A Generalized Reduced Linear Program for Markov Decision Processes

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

Abstract

We present *the first* sufficient conditions that guarantee stability of two-timescale stochastic approximation. Our analysis is based on the ordinary differential equation (ODE) method and is an extension of the results in **?** for single-timescale recursions. Our results extend the stability analysis for one-timescale schemes to multi-timescale stochastic approximation algorithms. As an application of our result, we show the stability of iterates in a two-timescale stochastic approximation algorithm arising in reinforcement learning.

Introduction

I. Markov Decision Processes

Markov Decision Process (MDP) is an useful framework to cast problems involving optimal sequential decision making under uncertainity. Such problems often occur in science, engineering and economics, with planning by autonomous agents, control of large queuing systems and inventory control being some specific examples. Informally, the decision maker/planner wishes to maximize its objective, and in order to achieve this end, needs to perform an appropriate *action/control* at each decision epoch based on the *state* of the underlying *system*. The sequence of actions/control is also known as *policy* and the planner is interested in computing the *optimal* policy. In most cases, in order to compute the optimal policy, the planner also needs to *predict* the value of each state of any given control. Thus, with any given MDP, there are two natural sub-problems namely the problem of prediction and the problem of control.

Dynamic Programming (DP) is a general principle to solve MDPs, the core of which constitutes the *Bellman* equation (BE). The Bellman equation relates the optimal policy to its corresponding value known as the optimal *value function*, i.e., it relates the prediction and control problems of a given MDP. A host of analytical and iterative methods exists to solve the Bellman equation for MDPs. These methods are known as exact dynamic programming methods since they spell out the optimal control and its value function exactly.

The three important exact DP methods are

- Value Iteration.
- Policy Iteration.
- Linear Programming.

Of these methods, value iteration and linear programming are termed as value function based DP methods since they compute the optimal value function first and the optimal policy is obtained by substituting it in the Bellman Equation. On the other hand, policy iteration predicts the value of a policy at each iterative step and then improves it to yeild a better policy in the next step eventually converging to the optimal policy. It is important to emphasize that these exact DP methods solve both the control and prediction problems.

II. Practical Issues

MDPs arising in practical problems are faced with two important issues. First issue is due to the *curse-of-dimensionality* (or simply the *curse*), a term which signifies the fact that the number of states of the MDP grows exponentially in the number of state variables. As the number of state variables increase, it becomes difficult to solve the MDP employing exact DP methods due to the computational overhead involved. Moreover, it also difficult to store and retrieve the exact value function and policy when the number of states are large.

The second issue is the lack of model information, wherein, the the model parameters of the MDP are not available explicitly. However, the the underlying system can be simulator or sample trajectories via direct interaction with the system. This scenario is known as the *reinforcement learning* setting since the model parameters have to be *learned* by using the feedback obtained via direct interaction with the environment.

A wide range of methods/algorithms exist in literature to address the issues of the curse and the lack of model information. In particular, the methods that tackle the curse are broadly termed as

approximate dynamic programming methods and methods that handle the case of lack of model information are called reinforcement learning algorithms.

III. Approximate Dynamic Programming

Approximate Dynamic Programming (ADP) is the name given to a class of approximate solution methods that tackle the curse. A major bottleneck the ADP methods face in the case of MDPs having a large number of states is that the value corresponding to each and every state cannot be stored, retrieved and computed. Most often ADP methods adopt an approximation architecture that enables compact representation of the value function in higher dimension so as to ease the storage and computation. Once the approximation architecture is fixed it is also important to devise a scheme that will choose the right candidate function that approximates the value function well. Thus central to any ADP method are

1) Approximation architecture.
2) Scheme to compute the right function within the chosen approximation architecture.

In most cases, ADP methods are obtained by combining exact DP methods with a given approximation architecture. A given ADP scheme can be said to belong to either of two approaches based on the way the prediction and the control problems are addressed. The two distinct approaches are:

1) The *value function* based approach, wherein a direct approximation to the optimal value function is obtained and a *greedy/sub-optimal* policy is computed by substituing the approximate value function in the Bellman equation.
2) The *approximate policy iteration* based approach, wherein, the critic evaluates the current policy, i.e., computes an approximation of the value function of the current policy. The actor on the other hand, makes use of the approximate value function to imporve the current policy.

Further, the performance of a given ADP method can be quantified by the following metrics namely

- Prediction Error, i.e., the difference between the exact value function and the approximate value function.
- Control Error, i.e., the loss in performance due to the sub-optimal policy as compared to the optimal policy.

It is a known result that if the prediction error is bounded in the $\max$-norm (or L_∞-norm), the control error can be bounded as well. The necessacity of L_∞-norm arises due to the fact that the Bellman operator is only a contraction map in the L_∞ norm.

An ADP method is said to address both the prediction and control problems if it can offer guaranteed performance levels measured in terms of the aforementioned metrics. Given an ADP method, it a major theoretical challenge to come up with an analytical expression to bound the aforementioned performance metrics and it is in particlar difficult to bound the control error in many ADP schemes.

IV. Linear Function Approximation

The first step in any ADP method is the choice of approximation architecture. In particular, a compact way of representing an approximation for the value function is necessary in most cases. A paramtererized function class is an useful approximation architecture since every function can be specified by the parameter it is enough to store only the parameters. Computing the right function that best approximates the exact value function then boils down to computations involving only the parameters and the curse can be tackled by choosing the number of parameters to be much lesser than the number of states. The most widely used parameterized class is the linear function approximator (LFA). Under a LFA, each function is written as a linear combination of pre-selected *basis* functions. The exact value function is then approximated by computing/learning the weights of the various basis functions in the linear combination.

The method used to compute the right weights of the linear combination affects the quality of the approximation. The projected Bellman equation and the approximate linear programming are two different ways of computing the weights. These two differing approaches have their advantages/disadvantages and provide interesting research problems.

A. Projected Bellman Equation

Once the basis functions of the LFA have been fixed, the right candidate function from the LFA has to be picked as the approximate value function. A candidate for the approximate value function could be that function which is at the least distance from the exact value function. Minimizing the distance between the linear combination of basis functions and the exact value function is similar to the idea of conventional *linear regression* wherein, the target function is *projected* onto the linear sub-space spanned by the basis functions. However, the idea of

linear regression cannot be applied in a straightforward manner to compute the approximate value function because the exact value function (which is the target function in this case) is not known.

The Projected Bellman Equation (PBE) combines the idea of linear least squares projection and the Bellman equation. The central idea underlying the PBE is to find a fixed point in the linear sub-space spanned by the basis functions. However, existence of such a fixed point can be ensured only under a restricted setting. In particular, the Bellam operator is *non-linear* and is a contraction map in the $\max$-norm (L_∞-norm). On the other hand, the linear least squares projection operator is non-expansive only in the L_2-norm. Hence the Bellman operator cannot be combined as such with the least squares projection operator. Neverthless, this issue can be side stepped by considering the Bellman operator restricted to a given policy which can be shown to be a contraction map in a generalized L_2-norm. The Bellman operator restricted to a given policy can be combined with the linear least squares projection operator to find a fixed point. However, such a fixed point can approximate only the value function of the particular policy (with respect to which the Bellman operator has been restricted) and not the optimal value function. As a consequence, the PBE can only be used for approximate policy evaluation, i.e., it can be used to compute an approximation to the value function of a given policy. This means that the PBE based methods fall into the category of approximate policy iteration approach, i.e., the approximate policy evaluation should be followed by a policy improvement step. However, there is a 'norm-mismatch' between the projection operator that minimizes the error in L_2-norm and the $\max/L_\infty$-norm required to guarantee policy improvement. As a result, the sub-optimal policy obtained from the approximate value function computed by the PBE is not guaranteed to be an improvement.

The major disadvantage of the PBE based methods is that they do not address the control problem completely. In fact, there are simple MDPs for which the PBE based solution methods are known to produce a sequence of policies that oscillate within a set of bad policies. This phenomenon is called as *policy-chattering* and is a direct consequence of the norm mismatch.

Problem 1: Projected Bellman Equation in the $(\min, +)$ linear basis

It is known that **?** the issue of norm-mismatch can be alleviated if the projection operator preserves *monotonicity*, i.e., if given two functions with one of them greater (component-wise) that the other, then the same holds for their projections as well. It is also known that the projection operator arising in the $(\min, +)$ linear algebra preserves this monotonicity property. The first part

of the thesis deals with developing convergent ADP methods based on $(\min, +)$ linear algebra. In particular, the monotonicity property helps in

- Building convergent value function based ADP methods. Since the optimal value function is approximated directly, the issue of policy-chattering is absent.
- Providing performance guarantees for the greedy/sub-optimal policy. This is possible since there is no issue of 'norm-mismatch'.

B. Approximate Linear Programming

The approximate linear programming (ALP) formulation is obtained by introducing linear function approximation in the linear programming formulation (LP) of the given MDP. In contrast to the PBE, the ALP does not rely on the linear least squares projection operator, but instead, computes the approximate value function by optimizing a linear objective over a set of linear inequality constraints. In particular, the ALP restricts its search to a space of linear functions that upper bound the optimal value function. The ALP is a value function method as it computes an approximation to the optimal value function. Since, the corresponding greedy policy can be derived in a straightforward manner, the ALP does not suffer from issues of policy-chattering. Further, the ALP also offers good performance guarantees for both the prediction as well as the control problems, and thus addresses both the problems.

