLP is trivially a GRLP.*

As a result of constraint reduction the feasible region of the GRLP is a superset of the feasible region of the ALP (see Figure 1). In order to bound $||J^* - \tilde{J}_c||$, (de Farias and Roy 2003) makes use of the property that $\Phi\tilde{r}_c \geq T\Phi\tilde{r}_c$.

[2] $J \geq TJ$ is a shorthand for the nd constraints in (2). It is also understood that constraints $(i-1)n+1,\ldots,in$ correspond to the i^{th} action.

[3] $\mathbf{1}$ is a vector with all components equal to 1.

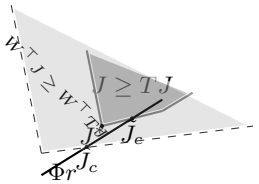

Figure 1: The outer lightly shaded region corresponds to GRLP constraints and the inner dark shaded region corresponds to the original constraints. The main contribution of the paper is to provide a bound for $||J^* - \hat{J}_c||$.

However, in the case of the GRLP, this property does not hold anymore and hence it is a challenge to bound the error $||J^* - \hat{J}_c||$. We tackle this challenge by introducing two novel max-norm contraction operators called the least upper bound projection (LUBP) and approximate least upper bound projection (ALUBP) operators denoted by Γ and $\tilde{\Gamma}$ respectively. We first present some definitions before the main result and a sketch of its proof. The least upper bound (LUB) projection operator $\Gamma\colon \mathbf{R}^n \to \mathbf{R}^n$ is defined as below:

Definition 2 *Given $J \in \mathbf{R}^n$, its least upper bound projection is denoted by ΓJ and is defined as*

$$(\Gamma J)(i) \triangleq \min_{j=1,\ldots,k} (\Phi r_{e_j})(i),\ \forall i = 1,\ldots,n, \qquad (6)$$

where $V(i)$ denotes the i^{th} component of the vector $V \in \mathbf{R}^n$. Also in (6)*, e_j is the vector with 1 in the j^{th} place and zeros elsewhere, and r_{e_j} is the solution to the linear program in* (7) *for $c = e_j$.*

$$r_c \triangleq \min_{r \in \chi} c^\top \Phi r,$$
$$s.t\ \Phi r \geq TJ. \qquad (7)$$

Remark 1

1. *Given Φ and $J \in \mathbf{R}^n$, define $\mathcal{F} \triangleq \{\Phi r | \Phi r \geq TJ\}$. Thus $\mathcal{F}$ is the set of all vectors in the span of Φ that upper bound TJ. By fixing c in the linear program in* (7) *we select a unique vector $\Phi r_c \in \mathcal{F}$. The LUB projection operator Γ picks n vectors $\Phi r_{e_i}, i = 1,\ldots,n$ from the set $\mathcal{F}$ and ΓJ is obtained by computing their component-wise minimum.*
2. *Even though ΓJ does not belong to the span of Φ, ΓJ in some sense collates the various best upper bounds that can be obtained via the linear program in* (7)*.*
3. *The LUB operator Γ in* (6) *bears close similarity to the ALP in* (4)*.*

We define an approximate least upper bound (ALUB) projection operator which has a structure similar to the GRLP and is an approximation to the LUB operator.

Definition 3 *Given $J \in \mathbf{R}^n$, its approximate least upper bound (ALUB) projection is denoted by $\tilde{\Gamma} J$ and is defined as*

$$(\tilde{\Gamma} J)(i) \triangleq \min_{j=1,\ldots,k} (\Phi r_{e_j})(i),\ \forall i = 1,\ldots,n, \qquad (8)$$

where r_{e_j} is the solution to the linear program in (9) *for $c = e_j$, and e_j is same as in Definition 2.*

$$r_c \triangleq \min_{r \in \chi} c^\top \Phi r,$$
$$s.t\ W^\top \Phi r \geq W^\top TJ, W \in \mathbf{R}_+^{nd \times m}. \qquad (9)$$

