Self-guided Approximate Linear Programs

Parshan Pakiman[†], **Selvaprabu Nadarajah**[†], **Negar Soheili**[†], **Qihang Lin**[‡]

[†] College of Business Administration, University of Illinois at Chicago, Chicago, IL 60607, USA
[‡] Tippie College of Business, The University of Iowa, Iowa City, IA 52242, USA

ppakim2@uic.edu, selvan@uic.edu, nazad@uic.edu, qihang-lin@uiowa.edu

Abstract

Approximate linear programs (ALPs) are well-known models based on value function approximations (VFAs) to obtain heuristic policies and lower bounds on the optimal policy cost of Markov decision processes (MDPs). The ALP VFA is a linear combination of predefined basis functions that are chosen using domain knowledge and updated heuristically if the ALP optimality gap is large. We side-step the need for such basis function engineering in ALP – an implementation bottleneck – by proposing a sequence of ALPs that embed increasing numbers of random basis functions obtained via inexpensive sampling. We provide a sampling guarantee and show that the VFAs from this sequence of models converge to the exact value function. We also show under mild conditions that our ALP policy cost is near-optimal when the number of sampled random bases is sufficiently large. Nevertheless, the performance of this ALP policy can fluctuate significantly as more basis functions are sampled. To mitigate these fluctuations, we "self-guide" our convergent sequence of ALPs using past VFA information such that a worst-case measure of policy performance is improved. Moreover, our method provides application-agnostic policies and bounds to benchmark approaches that exploit application structure.

1 Introduction

Approximate linear programming [19, 6] is a popular approach to compute value function approximations (VFAs) for Markov decision processes (MDPs; [17]) that has been applied to various applications [7, 2, 20, 15, 14, 3, 5]. VFAs in an approximate linear program (ALP) are represented as a linear combination of functions, referred to as basis functions, such that solving ALP provides their weights, the ALP VFA and policy, and the ALP lower bound on the optimal policy cost. Figure 1(a) depicts the steps involved in a standard implementation of ALP. Step (i) selects basis functions using domain knowledge. Step (ii) solves the ALP formulated using these bases. Step (iii) evaluates the value of the ALP policy in simulation and computes its optimality gap. Step (iv) modifies the bases and repeats the process from Step (ii) if the optimality gap is large; otherwise, this process is terminated and the incumbent VFA is returned. Tackling an ALP with a fixed set of bases in Step (ii) is challenging since it has a large number of constraints and has been a topic of active research (see [11] for a recent overview of ALP solution techniques). Steps (i) and (iv) are ALP implementation bottlenecks but this issue has received limited attention in the literature [10, 1, 4]. This workshop paper summarizes the main ideas of the original paper (OP[1]), that is under revision at Management Science journal and is available at https://www.dropbox.com/s/4rsh9srmbjrdzom/NeurIPS_2020_self_guided_ALPs, to tackle basis function engineering in ALP for a wide class of MDPs with continuous state and action spaces.

Our starting point is a novel reformulation of a discounted-cost MDP, which we refer to as feature-based exact linear program (FELP). This model is intractable. We approximate it using random bases to obtain the feature-based approximate linear program (FALP), where a VFA is represented as a linear combination of randomly sampled basis functions. The randomized nature of FALP allows us to pursue the modified ALP implementation process illustrated in Figure 1(b). In this scheme, basis function selection and modification in Steps (i) and (iv) of the standard implementation approach (Figure 1(a)) have been replaced by inexpensive sampling. We establish that the FALP optimality gap converges to zero as the number of samples tends to infinity. Despite this asymptotic property, neither the FALP lower bound nor its policy cost may improve monotonically as more bases are sampled. While the former issue can be handled easily, the latter is undesirable and is harder to tackle. We propose a mechanism for the FALP sequence to self-guide its VFAs in a manner that addresses

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

Figure 1: ALP implementation strategies.

the non-monotonic behavior of bounds. We refer to FALP with this self-guiding mechanism as the feature-based guided linear program (FGLP) and embed it in lieu of FALP in the iterative process of Figure 1(b). Therefore, unlike FALP VFAs, the sequence of FGLP VFAs provides monotonically decreasing lower bounds and policy costs with a monotonically non-increasing worst case bound.

Novelty and Contributions. Research on ALPs predominantly assumes a fixed set of basis functions. Work that relaxes this assumption, as we do, is limited. Two convergent basis function generation algorithms are developed in [10] and [1], where they require solving challenging nonlinear programs and the knowledge of problem-specific structure, respectively. Instead, our approach uses low-cost sampling and is agnostic to the application. To tackle basis function generation, the kernel trick is used by [4] to replace the inner-products of basis functions in the dual of a regularized ALP relaxation. Unlike the approximation guarantees in [4], that need an idealized sampling distribution depending on an optimal policy, our VFA guarantees are not linked to the knowledge of an optimal policy. Moreover, our work builds on the seminal research on random basis functions in [18]. There is extant literature applying this idea to machine learning applications [12, 13, 21], and a value iteration algorithm by [8]. Our investigation of the variance in the quality of policies obtained from VFAs based on random bases as well as the subsequent addition of self-guiding constraints to mitigate this issue are both novel.

Organization of Paper. In §2, we present the FELP reformulation of infinite horizon discounted-cost MDPs. In §3, we present FALP and our algorithm formalizing the iterative procedure in Figure 1(b). In §4, we discuss FGLP and conclude in §5. Appendix A provides additional details on FALP, FGLP, and policy cost fluctuations. Note that in §§5-6 of OP[1], we asses the performance of our policies on challenging instances of perishable inventory control and generalized joint replenishment applications.