A significant shortcoming of the ALP formulation is that the number of constraints is of the order of the state space. Most MDPs arising in practice have a large number of constraints and hence it is always not possible to solve the ALP with all the constraints. However, there are some special cases of factored MDPs with factored value function representations that enable elimination of variables to come up with a tractable number of constraints. A general approach is to handle the large number of constraints is constraint sampling, wherein a Reduced Linear Program (RLP) is formulated by sampling fewer number of constraints from the orginial constraints of the ALP. Though the RLP is known to perform well in experiments, the theoretical analysis is available only for a specific RLP formulated under idealized settings.

Problem 2: Framework to analyse constraint reduction in ALP:

The second part of the thesis deals with developing a novel theoretical framework to analyse constraint reduction/approximation in the ALP. The framework is built around studying a Generalized Reduced Linear Program (GRLP) whose constraints are obtained as positive linear combinations of the original constraints of the ALP. The salient features of the framework are

1) The analysis is based on two novel contraction maps and the error bounds are provided in a modified L_∞-norm. Both the prediction and control error are bounded, thus making the GRLP a complete ADP scheme.
2) Justification of linear function approximation of the Lagrange multipliers associated with the constraints of the ALP. This is a desirable outcome, since both the primal and dual variables have linear function approximation.

V. Reinforcement Learning

Reinforcement Learning (RL) algorithms can be viewed as 'on-line'/sample trajectory based solution methods for solving MDPs. In the case of MDPs with a large number of states, RL algorithms are obtained as on-line versions of ADP methods. In order to handle the noise in the sample trajectory RL algorithms make use of stochastic approximation (SA). Typically, RL algorithm employing stochastic approximation are iterative schemes which take a small *step* towards the desired value at each iteration. By making the right choice of the *step-size* schedule effect of noise can be nullified and convergence to the desired value can be guaranteed.

Actor-Critic algorithms form an important sub-class of RL algorithms, wherein, the critic is responsible for policy evalutaion and the actor is responsible for policy improvement. In order to tackle the curse, the actor-critic schemes parameterize the policy and the value function. In an actor-critic algorithm, the critic forms the inner-loop and the actor constitutes the outer-loop. This is due to the fact that the actor has to needs the critic to evaluate a given policy before the actor updates the policy parameters. This effect of having two seperate loops can instead by achieved in practice by adopting different *step-size* schedules for the actor and the critic. Specifically, the step-sizes used by the actor updates have to be much smaller than those used in the critic updates.

Stochastic approximation schemes that use different step-size schedules for different sets of iterates are known as multi timescale stochastic approximation schemes. The conditions under which the iterates of a multi timescale SA schemes converge to the desired value have been studied in literature. One of the conditions required to ensure the convergence of the iterates of a multi timescale SA scheme is that the iterates need to be stable, i.e., they should be bounded. However, the conditions that *imply* the stability of the iterates in a multi timescale SA scheme have not been understood.

Problem 3: Stability Criterion for Two Timescale Stochastic Approximation Schemes:

The third part of the thesis deals with providing conditions under which the stability of iterates in a two timescale stochastic approximation scheme follows. Salient features of the contribution are as follows:

1) The analysis is based on the ODE approach to stochastic approximation.
2) Stability of iterates of important actor-critic algorithms follow for the result.

Markov Decision Processes (MDPs) In this section, we briefly discuss the basics of Markov Decision Processes (MDPs) (the reader is referred to **??** for a detailed treatment).

An MDP is a 4-tuple $\langle S, A, P, g\rangle$, 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. We also assume that the number of actions is finite. Without the loss of generality, we set $S = \{1, 2, \ldots, n\}$ 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 $g_a(s)$.

By a policy, we mean a sequence $\pi = \{\pi_0, \ldots, \pi_n, \ldots\}$ of functions $\pi_i, i \geq 0$ that describe the manner in which an action is picked in a given state at time i. The most general class of policies is the class of *history-dependent randomized policies* denoted by π and is defined below.

Definition 1: Let $h_n = \{s_0, \ldots, s_n, \pi_0, \ldots, \pi_{n-1}\}$ denote the *history*. A policy π is said to be a *history-dependent randomized* policy if $\pi_n, n \geq 0$ is such that for every $s \in S$, $\pi_n(s, \cdot|h_n)$ is a probability distribution over the set of actions.

We denote the class of policies by Π. The infinite horizon discounted reward corresponding to state s under a policy π is denoted by $J_\pi(s)$ and is defined by

$$J_\pi(s) \stackrel{\Delta}{=} \mathbf{E}[\sum_{n=0}^{\infty} \alpha^n g_{a_n}(s_n)|\pi],$$

where $a_n \sim \pi_n(s, \cdot)$ and $\alpha \in (0, 1)$ is a given discount factor. Here $J_\pi(s)$ is known as the value of the state s under the policy π, and the vector quantity $J_\pi \stackrel{\Delta}{=} (J_\pi(s))_{s\in S} \in R^n$ is called the value-function corresponding to the policy π. The optimal value function J^* is defined as $J^*(s) \stackrel{def}{=} \max_{\pi\in\Pi} J_\pi(s)$.

A stationary deterministic policy (SDP) is one where $\pi_n \equiv u$ for all $n \geq 0$ for some $u \colon S \to A$. By abuse of notation we denote the SDP by u itself instead of π. In the setting that we consider, one can find an SDP that is optimal **??**, i.e., there exists an SDP u^* such that $J_{u^*}(s) = J^*(s), \forall s \in S$. In this paper, we restrict our focus to the class U of SDPs and without loss of generality we

assume that there exists a unique SDP u^* that is optimal. Under a stationary policy u (or π), the MDP is a Markov chain and we denote its probability transition kernel by $P_u = (p_{u(i)}(i,j), i,j = 1,\ldots,n)$ (or $P_\pi = (p_{\pi(i)}(i,j), i,j = 1,\ldots,n)$, where $p_{\pi(i)}(i,j) = \sum_{a\in A}\pi(i,a)p_a(i,j)$ and $\pi(i) = (\pi(i,a), a \in A)$).

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

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

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

If J^* is computed first, then u^* can be obtained by substituting J^* in (**??**). The Bellman operator T is defined using the model parameters of the MDP as follows:

Definition 2: The Bellman operator $T : \mathrm{R}^n \to \mathrm{R}^n$ is defined by

$$(TJ)(s) = \max_{a\in A}\left(g_a(s) + \alpha\sum_{s'} p_a(s,s')J(s')\right), \tag{2}$$

where $J \in \mathrm{R}^n$ and $J_s = J(s)$, $s \in S = \{1,\ldots,n\}$. Similarly one can define the Bellman operator $T_u : \mathrm{R}^n \to \mathrm{R}^n$ restricted to an SDP u as follows:

$$(T_uJ)(s) = g_{u(s)}(s) + \alpha\sum_{s'} p_{u(s)}(s,s')J(s')\,. \tag{3}$$

Given $J \in \mathrm{R}^n$, TJ is the 'one-step' greedy value function. It is also useful to define the notion of a one-step greedy policy as below:

Definition 3: A policy $\tilde{u}$ is said to be greedy with respect to $\tilde{J}$ if

$$\tilde{u}(s) = \arg\max_{a\in A}\big(g_a(s) + \alpha\sum_{s'} p_a(s,s')\tilde{J}(s')\big) \tag{4}$$

We also define the Bellman operator H for action values **?**:

Definition 4: Let $H : \mathrm{R}^n \to \mathrm{R}^{nd}$ be defined as follows: For $J \in \mathrm{R}^n$,

$$HJ = \begin{bmatrix} H_1J \\ \vdots \\ H_dJ \end{bmatrix} \in \mathrm{R}^{nd}, \text{where}$$

$$(H_aJ)(s) = g_a(s) + \alpha\sum_{s'} p_a(s,s')J(s'), \qquad s \in S, a \in A. \tag{5}$$

We now state without proof the important properties related to the Bellman operator. Though in these results, we make use of the Bellman operator T, the results also trivially hold for T_u as well.

A. Properties of T

Lemma 5: T is a $\max$-norm contraction operator, i.e., given $J_1, J_2 \in \mathbf{R}^n$,

$$||TJ_1 - TJ_2||_\infty \leq \alpha ||J_1 - J_2||_\infty. \tag{6}$$

Corollary 1: J^* is a unique fixed point of T, i.e., $J^* = TJ^*$.

The Bellman operator T exhibits two more important properties presented in the following lemmas (see **?** for proofs):

Lemma 6: T is a monotone map, i.e., given $J_1, J_2 \in \mathrm{R}^n$ such that $J_2 \geq J_1$, we have $TJ_2 \geq TJ_1$. Further, if $J \in \mathrm{R}^n$ is such that $J \geq TJ$, it follows that $J \geq J^*$.

Lemma 7: Given $J \in \mathrm{R}^n$, $t \in \mathrm{R}$ and $\mathbf{1} \in \mathrm{R}^n$, a vector with all entries 1, we have

$$T(J + t\mathbf{1}) = TJ + \alpha t \mathbf{1}. \tag{7}$$

Note that Lemmas **??**-**??** also hold for the Bellman operator H defined for the action values in Definition **??**.

Solving an MDP involves handling two sub-problems namely the problem of *control* and the problem of *prediction*. The problem of *control* deals with coming up with a good (and if possible the optimal) policy. Often, in order to solve the problem of *control*, one needs to solve the problem of *prediction*, which deals with computing the value function J_u of the policy u. The fact that the two problems are related is reflected in the Bellman equation in (**??**), where J^* from (**??**) is used in (**??**) to obtain the optimal policy u^*. Thus, the Bellman equation is at the heart of the solution methods to MDPs. Any solution method to MDP is said to be complete only if it satisfactorily (with provable performance guarantees) addresses both the prediction and the control problems.