Definition 4 *The LUB projection of J^* is denoted by $\bar{J} = \Gamma J^*$, and let $r^* \triangleq \arg\min_{r \in \mathbf{R}^k} ||J^* - \Phi r^*||$.*

Main Result

Theorem 5

$$||J^* - \hat{J}_c||_{1,c} \leq \frac{6||J^* - \Phi r^*||_\infty + 2||\Gamma \bar{J} - \tilde{\Gamma} \bar{J}||_\infty}{1 - \alpha}. \qquad (10)$$

Proof: Here we provide a sketch of the proof (see (Lakshminarayanan and Bhatnagar 2014) for the detailed proof). Figure 2 gives an idea of the steps that lead to the result. First, one shows that the operators Γ and $\tilde{\Gamma}$ have the max-norm contraction property with factor α. As a result, operators Γ and $\tilde{\Gamma}$ have fixed points $\tilde{V} \in \mathbf{R}^n$ and $\hat{V} \in \mathbf{R}^n$ respectively. This leads to the inequalities $\tilde{J}_c \geq \tilde{V} \geq J^*$ and $\hat{J}_c \geq \hat{V}$ (see Figure 2), followed by which one can bound the term $||J^* - \hat{V}||_\infty$ and then go on to show that any solution $\tilde{r}_c$ to the GRLP is also a solution to the program in (11).

$$\min_{r \in \chi} ||\Phi r - \hat{V}||_{1,c}$$
$$\text{s.t}\ W^\top \Phi r \geq W^\top T \Phi r. \qquad (11)$$

One then obtains the bound $||J^* - \hat{J}_c||_{1,c}$ as in (10) using the fact that $||J^* - \bar{J}||_\infty \leq 2||J^* - \Phi r^*||_\infty$ where r^* is as in Definition 4.

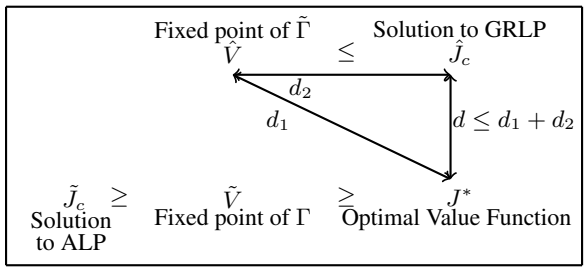

Figure 2: A schematic of the error analysis. Here $d = ||J^* - \hat{J}_c||_{1,c}$.

It is important to note that $\Gamma/\tilde{\Gamma}$ are only analytical constructs that lead us to the error bounds, and need not be calculated in practice for systems with large n.

Discussion

We now make various important qualitative observations about the result in Theorem 5.

Error Terms: The error term is split into two factors, the

first of which is related to the best possible projection while the second factor is related to constraint approximation. The second factor $||\Gamma\bar{J}-\tilde{\Gamma}\bar{J}||_\infty$ is completely defined in terms of Φ, W and T, and does not require knowledge of stationary distribution of the optimal policy. It makes intuitive sense since given that Φ approximates J^*, it is enough for W to depend on Φ and T without any additional requirements. Unlike the result in (de Farias and Roy 2004) which holds only for a specific RLP formulated under ideal assumptions, our bounds hold for any GRLP and as a result for any given RLP. Another interesting feature of our result is that it holds with probability 1. Also by making use of appropriate Lyapunov functions as in (de Farias and Roy 2003), the error bound in (10) can also be stated using a weighted L_∞-norm, thereby indicating the relative importance of states.

Additional insights on constraint sampling: It is easy to notice from Definitions 2, 3 and 4 that for any given state $s \in S, \Gamma\bar{J}(s) \geq J^*(s)$, and that $\Gamma\bar{J}(s) \geq \tilde{\Gamma}\bar{J}(s)$. If the state s is selected in the RLP, then it is also true that $\Gamma\bar{J}(s) \geq \tilde{\Gamma}\bar{J}(s) \geq J^*(s)$. Thus the additional error $|\Gamma\bar{J}(s) - \tilde{\Gamma}\bar{J}(s)|$ due to constraint sampling is less than the original projection error $|\Gamma\bar{J}(s) - J^*(s)|$ due to function approximation. This means that the RLP is expected to perform well whenever *important* states are retained after constraint sampling. Thus the sampling distribution need not be the stationary distribution of the optimal policy as long as it samples the important states, an observation that might theoretically explain the empirical successes of the RLP (de Farias and Roy 2003; Farias and Roy 2006; Desai, Farias, and Moallemi 2009).

