A Generalized Reduced Linear Program for Markov Decision Processes

Chandrashekar L†
Shalabh Bhatnagar§

IISc-CSA-SSL-TR-2014-17


Stochastic Systems Lab
Computer Science and Automation
Indian Institute of Science, India
March 2014
A Generalized Reduced Linear Program for Markov Decision Processes

Chandrashekar L†  Shalabh Bhatnagar§

July 28, 2014

Abstract

Approximate Linear Program (ALP) is a useful method for solving Markov Decision Processes with large number of states. An attractive feature of the ALP is that it offers performance guarantees for the approximate value function as well as the corresponding greedy policy. Nevertheless, a serious shortcoming is the presence of a large number of constraints. Techniques in the past have addressed this shortcoming either by selecting a subset of the original constraints to formulate a Reduced Linear Program (RLP) or by making use of problem structure to intelligently eliminate the constraints. Theoretical performance guarantees are available in the case of RLP if its constraints are sampled from the original constraints using the stationary distribution with respect to the optimal policy.

In this paper we deal with a constraint approximation method that we call the Generalized Reduced Linear Program (GRLP) which approximates the feasible set of the ALP by a larger set described by a fewer number of linear constraints. Each constraint of the GRLP is obtained as a positive linear combination of the constraints of the ALP. The main contribution of this paper is the analytical machinery we develop to obtain error bounds related to GRLP that are based on two novel max-norm contraction operators $\Gamma$ and $\tilde{\Gamma}$. As an example, we demonstrate the usage of the error bounds developed to come up with the right GRLP for a controlled queue. We discuss the implication of our result in the context of reinforcement learning algorithms and also point out interesting future directions.

1 Introduction

Solving a Markov Decision Process (MDP) involves computing the optimal value-function $J^*$ and the optimal policy $u^*$ which are vectors with dimension of the size of state space. The Bellman operator $T$ is defined in terms of the model parameters of the MDP and the Bellman equation states that $J^*$ is a unique fixed point of $T$, i.e., $J^* = TJ^*$. Typically, $J^*$ is computed first and $u^*$ is obtained as a greedy policy with respect to $J^*$. Conventional solution methods which are analytical/iterative in nature make use directly or indirectly of the fact that $J^*$ is a fixed point of $T$. These methods are effective in dealing with MDPs with small number of states.

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 the conventional solution methods fail to be effective for such MDPs. Function approximation is a dimensionality reduction technique that tackles the curse by computing an approximate

†Dept Of CSA, IISc chandrul@csa.iisc.ernet.in, §Dept Of CSA, IISc shalabh@csa.iisc.ernet.in
value function $\hat{J}$ which belongs to a parameterized class. The most widely used architecture for function approximation is the linear function approximation (LFA) architecture in which $J^*$ is approximated by $J^* \approx \tilde{J} = \Phi r^*$, where $\Phi$ is a feature matrix and $r^*$ is the learnt parameter vector. Here, the columns of $\Phi$ are the basis functions and dimensionality reduction is achieved by choosing a fewer number of basis functions compared to the number of states. Once $\tilde{J}$ is computed, a greedy policy $\hat{u}$ is obtained from the model parameters.

Approximate Linear Program (ALP) de Farias and Roy [2003, 2004], Desai et al. [2009], Borkar et al. [2009], Guestrin et al. [2003], Taylor et al. [2010], Pazis and Parr [2011] employs LFA in the linear programming formulation (Bertsekas [2013], Bertsekas and Tsitsiklis [1996]) of MDP. The ALP is a value function method, i.e., it computes an approximate value function $\tilde{J}$ that satisfies the Bellman inequality $\tilde{J} \geq T \tilde{J}$ (for the discounted reward criterion). A serious shortcoming of the ALP is the large number of (Bellman) inequality constraints (equal to the number of states times the number of actions). Since the ALP is in fewer number of variables compared to the LP our intuition suggests that only fewer number of constraints should matter and there is a scope for approximating the constraints as well. For any such constraint approximation/reduction technique it is important to understand and characterize the nature of additional errors introduced. A known technique in literature is constraint sampling de Farias and Roy [2004], Farias and Roy [2006] which involves solving a reduced linear program (RLP) which has a small number of constraints sampled from the constraints of the ALP. de Farias and Roy [2004] presents performance guarantees when the constraints are sampled with respect to the stationary distribution the optimal policy. However, such an ideal assumption on the availability of optimal policies is limiting in practice.