The basic solution methods namely value iteration, policy iteration and linear programming (LP) formulation **?** solve both the control and prediction problems. Of the three basic methods, in this paper, we are interested in the LP formulation given by

$$\begin{aligned} &\min_{J \in \mathrm{R}^n} c^\top J \\ &\quad \text{s.t } J(s) \geq g_a(s) + \alpha \sum_{s'} p_a(s, s') J(s'), \qquad s \in S, a \in A, \end{aligned} \tag{8}$$

where $c \in \mathrm{R}^n_+$ is any vector whose components are all non-negative. One can show that J^* is the solution to the LP formulation (**??**) **?**. Also, of the three methods, value iteration and LP formulation are value function based methods, i.e., they compute J^* directly and then u^* is

obtained by plugging J^* in (**??**).

While the basic methods (i.e., VI, PI and LP) can be used to compute exact values J^* and u^* for MDPs with a small number of states, they are computationally expensive in the case of MDPs with a large number of states.

VI. Approximate Dynamic Programming

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 arising in practical applications suffer from the curse, i.e., have large number of states and it is difficult to compute $J^*/J_u \in \mathrm{R}^n$ exactly in such scenarios. Approximate dynamic programming (ADP) **????** methods compute an approximate value function $\tilde{J}$ instead of J^*/J_u. In order to tackle the curse, ADP methods resort to dimensionality reduction by parameterizing the value function and/or the policy. Value-function based ADP methods form an important subclass of ADP methods whose brief overview is given below.

A. Value-Function based ADP

In the value-function based ADP schemes a parameterized family is chosen and the approximate value-function $\tilde{J}$ is picked from the chosen parameterized class. Once the approximate value function $\tilde{J}$ is computed, a sub-optimal policy $\tilde{u}$ can be obtained as the one-step greedy policy with respect to $\tilde{J}$ by making use of (**??**). The following lemma characterizes the degree of sub-optimality of the greedy policy $\tilde{u}$.

Lemma 8: Let $\tilde{J}$ be the approximate value function and $\tilde{u}$ be as in (**??**), then

$$||J^* - J_{\tilde{u}}||_\infty \leq \frac{2}{1-\alpha}||J^* - \tilde{J}||_\infty. \tag{9}$$

The quality of any ADP method depends on the approximation guarantees it offers for the quantities $||J^* - \tilde{J}||$ and $||J^* - J_{\tilde{u}}||$, where $||\cdot||$ is an appropriate norm. The term $||J^* - \tilde{J}||$ denotes the error in prediction and $||J^* - J_{\tilde{u}}||$ denotes the loss in performance resulting from the sub-optimal policy $\tilde{u}$ with respect to the optimal policy u^*. Of the two error terms, $||J^* - J_{\tilde{u}}||$ is more important because ultimately we are interested in coming up with a useful policy. In the context of ADP methods, the control and prediction problems are said to be addressed when the error terms $||J^* - \tilde{J}||$ and $||J^* - J_{\tilde{u}}||$ are bounded by "small" constants.

B. Linear Function Approximation

The most widely used parameterized class to approximate the value-function is the linear function approximator (LFA). Under LFA, the value-function is approximated as $\tilde{J} = \Phi r^*$, with $\Phi = [\phi_1 | \dots | \phi_k]$ being an $n \times k$ feature matrix and r^* is the parameter to be learnt.

There are two important approaches to value function approximation. Both the approaches start out with a basic solution method and appropriately introduce function approximation in it. The two approaches are:

1) The Projected Bellman Equation (PBE) which combines the BE and the linear least squares projection operator to project high dimensional quantities onto the subspace of the LFA.
2) The approximate linear programming formulation which introduces the LFA in the linear programming formulation.

A host of ADP methods are based on the PBE and have been found to be useful in practice. The main application of the PBE is for approximate policy evaluation, i.e., to compute $\tilde{J}_u$, an approximation to the value function J_u of policy u. Due to the mismatch in the norms, i.e, the linear least squares projection operator based on the L_2-norm and the L_∞-norm of the Bellman operator T, one cannot use the PBE to obtain a direct approximation to J^*. Thus in order to solve the problem of control, $\tilde{J}_u$ is used to compute a one-step greedy policy. However, again due to the mismatch in the norms, i.e., L_2-norm of the linear least squares projection and the L_∞ norm required for policy improvement (Lemma **??**), the one-step greedy policy need not necessarily be an improvement. This leads to a phenomenon called *policy-chattering* **?** where looping within of bad policies can occur. Further, such policy-chattering can be demonstrated in simple examples as well **?**. Thus, though the approximate value function obtained by solving the PBE offers guarantees for prediction it does not offer any guarantees for the control problem, a significant shortcoming of the PBE based methods.

The ALP formulation **?** on the other hand does not suffer from issues such as policy-chattering, for the simple reason that it computes $\tilde{J}$ which is an approximation to J^* and a one-step greedy policy $\tilde{u}$ obtained using $\tilde{J}$. In short, since there is only one policy that the ALP outputs there is no question of policy-chattering. Further, the ALP offers performance guarantees for both the error terms $||J^* - \tilde{J}||$ and $||J^* - J_{\tilde{u}}||$. Though the ALP is a complete method addressing both the control and prediction problems, it nevertheless suffers from an important limitation in the form of large number of constraints (as large as the size of the state space). This limitation

has been addressed in literature by sampling only a fewer tractable constraints to formulate a reduced linear program (RLP). The RLP has been shown to perform well in practice **???**, but theoretical performance guarantees **?** are available for a specific RLP obtained under idealized assumptions. In this paper, by providing a sound theoretical analysis of the RLP, we aim to show that RLP is a complete method that addresses both the prediction and the control problems. We achieve this by developing and presenting a comprehensive theoretical framework to understand the constraint reduction/approximation procedure.

In the next section, we discuss the approximate linear programming (ALP) formulation, the basic results and present prior results in literature as well as motivate the problem that we address in this paper.

VII. Approximate Linear Programming

The LP formulation in (**??**) can be represented in short as,

$$\min_{J\in\mathrm{R}^n} c^\top J$$
$$\text{s.t } J \geq TJ, \tag{10}$$

or

$$\min_{J\in\mathrm{R}^n} c^\top J$$
$$\text{s.t } EJ \geq HJ, \tag{11}$$
$$\tag{12}$$

where $J \geq TJ$ is a shorthand for the nd constraints in (**??**) and E is an $nd \times n$ matrix given by $E = [I, \ldots, I]^\top$, i.e., E is obtained by stacking d identical $n \times n$ identity matrices one over the other. Note that (**??**) and (**??**) are identical programs and differ only in notation. We use notation of type (**??**) whenever we prefer brevity and we use notation (**??**) in some definitions and proof for the sake of clarity. The approximate linear program (ALP) is obtained by making use of LFA in the LP, i.e., by letting $J = \Phi r$ in (**??**) and is given as

$$\min_{r\in\mathrm{R}^k} c^\top \Phi r$$
$$\text{s.t } \Phi r \geq T\Phi r. \tag{13}$$

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. We now state the assumptions and definitions used for the rest of the paper.

Assumption 1: The first column of the feature matrix Φ (i.e., ϕ_1) is $\mathbf{1} \in \mathrm{R}^n$. In other words, the constant function is part of the basis.

Assumption 2: $c = (c(i), i = 1, \ldots, n) \in \mathrm{R}^n$ is a probability distribution, i.e., $c(i) \geq 0$ and $\sum_{i=1}^{n} c(i) = 1$.

Definition 9: Given a function $\chi \colon S \to \mathrm{R}^+$ we define the quantity β_χ as

$$\beta_\chi = \max_{s \in S} \frac{\max_{a \in A}\big(\alpha \sum_{s'} p_a(s, s')\chi(s')\big)}{\chi(s)}. \tag{14}$$

Definition 10: The function χ is then said to be a *Lyapunov* function if $\beta_\chi < 1$.

Assumption 3: $\psi \colon S \to \mathrm{R}^+$ is a Lyapunov function and is present in the column span of the feature matrix Φ.

It is straightforward to check that the function $\mathbf{1}$ is a Lyapunov function and trivially is present in the column span of Φ.

Definition 11: The modified L_1-norm is defined as

$$||J||_{1,c} = \sum_{s \in S} c(s)|J(s)|, \tag{15}$$

where c obeys Assumption **??**.

Definition 12: The modified L_∞-norm is defined as follows:

$$||J||_{\infty,\rho} = \max_{s \in S} \rho(s)|J(s)|, \tag{16}$$

where $\rho \colon \mathcal{S} \to \mathrm{R}^+$, $J \in \mathrm{R}^n$.

In Definition **??**, the use of the weighting function ρ allows us to measure the error taking into account the relative importance of the various states. A lower value of $\rho(s)$ means that the state s is less important and vice-versa.

We recall below Theorem 4.2 of **?** which bounds the error in the approximate value function.

Theorem 13: Let $\tilde{r}_c$ be the solution to the ALP in (**??**), $\tilde{J}_c = \Phi\tilde{r}_c$, ψ be a Lyapunov function and c be a distribution as in (**??**), then

$$||J^* - \tilde{J}_c||_{1,c} \leq \frac{2c^\top \psi}{1 - \beta_\psi} \min_r ||J^* - \Phi r||_{\infty,1/\psi}. \tag{17}$$

We now recall Theorem 3.1 of **?** that characterizes the loss in performance of the greedy policy.