Relation to other ADP methods:

ADP Method	Empirical	Theoretical
Projected Bellman Equation	✓	✗-Policy Chattering
ALP	✗-Large number of Constraints	✓
RLP	✓	✗- Only under ideal assumptions

A host of the ADP methods such as (Lagoudakis and Parr 2003; Nedić and Bertsekas 2003; Boyan 1999; Tsitsiklis and Roy 1997) are based on solving the projected Bellman equation (PBE). The PBE based methods have been empirically successful and also have theoretical guarantees for the approximate value function. However, a significant shortcoming is that they suffer from the issue of *policy-chattering* (see section 6.4.3 of (Bertsekas 2013)), i.e., the sequence of policies might oscillate within a set of bad policies. A salient feature of the ALP based methods is that they find only one approximate value function $\tilde{J}_c$ and one sub-optimal policy derived as a greedy policy with respect to $\tilde{J}_c$. As a result there is no such issue of policy-chattering for the ALP based methods. By providing the error bounds for the GRLP, our paper provides the much required theoretical support for the RLP. Our GRLP framework closes the long-standing gap in the literature of providing a theoretical framework to bound the error due to constraint reduction in ALP based schemes.

GRLP is linear function approximation of the constraints: In order to appreciate this fact consider the Lagrangian of the ALP and GRLP in (12) and (13), respectively, i.e.,

$$\tilde{L}(r,\lambda) = c^\top \Phi r + \lambda^\top (T\Phi r - \Phi r), \quad (12)$$

$$\hat{L}(r,q) = c^\top \Phi r + q^\top W^\top (T\Phi r - \Phi r). \quad (13)$$

The insight that the GRLP is linear function approximation of constraints (i.e., the Lagrangian multipliers) can be obtained by noting that $Wq \approx \lambda$ in (13). Note that while the ALP employs LFA in its objective, the GRLP employs linear approximation both in the objective as well as the constraints. This has significance in the context of the reinforcement learning setting (Sutton and Barto 1998) wherein the model information is available in the form of noisy sample trajectories. RL algorithms make use of stochastic approximation (SA) (Borkar 2008) and build on ADP methods to come up with incremental update schemes to learn from noisy samples presented to them and linear approximation architectures are found to be useful in this setting. An SA scheme to solve the GRLP in the RL setting can be derived in a manner similar to (Borkar, Pinto, and Prabhu 2009).

Application to Controlled Queues

We take up an example in the domain of controlled queues to show that experiments are in agreement with the theory developed. More specifically, we look at the error bounds for different constraints reduction schemes to demonstrate the fact that whenever value of $||\Gamma\bar{J} - \tilde{\Gamma}\bar{J}||_\infty$ is less, the GRLP solution is close to the optimal value function.

The queuing system consists of $n = 10^4$ states and $d = 4$ actions. We chose $n = 10^4$ because it was possible to solve both the GRLP and the exact LP (albeit with significant effort) so as to enumerate the approximation errors. We hasten to mention that while we could run the GRLP for queuing systems with $n > 10^4$ without much computational overhead, solving the exact LP was not possible for $n > 10^4$, as a result of which the approximation error could not be computed.