In this paper we propose a constraint approximation method and provide a novel analytical framework to characterize the approximation error. Our specific contributions are listed below:

1. We develop a generalized reduced linear program (GRLP) which has a tractable number of constraints. Each constraint of the GRLP is a positive linear combination of the original constraints of the ALP. This way of constraint generation can be represented as a pre-multiplication of the constraints of the ALP by a matrix $W$ whose entries are all positive. We show that constraint sampling (de Farias and Roy [2004], Farias and Roy [2006]) is a special case of the GRLP.

2. In order to relate $\hat{J}$, the solution to the GRLP, and the optimal value function $J^*$ we develop a novel analytical machinery as follows:
   - We introduce a novel least upper bound (LUB) projection operator $\Gamma$ whose definition involves the feature matrix $\Phi$ used in the ALP. Given a target function $V$, $\Gamma V$ is the least upper bound for $TV$ and is obtained by taking a piecewise minimum of functions chosen from the column span of $\Phi$. The ALP and the LUB operator $\Gamma$ are seen to be closely related via the feature matrix $\Phi$ and the Bellman operator $T$.
   - We also introduce an approximate least upper bound (ALUB) projection operator $\tilde{\Gamma}$ whose definition involves the feature matrix $\Phi$ and the positive matrix $W$. The $\tilde{\Gamma}$ is an approximation to $\Gamma$ and is closely related to GRLP via $\Phi$ and $W$.
   - We observe that the operators $\Gamma$ and $\tilde{\Gamma}$ exhibit the max-norm contraction property with fixed points $\tilde{V}$ and $\hat{V}$, respectively. Quantities $\tilde{V}$ and $\hat{V}$ play a critical role in bounding the approximation error $\|J^* - \hat{J}\|$.

3. We show that $\|J^* - \hat{J}\|_{1,c} \leq (c_1 + c_2)$, where $\|\cdot\|_{1,c}$ is a generalized $L_1$ norm, and $c_1$, $c_2$ are constants. In particular, the term $c_1$ is related to $\|J^* - \Phi r^*\|_{\infty}$, where $\Phi r^*$ is the best approximation
in the max-norm i.e., \( r^* = \arg \min_{r} \| J^* - \Phi r \|_{\infty} \). The term \( c_2 \), on the other hand, is related to the approximation error \( \| \Gamma \bar{J} - \bar{J} J^* \|_{\infty} \), where \( \bar{J} = \Gamma J^* \). Thus \( c_2 \) captures the error due to the projection operator \( \bar{J} \) related to constraint approximation.

4. We make use of the error bounds obtained in this paper to come up with the right GRLP for a widely studied numerical example in controlled queues.

2 Markov Decision Processes

In this section, we briefly discuss the basics of Markov Decision Processes (MDPs) (please refer to Bertsekas [2013], Puterman [1994] for a detailed treatment). 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 \( a \). The reward function \( g \) is a map \( g: S \times A \to \mathbb{R} \) that specifies the reward obtained for performing action \( a \in A \) in state \( s \in S \) and is denoted by \( g_a(s) \).

A policy \( \mu \) specifies the action selection mechanism, and is described by the sequence \( \mu = \{u_1, u_2, \ldots, u_n, \ldots\} \), where \( u_n: S \to A, \ \forall n \geq 0 \). A stationary deterministic policy (SDP) is one where \( u_n \equiv u, \ \forall n \geq 0 \) for some \( u: S \to A \). By abuse of notation we denote the SDP by \( u \) instead of \( \mu \). 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. Given an SDP \( u \), the infinite horizon discounted reward corresponding to state \( s \) under \( u \) is denoted by \( J_u(s) \) and is defined as

\[
J_u(s) \triangleq \mathbb{E} \left[ \sum_{n=0}^{\infty} \alpha^n g_{a_n}(s_n) | s_0 = s, a_n = u(s_n) \ \forall n \geq 0 \right],
\]

(1)

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 \( J_u \triangleq (J_u(s), \forall s \in S) \in \mathbb{R}^n \) is called the value-function corresponding to the SDP \( u \). Under an SDP \( u \), the MDP is a Markov chain with probability transition kernel \( P_u \).