2 Exact Linear Programs for MDPs

Standard Exact Linear Program. Consider a decision maker controlling a system over an infinite horizon. A (stationary and deterministic) policy $\pi : \mathcal{S} \mapsto \mathcal{A}_s$ assigns an action $a \in \mathcal{A}_s$ to each state $s \in \mathcal{S}$ where $\mathcal{S} \subseteq \Re^{d_S}$ denotes the MDP state space and $\mathcal{A}_s$ represents the feasible action space at state s. We assume $\mathcal{S}$ and $\mathcal{A}_s$ for all $s \in \mathcal{S}$ are continuous and compact sets. An action $a \in \mathcal{A}_s$ taken at state $s \in \mathcal{S}$ results in an immediate cost of $c(s,a)$ and the transition of the system to state $s' \in \mathcal{S}$ with probability $P(s'|s,a)$. The decision maker's objective is to find an optimal policy $\pi^* \in \Pi$, where Π is the set of feasible policies, that minimizes long-run discounted expected costs, which is,

$$\pi^* \in \arg\min_{\pi\in\Pi} \mathbb{E}_\chi[\mathrm{PC}(s,\pi)] \quad \text{s.t.} \quad \mathrm{PC}(s,\pi) \coloneqq \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t c(s_t^\pi, \pi(s_t^\pi)) \;\middle|\; s_0 = s\right]. \tag{1}$$

Distribution χ over $\mathcal{S}$ is on the initial state s_0, expectation $\mathbb{E}$ is taken with respect to the state-action probability distribution induced by $P(\cdot|s,a)$ and π, $\gamma \in (0,1)$ is a discount factor, and s_t^π is the state reached at stage t when following this policy. Notation $\mathrm{PC}(s,\pi)$ shows the long-run discounted expected cost of a policy π starting from $s_0 = s$. There are known conditions that ensure the existence of π^* and we take them to be true in this paper (see pages 46-47 of [9] and EC.1.1 of OP[1]).

Let $\mathcal{C}$ be the class of continuous maps over $\mathcal{S}$ and define the MDP value function $V^*(s) \coloneqq \mathrm{PC}(s,\pi^*)$. Throughout, we assume V^* is continuous. It is known that value function can be conceptually computed from the exact linear program (ELP; see, e.g., pages 131-143 of [9]),

$$\max_{V'\in\mathcal{C}} \mathbb{E}_\nu[V'(s)] \text{ s.t. } V'(s) - \gamma\mathbb{E}[V'(s')|s,a] \le c(s,a), \ \forall (s,a) \in \mathcal{S}\times\mathcal{A}_s,$$

where ν is a state-relevance distribution that specifies the relative importance of each state in the state space. ELP is intractable since it has continuums of decision variables and constraints.

Feature-based Exact Linear Programs. Consider random basis function $\varphi(\cdot;\theta) : \mathcal{S} \mapsto \Re$ and its associated sampling distribution $\rho(\theta)$ that are parameterized by $\theta \in \Theta$. Define the class of functions,

$$\mathcal{R}_C(\varphi,\rho) \coloneqq \left\{V : \mathcal{S} \mapsto \Re \;\middle|\; \exists (b_0, \boldsymbol{b}) \text{ with } V(s) = b_0 + \langle \boldsymbol{b}, \varphi(s)\rangle \text{ and } \|\boldsymbol{b}\|_{\infty,\rho} \le C\right\},$$

where $b_0 \in \Re$ is an intercept, $\boldsymbol{b} : \Theta \mapsto \Re$ is a weighting function, $C \in \Re_+$ is a constant, $\langle \boldsymbol{b}, \varphi(s)\rangle \coloneqq \int_\Theta \boldsymbol{b}(\theta)\varphi(s;\theta)\mathrm{d}\theta$ is an inner product, and $\|\boldsymbol{b}\|_{\infty,\rho} \coloneqq \sup_\theta |{}^{\boldsymbol{b}(\theta)}/_{\rho(\theta)}|$ is referred to as the

(∞, ρ)-norm. Random Fourier basis functions defined as $\varphi(s;\theta) = \cos(\theta_0 + \theta_1 s_1 + \cdots + \theta_{d_S} s_{d_S})$ are a popular choice where θ^0 and each θ^i for $i \geq 1$ are sampled, respectively, from a uniform distribution over the interval $[-\pi, \pi]$ and a mean-zero normal distribution with the standard deviation of $\sigma > 0$. The reformulation of ELP relies on "universal" random basis functions. We say φ with sampling density ρ is universal if for every $V \in \mathcal{C}$ and $\varepsilon > 0$, there is a constant $C > 0$ such that $\bar{V} \in \mathcal{R}_C(\varphi, \rho)$ and $\|V - \bar{V}\|_\infty := \max_s |V(s)| < \varepsilon$. Random Fourier basis functions are universal.

Since V^* is continuous in our setting and φ is universal, replacing variables $V'(s)$ modeling $V^*(s)$ in ELP by the inner product $b_0 + \langle \boldsymbol{b}, \varphi(s) \rangle$ should intuitively not result in any significant error. Performing this replacement and requiring $b_0 + \langle \boldsymbol{b}, \varphi(s) \rangle \in \mathcal{R}_C(\varphi, \rho)$ gives the following program, FELP,

$$\begin{array}{lll} \sup\limits_{b_0, \boldsymbol{b}} & b_0 + \langle \boldsymbol{b}, \mathbb{E}_\nu[\varphi(s)] \rangle & \\ \|\boldsymbol{b}\|_{\infty,\rho} \leq C & \text{s.t. } (1-\gamma) b_0 + \langle \boldsymbol{b}, \varphi(s) \; - \; \gamma \mathbb{E}[\varphi(s') \mid s, a] \rangle & \leq c(s,a), \qquad \forall (s,a) \in \mathcal{S} \times \mathcal{A}_s \end{array}$$

We show in Appendix A.1 that the optimal FELP solution approximates V^* with arbitrary accuracy.