Queuing Model: The queuing model used here is similar to the one in Section 5.2 of (de Farias and Roy 2003). We consider a single queue with arrivals and departures. The state of the system is the queue length with the state space given by $S = \{0, \ldots, n-1\}$, where $n-1$ is the buffer size of the queue. The action set $A = \{1, \ldots, d\}$ is related to the service rates. We let s_t denote the state at time t. The state at time $t+1$ when action $a_t \in A$ is chosen is given by $s_{t+1} = s_t + 1$ with probability p, $s_{t+1} = s_t - 1$ with probability $q(a_t)$ and $s_{t+1} = s_t$, with probability $(1 - p - q(a_t))$. For states $s_t = 0$ and $s_t = n-1$, the system dynamics is given by $s_{t+1} = s_t + 1$ with probability p when $s_t = 0$ and $s_{t+1} = s_t - 1$ with probability $q(a_t)$ when $s_t = n-1$. The service rates satisfy $0 < q(1) \leq \ldots \leq q(d) < 1$ with $q(d) > p$ so as to ensure 'stabilizability' of the queue. The reward associated with the action $a \in A$ in state $s \in S$ is given by $g_a(s) = -(s + 60q(a)^3)$.

Choice of Φ : We make use of polynomial features in Φ (i.e., $1, s, \ldots, s^{k-1}$) since they are known to work well for this domain (de Farias and Roy 2003). This takes care of the term $||J^* - \Phi r^*||_\infty$ in (10).

Selection of W: For our experiments, we choose two contenders for the W-matrix:

(**i**) W_c- matrix that corresponds to sampling according to c. This is justified by the insights obtained from the error term $||\Gamma\bar{J} - \tilde{\Gamma}\bar{J}||_\infty$ and the idea of selecting the important states. (**ii**) W_a state-aggregation matrix, a heuristic derived by interpreting W to be the feature matrix that approximates the Lagrange multipliers as $\lambda \approx Wq$, where $\lambda \in \mathbf{R}^{nd}, r \in \mathbf{R}^m$. One can show (Dolgov and Durfee 2006) that the optimal Lagrange multipliers are the discounted number of visits to the 'state-action' pairs under the optimal policy u^*, i.e., $\lambda^*(s,u^*(s)) = \big(c^\top (I - \alpha P_{u^*})^{-1}\big)(s) = \big(c^\top (I + \alpha P_{u^*} + \alpha^2 P_{u^*}^2 + \ldots)\big)(s), \lambda^*(s,u^*(s)) = 0, \forall a \neq u^*(s)$, where P_{u^*} is the probability transition matrix with respect to the optimal policy. Even though we might not have the optimal policy u^* in practice, the fact that λ^* is a linear combination of $\{P_{u^*}, P_{u^*}^2, \ldots\}$ hints at the kind of features that might be useful for the W matrix. Our choice of W_a matrix to correspond to aggregation of nearby states is motivated by the observation that P^n captures n^{th} hop connectivity/neighborhood information. The aggregation matrix W_a is defined as below: $\forall i = 1, \ldots, m$,

$$W_a(i,j) = 1, \ \forall j \text{ s.t } j = (i-1)\frac{n}{m} + k + (l-1)n,$$
$$k = 1, \ldots, \frac{n}{m}, l = 1, \ldots, d,$$
$$= 0, \text{ otherwise.} \quad (14)$$

In order to provide a contrast between good and bad choices of W matrices we also make use of two more matrices, an ideal matrix W_i generated by sampling according to the stationary distribution of the optimal policy as in (de Farias and Roy 2004) and W_c generated by sampling using c, as well as W_r, a random matrix in $\mathbf{R}_+^{nd \times m}$. For the sake of comparison, we compute $||\Gamma\bar{J} - \tilde{\Gamma}\bar{J}||_\infty$ for the different W matrices. Though computing $||\Gamma\bar{J} - \tilde{\Gamma}\bar{J}||_\infty$ might be hard in the case of large n, since $||\Gamma\bar{J} - \tilde{\Gamma}\bar{J}||_\infty$ is completely dependent on the structure of Φ, T and W, we can compute it for small n instead and use it as a surrogate. Accordingly, we first chose a smaller system, Q_S, with $n = 10$, $d = 2$, $k = 2$, $m = 5$, $q(1) = 0.2$, $q(2) = 0.4$, $p = 0.2$ and $\alpha = 0.98$. In the case of Q_S, W_a ((14) with $m = 5$) turns out to be a 20×5 matrix where the i^{th} constraint of the GRLP is the average of all constraints corresponding to states $(2i-1)$ and $2i$ (there are four constraints corresponding to these two states). The various error terms are listed in Table 1 and plots are shown in Figure 3. It is clear from Table 1 that W_a, W_i and W_c have much better $||\Gamma\bar{J} - \tilde{\Gamma}\bar{J}||_\infty$ than randomly generated positive matrices. Since each constraint is a hyperplane, taking linear combinations of non-adjacent hyperplanes might drastically affect the final solution. This could be a reason why W_r (random matrix) performs poorly in comparison with other W matrices.