Given an MDP, our aim is to find the optimal policy \( u^* \) which obtains the maximum reward\(^2\). Formally, \( u^*(s) \triangleq \arg \max_{u \in U} J_u(s) \), and the optimal value function \( J^* = J_{u^*} \). The optimal policy and value function obey the Bellman equation (BE) given below: \( \forall s \in S \),

\[
J^*(s) = \max_{a \in A} \{ g_a(s) + \alpha \sum_{s'} p_a(s, s') J^*(s') \},
\]

(2a)

\[
u^*(s) = \arg \max_{a \in A} \{ g_a(s) + \alpha \sum_{s'} p_a(s, s') J^*(s') \}.
\]

(2b)

Typically \( J^* \) is computed first and \( u^* \) is obtained by substituting \( J^* \) in (2b). The Bellman operator \( T: \mathbb{R}^n \to \mathbb{R}^n \) is defined using the model parameters of the MDP as follows:

\[
(T(J))(s) = \max_{a \in A} \{ g_a(s) + \alpha \sum_{s'} p_a(s, s') J(s') \}, \quad \text{where } J \in \mathbb{R}^n.
\]

(3)

\(^2\)Such \( u^* \) exists and is well defined in the case of infinite horizon discounted reward MDP, for more details see Puterman [1994].
3 Approximate Linear Programming Formulation

We now present the linear programming formulation of the MDP which forms the basis for ALP. The LP formulation is obtained by unfurling the max operator in the BE in (2) into a set of linear inequalities as follows:

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

(4)

where $$c \in \mathbb{R}_+^n$$ is any vector whose components are all non-negative. One can show that $$J^*$$ is the solution to (4) Bertsekas [2013]. The LP formulation in (4) can be represented in short as,

$$\min_{J \in \mathbb{R}^n} c^T J$$
$$\text{s.t. } J \geq TJ.$$

(5)

The approximate linear program (ALP) is obtained by letting $$J = \Phi r$$ in (5) and is given as

$$\min_{r \in \mathbb{R}^k} c^T \Phi r$$
$$\text{s.t. } \Phi r \geq T \Phi r,$$

(6)

where $$\Phi = [\phi_1 \ldots \phi_k]$$ is $$n \times k$$ feature matrix with $$\phi_1, \ldots, \phi_k$$ as the $$k'$$'n-dimensional' feature (column) vectors and $$r \in \mathbb{R}^k$$. 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. Note that the ALP is a linear program in $$k (<< n)$$ variables as opposed to the LP in (5) 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.

4 Generalized Reduced Linear Program

The idea of constraint approximation is based on the intuition that since the ALP has fewer number of variables, only a fewer number of constraints should matter. Formally, since the ALP is a linear program in $$k$$ variables, we know that an optimal solution exists on a vertex, and any such vertex is given by $$k$$ active constraints. However, we do not know a priori which of the $$k$$ out of $$nd$$ constraints will be active. de Farias and Roy [2004] suggests constraint sampling, i.e., selecting $$m$$ constraints from the original constraints of the ALP to write down a Reduced Linear Program (RLP), and goes on to establish approximation error bounds when constraints are sampled using the stationary distribution of the optimal policy.

In this paper we take a more generalized view towards constraint approximation in that, instead of just considering a subset of the original constraints, we approximate the feasible set by a bigger set specified by a fewer number of linear constraints. To this end, we define the generalized reduced linear program

---

3. J \geq TJ is a shorthand for the nd constraints in (4). It is also understood that constraints $$(i-1) \times n + 1, \ldots, i \times n$$ correspond to the ith action.
(GRLP) as below:

\[
\min_{r \in \mathbb{R}^k} c^T \Phi r, \\
\text{s.t } W^T \Phi r \geq W^T T \Phi r, \quad W \in \mathbb{R}^{nd \times m}_+ \tag{7}
\]

with \(W\) being an \(nd \times m\) matrix. 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 (7) 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 \(W \in \mathbb{R}^{nd \times m}_+\) is a full rank \(nd \times m\) matrix with all non-negative entries. The first column of the feature matrix \(\Phi\) (i.e., \(\phi_1\)) is \(1^d \in \mathbb{R}^n\) and that \(c = (c(i), i = 1, \ldots, n) \in \mathbb{R}^n\) is a probability distribution, i.e., \(c(i) \geq 0\) and \(\sum_{i=1}^n c(i) = 1\).

The rest of the paper develops analytically various performance bounds and our main result provides a bound for the error between the approximate value function \(\hat{J}_c\) as computed by the GRLP and the optimal value function \(J^*\).