Theorem 14: Let $\tilde{u}$ be the greedy policy with respect to the solution $\tilde{J}_c$ of the ALP, then

$$||J^* - J_{\tilde{u}}||_{1,c} \leq \frac{1}{1-\alpha}||J^* - \tilde{J}_c||_{1,c'}, \tag{18}$$

where $c' = (1-\alpha)c^\top(I - \alpha P_{\tilde{u}})^{-1}$.

Theorems **??** and **??** together imply that the ALP addresses both the control and prediction problems. Please refer to **?** for a more detailed treatment of the ALP.

Note that the ALP is a linear program in k ($<< n$) variables as opposed to the LP in (**??**) 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.

VIII. Constraint sampling

The most important work in the direction of constraint reduction is constraint sampling **?** wherein a reduced linear program (RLP) is solved instead of the ALP. While the objective function of the RLP is the same as that of the ALP, the RLP has only $m << nd$ constraints sampled from the original ALP according to a given probability distribution. The reduced linear program is then given by

$$\begin{aligned} &\min_{J \in \mathrm{R}^n} c^\top J \\ &\text{s.t } J(s) \geq g_a(s) + \alpha \sum_{s'} p_a(s,s')J(s'),\ \forall (s,a) \in \mathcal{I}, \end{aligned} \tag{19}$$

where $\mathcal{I}$ is the index set containing the m sampled state-action pairs. Using the index set $\mathcal{I}$ we define an $nd \times m$ matrix $\mathcal{M}$, called the constraint sampling matrix as below.

Definition 15: Let $\mathcal{I} = \{(s_1, a_1), \ldots, (s_m, a_m)\}$ be the sampled state-action pairs and let $q_i \stackrel{\Delta}{=} s_i + (a_i - 1) \times n, \forall i = 1, \ldots, m$. Then the constraint sampling matrix associated with the index set $\mathcal{I}$ is given by

$$\begin{aligned} \mathcal{M}(i,j) &= 1, \text{ if } q_i = j \\ &= 0, \text{ otherwise.} \end{aligned} \tag{20}$$

The RLP can then be represented in short as

$$\begin{aligned} &\min_{r \in \mathcal{N}} c^\top \Phi r \\ &\text{s.t } \mathcal{M}^\top E \Phi r \geq \mathcal{M}^\top H \Phi r, \end{aligned} \tag{21}$$

where $\mathcal{M}$ is a constraint sampling matrix as in Definition **??** and $\mathcal{N} \subset \mathrm{R}^k$ is a bounded set such that $\tilde{r}_c \in \mathcal{N}$. Note that the feasible set of the RLP is a superset of the feasible set of the ALP.

The RLP based on constraint sampling has been found to perform well empirically in application domains ranging from controlled queuing networks (see section 6.2 of **?** and section 7 of **?**) to Tetris (see **?** and section 6 of **?**). However, the theoretical guarantees for the RLP are available only under a restricted setting **?**. We present the main result of **?** on constraint sampling, and to that end define the following:

Definition 16: Given a policy u and Lyapunov function ψ in the column span of Φ, we define probability distributions μ_u, $\mu_{u,\psi}$ and constant θ as

$$\begin{aligned} \mu^\top &\triangleq (1-\alpha)c^\top (I-\alpha P_u)^{-1}, \\ \mu_{u,\psi}(s,a) &\triangleq \frac{\mu_u(s)\psi(s)}{\mu_u^\top \psi d}, \\ \theta &\triangleq \frac{(1+\beta_\psi)}{2}\frac{\mu_{u^*}^\top \psi}{c^\top J^*}\sup_{r\in\mathcal{N}}||J^*-\Phi r||_{\infty,1/\psi}. \end{aligned} \tag{22}$$

We now recall Theorem 3.1 of **?** below.

Theorem 17: Let ϵ and δ be scalars in $(0,1)$. Let u^* be the optimal policy and $\mathcal{I}$ an index set containing the m state-action pairs sampled independently according to the distribution $\psi_{u^*,\psi}$, for some Lyapunov function ψ, where

$$m \geq \frac{16d\theta}{(1-\alpha)\epsilon}\big(k\ln\frac{48d\theta}{(1-\alpha)\epsilon}+\ln\frac{2}{\delta}\big). \tag{23}$$

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

$$||J^*-\Phi\hat{r}||_{1,c} \leq ||J^*-\Phi\tilde{r}||_{1,c}+\epsilon||J^*||_{1,c}. \tag{24}$$

Motivation for our work:

The result in Theorem **??** has a significant limitation in that the sampling distribution $\mu_{u^*,\psi}$ requires the knowledge of u^*. The optimal policy u^* might not be available in practice and hence it would not be possible to even formulate the specific RLP for which the bound in Theorem **??** applies. However, as mentioned before, the RLP has performed reasonably well even when the sampling distribution is not $\mu_{u^*,\psi}$. Thus there is a gap between the theoretical understanding of the RLP and its practical efficacy which merits our attention. The gap also indicates that the RLP might be a special case of a generalized constraint reduction scheme with provable performance guarantees. Understanding such a generalized method would result in an ALP based ADP techinque that would have theoretical performance guarantees while being practically useful. In particular, we wish to answer the following open questions.

- 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 **?** 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.

IX. Generalized Reduced Linear Program

In this section we present the generalized reduced linear program (GRLP) which is obtained by appropriately extending the definition of the RLP. An important property that should carry over from the RLP is that the feasible set of the GRLP should also be a superset of the feasible set of the ALP. A natural way to achieve this is to replace the set of sampled constraints in the RLP by a set of constraints which are obtained as linear combinations of the original constraints of the ALP. Formally, we define the generalized reduced linear program (GRLP) as below:

$$\begin{aligned} &\min_{r \in \chi} c^\top \Phi r, \\ &\text{s.t } W^\top E \Phi r \geq W^\top H \Phi r, \end{aligned} \tag{25}$$

where $W \in \mathrm{R}_+^{nd \times m}$ is an $nd \times m$ matrix with all nonnegative entries and $\chi \subset \mathrm{R}^k$ 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. Constraint reduction is achieved by choosing $m << nd$. The key difference between the RLP in (**??**) and the GRLP in (**??**) despite their similar structure is that while $\mathcal{M}$ is a matrix of only zeros and ones, W is a matrix of positive entries alone. Also note that an RLP is trivially a GRLP as well. Unless specified otherwise we use $\hat{r}_c$ to denote the solution to the GRLP in (**??**), $\hat{J}_c = \Phi \hat{r}_c$, to denote the corresponding approximate value function and $\hat{u}$ to denote the greedy policy with respect to $\hat{J}_c$.

Note that we want to avoid certain uninteresting and trivial cases of W matrix such as $W = 0$ or an entire column of W being zero (which means no constraint is generated with respect to that column). Thus it is intuitive to demand that every column of W should be non-negative and have at least one entry which is strictly positive. Also, normalizing columns of W so that they sum to 1 does not make any difference to the constraints of the RLP. Keeping these in mind we also assume the following throughout the rest of the paper:

Assumption 4: $W \in \mathrm{R}_{+}^{nd \times m}$ is a full rank $nd \times m$ matrix (where $m << nd$) and each of its column sums equals 1.

The above assumption is just a technical condition that eliminates uninteresting choices such as $W = 0$ or cases when certain columns of W have all zeros, which implies that the corresponding column generates no constraint. The rest of the paper develops analytically various performance bounds and our main results provide the following:

1) A bound for $||J^* - \hat{J}_c||$, the error between the approximate value function $\hat{J}_c$ as computed by the GRLP and the optimal value function J^*;
2) a bound for $||J^* - J_{\hat{u}}||$, the loss in performance due to the greedy policy $\hat{u}$ measured with respect to the optimal policy; and
3) an important result on constraint sampling.

We achieve the above via two novel $\max$-norm contraction operators namely the least upper bound (LUB) projection operator (denoted by Γ) and the approximate least upper bound (ALUB) projection operator (denoted by $\hat{\Gamma}$). We bound the error due to constraint approximation by analyzing the fixed points of the operators Γ and $\hat{\Gamma}$. We first establish our results in the L_∞-norm and then extend the same in a modified L_∞-norm. The schematic in Fig. **??** provides a pictorial representation of what shall follow in the next three sections.

X. Least Upper Bound Projection

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

Definition 18: Given $J \in \mathrm{R}^n$, its least upper bound projection is denoted by ΓJ and is defined as

$$(\Gamma J)(i) \stackrel{\Delta}{=} \min_{j=1,\ldots,k} (\Phi r_{e_j})(i), \ \forall i = 1, \ldots, n, \tag{26}$$

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

$$\begin{aligned} r_c \stackrel{\Delta}{=} \min_{r \in \chi} & c^\top \Phi r, \\ \text{s.t } & \Phi r \geq TJ. \end{aligned} \tag{27}$$

Remark 1:

1) The definition of LUB operator $\Gamma\colon \mathrm{R}^n \to \mathrm{R}^n$ involves n associated linear programs.
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 $a, b, c \in \mathrm{R}$).
3) Given Φ and $J \in \mathrm{R}^n$, define $\mathcal{F} \stackrel{\Delta}{=} \{\Phi r | \Phi r \geq TJ\}$. 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 (**??**) 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.
4) Even though ΓJ does not belong to the span of Φ, ΓJ collates the various best upper bounds that can be obtained via the linear program in (**??**).
5) The LUB operator Γ in (**??**) bears close similarity to the ALP in (**??**).

Definition 19: The LUB projection of J^* is denoted by $\bar{J} = \Gamma J^*$.