3 Feature-based Approximate Linear Programs

In the literature, an ALP is derived by replacing V' in ELP by a VFA with pre-specified basis functions. Instead, we obtain an ALP by replacing $b_0 + \langle \boldsymbol{b}, \varphi(s) \rangle$ in FELP by the sampled VFA $V(s;\boldsymbol{\beta}) := \beta_0 + \sum_{i=1}^N \beta_i \varphi(s;\theta_i)$ where each θ_i is an iid samples from ρ and $\boldsymbol{\beta} := (\beta_0, \beta_1, \ldots, \beta_N)$. We refer to the ALP constructed using these N samples as feature-based approximate linear program, $\text{FALP}_{(\text{N})}$,

$$\begin{aligned} \max_{\boldsymbol{\beta}} \quad & \beta_0 + \sum_{i=1}^N \beta_i \mathbb{E}_\nu \big[\varphi(s;\theta_i)\big] \\ \text{s.t.} \quad & (1-\gamma)\beta_0 + \sum_{i=1}^N \beta_i \left(\varphi(s;\theta_i) - \gamma \mathbb{E}\big[\varphi(s';\theta_i) | s, a\big]\right) \leq c(s,a), \quad \forall (s,a) \in \mathcal{S} \times \mathcal{A}_s \end{aligned} \tag{2}$$

Let $\boldsymbol{\beta}_{\text{N}}^{\text{FA}} := (\beta_{N,0}^{\text{FA}}, \ldots, \beta_{N,N}^{\text{FA}})$ be an $\text{FALP}_{(\text{N})}$ optimal solution and define shorthand $V(\boldsymbol{\beta}) \equiv V(\cdot;\boldsymbol{\beta})$. It is known that ALP provides VFAs that are state-wise lower bound on V^*. Similarly, in our setting, we have $V(\boldsymbol{\beta}_{\text{N}}^{\text{FA}}) \leq V^*$ for a any N. Also, for $N' > N$, if $\text{FALP}_{(\text{N}')}$ contains the same random bases as $\text{FALP}_{(\text{N})}$ and $N' - N$ additional iid sampled basis functions, then the VFA from $\text{FALP}_{(\text{N}')}$ improves the $\text{FALP}_{(\text{N})}$ VFA with respect to $(1,\nu)$-norm, that is, $\|V^* - V(\boldsymbol{\beta}_{\text{N}'}^{\text{FA}})\|_{1,\nu} \leq \|V^* - V(\boldsymbol{\beta}_{\text{N}}^{\text{FA}})\|_{1,\nu}$. Moreover, we derive a finite sampling bound on N for VFA $V(\boldsymbol{\beta}_{\text{N}}^{\text{FA}})$ to be close to V^* in Appendix A.2.

Algorithm 1: Random Basis Function Generation for VFA Computation using Math Programs

Require: distribution ν, pair (φ, ρ), math program $\mathcal{M}_{(\text{N})}$, optimality tolerance τ, and batch size B.
Initialize: $N \leftarrow 0$, $\vartheta \leftarrow \{\}$, $\boldsymbol{\beta}^{\text{UB}} \leftarrow \mathbf{0} \in \mathbb{R}^B$, $\boldsymbol{\beta}^{\text{LB}} \leftarrow \mathbf{0} \in \mathbb{R}^B$, and $\tau^* \leftarrow 1$.
while $\tau^* > \tau$ **do**
- (i) Update $N = N + B$.
- (ii) Sample B independent samples $\{\theta_1, \ldots, \theta_B\}$ from $\rho(\theta)$ and set $\vartheta = \vartheta \cup \{\theta_1, \ldots, \theta_B\}$.
- (iii) Solve $\mathcal{M}_{(\text{N})}$ to obtain coefficents $\boldsymbol{\beta}_{\text{N}} \in \Re^{N+1}$ and compute $\text{PC}(\boldsymbol{\beta}_{\text{N}})$ and $\text{LB}(\boldsymbol{\beta}_{\text{N}})$.
- (iv) **if** $\text{LB}(\boldsymbol{\beta}_{\text{N}}) \geq \text{LB}(\boldsymbol{\beta}^{\text{LB}})$ **do** redefine $\boldsymbol{\beta}^{\text{LB}}$ as $\boldsymbol{\beta}_{\text{N}}$.
- (v) **if** $\text{PC}(\boldsymbol{\beta}_{\text{N}}) \leq \text{PC}(\boldsymbol{\beta}^{\text{UB}})$ **do** redefine $\boldsymbol{\beta}^{\text{UB}}$ as $\boldsymbol{\beta}_{\text{N}}$.
- (vi) Compute $\tau^* = 1 - \text{LB}(\boldsymbol{\beta}^{\text{LB}})/\text{PC}(\boldsymbol{\beta}^{\text{UB}})$.

Return: coefficients $\boldsymbol{\beta}^{\text{LB}}$ and $\boldsymbol{\beta}^{\text{UB}}$.

Algorithm. Given the worst case nature of the bound on N in Theorem 1, it is likely too large to be used in practice. We thus evaluate if a particular N is large enough to obtain near-optimal policies and bounds from $\text{FALP}_{(\text{N})}$. For a function $V(\boldsymbol{\beta}) \leq V^*$, the lower bound on the optimal policy cost $\text{PC}(\pi^*) \equiv \mathbb{E}_\chi[V^*(s)]$ is $\text{LB}(\boldsymbol{\beta}) := \mathbb{E}_\chi[V(s;\boldsymbol{\beta})]$. The upper bound is defined with respect to the so-called greedy policy $\pi_g(\boldsymbol{\beta}_{\text{N}}^{\text{FA}})$ associated with $V(\boldsymbol{\beta}_{\text{N}}^{\text{FA}})$ (see, e.g., [16]). The action taken by this policy at state $s \in \mathcal{S}$ solves program $\min_{a \in \mathcal{A}_s}\{c(s,a) + \gamma \mathbb{E}[V(s';\boldsymbol{\beta}_{\text{N}}^{\text{FA}})|s,a]\}$. Algorithm 1 employs $\text{FALP}_{(\text{N})}$ in an iterative procedure reflecting the scheme in Figure 1(b). We use $\mathcal{M}_{(\text{N})}$ to represent a generic math program (= $\text{FALP}_{(\text{N})}$ in this section) parameterized by ν and the number of samples N. The main steps of the algorithm are the following. In Step (i), the number of sampled bases is incremented by B. In Step (ii), B independent basis function parameters are sampled from $\rho(\theta)$ and appended to ϑ. In Step (iii), the math program $\mathcal{M}_{(\text{N})}$ embedding the random bases of set ϑ is solved and the resulting VFA coefficient vector $\boldsymbol{\beta}_{\text{N}}$ is used to compute the greedy policy cost $\text{PC}(\boldsymbol{\beta}_{\text{N}}) := \text{PC}(\pi_g(\boldsymbol{\beta}))$ and lower bound $\text{LB}(\boldsymbol{\beta}_{\text{N}})$. Steps (iv) and (v) update $\boldsymbol{\beta}^{\text{LB}}$ and $\boldsymbol{\beta}^{\text{UB}}$ if there is improvement in the lower bound and policy cost, respectively. The optimality gap percentage τ^* is updated in Step (vi) using $\text{LB}(\boldsymbol{\beta}^{\text{LB}})$ and $\text{PC}(\boldsymbol{\beta}^{\text{UB}})$. If $\tau^* \leq \tau$, Algorithm 1 terminates and returns the VFA vectors $\boldsymbol{\beta}^{\text{LB}}$ and $\boldsymbol{\beta}^{\text{UB}}$ corresponding to the tightest lower bound and best policy, respectively.

