MATHEMATICS OF OPERATIONS RESEARCH
Vol. 31, No. 2, May 2006, pp. 234–244
ISSN 0364-765X | EISSN 1526-5471 | 06 | 3102 | 0234

DOI 10.1287/moor.1060.0188

Performance Loss Bounds for Approximate Value Iteration with State Aggregation

Benjamin Van Roy
Stanford University, Stanford, California 94305, bvr@stanford.edu

We consider approximate value iteration with a parameterized approximator in which the state space is partitioned and the optimal cost-to-go function over each partition is approximated by a constant. We establish performance loss bounds for policies derived from approximations associated with fixed points. These bounds identify benefits to using invariant distributions of appropriate policies as projection weights. Such projection weighting relates to what is done by temporal-difference learning. Our analysis also leads to the first performance loss bound for approximate value iteration with an average-cost objective.

Key words: approximate value iteration; state aggregation; temporal-difference learning
MSC2000 subject classification: Primary: 90C39, 90C40; secondary: 68T05, 68T37
OR/MS subject classification: Primary: dynamic programming/optimal control; Markov; finite state
History: Received August 2, 2004; revised August 12, 2005.

1. Preliminaries. Consider a discrete-time communicating Markov decision process (MDP) with a finite state space $\mathcal{S} = \{1, \ldots, |\mathcal{S}|\}$. At each state $x \in \mathcal{S}$, there is a finite set $\mathcal{U}_x$ of admissible actions. If the current state is x and an action $u \in \mathcal{U}_x$ is selected, a cost of $g_u(x)$ is incurred and the system transitions to a state $y \in \mathcal{S}$ with probability $p_{xy}(u)$. For any $x \in \mathcal{S}$ and $u \in \mathcal{U}_x$, $\sum_{y \in \mathcal{S}} p_{xy}(u) = 1$. Costs are discounted at a rate of $\alpha \in (0, 1)$ per period. Each instance of such an MDP is defined by a quintuple $(\mathcal{S}, \mathcal{U}, g, p, \alpha)$.

A (stationary deterministic) policy is a mapping μ that assigns an action $u \in \mathcal{U}_x$ to each state $x \in \mathcal{S}$. If actions are selected based on a policy μ, the state follows a Markov process with transition matrix P_μ, where each (x, y)th entry is equal to $p_{xy}(\mu(x))$. The restriction to communicating MDPs ensures that it is possible to reach any state from any other state.

Each policy μ is associated with a cost-to-go function $J_\mu \in \Re^{|\mathcal{S}|}$, defined by

$$J_\mu = \sum_{t=0}^{\infty} \alpha^t P_\mu^t g_\mu = (I - \alpha P_\mu)^{-1} g_\mu,$$

where, with some abuse of notation, $g_\mu(x) = g_{\mu(x)}(x)$ for each $x \in \mathcal{S}$. A policy μ is said to be *greedy* with respect to a function J if

$$\mu(x) \in \operatorname*{argmin}_{u \in \mathcal{U}_x} \left(g_u(x) + \alpha \sum_{y \in \mathcal{S}} p_{xy}(u) J(y) \right),$$

for all $x \in \mathcal{S}$.

The optimal cost-to-go function $J^* \in \Re^{|\mathcal{S}|}$ is defined by $J^*(x) = \min_\mu J_\mu(x)$, for all $x \in \mathcal{S}$. A policy μ^* is said to be optimal if $J_{\mu^*} = J^*$. It is well known that an optimal policy exists. Further, a policy μ^* is optimal if and only if it is greedy with respect to J^*. Hence, given the optimal cost-to-go function, optimal actions can be computed minimizing the right-hand side of the above inclusion.

Value iteration generates a sequence J_l converging to J^* according to $J_{l+1} = TJ_l$, where T is the dynamic programming operator, defined by

$$(TJ)(x) = \min_{u \in \mathcal{U}_x} \left(g_u(x) + \alpha \sum_{y \in \mathcal{S}} p_{xy}(u) J(y) \right),$$

for all $x \in \mathcal{S}$ and $J \in \Re^{|\mathcal{S}|}$. This sequence converges to J^* for any initialization of J_0.

2. Preview of results. Let the state space $\mathcal{S}$ be partitioned into K subsets, represented by a matrix $\Phi \in \Re^{|\mathcal{S}| \times K}$ in which each kth column is made up of binary-valued components indicating whether or not each state is in the kth partition. Approximate value iteration computes a vector $\tilde{r}$ that solves $\Phi\tilde{r} = \Pi_\pi T \Phi \tilde{r}$, where T is the dynamic programming operator and the matrix Π_π projects onto the column space of Φ with respect to a

weighted Euclidean norm, with weights $\pi \in \Re_+^{|\mathcal{S}|}$. If the support of π intersects every partition, the cost-to-go function $J_{\mu_{\tilde{r}}}$ of each policy $\mu_{\tilde{r}}$ that is greedy with respect to $\Phi\tilde{r}$ satisfies

$$(1-\alpha)\|J_{\mu_{\tilde{r}}} - J^*\|_\infty \leq \frac{4\alpha}{1-\alpha} \min_{r\in\Re^K} \|J^* - \Phi r\|_\infty, \tag{1}$$

as established in Tsitsiklis and Van Roy [46].

The left-hand side of Equation (1) is a measure of performance loss in which the coefficient $1-\alpha$ serves as a normalizing constant. In particular, $(1-\alpha)J_\mu(x)$ represents an exponentially weighted average of expected future costs, given that the process starts in state x and is controlled by policy μ. A natural question is why the term $\min_{r\in\Re^K} \|J^* - \Phi r\|_\infty$ should not be similarly normalized. The short answer is that the part of $J^*(x)$ that grows with $1/(1-\alpha)$ is constant over x and therefore offset by part of Φr. We will discuss this later in greater depth.

Typically, α is close to 1 and, therefore, the coefficient $4\alpha/(1-\alpha)$ on the right-hand side of Equation (1) is large. Unfortunately, this dependence on α is necessary because, as we will establish, for any $\alpha \in (0,1)$ and value taken by $\min_{r\in\Re^K} \|J^* - \Phi r\|_\infty$, the bound is sharp.