Error Term	W_i	W_c	W_a	W_r
$\|\|\Gamma\bar{J} - \tilde{\Gamma}\bar{J}\|\|_\infty$	39	84	54.15	251.83

Table 1: Shows various error terms for Q_S.

Next we consider a moderately large queuing system Q_L with $n = 10^4$ and $d = 4$ with $q(1) = 0.2$, $q(2) = 0.4$, $q(3) = 0.6$, $q(4) = 0.8$, $p = 0.4$ and $\alpha = 0.98$. In the case of Q_L, we chose $k = 4$ (i.e., we used $1, s, s^2$ and s^3 as basis vectors) and we chose W_a (14), W_c, W_i and W_r with $m = 50$. We set $c(s) = (1-\zeta)\zeta^s$, $\forall s = 1, \ldots, 9999$, with $\zeta = 0.9$ and $\zeta = 0.999$ respectively. The results in Table 2 show that performance exhibited by W_a and W_c is better by several orders of magnitude over 'random' in the case of the large system Q_L and is close to the ideal sampler W_i. Also note that a better performance of W_a and W_c in the larger system Q_L is in agreement with a lower value of $||\Gamma\bar{J} - \tilde{\Gamma}\bar{J}||_\infty$ in the smaller system Q_S.

Error Terms	W_i	W_c	W_a	W_r
$\|\|J^* - \hat{J}_c\|\|_{1,c}$ for $\zeta = 0.9$	32	32	220	5.04×10^4
$\|\|J^* - \hat{J}_c\|\|_{1,c}$ for $\zeta = 0.999$	110	180.5608	82	1.25×10^7

Table 2: Shows performance metrics for Q_L.

Conclusion

Solving MDPs with large number of states is of practical interest. However, when the number of states is large, it is difficult to calculate the exact value function. ALP is a widely studied ADP scheme that computes an approximate value function and offers theoretical guarantees. Nevertheless, the ALP is difficult to solve due to its large number of constraints and in practice a reduced linear program (RLP) is solved. Though RLP has been shown to perform well empirically, theoretical guarantees are available only for a specific RLP formulated under idealized assumptions. This paper provided a more elaborate treatment of constraint reduction/approximation. Specifically, we generalized the RLP to formulate a generalized reduced linear program (GRLP) and provided error bounds. Our results addressed a major long-standing open problem of analytically justifying linear function approximation of the constraints. We discussed the implications of our results in the contexts of ADP and reinforcement learning. We found that our experiments conform to the theory developed in this paper on an example in the domain of controlled queues. Future directions include providing more sophisticated error bounds based on Lyapunov functions, a two-time scale actor-critic scheme to solve the GRLP, and basis function adaptation schemes to tune the W matrix.

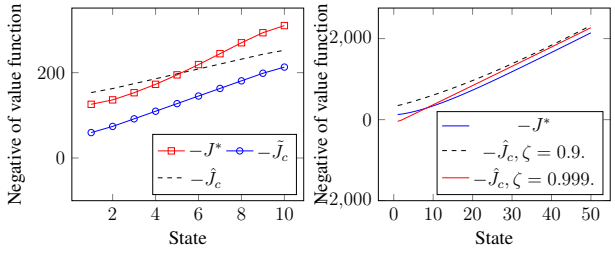

Figure 3: Plot corresponding to Q_S on the left and Q_L on the right. The GRLP here used W_a in (14) with $m = 5$ for Q_S and $m = 50$ for Q_L.