In Appendix A.3 we show under a mild condition that Algorithm 1 converges. Despite the asymptotic property of lower and upper bounds in Algorithm 1, $\mathrm{LB}(\boldsymbol{\beta}_{\mathrm{N}})$ and $\mathrm{PC}(\boldsymbol{\beta}_{\mathrm{N}})$ may not monotonically improve with N. For $\nu = \chi$, the sequence of lower bounds becomes monotonic because $\mathrm{LB}(\boldsymbol{\beta}_{\mathrm{N}}) \equiv \mathbb{E}_{\chi}[V(\boldsymbol{\beta}_{\mathrm{N}})]$ and the objective function $\mathbb{E}_{\nu}[V(\boldsymbol{\beta}_{\mathrm{N}})]$ of $\mathrm{FALP}_{(\mathrm{N})}$ coincide. However, even for $\nu = \chi$, $\mathrm{PC}(\boldsymbol{\beta}^{\mathrm{FA}}_{\mathrm{N}})$ may worsen as more random bases are added. We refer to this undesirable behavior as policy cost fluctuation. Let $\mu_{\chi}(\mathcal{S}_1; \boldsymbol{\beta})$ be the state visit frequency that encodes the probability of visiting subset $\mathcal{S}_1 \subseteq \mathcal{S}$ when actions are taken under $\pi_g(\boldsymbol{\beta})$ and the initial state s_0 is distributed according to χ. We formalize the definition of $\mu_{\chi}(\boldsymbol{\beta}) \equiv \mu_{\chi}(\mathcal{S}_1; \boldsymbol{\beta})$ in Appendix A.3. Proposition 1 shows that for a VFA satisfying $V(\boldsymbol{\beta}) \leq V^*$, the extra cost incurred by using the greedy policy $\pi_g(\boldsymbol{\beta})$ instead of π^* is bounded by the $(1, \mu_{\chi}(\boldsymbol{\beta}))$-norm difference between $V(\boldsymbol{\beta})$ and V^*.

Proposition 1 (Theorem 1 in [6]) *For a VFA $V(\boldsymbol{\beta})$ satisfying $V(\boldsymbol{\beta}) \leq V^*$, on has that* $\mathrm{PC}(\boldsymbol{\beta}) - \mathrm{PC}(\pi^*) \leq \|V(\boldsymbol{\beta}) - V^*\|_{1,\mu_{\chi}(\boldsymbol{\beta})}/(1-\gamma)$.

This result implies if $\nu = \mu_{\chi}(\boldsymbol{\beta})$, then an FALP VFA with a small $(1, \nu)$-norm error also guarantees good greedy policy performance, but such a performance guarantee does not hold when these frequencies differ. Therefore, it may be possible to mitigate policy cost fluctuation (see Appendix A.4) by improving the VFA at each iteration of Algorithm 1 with respect to both $(1, \mu_{\chi}(\boldsymbol{\beta}))$- and $(1, \nu)$- norms.

4 Self-guided Approximate Linear Programs

Motivated by Proposition 1, we explore the strategy of mitigating policy cost fluctuation by improving the term $\|V(\boldsymbol{\beta}) - V^*\|_{1,\mu_{\chi}(\boldsymbol{\beta})}$. We begin by presenting a modification of $\mathrm{FALP}_{(\mathrm{N})}$ to be used in conjunction with Algorithm 1, which we dub feature-based guided linear program and abbreviate $\mathrm{FGLP}_{(\mathrm{N})}$. We then describe how this linear program improves the aforementioned bound. Denoting by $\boldsymbol{\beta}^{\mathrm{FG}}_{\mathrm{N-B}}$ an optimal solution to $\mathrm{FGLP}_{(\mathrm{N-B})}$ and letting $V(\boldsymbol{\beta}^{\mathrm{FG}}_{0}) \equiv -\infty$, $\mathrm{FGLP}_{(\mathrm{N})}$ entails the same decision variables, objective function, and constraints as $\mathrm{FALP}_{(\mathrm{N})}$, in addition to self-guiding constraints,

$$\beta_0 + \beta_1\varphi(s;\theta_1) + \cdots + \beta_N\varphi(s;\theta_N) \;\geq\; V\big(s; \boldsymbol{\beta}^{\mathrm{FG}}_{\mathrm{N-B}}\big), \qquad \forall s \in \mathcal{S}. \tag{3}$$

Proposition 2 *Suppose $\mathcal{M}_{(\mathrm{N})} = FGLP_{(\mathrm{N})}$ in Algorithm 1. Then, for any given $n > 1$, the sequence of VFAs generated by this algorithm up to iteration n satisfies*

$$V(s;\boldsymbol{\beta}^{\mathrm{FA}}_{\mathrm{B}}) \;=\; V(s;\boldsymbol{\beta}^{\mathrm{FG}}_{\mathrm{B}}) \;\leq\; V(s;\boldsymbol{\beta}^{\mathrm{FG}}_{\mathrm{2B}}) \;\leq\; \cdots \;\leq\; V(s;\boldsymbol{\beta}^{\mathrm{FG}}_{\mathrm{nB}}) \;\leq\; V^*(s), \quad \forall s \in \mathcal{S}. \tag{4}$$