Let $\pi_{\tilde{r}}$ denote an invariant state distribution when the process is controlled by policy $\mu_{\tilde{r}}$. We consider a vector $\tilde{r}$ that solves $\Phi\tilde{r} = \Pi_{\pi_{\tilde{r}}}\Phi\tilde{r}$ and for which the support of $\pi_{\tilde{r}}$ intersects every partition. We will establish that if a solution $\tilde{r}$ exists, each associated greedy policy $\mu_{\tilde{r}}$ satisfies

$$(1-\alpha)\pi_{\tilde{r}}^T(J_{\mu_{\tilde{r}}} - J^*) \leq 2\alpha \min_{r\in\Re^K} \|J^* - \Phi r\|_\infty. \tag{2}$$

For any μ, $\lim_{\alpha\uparrow 1}(1-\alpha)(J_\mu(x) - J^*(x))$ is nonnegative and independent of x and, therefore, for any probability distribution π,

$$\lim_{\alpha\uparrow 1}(1-\alpha)\|J_\mu - J^*\|_\infty = \lim_{\alpha\uparrow 1}(1-\alpha)\pi^T(J_\mu - J^*).$$

As such, for α close to one, the left-hand sides of Equations (1) and (2) are comparable. Relative to the right-hand side of Equation (1), the omission of a factor of $2/(1-\alpha)$ in Equation (2) represents a major improvement.

The dependence of the projection on $\tilde{r}$ is motivated by a version of temporal-difference learning that can be viewed as trying to compute a solution to $\Phi\tilde{r} = \Pi_{\pi_{\tilde{r}}}T\Phi\tilde{r}$. A solution need not exist. As established in de Farias and Van Roy [20], one approach to ensuring existence involves incorporating exploration in the learning process and using a modified dynamic programming operator T^ϵ that accounts for this. We will show that a bound similar to Equation (2) holds for an approximately greedy policy $\mu_{\tilde{r}}^\epsilon$, which we will refer to as the ϵ-greedy Boltzmann exploration policy, when $\Phi\tilde{r} = \Pi_{\pi_{\tilde{r}}^\epsilon}T^\epsilon\Phi\tilde{r}$, where $\pi_{\tilde{r}}^\epsilon$ is the corresponding invariant state distribution.

As we will show, the analysis behind Equation (2) can be extended to accommodate an average-cost objective. This results in the first performance loss bound for approximate value iteration with an average-cost objective. In particular, the natural generalization of Equation (1) to the average-cost case gives rise to an infinite right-hand side.

Our results indicate that weighting a Euclidean norm projection by the invariant distribution of a greedy (or approximately greedy) policy can lead to a dramatic performance gain. It is intriguing that temporal-difference learning implicitly carries out such a projection and, consequently, any limit of convergence obeys the stronger performance loss bound.

This is not the first time that the invariant distribution has been shown to play a critical role in approximate value iteration and temporal-difference learning. In prior work involving approximation of a cost-to-go function for a fixed policy (no control) and a general linearly parameterized approximator (arbitrary matrix Φ), it was shown that weighting by the invariant distribution is key to ensuring convergence and an approximation error bound (Tsitsiklis and Van Roy [47, 48], Van Roy [52]). Earlier empirical work anticipated this (Sutton [42, 43]).

An important caveat to the line of work presented in this paper is that no known algorithms are guaranteed to solve the equations $\Phi\tilde{r} = \Pi_{\pi_{\tilde{r}}}T\Phi\tilde{r}$ or $\Phi\tilde{r} = \Pi_{\pi_{\tilde{r}}^\epsilon}T^\epsilon\Phi\tilde{r}$ for a broad class of relevant problem instances. One can apply versions of the temporal-difference learning algorithm in such a way that in the event of convergence the limit solves the latter equation. However, there are no convergence let alone efficiency guarantees.

3. State aggregation.

The state spaces of relevant MDPs are typically so large that computation and storage of a cost-to-go function is infeasible. One approach to dealing with this obstacle involves partitioning the state space $\mathcal{S}$ into a manageable number K of disjoint subsets $\mathcal{S}_1, \ldots, \mathcal{S}_K$ and approximating the optimal cost-to-go function with a function that is constant over each partition. This can be thought of as a form of state aggregation—all states within a given partition are assumed to share a common optimal cost-to-go.

There is substantial literature on the use of state aggregation to accelerate computation of effective policies for Markov decision processes (Fox [23], Bertsekas [4], Whitt [57], Morin [34], Hinderer [27], Mendelssohn [32], Axsäter [1], Birge [11], Bean et al. [3], Bertsekas and Castañon [6], Chow and Tsitsiklis [18, 19], Kushner and Dupuis [30], Gordon [24, 25, 26], Moore and Atkeson [33], Barraquand and Martineau [2], Tsitsiklis and Van Roy [46], Rust [38], Lambert et al. [31]).

To represent an approximation, we define a matrix $\Phi \in \Re^{|\mathcal{S}|\times K}$ such that each kth column is an indicator function for the kth partition $\mathcal{S}_k$. Hence, for any $r \in \Re^K$, k, and $x \in \mathcal{S}_k$, $(\Phi r)(x) = r_k$. In this paper, we study variations of value iteration, each of which aims to compute a vector r so that Φr approximates J^*. Another important issue is how to partition the space. Though we will not make this a central topic of the paper, we discuss it briefly in the closing section.

The use of a policy μ_r that is greedy with respect to Φr is justified by the following result:

THEOREM 3.1. *If μ is a greedy policy with respect to a function $\tilde{J} \in \Re^{|\mathcal{S}|}$, then*

$$\|J_\mu - J^*\|_\infty \le \frac{2\alpha}{1-\alpha}\|J^* - \tilde{J}\|_\infty.$$

This result is easy to prove and has appeared in multiple contexts (see Singh and Yee [39] for a proof).