References

[Bertsekas 2013] Bertsekas, D. P. 2013. *Dynamic Programming and Optimal Control*, volume II. Belmont,MA: Athena Scientific, 4^{th} edition.

[Borkar, Pinto, and Prabhu 2009] Borkar, V. S.; Pinto, J.; and Prabhu, T. 2009. A new learning algorithm for optimal stopping. *Discrete Event Dynamic Systems* 19(1):91–113.

[Borkar 2008] Borkar, V. S. 2008. *Stochastic Approximation: A Dynamical Systems Viewpoint*. TRIM.

[Boyan 1999] Boyan, J. A. 1999. Least-squares temporal difference learning. In *ICML*, 49–56. Citeseer.

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

[de Farias and Roy 2004] de Farias, D. P., and Roy, B. V. 2004. On constraint sampling in the linear programming approach to approximate dynamic programming. *Math. Oper. Res.* 29(3):462–478.

[Desai, Farias, and Moallemi 2009] Desai, V. V.; Farias, V. F.; and Moallemi, C. C. 2009. A smoothed approximate linear program. In *NIPS*, 459–467.

[Dolgov and Durfee 2006] Dolgov, D. A., and Durfee, E. H. 2006. Symmetric approximate linear programming for factored mdps with application to constrained problems. *Annals of Mathematics and Artificial Intelligence* 47(3-4):273–293.

[Farias and Roy 2006] Farias, V. F., and Roy, B. V. 2006. Tetris: A study of randomized constraint sampling. In *Probabilistic and Randomized Methods for Design Under Uncertainty*. Springer. 189–201.

[Guestrin et al. 2003] Guestrin, C.; Koller, D.; Parr, R.; and Venkataraman, S. 2003. Efficient solution algorithms for factored MDPs. *J. Artif. Intell. Res.(JAIR)* 19:399–468.

[Konidaris, Osentoski, and Thomas 2011] Konidaris, G.; Osentoski, S.; and Thomas, P. S. 2011. Value function approximation in reinforcement learning using the Fourier basis. In *AAAI*.

[Lagoudakis and Parr 2003] Lagoudakis, M. G., and Parr, R. 2003. Least-squares policy iteration. *Journal of Machine Learning Research* 4:1107–1149.

[Lakshminarayanan and Bhatnagar 2014] Lakshminarayanan, C., and Bhatnagar, S. 2014. A generalized reduced linear program for markov decision processes. *CoRR* abs/1409.3536v2.

[Mahadevan and Liu 2010] Mahadevan, S., and Liu, B. 2010. Basis construction from power series expansions of value functions. In *Advances in Neural Information Processing Systems*, 1540–1548.

[Mahadevan and Maggioni 2007] Mahadevan, S. S., and Maggioni, M. 2007. Proto-value functions: A Laplacian framework for learning representation and control in Markov decision Processes. *Journal of Machine Learning Research* 8(16):2169–2231.

[Morrison and Kumar 1997] Morrison, J. R., and Kumar, P. R. 1997. New linear program performance bounds for queueing networks. Technical Report 3, Journal of Optimization Theory and Applications.

[Nedić and Bertsekas 2003] Nedić, A., and Bertsekas, D. P. 2003. Least squares policy evaluation algorithms with linear function approximation. *Discrete Event Dynamic Systems* 13(1-2):79–110.

[Petrik and Zilberstein 2009] Petrik, M., and Zilberstein, S. 2009. Constraint relaxation in approximate linear programs. In *Proceedings of the 26th Annual International Conference on Machine Learning*, 809–816. ACM.

[Puterman 1994] Puterman, M. L. 1994. *Markov Decision Processes: Discrete Stochastic Programming*. New York: John Wiley.

[Sutton and Barto 1998] Sutton, R. S., and Barto, A. G. 1998. *Introduction to Reinforcement Learning*. Cambridge, MA, USA: MIT Press, 1st edition.

[Tsitsiklis and Roy 1997] Tsitsiklis, J. N., and Roy, B. V. 1997. An analysis of temporal-difference learning with function approximation. Technical report, IEEE Transactions on Automatic Control.