Proposition 2 establishes a key property of FGLP. The equality in (4) follows from our assumption that $V(\boldsymbol{\beta}^{\mathrm{FG}}_{0}) = -\infty$. It is easy to show that inequality $V(s;\boldsymbol{\beta}^{\mathrm{FG}}_{\bar{n}\mathrm{B}}) \leq V^*(s)$ holds for all $s \in \mathcal{S}$, and the inequalities of the type $V(s;\boldsymbol{\beta}^{\mathrm{FG}}_{\mathrm{N-B}}) \leq V(s;\boldsymbol{\beta}^{\mathrm{FG}}_{\mathrm{N}})$ are implied by constraints (3). Proposition 2 suggests that Algorithm 1 with FGLP generates a sequence of VFAs that gets (weakly) closer to V^* at all states. As a result, unlike FALP, the FGLP lower bound $\mathrm{LB}(\boldsymbol{\beta}_{\mathrm{N}})$ is non-decreasing with N even when the state-relevance distribution ν is not equal to the initial-state distribution χ. Also, VFAs $V(\boldsymbol{\beta}^{\mathrm{FG}}_{\mathrm{N}})$ and $V(\boldsymbol{\beta}^{\mathrm{FG}}_{\mathrm{N-B}})$ fulfill $\|V(\boldsymbol{\beta}^{\mathrm{FG}}_{\mathrm{N}}) - V^*\|_{1,\mu} \leq \|V(\boldsymbol{\beta}^{\mathrm{FG}}_{\mathrm{N-B}}) - V^*\|_{1,\mu}$ for any proper distribution μ defined over $\mathcal{S}$. Thus, for a fixed $\bar{n}$ and its corresponding distribution $\mu_{\chi}(\boldsymbol{\beta}^{\mathrm{FA}}_{\bar{n}\mathrm{B}})$, $\|V(\boldsymbol{\beta}^{\mathrm{FA}}_{\bar{n}\mathrm{B}}) - V^*\|_{1,\mu_{\chi}(\boldsymbol{\beta}^{\mathrm{FA}}_{\bar{n}\mathrm{B}})}$ is non-increasing in n. This property tackles policy cost fluctuation but does not hold when using FALP. We illustrate how FGLP mitigates policy cost fluctuation in Appendix A.4.

Studying the quality of the sequence of FGLP VFAs generated by Algorithm 1 is challenging because consecutive VFAs in this sequence are coupled by the self-guiding ALP constraints (3). Analogous to Theorem 1, we investigate a finite sampling bound for $\mathrm{FGLP}_{(\mathrm{N})}$. The techniques used to obtain a sampling bound for $\mathrm{FALP}_{(\mathrm{N})}$ in Theorem 1 (understandably) do not factor in the effect of $V(\boldsymbol{\beta}^{\mathrm{FG}}_{\mathrm{N}})$ (see EC.3 in OP[1]) and thus do not provide a useful lower bound on H for FGLP of the type described above. We therefore develop a new projection-based analysis to bound H. We show that for a given VFA $V(s;\boldsymbol{\beta}^{\mathrm{FG}}_{\mathrm{N}})$, there is a finite H and vector $\boldsymbol{\beta}^{\mathrm{FG}}_{\mathrm{N+H}}$ such that $\boldsymbol{\beta}^{\mathrm{FG}}_{\mathrm{N+H}}$ is feasible to constraints (2) and is near-feasible to constraints (3), and VFA $V(s;\boldsymbol{\beta}^{\mathrm{FG}}_{\mathrm{N+H}})$ is "close" to $V^*(s)$. We refer a reader to §4.2 of OP[1] for additional details and explanations on the FGLP sampling bound.

5 Conclusions

We propose a procedure for basis function generation in approximate linear programming, which is an established approach to obtain value function approximations (VFAs) for high dimensional Markov decision processes (MDPs). Our application-agnostic procedure embeds random basis functions generated via inexpensive sampling in an approximate linear program (ALP), which we refer to as random feature-based ALP (FALP). FALP side-steps the implementation task of basis function engineering when using ALP, which is typically both ad-hoc and application-specific. We provide a sampling guarantee for the VFA generated by FALP to be arbitrarily close to the MDP value function. Despite this worst-case sampling guarantee, the FALP policy performance can fluctuate significantly

as random basis functions are added to FALP iteratively. We modify FALP, dubbed feature-based guided linear program (FGLP), to circumvent this issue. FGLP adds constraints to FALP requiring its VFA to be a pointwise upper bound on a previously constructed FGLP with fewer random bases. Similar to FALP, we develop a finite sampling bound for FGLP.

Notes

OP[1]. The original paper associated with this workshop paper is currently under revision at Management Science journal and is available at https://www.dropbox.com/s/4rsh9srmbjrdzom/NeurIPS_2020_self_guided_ALPs.

References

[1] D. Adelman and D. Klabjan. Computing near-optimal policies in generalized joint replenishment. *INFORMS Journal on Computing*, 24(1):148–164, 2012.

[2] D. Adelman and A. J. Mersereau. Dynamic capacity allocation to customers who remember past service. *Management Science*, 59(3):592–612, 2013.

[3] S. R. Balseiro, H. Gurkan, and P. Sun. Multiagent mechanism design without money. *Operations Research*, 67(5):1417–1436, 2019.

[4] N. Bhat, V. Farias, and C. C. Moallemi. Non-parametric approximate dynamic programming via the kernel method. In *Advances in Neural Information Processing Systems*, pages 386–394, 2012.

[5] D. Blado and A. Toriello. Relaxation analysis for the dynamic knapsack problem with stochastic item sizes. *SIAM Journal on Optimization*, 29(1):1–30, 2019.

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

[7] V. V. Desai, V. F. Farias, and C. C. Moallemi. Approximate dynamic programming via a smoothed linear program. *Operations Research*, 60(3):655–674, 2012.