4. Approximate value iteration. One common way of approximating a function $J \in \Re^{|\mathcal{S}|}$ with a function of the form Φr involves projection with respect to a weighted Euclidean norm $\|\cdot\|_\pi$. The weighted Euclidean norm of a function J is defined by

$$\|J\|_{2,\pi} = \left(\sum_{x\in\mathcal{S}} \pi(x)J^2(x)\right)^{1/2}.$$

Here, $\pi \in \Re_+^{|\mathcal{S}|}$ is a vector of weights that assign relative emphasis among states. The projection $\Pi_\pi J$ is the function Φr that attains the minimum of $\|J - \Phi r\|_{2,\pi}$; if there are multiple functions Φr that attain the minimum, they must form an affine space and the projection is taken to be the one with minimal norm $\|\Phi r\|_{2,\pi}$. Note that in our context, where each kth column of Φ represents an indicator function for the kth partition, for any π, J, and $x \in \mathcal{S}_k$,

$$(\Pi_\pi J)(x) = \frac{\sum_{y\in\mathcal{S}_k} \pi(y)J(y)}{\sum_{y\in\mathcal{S}_k} \pi(y)}.$$

Approximate value iteration begins with a function $\Phi r^{(0)}$ and generates a sequence according to

$$\Phi r^{(l+1)} = \Pi_\pi T \Phi r^{(l)}.$$

It is well known that the dynamic programming operator T is a contraction mapping with respect to the maximum norm. Further, Π_π is maximum-norm nonexpansive (Tsitsiklis and Van Roy [46], Gordon [24, 25, 26]). (This is not true for general Φ, but is true in our context in which columns of Φ are indicator functions for partitions.) It follows that the composition $\Pi_\pi T$ is a contraction mapping. By the contraction mapping theorem, $\Pi_\pi T$ has a unique fixed point $\Phi\tilde{r}$, which is the limit of the sequence $\Phi r^{(l)}$. Further, the following result holds:

THEOREM 4.1. *For any MDP, partition, and weights π with support intersecting every partition, if $\Phi\tilde{r} = \Pi_\pi T \Phi\tilde{r}$, then*

$$\|\Phi\tilde{r} - J^*\|_\infty \le \frac{2}{1-\alpha}\min_{r\in\Re^K}\|J^* - \Phi r\|_\infty,$$

and

$$(1-\alpha)\|J_{\mu_{\tilde{r}}} - J^*\|_\infty \le \frac{4\alpha}{1-\alpha}\min_{r\in\Re^K}\|J^* - \Phi r\|_\infty.$$

The first inequality of the theorem is an *approximation error bound*, established in Tsitsiklis and Van Roy [46] and Gordon [24, 25, 26] for broader classes of approximators that include state aggregation as a special case. The second is a *performance loss bound*, derived by simply combining the approximation error bound and Theorem 3.1. Closely related results have been established in the context of observation-based control of partially observable MDPs (Jaakkola et al. [29]) and in the context of state aggregation methods for discretization in an MDP with a continuous state space (Whitt [57]).

Note that $J_{\mu_{\tilde{r}}}(x) \ge J^*(x)$ for all x, so the left-hand side of the performance loss bound is the maximal increase in cost-to-go, normalized by $1-\alpha$. This normalization is natural, since a cost-to-go function is a linear combination of expected future costs, with coefficients $1, \alpha, \alpha^2, \ldots$ which sum to $1/(1-\alpha)$.

Our motivation of the normalizing constant begs the question of whether, for fixed MDP parameters $(\mathcal{S}, \mathcal{U}, g, p)$ and fixed Φ, $\min_r \|J^* - \Phi r\|_\infty$ also grows with $1/(1-\alpha)$. It turns out that $\min_r \|J^* - \Phi r\|_\infty = O(1)$. To see why, note that for any μ,

$$J_\mu = (I - \alpha P_\mu)^{-1} g_\mu = \frac{1}{1-\alpha}\lambda_\mu + h_\mu,$$

where $\lambda_\mu(x)$ is the *expected average cost* if the process starts in state x and is controlled by policy μ,

$$\lambda_\mu = \lim_{\tau\to\infty} \frac{1}{\tau} \sum_{t=0}^{\tau-1} P_\mu^t g_\mu,$$

and h_μ is the *discounted differential cost function*

$$h_\mu = (I - \alpha P_\mu)^{-1}(g_\mu - \lambda_\mu).$$

Both λ_μ and h_μ converge to finite vectors as α approaches 1 (Blackwell [12]). For an optimal policy μ^*, $\lim_{\alpha\uparrow 1} \lambda_{\mu^*}(x)$ does not depend on x (in our context of a communicating MDP). Since constant functions lie in the range of Φ,

$$\lim_{\alpha\uparrow 1} \min_{r\in\Re^K} \|J^* - \Phi r\|_\infty \le \lim_{\alpha\uparrow 1} \|h_{\mu^*}\|_\infty < \infty.$$

The performance loss bound still exhibits an undesirable dependence on α through the coefficient $4\alpha/(1-\alpha)$. In most relevant contexts, α is close to 1; a representative value might be 0.99. Consequently, $4\alpha/(1-\alpha)$ can be very large. Unfortunately, the bound is sharp, as expressed by the following theorem. We will denote by $\mathbf{1}$ the vector with every component equal to 1.

THEOREM 4.2. *For any $\delta > 0$, $\alpha \in (0,1)$, and $\Delta \ge 0$, there exists MDP parameters $(\mathcal{S}, \mathcal{U}, g, p)$ and a partition such that $\min_{r\in\Re^K} \|J^* - \Phi r\|_\infty = \Delta$ and, if $\Phi\tilde{r} = \Pi_\pi T \Phi \tilde{r}$ with $\pi = \mathbf{1}$,*

$$(1-\alpha)\|J_{\mu_{\tilde{r}}} - J^*\|_\infty \ge \frac{4\alpha}{1-\alpha} \min_{r\in\Re^K} \|J^* - \Phi r\|_\infty - \delta.$$

This theorem can be established by a variation of an example from Tsitsiklis and Van Roy [46]. We will discuss this new example in §6. The choice of uniform weights ($\pi = \mathbf{1}$) is meant to point out that even for such a simple, perhaps natural, choice of weights the performance loss bound is sharp.