5 Least Upper Bound Projection

The least upper bound (LUB) projection operator \(\Gamma: \mathbb{R}^n \to \mathbb{R}^n\) is defined as below:

Definition 1 Given \(J \in \mathbb{R}^n\), its least upper bound projection is denoted by \(\Gamma J\) and is defined as

\[
(\Gamma J)(i) \triangleq \min_{j = 1, \ldots, k} (\Phi r_{e_j})(i), \quad \forall i = 1, \ldots, n, \tag{8}
\]

where \(V(i)\) denotes the \(i^{th}\) component of the vector \(V \in \mathbb{R}^n\). Also in (8), \(e_j\) 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 (9) for \(c = e_j\).

\[
r_{e_j} \triangleq \min_{r \in \mathbb{R}^k} \Phi r, \\
\text{s.t } \Phi r \geq TJ. \tag{9}
\]

Remark 1

1. The definition of LUB operator \(\Gamma: \mathbb{R}^n \to \mathbb{R}^n\) involves solving \(n\) associated linear programs, i.e., one needs to compute \(r_{e_i}, \forall i = 1, \ldots, n\).

2. Observe that \(\Gamma J \geq TJ\) (follows from the fact that if \(a \geq c\) and \(b \geq c\) then \(\min(a, b) \geq c\), where \(a, b, c \in \mathbb{R}\)).

3. Given \(\Phi\) and \(J \in \mathbb{R}^n\), define \(\mathcal{F} \triangleq \{\Phi r | \Phi r \geq TJ\}\). Thus \(\mathcal{F}\) is the set of all vectors in the span of \(\Phi\) that upper bound \(TJ\). By fixing \(c\) in the linear program in (9) we select a unique vector \(\Phi r_{e_i} \in \mathcal{F}\). The LUB projection operator \(\Gamma\) picks \(n\) vectors \(\Phi r_{e_i}, i = 1, \ldots, n\) from the set \(\mathcal{F}\) and \(\Gamma J\) is obtained by computing their component-wise minimum.

\(^4\)1 is vector with all components equal to 1. This definition of 1 is used throughout the paper.
4. Even though $\Gamma J$ does not belong to the span of $\Phi$, $\Gamma J$ in some sense collates the various best upper bounds that can be obtained via the linear program in (9).

5. The LUB operator $\Gamma$ in (8) bears close similarity to the ALP in (6).

**Definition 2** The best possible LUB projection of $J^*$ is given by $\bar{J} = \Gamma J^*$.

We now characterize the LUB projection operator $\Gamma$ in the following lemmas (all the proofs are presented in the Appendix).

**Lemma 3** Let $r^* \in \mathbb{R}^k$ be defined as $r^* \triangleq \arg\min_{r \in \mathbb{R}^k} ||J^* - \Phi r||_{\infty}$, then

$$||J^* - \bar{J}||_{\infty} \leq 2||J^* - \Phi r^*||_{\infty}. \quad (10)$$

**Theorem 4** The operator $\Gamma: \mathbb{R}^n \rightarrow \mathbb{R}^n$ obeys the max-norm contraction property with factor $\alpha$.

**Corollary 1** The iterative scheme in (11) based on the LUB projection operator $\Gamma$ in (8) converges to a unique fixed point $\tilde{V}$.

$$V_{n+1} = \Gamma V_n, \quad \forall n \geq 0. \quad (11)$$

**Lemma 5** $\tilde{V}$, the unique fixed point of the iterative scheme (11), obeys $\tilde{V} \geq TV$.

**Lemma 6** $\tilde{V}$, the unique fixed point of the iterative scheme (11), and the solution $\hat{J}_c$ to the ALP in (6), obey the relation $\hat{J}_c \geq \tilde{V} \geq J^*$.

**Theorem 7** Let $\tilde{V}$ be the fixed point of the iterative scheme in (11) and let $\bar{J}$ be the best possible projection of $J^*$ as in Definition 2, then

$$||J^* - \tilde{V}||_{\infty} \leq \frac{1}{1 - \alpha}||J^* - \bar{J}||_{\infty}. \quad (12)$$

6 Approximate Least Upper Bound Projection

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

$$(\tilde{\Gamma} J)(i) \triangleq \min_{j=1,\ldots,k} (\Phi r_{e_j})(i), \quad \forall i = 1, \ldots, n, \quad (13)$$

where $r_{e_j}$ is the solution to the linear program\(^3\) in (14) for $c = e_j$, and $e_j$ is same as in Definition 1.