[8] W. B. Haskell, R. Jain, H. Sharma, and P. Yu. A universal empirical dynamic programming algorithm for continuous state MDPs. *IEEE Transactions on Automatic Control*, 65(1):115–129, Jan 2020. ISSN 2334-3303.

[9] O. Hernández-Lerma and J. B. Lasserre. *Discrete-time Markov Control Processes: Basic Optimality Criteria*, volume 30. Springer Science & Business Media, New York, NY, 1996.

[10] D. Klabjan and D. Adelman. An infinite-dimensional linear programming algorithm for deterministic semi-Markov decision processes on Borel spaces. *Mathematics of Operations Research*, 32(3):528–550, 2007.

[11] Q. Lin, S. Nadarajah, and N. Soheili. Revisiting approximate linear programming: Constraint-violation learning with applications to inventory control and energy storage. *Management Science*, page (forthcoming), 2019.

[12] Y. Lu, P. Dhillon, D. P. Foster, and L. Ungar. Faster ridge regression via the subsampled randomized hadamard transform. In *Advances in Neural Information Processing Systems*, pages 369–377, 2013.

[13] B. McWilliams, D. Balduzzi, and J. M. Buhmann. Correlated random features for fast semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 440–448, 2013.

[14] M. Mladenov, C. Boutilier, D. Schuurmans, G. Elidan, O. Meshi, and T. Lu. Approximate linear programming for logistic Markov decision processes. In *Proceedings of the Twenty-sixth International Joint Conference on Artificial Intelligence (IJCAI-17)*, pages 2486–2493, 2017.

[15] S. Nadarajah, F. Margot, and N. Secomandi. Relaxations of approximate linear programs for the real option management of commodity storage. *Management Science*, 61(12):3054–3076, 2015.

[16] W. B. Powell. *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. John Wiley & Sons, Hoboken, NJ, 2007.

[17] M. L. Puterman. *Markov Decision Processes: Discrete Dtochastic Dynamic Programming*. John Wiley & Sons, Hoboken, NJ, 1994.

[18] A. Rahimi and B. Recht. Uniform approximation of functions with random bases. In *2008 46th Annual Allerton Conference on Communication, Control, and Computing*, pages 555–561, Sep. 2008.

[19] P. J. Schweitzer and A. Seidmann. Generalized polynomial approximations in Markovian decision processes. *Journal of Mathematical Analysis and Applications*, 110(2):568–582, 1985.

[20] C. Tong and H. Topaloglu. On the approximate linear programming approach for network revenue management problems. *INFORMS Journal on Computing*, 26(1):121–134, 2013.

[21] L. Wu, P.-Y. Chen, I. E.-H. Yen, F. Xu, Y. Xia, and C. Aggarwal. Scalable spectral clustering using random binning features. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2506–2515, 2018.

A Appendix

A.1 FELP Reformulation

Similar to [18], we make the following technical assumptions on random bases to establish our theoretical results.

Assumption 1 *The class of random basis functions φ is universal, and its sampling distribution ρ has a finite second moment and satisfies $\rho(\theta) \in (0, U_\rho]$ for all $\theta \in \Theta$ for a finite positive constant U_ρ. Moreover, $\varphi(s;\theta) = \bar{\varphi}\big(q + \sum_{i=1}^{d_\mathcal{S}} \omega_i s_i\big)$, where $\theta = (q, \omega_1, \ldots, \omega_{d_\mathcal{S}})$ and $\bar{\varphi} : \mathbb{R} \mapsto \mathbb{R}$ is a mapping with finite Lipschitz constant L that satisfies $\|\bar{\varphi}\|_\infty \leq 1$ and $\bar{\varphi}(0) = 0$.*

Let $(b_0^{\mathrm{FE}}, \boldsymbol{b}^{\mathrm{FE}})$ denotes an FELP optimal solution and define the function $V^{\mathrm{FE}}(s) \coloneqq b_0^{\mathrm{FE}} + \langle \boldsymbol{b}^{\mathrm{FE}}, \varphi(s) \rangle$. Also, define $(1,\nu)$-norm of a function V as $\|V\|_{1,\nu} \coloneqq \mathbb{E}_\nu[|V|]$. Proposition 3 establishes that the function V^{FE} defined by an optimal FELP solution approximates V^* arbitrarily closely with respect to the $(1,\nu)$-norm.

Proposition 3 *Fix $\varepsilon > 0$. There exists a finite constant $C \in \Re_+$ such that any optimal solution $(b_0^{\mathrm{FE}}, \boldsymbol{b}^{\mathrm{FE}})$ to FELP with parameter C satisfies $\|V^* - V^{\mathrm{FE}}\|_{1,\nu} \leq 2\varepsilon/(1-\gamma)$.*

A.2 FALP Sampling Bound

Theorem 1 establishes an FALP$_{(\mathrm{N})}$ sampling bound relying on constants,

$$\Omega \coloneqq 4\mathrm{L}(\mathrm{diam}(\mathcal{S}) + 1)(\mathbb{E}_\rho\left[\langle \theta, \theta \rangle\right])^{1/2} \quad \text{and} \quad \Delta_\delta \coloneqq (2\ln(1/\delta))^{1/2},$$

where $\delta \in (0,1]$ is a fixed number, $\mathrm{L} \in \Re_+$ is a constant in Assumption 1, $\mathbb{E}_\rho$ is expectation under ρ, and $\mathrm{diam}(\mathcal{S}) \coloneqq \max_s \|s\|_2$ is the diameter of $\mathcal{S}$.

Theorem 1 *Given $\varepsilon > 0$, $\delta \in (0,1]$, and*

$$N \geq \left\lceil \varepsilon^{-2} \, \|\boldsymbol{b}^{\mathrm{FE}}\|_{\infty,\rho}^2 ((1+\gamma)\Omega/2 + \Delta_\delta)^2 \right\rceil,$$

any FALP$_{(\mathrm{N})}$ optimal solution $\boldsymbol{\beta}_{\mathrm{N}}^{\mathrm{FA}}$, satisfies $\|V^* - V(\boldsymbol{\beta}_{\mathrm{N}}^{\mathrm{FA}})\|_{1,\nu} \leq {}^{4\varepsilon}/_{(1-\gamma)}$ with a probability of $1-\delta$.