We now characterize the LUB projection operator Γ in the following lemmas (all the proofs are presented in the Appendix). As mentioned earlier, the error analysis depends on two $\max$-norm contraction operators the first of which is Γ. The important result of this section is Theorem **??** and it relates the fixed point $\tilde{V}$ of Γ to J^*.

Lemma 20: Let $r^* \in \mathrm{R}^k$ be defined as $r^* \stackrel{\Delta}{=} \arg\min_{r \in R^k} ||J^* - \Phi r||_\infty$, then

$$||J^* - \bar{J}||_\infty \leq 2||J^* - \Phi r^*||_\infty. \tag{28}$$

Proof: The result follows from the definition of Γ in (**??**) and the construction of V_0, Assumption **??**, and the fact that $\Phi r^* + ||J^* - \Phi r^*||_\infty \mathbf{1} \geq TJ^*$. To see this, note that

$$\Gamma J^* = \hat{J}_c \geq J^*,$$

$$\Phi r^* + ||J^* - \Phi r^*||_\infty \geq TJ^* = J^*.$$

Thus,

$$\Phi r^* + ||J^* - \Phi r^*||_\infty \geq \Gamma J^* \geq TJ^*. \tag{29}$$

Lemma 21: For $J_1, J_2 \in \mathrm{R}^n$ such that $J_1 \geq J_2$, we have $\Gamma J_1 \geq \Gamma J_2$.

Proof: Choose any $i \in \{1, \ldots, n\}$ and let $r^1_{e_i}$ and $r^2_{e_i}$ be solutions to the linear program in (**??**) for $c = e_i$ with $J = J_1$ and $J = J_2$ respectively. Since $J_1 \geq J_2$, we have $TJ_1 \geq TJ_2$ and $e_i^\top \Phi r^1_{e_i} \geq e_i^\top \Phi r^2_{e_i}$, i.e., $(\Phi r^1_{e_i})(i) \geq (\Phi r^2_{e_i})(i)$. The result follows from the fact that $(\Gamma J)(i) = (\Phi r_{e_i})(i),\ \forall J \in \mathrm{R}^n$, and our choice of i was arbitrary.

Lemma 22: Let $A \in \mathrm{R}^{u\times v}$, $b, c \in R^u$, $x_0 \in R^v$ and $b_0 = Ax_0$. Then

$$\min\{c^\top Ax : Ax \geq b + b_0\} = \min\{c^\top Ax : Ax \geq b\} + c^\top b_0. \tag{30}$$

Proof: The claim can be shown by a simple change of variables.

Lemma 23: Let $J_1 \in \mathrm{R}^n$ and $t \in \mathrm{R}$ be a constant. If $J_2 = J_1 + k\mathbf{1}$, then $\Gamma J_2 = \Gamma J_1 + \alpha t\mathbf{1}$.

Proof: Consider the i^{th} linear programs associated with ΓJ_1 and ΓJ_2. The result follows by using Lemma **??** with $A = \Phi$, $b = TJ$, $c = e_i$, $b_0 = \alpha t\mathbf{1}$ and $x_0 = \alpha t e_i$.

Theorem 24: The operator $\Gamma\colon \mathrm{R}^n \to \mathrm{R}^n$ obeys the $\max$-norm contraction property with factor α.

Proof: Given $J_1, J_2 \in \mathrm{R}^n$, let $\epsilon = ||J_1 - J_2||_\infty$. Thus,

$$J_2 - \epsilon\mathbf{1} \leq J_1 \leq J_2 + \epsilon\mathbf{1}. \tag{31}$$

From Lemmas **??** and **??**, we can write

$$\Gamma J_2 - \alpha\epsilon\mathbf{1} \leq \Gamma J_1 \leq \Gamma J_2 + \alpha\epsilon\mathbf{1}. \tag{32}$$

Corollary 2: The iterative scheme in (**??**) based on the LUB projection operator Γ in (**??**) converges to a unique fixed point $\tilde{V}$.

$$V_{n+1} = \Gamma V_n, \ \forall n \geq 0. \tag{33}$$

Lemma 25: $\tilde{V}$, the unique fixed point of the iterative scheme (**??**), obeys $\tilde{V} \geq T\tilde{V}$.

Proof: Consider the i^{th} linear program associated with $\Gamma\tilde{V}$. We know that $\Phi r_{e_i} \geq T\tilde{V}$, $\forall i = 1, \ldots, n$. The result follows from noting that $\tilde{V}$ is the unique fixed point of Γ and that $\tilde{V}(i) = \min\limits_{j=1,\ldots,n}(\Phi r_{e_j})(i)$.

Lemma 26: $\tilde{V}$, the unique fixed point of the iterative scheme (**??**), and the solution $\tilde{J}_c$ to the ALP in (**??**), obey the relation $\tilde{J}_c \geq \tilde{V} \geq J^*$.

Proof: Since $\tilde{V} \geq T\tilde{V}$ it follows that $\tilde{V} \geq J^*$. Let $\Phi r_1, \Phi r_2, \ldots, \Phi r_n$ be solutions to the ALP in (**??**) for $c = e_1, e_2, \ldots, e_n$ respectively. Now consider the iterative scheme in (**??**) with $V_0(i) =$

$\min_{j=1,\ldots,n}(\Phi r_j)(i)$. It is clear from the definition of V_0 that $\tilde{J_c}(i) \geq \Phi r_i(i) \geq V_0(i),\ \forall i = 1,\ldots,n$. Also from the monotone property of T, we have

$$\begin{aligned}
\Phi r_i &\geq V_0, \\
T\Phi r_i &\geq TV_0, \text{we also have} \\
\Phi r_i \geq T\Phi r_i &\geq TV_0, \text{ by taking component-wise minimum,} \\
V_0 &\geq TV_0. \qquad (34)
\end{aligned}$$

From the first three inequalitites in (**??**), $\Phi r_i \geq T\Phi r_i \geq TV_0,\ \forall i = 1 \rightarrow n$ and hence $V_0 \geq TV_0$. Since $V_1 = \Gamma V_0$, from the definition of Γ in (**??**) we have $V_0 \geq V_1$, and recursively $V_n \geq V_{n+1},\ \forall n \geq 0$. So it follows that $\tilde{J_c} \geq V_0 \geq V_1 \ldots \geq \tilde{V}$.

Theorem 27: Let $\tilde{V}$ be the fixed point of the iterative scheme in (**??**) and let $\bar{J}$ be the best possible projection of J^* as in Definition **??**, then

$$||J^* - \tilde{V}||_\infty \leq \frac{1}{1-\alpha}||J^* - \bar{J}||_\infty. \qquad (35)$$

Proof: Let $\epsilon = ||J^* - \bar{J}||_\infty$, and $\{V_n\}, n \geq 0$ be the iterates of the scheme in (**??**) with $V_0 = \bar{J}$, then

$$\begin{aligned}
||J^* - \tilde{V}||_\infty &\leq ||J^* - V_0 + V_0 - V_1 + V_1 \ldots - \tilde{V}||_\infty \\
&\leq ||J^* - V_0||_\infty + ||V_0 - V_1||_\infty + \ldots
\end{aligned}$$

Since $||V_1 - V_0||_\infty = ||\Gamma\bar{J} - \Gamma J^*||_\infty \leq \alpha||\bar{J} - J^*||_\infty$, from Theorem **??**,

$$\begin{aligned}
||J^* - \tilde{V}||_\infty &\leq \epsilon + \alpha\epsilon + \alpha^2\epsilon + \ldots \\
&= \frac{\epsilon}{1-\alpha}. \qquad (36)
\end{aligned}$$

XI. 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 28: Given $J \in \mathrm{R}^n$, its approximate least upper bound (ALUB) projection is denoted by $\hat{\Gamma}J$ and is defined as

$$(\hat{\Gamma}J)(i) \stackrel{\Delta}{=} \min_{j=1,\ldots,k}(\Phi r_{e_j})(i),\ \forall i = 1,\ldots,n, \qquad (37)$$

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

$$r_c \stackrel{\Delta}{=} \min_{r \in \chi} c^\top \Phi r,$$
$$\text{s.t } W^\top E \Phi r \geq W^\top H J, W \in \mathrm{R}_+^{nd \times m}. \tag{38}$$

Note that W in (**??**) is the same matrix that is used in (**??**) and satisfies Assumption **??**.

Lemma 29: For $J_1, J_2 \in \mathrm{R}^n$ such that $J_1 \geq J_2$, we have $\hat{\Gamma} J_1 \geq \hat{\Gamma} J_2$.

Proof: The proof follows from Assumptions **??** and **??** using arguments along the lines of Lemma **??**.

Lemma 30: Let $J_1 \in \mathrm{R}^n$ and $t \in \mathrm{R}$ be a constant. If $J_2 = J_1 + t\mathbf{1}$, then $\hat{\Gamma} J_2 = \hat{\Gamma} J_1 + \alpha t \mathbf{1}$.

Proof: The proof follows from Assumption **??** and **??**, as well as Lemma **??** using arguments along the lines of Lemma **??**. In particular, consider the i^{th} linear program corresponding to $\hat{\Gamma} J_1$ and $\hat{\Gamma} J_2$. Now, the result follows by letting $A = W^\top E \Phi$, $b = W^\top H J$, $c = e_i$, $b_0 = \alpha t \mathbf{1}$, $x_0 = \alpha t e_i$.

Theorem 31: The operator $\hat{\Gamma} \colon \mathrm{R}^n \to \mathrm{R}^n$ obeys the $\max$-norm contraction property with factor α and the following iterative scheme based on the ALUB projection operator $\hat{\Gamma}$, see (**??**), converges to a unique fixed point $\hat{V}$.

$$V_{n+1} = \hat{\Gamma} V_n, \ \forall n \geq 0. \tag{39}$$

Proof: Follows along similar lines as the proof of Theorem **??**.