$$r_c \triangleq \min_{r \in \mathbb{R}^k} \Phi r, \quad \text{s.t.} \quad W^\top \Phi r \geq W^\top T J, \quad W \in \mathbb{R}_+^{nd \times m} \quad (14)$$

\(^3\)We assume that the linear program in (14) is bounded for any $c$ satisfying Assumption 1. This technical condition can be taken care by letting $r \in \chi \subset \mathbb{R}^k$, for any bounded set $\chi$ such that $\hat{J}_c \in \chi$ and the analysis presented here will still hold.
Theorem 9 The operator $\tilde{\Gamma}: \mathbb{R}^n \rightarrow \mathbb{R}^n$ obeys the max-norm contraction property with factor $\alpha$ and the following iterative scheme based on the ALUB projection operator $\tilde{\Gamma}$, see (15), converges to a unique fixed point $\hat{V}$.

$$V_{n+1} = \tilde{\Gamma}V_n, \forall n \geq 0.$$ \hspace{1cm} (15)

Lemma 10 The unique fixed point $\hat{V}$ of the iteration in (15) and the solution $\hat{J}_c$ of the GRLP obey $\hat{J}_c \geq \hat{V}$.

Theorem 11 Let $\hat{V}$ be the fixed point of the iterative scheme in (15) and let $\hat{J}$ be the best possible approximation of $J^*$ as in Definition 2, then

$$||J^* - \hat{V}||_\infty \leq \frac{||J^* - \hat{J}||_\infty + ||\Gamma \hat{J} - \tilde{\Gamma} \hat{J}||_\infty}{1 - \alpha}.$$ \hspace{1cm} (16)

Corollary 2 Let $\hat{V}$, $\hat{J}$ be as in Theorem 11 and let $r^* \overset{\Delta}{=} \arg \min_{r \in \mathbb{R}^k} ||J^* - \Phi r||_\infty$ then

$$||J^* - \hat{V}||_\infty \leq \frac{2||J^* - \Phi r^*||_\infty + ||\Gamma \hat{J} - \tilde{\Gamma} \hat{J}||_\infty}{1 - \alpha}.$$ \hspace{1cm} (17)

It is important to note that computing $\Gamma/\tilde{\Gamma}$ and $\hat{V}/\tilde{\hat{V}}$ involve solving $n$ linear programs which is easy when $n$ is small, however, is difficult and impractical when $n$ is large. Nevertheless, we hasten to point out that these quantities are only analytical constructs that lead us to the error bounds.

7 State Relevance Weights of GRLP

The distribution $c$ in (6) and (7) is known as the state relevance weight and signifies the importance of the various states. Lemma 13 is similar to Lemma 3.1 of de Farias and Roy [2003].

Definition 12 We let $|| \cdot ||_{1,c}$ denote the weighted $L_1$ norm with weight vector $c$ as in Assumption 1. Thus for any $w \in \mathbb{R}^n$, $||w||_{1,c} = \sum_{i=1}^n c(i)|w(i)|$.

Lemma 13 $\hat{r} \in \mathbb{R}^k$ is a solution to GRLP in (7) iff it solves the following program:

$$\min_{r \in \mathbb{R}^k} ||\Phi r - \hat{V}||_{1,c}$$

$$s.t \ W^T\Phi r \geq W^T T\Phi r.$$ \hspace{1cm} (18)

Theorem 14 Let $\hat{V}$ be the solution to the iterative scheme in (15) and let $\hat{J}_c = \Phi \hat{r}_c$ be the solution to the GRLP. Let $\hat{J}$ be the best possible approximation to $J^*$ as in Definition 2, and $||\Gamma \hat{J} - \tilde{\Gamma} \hat{J}||_\infty$ be the error due to ALUB projection and let $r^* \overset{\Delta}{=} \arg \min_{r \in \mathbb{R}^k} ||J^* - \Phi r||_\infty$, then

$$||\hat{J}_c - \hat{V}||_{1,c} \leq \frac{4||J^* - \Phi r^*||_\infty + ||\Gamma \hat{J} - \tilde{\Gamma} \hat{J}||_\infty}{1 - \alpha}.$$ \hspace{1cm} (19)
7.1 Main Result

**Corollary 3** Let \( J_c, V, r^* \) and \( J^* \) be as in Theorem 14, then

\[
||J^* - J_c||_{1,c} \leq \frac{6||J^* - \Phi r^*||_{\infty}}{1-\alpha} + 2||\Gamma J - \tilde{\Gamma} J||_{\infty}. \tag{20}
\]

There are couple of interesting observations about the main result in Corollary 3.