The bound in Theorem 1 is based on concentration arguments analogous to [18] but augmented to a constrained setting by leveraging the structure of FALP$_{(\mathrm{N})}$. First, a given infeasible solution to this program can be made feasible by scaling β_0 of the VFA. Second, a $(1,\nu)$-norm guarantee between $V(\boldsymbol{\beta}_{\mathrm{N}}^{\mathrm{FA}})$ and V^* is intuitively possible without knowledge of V^* because FALP$_{(\mathrm{N})}$ is equivalent to,

$$\min_{\boldsymbol{\beta}} \|V(\boldsymbol{\beta}) - V^*\|_{1,\nu} \;\text{ s.t. }\; V(s;\boldsymbol{\beta}) - \gamma \mathbb{E}\big[V(s';\boldsymbol{\beta}) \mid s, a\big] \leq c(s,a), \;\; \forall (s,a) \in \mathcal{S} \times \mathcal{A}_s,$$

by the virtue of Lemma 1 in [6]. Finally, we sharpen the constant in the original bound of [18] using ALP property $V(\boldsymbol{\beta}_{\mathrm{N}}^{\mathrm{FA}}) \leq V^*$ (see EC.2 of OP[1] for details).

A.3 Convergence of Algorithm 1

Proposition 4 shows that Algorithm 1 with $\mathcal{M}_{(\mathrm{N})} = \mathrm{FALP}_{(\mathrm{N})}$ terminates under mild conditions. For a given greedy policy $\pi_g(\boldsymbol{\beta})$, define its state visit frequency $\mu_\chi(\boldsymbol{\beta})$ as follows (see pages 132–133 in [9]):

$$\mu_\chi(\mathcal{S}_1;\boldsymbol{\beta}) \coloneqq \chi(\mathcal{S}_1) \; + \; \sum_{t=0}^{\infty} \gamma^{t+1} \mathbb{E}\Big[P\big(s_{t+1}^{\pi_g(\boldsymbol{\beta})} \in \mathcal{S}_1 \mid s_t, \pi_g(s_t;\boldsymbol{\beta})\big)\Big] \tag{5}$$

where state $s_{t+1}^{\pi_g(\boldsymbol{\beta})}$ and probability P retain their definitions from §2, and $\chi(\mathcal{S}_1)$ is the probability of the initial state belonging to $\mathcal{S}_1$. Exception $\mathbb{E}$ is taken with respect to policy $\pi_g(\boldsymbol{\beta})$ and distribution χ.

Proposition 4 *Suppose that $\mathcal{M}_{(\mathrm{N})} = $ FALP$_{(\mathrm{N})}$ in Algorithm 1, the distribution ν assigns positive mass to all non-zero measure subsets of the state space, and the state-visit frequency $\mu_\chi(\boldsymbol{\beta}_{\mathrm{N}})$ is bounded above by a constant for all N. Then, for a given $\delta \in (0,1]$ and $\tau \in (0,1]$, Algorithm 1 terminates after a finite number of iterations with a probability of at least $1-\delta$.*

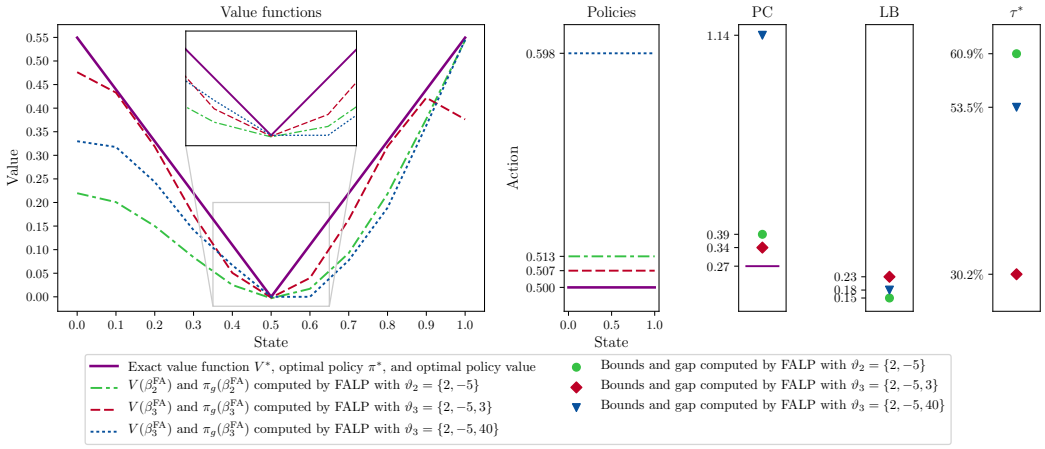
Proposition 4 establishes that the lower bounds and policy costs generated by Algorithm 1 with $\mathcal{M}_{(\mathrm{N})} = \mathrm{FALP}_{(\mathrm{N})}$ converge towards each other as the number of samples tends to infinity.

A.4 Illustrating Policy Cost Fluctuation

Consider a simple version of MDP (1) with $\mathcal{S} = \mathcal{A}_s = [0,1]$. State transitions are governed by the discrete distribution $P(s' = s|s,a) = 0.1$, $P(s' = a|s,a) = 0.9$, and $P(s' \notin \{s,a\}|s,a) = 0$. The immediate cost function is $c(s,a) = |s - 0.5|$ for all $(s,a) \in \mathcal{S} \times \mathcal{A}_s$ and future costs are discounted by $\gamma = 0.9$. Initial distribution χ is chosen to be uniform over $\mathcal{S}$. The optimal action $\pi^*(s)$ for all $s \in \mathcal{S}$ equals 0.5 since $c(s,a)$ equals 0, if $s = 0.5$, and is strictly positive, otherwise. Therefore, we have $V^*(s) = c(s,0.5)/(1-0.1\gamma)$ for all $s \in \mathcal{S}$. The optimal policy cost $\mathrm{PC}(\pi^*)$ is 0.27. Figure 2 displays this information using thick (purple) solid lines. For this MDP, greedy policy optimization reduces to $\min_{s \in [0,1]} V(s;\boldsymbol{\beta})$, which means $\pi_g(s,\boldsymbol{\beta}) = \bar{s}$ for all $s \in \mathcal{S}$ where $\bar{s}$ minimizes $V(\cdot;\boldsymbol{\beta})$.