Based on Theorems 4.1 and 4.2, one might expect that there exists MDP parameters $(\mathcal{S}, \mathcal{U}, g, p)$ and a partition such that with $\pi = \mathbf{1}$,

$$(1-\alpha)\|J_{\mu_{\tilde{r}}} - J^*\|_\infty = \Theta\left(\frac{1}{1-\alpha} \min_{r\in\Re^K} \|J^* - \Phi r\|_\infty\right),$$

in other words, that the performance loss is both lower and upper bounded by $1/(1-\alpha)$ times the smallest possible approximation error. It turns out that this is not true, at least if we restrict to a finite state space. However, as the following theorem establishes, the coefficient multiplying $\min_{r\in\Re^K} \|J^* - \Phi r\|_\infty$ can grow arbitrarily large as α increases, keeping all else fixed.

THEOREM 4.3. *For any L and $\Delta \ge 0$, there exist MDP parameters $(\mathcal{S}, \mathcal{U}, g, p)$ and a partition such that $\lim_{\alpha\uparrow 1} \min_{r\in\Re^K} \|J^* - \Phi r\|_\infty = \Delta$ and, if $\Phi\tilde{r} = \Pi_\pi T \Phi \tilde{r}$ with $\pi = \mathbf{1}$,*

$$\liminf_{\alpha\uparrow 1} (1-\alpha)(J_{\mu_{\tilde{r}}}(x) - J^*(x)) \ge L \lim_{\alpha\uparrow 1} \min_{r\in\Re^K} \|J^* - \Phi r\|_\infty,$$

for all $x \in \mathcal{S}$.

This theorem is also established by the example we will present in §6.

For any μ and x,

$$\lim_{\alpha\uparrow 1}((1-\alpha)J_\mu(x) - \lambda_\mu(x)) = \lim_{\alpha\uparrow 1}(1-\alpha)h_\mu(x) = 0.$$

Combined with Theorem 4.3, this yields the following corollary.

COROLLARY 4.1. *For any L and $\Delta \ge 0$, there exist MDP parameters $(\mathcal{S}, \mathcal{U}, g, p)$ and a partition such that $\lim_{\alpha\uparrow 1} \min_{r\in\Re^K} \|J^* - \Phi r\|_\infty = \Delta$ and, if $\Phi\tilde{r} = \Pi_\pi T \Phi \tilde{r}$ with $\pi = \mathbf{1}$,*

$$\liminf_{\alpha\uparrow \mathbf{1}} (\lambda_{\mu_{\tilde{r}}}(x) - \lambda_{\mu^*}(x)) \ge L \lim_{\alpha\uparrow 1} \min_{r\in\Re^K} \|J^* - \Phi r\|_\infty,$$

for all $x \in \mathcal{S}$.

5. Using the invariant distribution. In the previous section, we considered an approximation $\Phi\tilde{r}$ that solves $\Pi_\pi T\Phi\tilde{r} = \Phi\tilde{r}$ for some arbitrary pre selected weights π. We now turn to consider use of an invariant state distribution $\pi_{\tilde{r}}$ of $P_{\mu_{\tilde{r}}}$ as the weight vector.[1] This leads to a circular definition: The weights are used in defining $\tilde{r}$ and now we are defining the weights in terms of $\tilde{r}$. What we are really after here is a vector $\tilde{r}$ that satisfies $\Pi_{\pi_{\tilde{r}}} T\Phi\tilde{r} = \Phi\tilde{r}$. The following theorem captures the associated benefits.

THEOREM 5.1. *For any MDP and partition, if $\Phi\tilde{r} = \Pi_{\pi_{\tilde{r}}} T\Phi\tilde{r}$ and $\pi_{\tilde{r}}$ has support intersecting every partition,*

$$(1-\alpha)\pi_{\tilde{r}}^T(J_{\mu_{\tilde{r}}} - J^*) \le 2\alpha \min_{r\in\Re^K} \|J^* - \Phi r\|_\infty.$$

PROOF. We first set out to prove that $\pi_{\tilde{r}}^T T\Phi r = \pi_{\tilde{r}}^T J_{\mu_{\tilde{r}}}$. A simple argument will then be used to show that this implies the theorem. First, note that

$$\pi_{\tilde{r}}^T J_{\mu_{\tilde{r}}} = \pi_{\tilde{r}}^T \sum_{t=0}^{\infty} \alpha^t P_{\mu_{\tilde{r}}}^t g_{\mu_{\tilde{r}}} = \frac{1}{1-\alpha}\pi_{\tilde{r}}^T g_{\mu_{\tilde{r}}}.$$

Let $D_{\tilde{r}} = \mathrm{diag}(\pi_{\tilde{r}})$ and note that $D_{\tilde{r}}\Pi_{\mu_{\tilde{r}}} = \Pi_{\mu_{\tilde{r}}}^T D_{\tilde{r}}$ and $\Pi_{\mu_{\tilde{r}}}\mathbf{1} = \mathbf{1}$ (projections are self-adjoint and $\mathbf{1}$ is in the range of ours). Using these relations, we have

$$\begin{aligned}
\pi_{\tilde{r}}^T\Phi\tilde{r} &= \pi_{\tilde{r}}^T\Pi_{\pi_{\tilde{r}}} T\Phi\tilde{r}\\
&= \mathbf{1}^T D_{\tilde{r}}\Pi_{\pi_{\tilde{r}}} T\Phi\tilde{r}\\
&= (\Pi_{\pi_{\tilde{r}}}\mathbf{1})^T D_{\tilde{r}} T\Phi\tilde{r}\\
&= \mathbf{1}^T D_{\tilde{r}} T\Phi\tilde{r}\\
&= \pi_{\tilde{r}}^T T\Phi\tilde{r}\\
&= \pi_{\tilde{r}}^T(g_{\mu_{\tilde{r}}} + \alpha P_{\mu_{\tilde{r}}}\Phi\tilde{r})\\
&= \pi_{\tilde{r}}^T g_{\mu_{\tilde{r}}} + \alpha\pi_{\tilde{r}}^T\Phi\tilde{r}.
\end{aligned}$$