1. The error term is split into two factors, the first of which is related to the best possible projection and the second factor is related to constraint approximation. The second factor \( ||\Gamma J - \tilde{\Gamma} J||_{\infty} \) is completely defined in terms of \( \Phi, W \) and \( T \), and does not require knowledge of stationary distribution of the optimal policy. It makes intuitive sense since given that \( \Phi \) is suited to \( J^* \), it is enough for \( W \) to be suited to \( \Phi \) and \( T \) without any additional requirements.

2. The error terms rely on the \textit{max}-norm contraction property of operators \( \Gamma \) and \( \tilde{\Gamma} \) and not on concentration inequalities as in the case of RLP in de Farias and Roy [2004]. Also quantities \( \tilde{V} \), and \( \hat{V} \) do not play a direct role in the final bound in (20).

For any given problem one can arrive at the right ALP in a step by step manner based on (20).

Choose right \( c \) & \( \Phi \) to reduce \( ||J^* - \Phi r^*||_{\infty} \) → Obtain the right ALP → Choose a good \( W \) by computing \( ||\Gamma J - \tilde{\Gamma} J||_{\infty} \) for various \( W \)s → Arrive at the right GRLP

8 Example

We take up an example in the domain of controlled queues and demonstrate the usefulness of the bounds developed in this paper to come up with the right GRLP. The queuing model we discuss 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 state at time \( t \). The state at time \( t + 1 \) when action \( a_t \in A \) is chosen is given by
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 $g_a(s) = -(s + 60q(a)^3)$.

From the final error bound in (20) it is clear that for the GRLP to be effective we need to choose $\Phi$ and $W$ carefully. We make use of polynomial features in $\Phi$ (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 (20). We chose a $W$ matrix denoted by $W_G$ (where G stands for good) and its entries are described as follows: $\forall i = 1, \ldots, m$,

$$W_G(i,j) = 1, \forall j \text{ s.t } j = (i-1) \times \frac{n}{m} + k + (l-1) \times n, k = 1, \ldots, \frac{n}{m}, l = 1, \ldots, d,$$

$$= 0, \text{ otherwise.} \quad (21)$$

The idea behind the above definition of $W_G$ is to average constraints corresponding to adjacent states. For the sake of comparison we compute $\|\Gamma J - \bar{\Gamma} J\|_\infty$ for $W_G$ (in (21)) and compare with random choices of W matrix. Though computing $\|\Gamma J - \bar{\Gamma} J\|_\infty$ might be hard in the case of large $n$, since $\|\Gamma J - \bar{\Gamma} J\|_\infty$ is completely dependent on the structure of $\Phi$, $T$ 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 $Q_S$, $W_G$ ((21) with $m = 5$) turns out to be a $20 \times 5$ 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. An illustration of the feasible region and the various associated constraints are presented in the Appendix. It is clear from Table 1 that $W_G$ performs much better than randomly generated positive matrices. Also, note that a higher $\|\Gamma J - \bar{\Gamma} J\|_\infty$ implies a higher $\|J^* - J_c\|_{1,c}$.

<table>
<thead>
<tr>
<th>Choice of W</th>
<th>$|J^* - V|_\infty$</th>
<th>$|J^* - \bar{V}|_\infty$</th>
<th>$|\Gamma J - \bar{\Gamma} J|_\infty$</th>
<th>$|J^* - J_c|_{1,c}$</th>
<th>$|J^* - J_c|_{1,c}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$W_G$ (21)</td>
<td>42.25</td>
<td>172.4</td>
<td>54.15</td>
<td>97.27</td>
<td>29.43</td>
</tr>
<tr>
<td>RANDOM</td>
<td>42.25</td>
<td>1366</td>
<td>251.83</td>
<td>97.27</td>
<td>112</td>
</tr>
</tbody>
</table>

Table 1: Shows various error terms for $Q_S$. Since $n = 10$ we let $c$ to have a uniform distribution (see Appendix), the values of $\|J^* - \bar{V}\|_\infty$ and $\|J^* - J_c\|_{1,c}$ are same across $W_G$ and the random matrices since they correspond to $\bar{\Gamma}$ operator and ALP respectively and are independent of $W$ matrix.