Next, we analyze the VFAs and greedy policies resulting from the application of Algorithm 1 for three consecutive iterations with the parameter B set to 1. Specifically, we compare iterations two and three, which correspond to FALP with two and three random bases, respectively. We let $\nu = \chi$ so that the lower bound $\mathrm{LB}(\boldsymbol{\beta}_{\mathrm{N}})$ is non-decreasing in N and we can focus only on the fluctuation of the policy cost $\mathrm{PC}(\boldsymbol{\beta}_{\mathrm{N}})$. We use Fourier random

Figure 2: Results from executing Algorithm 1 with FALP on example.



basis functions $\varphi(s;\theta) = \cos(\theta s)$, where $\theta \in \Re$, to define FALP VFA. At the end of iteration two, suppose the random bases correspond to the sampled parameters in set $\vartheta = \{2, -5\}$ and $\boldsymbol{\beta}^{\mathrm{LB}} = \boldsymbol{\beta}^{\mathrm{UB}} = \boldsymbol{\beta}_2^{\mathrm{FA}}$. Figure 2 plots $V(\boldsymbol{\beta}_2^{\mathrm{FA}})$ and $\pi_g(\boldsymbol{\beta}_2^{\mathrm{FA}})$ in (green) dashed-dotted lines and displays the associated bounds using circular markers. The minimum of $V(\cdot;\boldsymbol{\beta}_2^{\mathrm{FA}})$ is attained at 0.513 and $\mathrm{LB}(\boldsymbol{\beta}_2^{\mathrm{FA}})$ equals 0.15. Moreover, $\pi_g(s, \boldsymbol{\beta}_2^{\mathrm{FA}})$ equals 0.513 for all $s \in \mathcal{S}$ and the greedy policy cost $\mathrm{PC}(\boldsymbol{\beta}_2^{\mathrm{FA}})$ equals 0.39. The corresponding optimality gap τ^* is 60.9%. At iteration three, we consider two scenarios for the sample θ_3.

Scenario 1 ($\theta_3 = 3$). VFA $V(\boldsymbol{\beta}_3^{\mathrm{FA}})$ with parameters in $\vartheta = \{2, -5, 3\}$, its greedy policy, and optimality gap are shown in Figure 2 using (red) dashed lines and diamond markers. The function $V(\cdot;\boldsymbol{\beta}_3^{\mathrm{FA}})$ attains its minimum over s at 0.507 and $\pi_g(\boldsymbol{\beta}_3^{\mathrm{FA}})$ equals this value at all states. In addition, $\mathrm{LB}(\boldsymbol{\beta}_3^{\mathrm{FA}})$ and $\mathrm{PC}(\boldsymbol{\beta}_3^{\mathrm{FA}})$ are 0.23 and 0.34, respectively, with both bounds improving over their respective iteration 2 values. Also, $\pi_g(\boldsymbol{\beta}_3^{\mathrm{FA}})$ becomes the best policy computed thus far with the optimality gap of 30.2%, which is significantly lower than the gap in iteration two because of the improvements in both the lower bound and policy cost.

Scenario 2 ($\theta_3 = 40$). The information displayed by (dark blue) dotted lines and triangular markers in Figure 2 corresponds to VFA $V(\boldsymbol{\beta}_3^{\mathrm{FA}})$ with parameters in $\vartheta_3 = \{2, -5, 40\}$. Both the minimum of $V(\cdot;\boldsymbol{\beta}_3^{\mathrm{FA}})$ and $\pi_g(\boldsymbol{\beta}_3^{\mathrm{FA}})$ equal 0.598. The lower bound $\mathrm{LB}(\boldsymbol{\beta}_3^{\mathrm{FA}})$ is 0.18 and improves on $\mathrm{LB}(\boldsymbol{\beta}_2^{\mathrm{FA}})$ as expected because ν equal to χ. Thus, $\boldsymbol{\beta}^{\mathrm{LB}} = \boldsymbol{\beta}_3^{\mathrm{FA}}$. In contrast, the upper bound $\mathrm{PC}(\boldsymbol{\beta}_3^{\mathrm{FA}})$ is 1.14, which is worse than $\mathrm{PC}(\boldsymbol{\beta}_2^{\mathrm{FA}})$, and $\boldsymbol{\beta}^{\mathrm{UB}}$ thus remains $\boldsymbol{\beta}_2^{\mathrm{FA}}$. In other words, we do not find an improved greedy policy. The optimality gap computed by Algorithm 1 equals 53.5% and is based on $\mathrm{PC}(\boldsymbol{\beta}_2^{\mathrm{FA}})$ and $\mathrm{LB}(\boldsymbol{\beta}_3^{\mathrm{FA}})$. This gap is smaller than the one of iteration 2 due to the stronger lower bound. If one instead computed the optimality gap of $\pi_g(\boldsymbol{\beta}_3^{\mathrm{FA}})$ with respect to $\mathrm{LB}(\boldsymbol{\beta}_3^{\mathrm{FA}})$, it would be 84.2% (i.e., $1 - {}^{0.18}/_{1.14}$), which highlights the significant worsening of the greedy policy in iteration 3.

Scenario 2 of iteration 3 makes concrete the notion of policy cost fluctuation in FALP. We analyzed both above scenarios when using Algorithm 1 with $\mathcal{M}_{(\mathrm{N})} = \mathrm{FGLP}_{(\mathrm{N})}$. As a result of the self-guiding constraints in FGLP, the worst-case bound on policy performance was roughly equal to 0.08 for both scenarios 1 and 2 of iteration 3, which is a significant improvement over the respective worst-case performance bounds of 0.99 and 9.98 when using FALP. Interestingly, accompanying this worst-case bound improvement, the policy cost improved in both the scenarios of iteration 3.