Lemma 32: The unique fixed point $\hat{V}$ of the iteration in (**??**) and the solution $\hat{J}_c$ of the GRLP obey $\hat{J}_c \geq \hat{V}$.

Proof: Follows in a similar manner as the proof of Lemma **??**. To elaborate, let $\Phi r_1, \Phi r_2, \ldots, \Phi r_n$ be solutions to the GRLP in (**??**) for $c = e_1, e_2, \ldots, e_n$ respectively. Now consider the iterative scheme in (**??**) with $V_0(i) = \min\limits_{j=1,\ldots,n} (\Phi r_j)(i)$. It is clear from the definition of V_0 that $\hat{J}_c(i) \geq \Phi r_i(i) \geq V_0(i), \ \forall i = 1, \ldots, n$. Also from the monotone property of T we have

$$\Phi r_i \geq V_0,$$
$$H \Phi r_i \geq H V_0, \text{we also have}$$
$$E \Phi r_i \geq H \Phi r_i \geq H V_0, \text{ by taking component-wise minimum,}$$
$$E V_0 \geq H V_0. \tag{40}$$

Since $V_1 = \hat{\Gamma} V_0$, from the definition of $\hat{\Gamma}$ in (**??**) and the construction of V_0, we have $V_0 \geq V_1$, and recursively $V_n \geq V_{n+1},\ \forall n \geq 0$. So it follows that $\hat{J}_c \geq V_0 \geq V_1 \ldots \geq \hat{V}$.

Theorem 33: Let $\hat{V}$ be the fixed point of the iterative scheme in (**??**) and let $\bar{J}$ be the best possible approximation of J^* as in Definition **??**, then

$$||J^* - \hat{V}||_\infty \leq \frac{||J^* - \bar{J}||_\infty + ||\Gamma J^* - \hat{\Gamma} J^*||_\infty}{1-\alpha}. \tag{41}$$

Proof: Let $\epsilon = ||J^* - \bar{J}||_\infty$, and $\{V_n\}, n \geq 0$ be the iterates of the scheme in (**??**) with $V_0 = \hat{\Gamma} J^*$, then

$$\begin{aligned} ||J^* - \hat{\Gamma} J^*||_\infty &\leq ||J^* - \Gamma J^*||_\infty + ||\Gamma J^* - \hat{\Gamma} J^*||_\infty \\ &= \epsilon + \beta, \end{aligned} \tag{42}$$

where $\beta = ||\Gamma J^* - \hat{\Gamma} J^*||_\infty$. Now

$$\begin{aligned} ||J^* - \hat{V}||_\infty &\leq ||J^* - V_0 + V_0 - V_1 + V_1 \ldots - \hat{V}||_\infty \\ &\leq ||J^* - V_0||_\infty + ||V_0 - V_1||_\infty + ||V_1 - V_2||_\infty + \ldots \\ &= ||J^* - V_0||_\infty + ||\hat{\Gamma} J^* - \hat{\Gamma} V_0||_\infty + \ldots \\ &\leq (\epsilon + \beta) + \alpha(\epsilon + \beta) + \ldots \\ &= \frac{\epsilon + \beta}{1-\alpha}. \end{aligned} \tag{43}$$

Corollary 3: Let $\hat{V}$, $\bar{J}$ be as in Theorem **??** and let $r^* \stackrel{\Delta}{=} \arg\min_{r \in \mathrm{R}^k} ||J^* - \Phi r||_\infty$, then

$$||J^* - \hat{V}||_\infty \leq \frac{2||J^* - \Phi r^*||_\infty + ||\Gamma J^* - \hat{\Gamma} J^*||_\infty}{1-\alpha}. \tag{44}$$

Proof: The result is obtained by using Lemma **??** to replace the term $||J^* - \bar{J}||_\infty$ in Theorem **??**.

XII. A SIMPLE BOUND

The following lemmas relate the fixed point $\hat{V}$ of $\hat{\Gamma}$ to the solution $\hat{J}_c$ of the GRLP in (**??**).

Lemma 34: $\hat{r} \in \mathrm{R}^k$ is a solution to GRLP in (**??**) iff it solves the following program:

$$\begin{aligned} &\min_{r \in \chi} ||\Phi r - \hat{V}||_{1,c} \\ &\text{s.t } W^\top \Phi r \geq W^\top T \Phi r. \end{aligned} \tag{45}$$

Proof: We know from Lemma **??** that $\hat{J}_c \geq \hat{V}$, and thus minimizing $||\Phi r - \hat{V}||_{1,c} = \sum_{i=1}^n c(i)|(\Phi r)(i) - \hat{V}(i)| = c^\top \Phi r - c^\top \hat{V}$, is the same as minimizing $c^\top \Phi r$.

Theorem 35: Let $\hat{V}$ be the solution to the iterative scheme in (**??**) and let $\hat{J}_c = \Phi\hat{r}_c$ be the solution to the GRLP. Let $\bar{J}$ be the best possible approximation to J^* as in Definition **??**, and $||\Gamma J^* - \hat{\Gamma}J^*||_\infty$ be the error due to ALUB projection and let $r^* \stackrel{\Delta}{=} \arg\min_{r\in \mathrm{R}^k} ||J^* - \Phi r||_\infty$, then

$$||\hat{J}_c - \hat{V}||_{1,c} \leq \frac{4||J^* - \Phi r^*||_\infty + ||\Gamma J^* - \hat{\Gamma}J^*||_\infty}{1-\alpha}. \tag{46}$$

Proof: Let $\gamma = ||J^* - \Phi r^*||_\infty$, then it is easy to see that

$$\begin{aligned} ||J^* - T\Phi r^*||_\infty &= ||TJ^* - T\Phi r^*||_\infty \leq \alpha\gamma, \text{ and} \\ ||T\Phi r^* - \Phi r^*||_\infty &\leq (1+\alpha)\gamma. \end{aligned} \tag{47}$$

From Assumption **??** there exists $r' \in \mathrm{R}^k$ such that $\Phi r' = \Phi r^* + \frac{(1+\alpha)\gamma}{1-\alpha}\mathbf{1}$ and r' is feasible to the ALP. Now

$$\begin{aligned} ||\Phi r' - J^*||_\infty &\leq ||\Phi r^* - J^*||_\infty + ||\Phi r' - \Phi r^*||_\infty \\ &\leq \gamma + \frac{(1+\alpha)\gamma}{1-\alpha} = \frac{2\gamma}{1-\alpha}. \end{aligned} \tag{48}$$

Since r' is also feasible for GRLP in (**??**) we have

$$\begin{aligned} ||\hat{J}_c - \hat{V}||_{1,c} &\leq ||\Phi r' - \hat{V}||_{1,c} \\ &\leq ||\Phi r' - \hat{V}||_\infty \text{ (Since } c \text{ is a distribution)} \\ &\leq ||\Phi r' - J^*||_\infty + ||J^* - \hat{V}||_\infty. \end{aligned}$$

The result follows from Corollary **??**.

Prediction Error bound in the L_∞-norm

Corollary 4: Let $\hat{J}_c$, $\hat{V}$, r^* and J^* be as in Theorem **??**, then

$$||J^* - \hat{J}_c||_{1,c} \leq \frac{6||J^* - \Phi r^*||_\infty + 2||\Gamma J^* - \hat{\Gamma}J^*||_\infty}{1-\alpha}. \tag{49}$$

Proof:

$$\begin{aligned} ||J^* - \hat{J}_c||_{1,c} &\leq ||J^* - \hat{V}||_{1,c} + ||\hat{V} - \hat{J}_c||_{1,c} \\ &\leq ||J^* - \hat{V}||_\infty + ||\hat{V} - \hat{J}_c||_{1,c} \end{aligned}$$

The result now follows from Corollary **??** and Theorem **??**. The results presented in Corollary **??** is in the L_∞-norm. In the next section, we use of Lyapunov functions to provide an improved bound in a modified L_∞-norm.

XIII. IMPROVED BOUNDS

In this section, we present improved error bounds by making use of Lyapunov functions.

Lemma 36: Let $r^* \in \mathrm{R}^k$ be defined as $r^* \stackrel{\Delta}{=} \arg\min_{r \in R^k} ||J^* - \Phi r||_{\infty,1/\psi}$, then

$$||J^* - \bar{J}||_{\infty,1/\psi} \leq 2||J^* - \Phi r^*||_{\infty,1/\psi}. \tag{50}$$

Proof: The result follows from the definition of Γ in (**??**), Assumption **??** and the fact that $\Phi r^* + ||J^* - \Phi r^*||_{\infty,1/\psi}\psi \geq TJ^*$.

Since most of our analysis in sections **??** and **??** depended on showing that Γ is a contraction map in the L_∞ norm we first show that Γ is also a contraction map in the modified L_∞ norm.

Lemma 37: Let $J_1 \in \mathrm{R}^n$ and $k \in \mathrm{R}$ be a constant. If $J_2 = J_1 + k\psi$, then $\Gamma J_2 \leq \Gamma J_1 + \beta_\psi k\psi$.

Proof: The result follows in a similar manner as the proofs for Lemmas **??** and **??** by using the result in Lemma **??**.

Theorem 38: The operator $\Gamma \colon \mathrm{R}^n \to \mathrm{R}^n$ is a contraction operator in modified L_∞ with factor β_ψ.