Having validated the choice of $W$ on $Q_S$ we then consider a larger queuing system (denoted by) $Q_L$ with $n = 1000$ 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 $W_G$ (21) with $m = 50$. We set $c(s) = (1 - \zeta)^s$, $\forall s = 1, \ldots, 999$, with $\zeta = 0.9$ and $\zeta = 0.999^7$, respectively.

The results in Table 2 show that performance exhibited by $W_G$ is better by several orders of magnitude over ‘Random’ in the case of the large system $Q_L$. Also note that the better performance exhibited by $W_G$ in the small system $Q_S$ continues in the case of the large system $Q_L$ as well. We refer the reader to

\footnotesize

6 Averaged across 10 random matrices, in case of both $Q_S$ and $Q_L$.

7 Please see Appendix for the rationale behind the two choices of $\zeta$. 

<table>
<thead>
<tr>
<th>$\zeta$</th>
<th>$|J^* - J_c|_{1,c}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9</td>
<td>29.43</td>
</tr>
<tr>
<td>0.999^7</td>
<td>112</td>
</tr>
</tbody>
</table>
negative of value function \(-J^* - \hat{\tilde{V}} - \hat{\tilde{J}} c - \hat{\tilde{J}} c\).

Figure 3: Plot corresponding to \(Q_s\) on the left and \(Q_L\) on the right.

<table>
<thead>
<tr>
<th>Choice of (W)</th>
<th>(|J^* - J_c|_{1,c}), with (\zeta = 0.9)</th>
<th>(|J^* - J_c|_{1,c}), with (\zeta = 0.999)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(W_G) (21) with (m = 50)</td>
<td>220</td>
<td>82</td>
</tr>
<tr>
<td>RANDOM</td>
<td>(5.04 \times 10^3)</td>
<td>(1.25 \times 10^4)</td>
</tr>
</tbody>
</table>

Table 2: Shows performance metrics for \(Q_L\).

the Appendix for more results on this example as well as detailed description of the setting and choice of parameters.

9 Discussion and Conclusion

The error bounds are in terms of \(\|J^* - \Phi r^*\|_{\infty}\) and \(\|\Gamma \tilde{J} - \hat{\Gamma} \tilde{J}\|_{\infty}\) and it can be argued that the basis functions might not approximate \(J^*\) uniformly over all the states. This problem can be alleviated easily by making use of Lyapunov functions as in de Farias and Roy [2003] and showing that the operators \(\Gamma\) and \(\hat{\Gamma}\) are contraction maps in a modified \(L_\infty\) norm. Since such a procedure requires additional algebra we have omitted its discussion in this paper and will provide the same in a longer version.

An important aspect of the GRLP is its amenability to the reinforcement learning (RL) Sutton and Barto [1998] setting. In the RL setting, the model information is limited to the knowledge of the sample trajectories and solution can be obtained by a primal/dual gradient scheme that makes use of the Lagrangian function. The Lagrangian function corresponding to ALP and GRLP can be written as

\[
\hat{\tilde{L}}(r, \lambda) = c^T \Phi r + \lambda^T (T \Phi r - \Phi r), \quad \text{and} \quad \hat{\tilde{L}}(r, q) = c^T \Phi r + q^T W^T (T \Phi r - \Phi r),
\]

respectively. The additional insight from (22) is that the GRLP can also be interpreted as linear function approximation of the Lagrangian multipliers, i.e., \(\lambda \approx W q\). Dolgov and Durfee [2006] takes such a view to formulate an approximate dual linear program (ADLP) as a dimensionality reduction technique. However, Dolgov and Durfee [2006] provides neither an error analysis for the ADLP nor an RL algorithm. de Farias and Roy [2003] deals with an RL algorithm based on the ALP, nevertheless, the Lagrangian multipliers are approximated by a non-linear function approximator.

An interesting direction is to extend the analytical machinery developed in this paper to constraint relaxation methods such as the smoothed approximate linear program (SALP) in Desai et al. [2009]. The key will be to appropriately modify the ALUB projection operator \(\hat{\Gamma}\) to suit the SALP. It will also be interesting to find the connection between the analytical arguments presented here based on contraction
maps and the ones in de Farias and Roy [2004] based on concentration inequalities. It is well known that constraint sampling works in the case of large MDPs like Tetris Farias and Roy [2006]. Since the RLP is a special case of GRLP the results obtained in this paper further bolster the validity of such a solution approach for large MDPs.
References