It follows from the fifth and final expressions that

$$\pi_{\tilde{r}}^T T\Phi\tilde{r} = \pi_{\tilde{r}}^T\Phi\tilde{r} = \frac{1}{1-\alpha}\pi_{\tilde{r}}^T g_{\mu_{\tilde{r}}} = \pi_{\tilde{r}}^T J_{\mu_{\tilde{r}}}.$$

Using this relation, we obtain

$$\begin{aligned}
\pi_{\tilde{r}}^T(J_{\mu_{\tilde{r}}} - J^*) &= \pi_{\tilde{r}}^T(T\Phi\tilde{r} - J^*)\\
&\le \|T\Phi\tilde{r} - J^*\|_\infty\\
&\le \alpha\|\Phi\tilde{r} - J^*\|_\infty\\
&\le \frac{2\alpha}{1-\alpha}\min_{r\in\Re^K}\|J^* - \Phi r\|_\infty,
\end{aligned}$$

where the final inequality follows from Theorem 4.1. □

When α is close to 1, which is typical, the right-hand side of our new performance loss bound is far less than that of Theorem 4.1. The primary improvement is in the omission of a factor of $1-\alpha$ from the denominator. However, for the bounds to be compared in a meaningful way, we must also relate the left-hand-side expressions. A relation can be based on the fact that for all μ,

$$\lim_{\alpha\uparrow 1}\|(1-\alpha)J_\mu - \lambda_\mu\|_\infty = 0,$$

as explained in §4. In particular, based on this we have

$$\begin{aligned}
\lim_{\alpha\uparrow 1}(1-\alpha)\|J_\mu - J^*\|_\infty &= |\lambda_\mu - \lambda^*|\\
&= \lambda_\mu - \lambda^*\\
&= \lim_{\alpha\uparrow 1}\pi^T(J_\mu - J^*),
\end{aligned}$$

[1] By an *invariant state distribution* of a transition matrix P, we mean any probability distribution π such that $\pi^T P = \pi^T$. In the event that $P_{\mu_{\tilde{r}}}$ has multiple invariant distributions, $\pi_{\tilde{r}}$ denotes an arbitrary choice.

for all policies μ and probability distributions π. Hence, the left-hand-side expressions from the two performance bounds become directly comparable as α approaches 1.

Another interesting comparison can be made by contrasting Corollary 4.1 against the following immediate consequence of Theorem 5.1.

COROLLARY 5.1. *For all MDP parameters* $(\mathcal{S}, \mathcal{U}, g, p)$ *and partitions, if* $\Phi\tilde{r} = \Pi_{\pi_{\tilde{r}}} T\Phi\tilde{r}$ *and* $\liminf_{\alpha\uparrow 1} \sum_{x\in\mathcal{S}_k} \pi_{\tilde{r}}(x) > 0$ *for all* k,

$$\limsup_{\alpha\uparrow 1} \|\lambda_{\mu_{\tilde{r}}} - \lambda_{\mu^*}\|_\infty \le 2 \lim_{\alpha\uparrow 1} \min_{r\in\Re^K} \|J^* - \Phi r\|_\infty.$$

The comparison suggests that solving $\Phi\tilde{r} = \Pi_{\pi_{\tilde{r}}} T\Phi\tilde{r}$ is strongly preferable to solving $\Phi\tilde{r} = \Pi_{\pi} T\Phi\tilde{r}$ with $\pi = \mathbf{1}$.

6. A simple example. We will present a simple example that serves two purposes. First, it provides a proof of Theorems 4.2 and 4.3. Second, it offers a concrete illustration of benefits afforded by use of the invariant distribution. The example we present builds on one used in Tsitsiklis and Van Roy [46].

Consider an MDP with states $\mathcal{S} = \{1, 2, \ldots, 2n\}$ for some positive integer n. With the exception of state 2, for which $\mathcal{U}_2 = \{1, 2\}$, $\mathcal{U}_x = \{1\}$ for $x \ne 2$. Let the transition probabilities be

$$\begin{aligned}
&p_{11}(1) = (1-\epsilon_1)(1-\epsilon_2) + \epsilon_1/2n,\\
&p_{12}(1) = (1-\epsilon_1)\epsilon_2 + \epsilon_1/2n,\\
&p_{1x}(1) = \epsilon_1/2n \quad \text{for } x \ne 1,\\
&p_{21}(1) = 1 - \epsilon_1 + \epsilon_1/2n,\\
&p_{22}(2) = 1 - \epsilon_1 + \epsilon_1/2n,\\
&p_{2x}(2) = \epsilon_1/2n \quad \text{for } x \ne 2,\\
&p_{x1}(1) = 1 - \epsilon_1 + \epsilon_1/2n \quad \text{for } x \ne 1, \text{odd},\\
&p_{xy}(1) = \epsilon_1/2n \quad \text{for } x \ne 1, \text{odd}, y \ne 1,\\
&p_{x2}(1) = 1 - \epsilon_1 + \epsilon_1/2n \quad \text{for } x \ne 2, \text{even},\\
&p_{xy}(1) = \epsilon_1/2n \quad \text{for } x \ne 2, \text{even}, y \ne 2,
\end{aligned}$$

for some scalars $\epsilon_1, \epsilon_2 \in (0, 1)$. Note that ϵ_1 can be thought of as a "reset probability;" upon reset, the process draws its next state from a uniform distribution. A positive reset probability ensures that the MDP is communicating. From state 1, if there is no reset, the system transitions to state 2 with probability ϵ_2 and otherwise remains in state 1.

Let the per-period costs be

$$\begin{aligned}
&g_1(1) = 0,\\
&g_1(2) = 0,\\
&g_2(2) = \kappa,\\
&g_1(x) = 2\Delta \quad \text{for } x \ne 1, \text{odd},\\
&g_1(x) = -2\Delta \quad \text{for } x \ne 2, \text{even},
\end{aligned}$$

for some scalars $\kappa \ge 0$ and $\Delta \ge 0$. Note that each instance of this MDP is identified by a quintuple $(n, \epsilon_1, \epsilon_2, \Delta, \kappa)$.