Proof: Given $J_1, J_2 \in \mathrm{R}^n$ let $\epsilon = ||J_1 - J_2||_{\infty,1/\psi}$. Thus

$$J_2 - \epsilon\psi \leq J_1 \leq J_2 + \epsilon\psi. \tag{51}$$

From Lemmas **??** and **??**, we can write

$$\Gamma J_2 - \beta_\psi \epsilon\psi \leq \Gamma J_1 \leq \Gamma J_2 + \beta_\psi \epsilon\psi. \tag{52}$$

Thus

$$||\Gamma J_1 - \Gamma J_2||_{\infty,1/\psi} \leq \beta_\psi ||J_1 - J_2||_{\infty,1/\psi}. \tag{53}$$

Corollary 5: $\hat{\Gamma}$ is also a contraction map in the modified L_∞ norm.

Proof: Follows from arguments similar to Theorem **??**.

Lemma 39: Let $\hat{V}$, $\bar{J}$ be as in Theorem **??** and let $r^* \stackrel{\Delta}{=} \arg\min_{r \in \mathrm{R}^k} ||J^* - \Phi r||_{\infty,1/\psi}$ then

$$||J^* - \hat{V}||_{\infty,1/\psi} \leq \frac{2||J^* - \Phi r^*||_{\infty,1/\psi} + ||\Gamma J^* - \hat{\Gamma} J^*||_{\infty,1/\psi}}{1 - \beta_\psi}. \tag{54}$$

Proof: The proof follows from Lemma **??**, Corollary **??** and by replacing the $||\cdot||_\infty$ norm by $||\cdot||_{\infty,1/\psi}$ in the arguments presented in sections **??** and **??** leading to Corollary **??**.

We now recall Lemma 4.3 of **?**.

Lemma 40: Let ψ be a Lyapunov function that belongs to the column span of Φ , $r \in \mathrm{R}^k$ be an arbitrary vector and let r' be such that

$$\Phi r' = \Phi r + ||J^* - \Phi r||_{\infty,1/\psi}(\frac{1+\beta_\psi}{1-\beta_\psi})\psi. \tag{55}$$

Then r' is feasible for the ALP in (**??**).

Theorem 41: Let $\hat{V}$ be the solution to the iterative scheme in (**??**) and let $\hat{J}_c = \Phi\hat{r}_c$ be the solution to the GRLP. Let $\bar{J}$ be the best possible approximation to J^* as in Definition **??**, and $||\Gamma J^* - \hat{\Gamma}J^*||_{\infty,1/\psi}$ be the error due to ALUB projection and let $r^* \stackrel{\Delta}{=} \arg\min_{r\in\mathrm{R}^k} ||J^* - \Phi r||_{\infty,1/\psi}$, then

$$||\hat{J}_c - \hat{V}||_{1,c} \leq \frac{c^\top\psi}{1-\beta_\psi}(4||J^* - \Phi r^*||_{\infty,1/\psi} + ||\Gamma J^* - \hat{\Gamma}J^*||_{\infty,1/\psi}). \tag{56}$$

Proof: Let $\gamma = ||J^* - \Phi r^*||_{\infty,1/\psi}$, then by choosing r' as in Lemma **??** we have

$$\begin{aligned}
||\Phi r' - J^*||_{\infty,1/\psi} &\leq ||\Phi r^* - J^*||_{\infty,1/\psi} + ||\Phi r' - \Phi r^*||_{\infty,1/\psi} \\
&= \gamma + \frac{1+\beta_\psi}{1-\beta_\psi}\gamma \\
&= \frac{2}{1-\beta_\psi}\gamma.
\end{aligned}$$

Since r' is also feasible for the GRLP in (**??**) we have

$$\begin{aligned}
||\hat{J}_c - \hat{V}||_{1,c} &\leq ||\Phi r' - \hat{V}||_{1,c} \\
&= \sum_{s\in S} c(s)\psi(s)\frac{|\Phi r'(s) - \hat{V}(s)|}{\psi(s)} \\
&\leq c^\top\psi||\Phi r' - \hat{V}||_{\infty,1/\psi} \\
&\leq c^\top\psi(||\Phi r' - J^*||_{\infty,1/\psi} + ||J^* - \hat{V}||_{\infty,1/\psi}).
\end{aligned} \tag{57}$$

The result follows from Corollary **??**.

Main Result 1**: Prediction Error bound in modified L_∞-norm**

Theorem 42: Let $\hat{J}_c$, $\hat{V}$, r^* and J^* be as in Theorem **??**, then

$$||J^* - \hat{J}_c||_{1,c} \leq \frac{c^\top\psi}{1-\beta_\psi}(6||J^* - \Phi r^*||_{\infty,1/\psi} + 2||\Gamma J^* - \hat{\Gamma}J^*||_{\infty,1/\psi}). \tag{58}$$

Proof:

$$\begin{aligned}
||J^* - \hat{J}_c||_{1,c} &\leq ||J^* - \hat{V}||_{1,c} + ||\hat{V} - \hat{J}_c||_{1,c} \\
&\leq c^\top\psi||J^* - \hat{V}||_{\infty,1/\psi} + ||\hat{V} - \hat{J}_c||_{1,c}.
\end{aligned}$$

The result now follows from Lemma **??** and Theorem **??**.

Main Result 2: Control Error bound in modified L_∞-norm

We now bound the performance of the greedy policy $\hat{u}$.

Theorem 43: Let $\hat{u}$ be the greedy policy with respect to the solution $\hat{J}_c$ of the GRLP and $J_{\hat{u}}$ be its value function. Let r^* be as in Theorem **??**, then

$$||J_{\hat{u}} - \hat{J}_c||_{1,c} \leq 2\big(\frac{c^\top \psi}{1-\beta_\psi}\big)^2\big(6||J^* - \Phi r^*||_{\infty,1/\psi} + 2||\Gamma J^* - \hat{\Gamma} J^*||_{\infty,1/\psi}\big). \tag{59}$$

Proof:

$$\begin{aligned}
||J_{\hat{u}} - \hat{J}_c||_{1,c} &= ||(I - \alpha P_{\hat{u}})^{-1}(T\hat{J}_c - \hat{J}_c)||_{1,c} \\
&\leq c^\top (I - \alpha P_{\hat{u}})^{-1}|T\hat{J}_c - \hat{J}_c| \\
&\leq c^\top (I - \alpha P_{\hat{u}})^{-1}\psi||T\hat{J}_c - \hat{J}_c||_{\infty,1/\psi} \\
&\leq \frac{c^\top \psi}{1-\beta_\psi}||T\hat{J}_c - \hat{J}_c||_{\infty,1/\psi} \\
&\leq \frac{c^\top \psi}{1-\beta_\psi}||T\hat{J}_c - TJ^* + J^* - \hat{J}_c||_{\infty,1/\psi} \\
&\leq \frac{c^\top \psi}{1-\beta_\psi}(||T\hat{J}_c - TJ^*||_{\infty,1/\psi} + ||J^* - \hat{J}_c||_{\infty,1/\psi}) \\
&\leq \frac{c^\top \psi}{1-\beta_\psi}(1+\beta_\psi)||J^* - \hat{J}_c||_{\infty,1/\psi},
\end{aligned} \tag{60}$$

where in the second inequality, for $x = (x_1, \ldots, x_n)^\top \in \mathrm{R}^n$, $|x| = (|x_1|, \ldots, |x_n|)^\top \in \mathrm{R}^n$. Now

$$\begin{aligned}
||J^* - J_{\hat{u}}||_{1,c} &\leq ||J^* - \hat{J}_c||_{1,c} + ||J_{\hat{u}} - \hat{J}_c||_{1,c} \\
&\leq c^\top \psi||J^* - \hat{J}_c||_{\infty,1/\psi} + c^\top \psi \frac{1+\beta_\psi}{1-\beta_\psi}||J^* - \hat{J}_c||_{\infty,1/\psi} \\
&= \frac{2c^\top \psi}{1-\beta_\psi}||J^* - \hat{J}_c||_{\infty,1/\psi}.
\end{aligned} \tag{61}$$

The result now follows by substituting the value of $||J^* - \hat{J}_c||_{\infty,1/\psi}$ from Corollary **??**.

Note 1: By letting $||\Gamma J^* - \hat{\Gamma} J^*||_{\infty,1/\psi} = ||\Gamma J^* - J^* + J^* - \hat{\Gamma} J^*||_{\infty,1/\psi} \leq 2||J^* - \Phi r^*||_{\infty,1/\psi} + ||J^* - \hat{\Gamma} J^*||_{\infty,1/\psi}$ (inequality follows from Lemma **??**), we can also modifiy the bounds in (**??**) and (**??**) as

$$||J^* - \hat{J}_c||_{1,c} \leq \frac{c^\top \psi}{1-\beta_\psi}(10||J^* - \Phi r^*||_{\infty,1/\psi} + 2||J^* - \hat{\Gamma} J^*||_{\infty,1/\psi}). \tag{62}$$

$$||J_{\hat{u}} - \hat{J}_c||_{1,c} \leq 2\big(\frac{c^\top \psi}{1-\beta_\psi}\big)^2\big(10||J^* - \Phi r^*||_{\infty,1/\psi} + 2||J^* - \hat{\Gamma} J^*||_{\infty,1/\psi}\big). \tag{63}$$

Here the term $||J^* - \hat{\Gamma} J^*||$ in (**??**) and (**??**) captures the error due to the use of both Φ and W. Though, (**??**) and (**??**) might be loser bounds than (**??**) and (**??**) respectively, the aim here is to capture the error due to function approximation as well as constraint reduction in a single term.

XIV. Discussion

In this section we discuss the implications and insights provided by the results presented in Theorems **??** and **??**.

A. On Error Terms

- The error bounds in the main results (Theorems **??** and **??**) contain two factors namely
  1) $\min_{r \in \mathrm{R}^k} ||J^* - \Phi r||_{\infty, 1/\psi}$,
  2) $||\Gamma J^* - \hat{\Gamma} J^*||_{\infty, 1/\psi}$.

  The first factor is related to the best possible approximation that can be achieved with the chosen feature matrix Φ. This term is inherent to the ALP formulation and it appears in the bounds provided by **?**.

  The second factor is related to constraint approximation and 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 **?** 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.
- A salient feature of the ALP formulation is the use of Lyapunov functions to control/shape the error across the states based on their relative importance. Since the error bounds are in a modified L_∞-norm, the GRLP framework retains this salient feature of the ALP.

The fact that both the prediction and control problems can be addressed by the GRLP makes it a complete ADP method, and by addressing the constraint approximation, the GRLP framework is an important addition to the theory of ALP.

B. On Constraint Reduction and Approximation

We claim the following based on the error bounds that we derived for the GRLP.

Claim 1) It is not always necessary to sample constraints according to the stationary distribution

of the optimal policy.

Claim 2) Constraint approximation is not only restricted to constraint sampling but also can be extended to include linear approximation of the constraints.

The following result (Theorem **??**) supports Claim 1 in the above.

Main Result 3: On Constraint Sampling

The error term $||\Gamma J^* - \hat{\Gamma} J^*||_{\infty,1/\psi}$ gives new insights into constraint sampling.

Theorem 44: Let $s \in S$ be a state whose constraint was sampled. Then

$$|\Gamma J^*(s) - \hat{\Gamma} J^*(s)| < |\Gamma J^*(s) - J^*(s)|. \tag{64}$$

Proof: Let r_{e_s} and $\hat{r}_{e_s}$ be solutions to the linear programs in (**??**) and (**??**) respectively for $c = e_s$ and $J = J^*$. It is easy to note that r_{e_s} is feasible for the linear program in (**??**) for $c = e_s$ and J^*, and hence it follows that $(\Phi r_{e_s})(s) \geq (\Phi \hat{r}_{e_s})(s)$. However, since all the constraints with respect to state s have been sampled we know that $(\Phi \hat{r}_{e_s})(s) \geq J^*$. The proof follows from noting that $(\Gamma J^*)(s) = (\Phi r_{e_s})(s)$ and $\hat{\Gamma} J^*(s) = (\Phi \hat{r}_{e_s})(s)$.

The expression in (**??**) in Theorem **??** says that the additional error $|\Gamma J^*(s) - \hat{\Gamma} J^*(s)|$ due to constraint sampling is less than the original projection error $|\Gamma J^*(s) - J^*(s)|$ due to function approximation. This means that for the RLP to perform well it is enough to retain those states for which the linear function approximation via Φ is known to perform well. The modified L_∞ norm in (**??**) comes to our rescue to control the error due to those states that are not sampled. 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 **???**.

To understand the implication of Claim 2 we need to look at the Lagrangian of the ALP and GRLP in (**??**) and (**??**) respectively, i.e.,

$$\tilde{L}(r, \lambda) = c^\top \Phi r + \lambda^\top (T\Phi r - \Phi r), \tag{65}$$

$$\hat{L}(r, q) = c^\top \Phi r + q^\top W^\top (T\Phi r - \Phi r). \tag{66}$$

The insight that the GRLP is a linear function approximation of the constraints (i.e., the Lagrangian multipliers) can be obtained by noting that $Wq \approx \lambda$ in (**??**). Note that while the ALP employs LFA in its objective function (i.e., use of Φr), the GRLP employs linear approximation both in the objective function (Φr) as well as the constraints (use of W). Further, W can be interpreted as the feature matrix that approximates the Lagrange multipliers as $\lambda \approx Wq$, where

$\lambda \in \mathrm{R}^{nd}, r \in \mathrm{R}^m$. One can show **?** that the optimal Lagrange multipliers are the discounted number of visits to the "state-action pairs" under the optimal policy u^*, i.e.,

$$\begin{aligned}\lambda^*(s,u^*(s)) &= \big(c^\top (I-\alpha P_{u^*})^{-1}\big)(s)\\ &= \big(c^\top (I+\alpha P_{u^*}+\alpha^2 P_{u^*}^2+\ldots)\big)(s),\\ \lambda^*(s,a) &= 0, \forall a \neq u^*(s),\end{aligned}$$

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 probability distribution and that it 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.

C. Numerical Illustration

We take up an example in the domain of controlled queues from **?** for which the ALP has been known to work well. For this domain, we make use of our results and observations to select various useful W matrices and present their performance.

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 (the latter 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 **?**. 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 **?**. This takes care of the term $||J^* - \Phi r^*||_\infty$ in (**??**).

Selection of W**:** For our experiments, we choose two contenders for the W-matrix and compare them with the ideal sampling matrix W_i (**?**) and random positive matrix W_r. Our choices of the W matrix are as below.

$(\mathbf{i})$ W_c- matrix that corresponds to sampling according to c. This is justified by the insights obtained from Theorem **??** on the error term $||\Gamma J^* - \hat{\Gamma} J^*||_\infty$, i.e., the idea of selecting the important states.

$(\mathbf{ii})$ W_a state-aggregation matrix, a heuristic derived using the fact that λ^* is a linear combination of $\{P_{u^*}, P_{u^*}^2, \ldots\}$. Our choice of the W_a matrix to correspond to aggregation of near by 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: $\forall i = 1, \ldots, m$,

$$\begin{aligned} W_a(i,j) = 1, \ \forall j \text{ s.t } j = (i-1) \times \frac{n}{m} + k + (l-1) \times n, \\ k = 1, \ldots, \frac{n}{m}, l = 1, \ldots, d, \\ = 0, \ \text{otherwise.} \end{aligned} \tag{67}$$

We ran our experiments on a moderately large queuing system denoted by 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 $\alpha = 0.98$. We chose $k = 4$ (i.e., we used $1, s, s^2$ and s^3 as basis vectors) and we chose W_a (**??**), 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 **??** show that the 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 closer to the ideal sampler W_i. Also note that a better performance of W_a and W_c in the larger system Q_L tallies with a lower value of $||\Gamma J^* - \hat{\Gamma} J^*||_\infty$ 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 I

SHOWS VALUES OF ERROR TERMS FOR Q_L.

Performance Metric	W_i	W_c	W_a
$\|\|J_{\hat{u}}\|\|_{1,c}$ for $\zeta = 0.9$	-441.25	-450.59	-446.49
$\|\|J_{\hat{u}}\|\|_{1,c}$ for $\zeta = 0.999$	$-2.0611e+04$	$-2.0611e+04$	$-2.0612e+04$

TABLE II

SHOWS PERFORMANCE METRICS FOR Q_L. HERE $||J^*||_{1,c} = -439.26$ FOR $\zeta = 0.9$ AND $||J^*||_{1,c} = -2.0603e+04$ FOR $\zeta = 0.999$ AND A RANDOM POLICY YIELDS A TOTAL REWARD OF $-1.2661e+03$.

Empirical evidence for the performance of RLP with various sampling distributions can also be found in **??**.

D. Reinforcement Learning

Reinforcement Learning (RL) algorithms are useful in scenarios where the system is available in the form of a simulator or only samples can be obtained via direct interaction. In particular, in the RL setting, the model parameters g and P are not known explicitly and the underlying MDP needs to be solved by using sample trajectories. In short, RL algorithms are sample trajectory based solution schemes for solving MDPs whose model information is not known. RL methods learn by filtering out the noisy sample via stochastic approximation and they also employ function approximation in order to handle MDPs with large number of states. Most RL algorithms are sample trajectory based extensions of ADP methods.

The RL extension of the ALP formulation has been applied to the optimal stopping problem in **?**. Function approximation is employed to approximate the square root of the Lagrange multipliers. However, since the approximation is not linear, convergence of the resulting RL algorithm cannot be guaranteed. Our results theoretically justify linear function approximation of the Lagrange multipliers, an immediate implication of which is that the RL extension of the ALP can be guaranteed to converge if the updates in **?** use LFA for the Lagrange multipliers instead of a non-linear approximator.

XV. CONCLUSION

The approximate linear programming (ALP) is an approximate dynamic programming method that addresses the prediction and control problems successfully. However, an important shortcoming of the ALP is that it has large number of constraints, which is tackled in practice by sampling a tractable number of constraints from the ALP to formulate and solve a reduced linear program (RLP). Though RLP has been found to work well empirically in various domains ranging from

queues to Tetris games, performance guarantees are available only in the case of a specific RLP formulated under idealized assumptions. Thus there has been a gap in the theory of constraint reduction.

In this paper, we introduced a novel framework based on the generalized reduced linear program formulation to study constraint reduction. The constraints of the GRLP were obtained as positive linear combinations of the original ALP. We provided an error bound that relates the optimal value function to the solution of the GRLP. Our error bound contained two terms, one inherent to the ALP formulation and the other due to constraint reduction. We also made qualitative and quantitative observations about the nature of the error term that arose due to constraint reduction. Our analysis also revealed the fact that it is not always necessary to sample according to the stationary distribution of the optimal policy and, in fact, potentially several different constraint sampling/approximation strategies might work. In particular, we also theoretically justified linear function approximation of the constraints. We also discussed the results and provided a numerical example in the domain of controlled queues. To conclude, we observe that by providing a novel theoretical framework to study constraint approximation, this paper provides important results that add to the theory of ALP.