It is easy to see that the policy μ^* which selects action 1 at state 2 is optimal and

$$J^*(x) = \begin{cases} 0 & \text{if } x \in \{1, 2\},\\ 2\Delta & \text{if } x \in \{3, 5, 7, \ldots, 2n-1\},\\ -2\Delta & \text{if } x \in \{4, 6, 8, \ldots, 2n\}. \end{cases}$$

The only other policy $\mu^\dagger$ selects action 2 at state 2.

Partition $\mathcal{S}$ into two subsets: $\mathcal{S}_1$ contains the odd indices and $\mathcal{S}_2$ the even ones. Parameters $\tilde{r}_1$ and $\tilde{r}_2$ will be used to approximate cost-to-go values in $\mathcal{S}_1$ and $\mathcal{S}_2$, respectively, with $\Phi\tilde{r}$ denoting the approximation. Note that

$$\mu_{\tilde{r}} = \begin{cases} \mu^* & \text{if } \alpha\tilde{r}_1 < \kappa/(1-\epsilon_1) + \alpha\tilde{r}_2,\\ \mu^\dagger & \text{if } \alpha\tilde{r}_1 > \kappa/(1-\epsilon_1) + \alpha\tilde{r}_2. \end{cases}$$

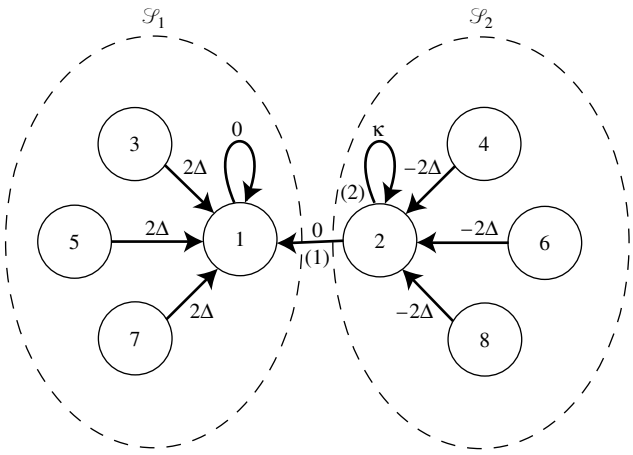

FIGURE 1. Illustration of the MDP and partitions for the case of $n = 8$ and $\epsilon_1 = \epsilon_2 = 0$. Numbers in parentheses represent actions.

Further,

$$\min_r \|J^* - \Phi r\|_\infty = \Delta.$$

Figure 1 offers an illustration of the MDP and partitions for the case of $n = 8$ and $\epsilon_1 = \epsilon_2 = 0$. In this case, all transitions are deterministic. A diagram with $\epsilon_1 > 0$ and $\epsilon_2 > 0$ would be too cluttered.

If $\pi = \mathbf{1}$, the equation $\Pi_\pi T \Phi \tilde{r} = \Phi \tilde{r}$ can be rewritten as

$$\tilde{r}_1 = \frac{n-1}{n} 2\Delta + (1-\epsilon_1)\left(\frac{n-1}{n}\alpha\tilde{r}_1 + \frac{1}{n}((1-\epsilon_2)\alpha\tilde{r}_1 + \epsilon_2\alpha\tilde{r}_2)\right) + \epsilon_1\alpha\frac{1}{2}(\tilde{r}_1 + \tilde{r}_2),$$

$$\tilde{r}_2 = \frac{n-1}{n}(-2\Delta) + (1-\epsilon_1)\left(\frac{n-1}{n}\alpha\tilde{r}_2 + \frac{1}{n}\min(\alpha\tilde{r}_1, \kappa/(1-\epsilon_1) + \alpha\tilde{r}_2)\right) + \epsilon_1\alpha\frac{1}{2}(\tilde{r}_1 + \tilde{r}_2).$$

Some algebra shows that for any $\kappa < 4\alpha\Delta/(1-\alpha)$ and $\epsilon_2 > 0$, for sufficiently small $\epsilon_1 > 0$ and sufficiently large n, $\alpha\tilde{r}_1 > \kappa/(1-\epsilon_1) + \alpha\tilde{r}_2$ and therefore $\mu_{\tilde{r}} = \mu^\dagger$. Theorem 4.2 follows. Fixing $\kappa = L$, Theorem 4.3 also follows.

Now consider solving $\Pi_{\pi_{\tilde{r}}} T \Phi \tilde{r} = \Phi \tilde{r}$. Fix n, $\Delta \geq 0$, and $\kappa > 0$. Under either policy (μ^* or $\mu^\dagger$), as ϵ_1 and ϵ_2 approach 0 the probability of being in state 1 conditioned on being in $\mathcal{S}_1$ and that of being in state 2 conditioned on being in $\mathcal{S}_2$ converge to 1. Hence, $\tilde{r}$ approaches the solution to

$$\tilde{r}_1 = \alpha r_1,$$

$$\tilde{r}_2 = \min(\alpha\tilde{r}_1, \kappa + \alpha\tilde{r}_2),$$

which is $\tilde{r} = 0$. It follows that for sufficiently small ϵ_1 and ϵ_2, $\mu_{\tilde{r}} = \mu^*$.

7. Exploration. If a vector $\tilde{r}$ solves $\Phi\tilde{r} = \Pi_{\pi_{\tilde{r}}} T \Phi \tilde{r}$ and the support of $\pi_{\tilde{r}}$ intersects every partition, Theorem 5.1 promises a desirable bound. However, there are two significant shortcomings to this solution concept which we will address in this section. First, in some cases, the equation $\Pi_{\pi_{\tilde{r}}} T \Phi \tilde{r} = \Phi \tilde{r}$ does not have a solution. It is easy to produce examples of this; though no example has been documented for the particular class of approximators we are using here, Bertsekas and Tsitsiklis [9] offer an example involving a different linearly parameterized approximator that captures the spirit of what can happen. Second, it would be nice to relax the requirement that the support of $\pi_{\tilde{r}}$ intersect every partition.

To address these shortcomings, we introduce a form of exploration. Exploration involves randomizing decisions so that at each state, each action is selected at least some of the time. Exploration has been a central issue in the reinforcement learning literature (see Sutton and Barto [44] for discussion).

To introduce exploration, we need to consider stochastic policies. A stochastic policy μ maps state-action pairs to probabilities. For each $x \in \mathcal{S}$ and $u \in \mathcal{U}_x$, $\mu(x,u)$ is the probability of taking action u when in state x. Hence, $\mu(x,u) \geq 0$ for all $x \in \mathcal{S}$ and $u \in \mathcal{U}_x$, and $\sum_{u \in \mathcal{U}_x} \mu(x,u) = 1$ for all $x \in \mathcal{S}$.

Given a scalar $\epsilon > 0$ and a function J, the ϵ-greedy Boltzmann exploration policy with respect to J is defined by

$$\mu(x,u) = \frac{e^{-(g_u(x) + \alpha\sum_{y\in\mathcal{S}} p_{xy}(u)J(y))(|\mathcal{U}_x|-1)/\epsilon e}}{\sum_{u\in\mathcal{U}_x} e^{-(g_u(x) + \alpha\sum_{y\in\mathcal{S}} p_{xy}(u)J(y))(|\mathcal{U}_x|-1)/\epsilon e}}.$$

For any $\epsilon > 0$ and r, let μ_r^ϵ denote the ϵ-greedy Boltzmann exploration policy with respect to Φr. Further, we define a modified dynamic programming operator that incorporates Boltzmann exploration:

$$(T^\epsilon J)(x) = \frac{\sum_{u\in\mathcal{U}_x} e^{-(g_u(x)+\alpha\sum_{y\in\mathcal{S}} p_{xy}(u)J(y))(|\mathcal{U}_x|-1)/\epsilon e}(g_u(x)+\alpha\sum_{y\in\mathcal{S}} p_{xy}(u)J(y))}{\sum_{u\in\mathcal{U}_x} e^{-(g_u(x)+\alpha\sum_{y\in\mathcal{S}} p_{xy}(u)J(y))(|\mathcal{U}_x|-1)/\epsilon e}}.$$

As ϵ approaches 0, ϵ-greedy Boltzmann exploration policies become greedy and the modified dynamic programming operators become the dynamic programming operator. More precisely, for all r, x, and J, $\lim_{\epsilon\downarrow 0}\mu_r^\epsilon(x,\mu_r(x)) = 1$ and $\lim_{\epsilon\downarrow 0} T^\epsilon J = TJ$. These are immediate consequences of the following result (see de Farias and Van Roy [20] for a proof).

LEMMA 7.1. *For any n, $v \in \Re^n$,*

$$\min_i v_i + \epsilon \geq \frac{\sum_i e^{-v_i(n-1)/\epsilon e} v_i}{\sum_i e^{-v_i(n-1)/\epsilon e}} \geq \min_i v_i.$$

Because we are only concerned with communicating MDPs, there is a unique invariant state distribution associated with each ϵ-greedy Boltzmann exploration policy μ_r^ϵ and the support of this distribution is $\mathcal{S}$. Let π_r^ϵ denote this distribution. We consider a vector $\tilde{r}$ that solves $\Phi\tilde{r} = \Pi_{\pi_{\tilde{r}}^\epsilon} T^\epsilon \Phi\tilde{r}$. For any $\epsilon > 0$, there exists a solution to this equation (this is an immediate extension of Theorem 5.1 from de Farias and Van Roy [20]).

We have the following performance loss bound which parallels Theorem 5.1 but with an equation for which a solution is guaranteed to exist and without any requirement on the resulting invariant distribution.

THEOREM 7.1. *For any MDP, partition, and $\epsilon > 0$, if $\Phi\tilde{r} = \Pi_{\pi_{\tilde{r}}^\epsilon} T^\epsilon \Phi\tilde{r}$, then*

$$(1-\alpha)(\pi_{\tilde{r}}^\epsilon)^T\left(J_{\mu_{\tilde{r}}^\epsilon} - J^*\right) \leq 2\alpha \min_{r\in\Re^K} \|J^* - \Phi r\|_\infty + \epsilon.$$

PROOF. Note that for any $\pi \in \Re_+^{|\mathcal{S}|}$, Π_π is nonexpansive with respect to the maximum norm and for any J,

$$\|J - \Pi_\pi J\|_\infty \leq 2\min_r \|J - \Phi r\|_\infty.$$

Using these properties and Lemma 7.1, we establish an error bound:

$$\begin{aligned}
\|\Phi\tilde{r} - J^*\|_\infty &= \|\Pi_{\pi_{\tilde{r}}^\epsilon} T^\epsilon \Phi\tilde{r} - J^*\|_\infty \\
&\leq \|\Pi_{\pi_{\tilde{r}}^\epsilon} T^\epsilon \Phi\tilde{r} - \Pi_{\pi_{\tilde{r}}^\epsilon} J^*\|_\infty + \|J^* - \Pi_{\pi_{\tilde{r}}^\epsilon} J^*\|_\infty \\
&\leq \|T^\epsilon \Phi\tilde{r} - J^*\|_\infty + 2\min_r \|J^* - \Phi r\|_\infty \\
&\leq \|T\Phi\tilde{r} - J^*\|_\infty + \epsilon + 2\min_r \|J^* - \Phi r\|_\infty \\
&\leq \alpha\|\Phi\tilde{r} - J^*\|_\infty + \epsilon + 2\min_r \|J^* - \Phi r\|_\infty,
\end{aligned}$$

and it follows that

$$\|\Phi\tilde{r} - J^*\|_\infty \leq \frac{2}{1-\alpha}\min_r \|J^* - \Phi r\|_\infty + \frac{\epsilon}{1-\alpha}.$$

An argument entirely analogous to one used in the proof of Theorem 5.1 establishes that

$$(\pi_{\tilde{r}}^\epsilon)^T T^\epsilon \Phi\tilde{r} = (\pi_{\tilde{r}}^\epsilon)^T \Phi\tilde{r} = (\pi_{\tilde{r}}^\epsilon)^T J_{\mu_{\tilde{r}}^\epsilon}.$$

Using this relation, we obtain

$$\begin{aligned}
(\pi_{\tilde{r}}^\epsilon)^T\left(J_{\mu_{\tilde{r}}^\epsilon} - J^*\right) &= (\pi_{\tilde{r}}^\epsilon)^T(T^\epsilon\Phi\tilde{r} - J^*) \\
&\leq \|T^\epsilon\Phi\tilde{r} - J^*\|_\infty \\
&\leq \|T\Phi\tilde{r} - J^*\|_\infty + \epsilon \\
&\leq \alpha\|\Phi\tilde{r} - J^*\|_\infty + \epsilon \\
&\leq \frac{2\alpha}{1-\alpha}\min_{r\in\Re^K}\|J^* - \Phi r\|_\infty + \frac{\alpha\epsilon}{1-\alpha} + \epsilon \\
&= \frac{2\alpha}{1-\alpha}\min_{r\in\Re^K}\|J^* - \Phi r\|_\infty + \frac{\epsilon}{1-\alpha}. \quad \square
\end{aligned}$$

8. Computation. Though computation is not a focus of this paper, we offer a brief discussion here. First, we describe a simple algorithm from Tsitsiklis and Van Roy [46] which draws on ideas from temporal-difference learning (Sutton [40, 41]) and Q-learning (Watkins [53], Watkins and Dayan [54]) to solve $\Phi\tilde{r} = \Pi_\pi T\Phi\tilde{r}$ (related methods for deterministic control problems are also proposed in Werbos [55]). It requires an ability to sample a sequence of states $x^{(0)}, x^{(1)}, x^{(2)}, \ldots$, each independent and identically distributed according to π. Also required is a way to efficiently compute

$$(T\Phi r)(x) = \min_{u\in\mathcal{U}_x}\left(g_u(x) + \alpha\sum_{y\in\mathcal{S}} p_{xy}(u)(\Phi r)(y)\right),$$

for any given x and r. This is typically possible when the action set $\mathcal{U}_x$ and the support of $p_{x\cdot}(u)$ (i.e., the set of states that can follow x if action u is selected) are not too large. The algorithm generates a sequence of vectors $r^{(l)}$ according to

$$r^{(l+1)} = r^{(l)} + \gamma_l\phi(x^{(l)})((T\Phi r^{(l)})(x^{(l)}) - (\Phi r^{(l)})(x^{(l)})),$$

where γ_l is a step size and $\phi(x)$ denotes the column vector made up of components from the xth row of Φ. In Tsitsiklis and Van Roy [46], using results from Tsitsiklis [45] and Jaakkola et al. [28], it is shown that under appropriate assumptions on the step size sequence, $r^{(l)}$ converges to a vector $\tilde{r}$ that solves $\Phi\tilde{r} = \Pi_\pi T\Phi\tilde{r}$.

We now move on to a version of temporal-difference learning that aims at solving $\Phi\tilde{r} = \Pi_{\pi^\epsilon_{\tilde{r}}} T^\epsilon\Phi\tilde{r}$. The algorithm requires simulation of a trajectory $x_0, x_1, x_2, \ldots$ of the MDP, with each action $u_t \in \mathcal{U}_{x_t}$ generated by the ϵ-greedy Boltzmann exploration policy with respect to $\Phi r^{(t)}$. The sequence of vectors $r^{(t)}$ is generated according to

$$r^{(t+1)} = r^{(t)} + \gamma_t\phi(x_t)((T^\epsilon\Phi r^{(t)})(x_t) - (\Phi r^{(t)})(x_t)).$$

Under suitable conditions on the step size sequence, if this algorithm converges, the limit satisfies $\Phi\tilde{r} = \Pi_{\pi^\epsilon_{\tilde{r}}} T^\epsilon\Phi\tilde{r}$. Whether such an algorithm converges and whether there are other algorithms that can effectively solve $\Phi\tilde{r} = \Pi_{\pi^\epsilon_{\tilde{r}}} T^\epsilon\Phi\tilde{r}$ for broad classes of relevant problems remain open issues.

There is a lot more to be said about algorithms of the sort we have described. We close this section briefly mentioning just two noteworthy issues. First, these algorithms serve as simple cases to study. There is a body of literature on variations that may converge more quickly (Werbos [56], Bradtke and Barto [15], Bertsekas and Ioffe [7], Boyan [13, 14], Choi and Van Roy [16, 17], Nedic and Bertsekas [35], Bertsekas et al. [10]). Second, when the support of $p_{x\cdot}(u)$ is too large, computation of $(T\Phi r)(x)$ becomes intractable. In this case, one may resort to variations of Q-learning (Watkins [53], Watkins and Dayan [54]), which approximate a cost-to-go function over state-action pairs rather than over states. This approach essentially uses a Monte-Carlo method to compute the expectation $\sum_{y\in S} p_{xy}(u)(\Phi r)(y)$ (see Bertsekas and Tsitsiklis [9], Sutton and Barto [44] for discussion). The variation of Q-learning that is closest in spirit to the second update procedure we described is called SARSA (Rummery and Niranjan [37], Rummery [36], Sutton and Barto [44]).

9. Extensions and open issues. The temporal-difference learning algorithm presented in §8 is a version of TD(0). This is a special case of TD(λ), which is parameterized by $\lambda \in [0, 1]$. It is not known whether the results of this paper can be extended to the general case of $\lambda \in [0, 1]$. Prior research has suggested that larger values of λ lead to superior results. In particular, an example of Bertsekas [5] and the approximation error bounds of Tsitsiklis and Van Roy [47, 48] and Van Roy [52], both of which are restricted to the case of a fixed policy, suggest that approximation error is amplified by a factor of $1/(1-\alpha)$ as λ is changed from 1 to 0. The results of §§5 and 7 suggest that this factor vanishes if one considers a controlled process and performance loss rather than approximation error.

Whether the results of this paper can be extended to accommodate approximate value iteration with general linearly parameterized approximators remains an open issue. In this broader context, error and performance loss bounds of the kind offered by Theorem 4.1 are unavailable, even when the invariant distribution is used to weight the projection. Such error and performance bounds *are* available, on the other hand, for the solution to a certain linear program (de Farias and Van Roy [21, 22]). Whether a factor of $1/(1-\alpha)$ can similarly be eliminated from *these* bounds is an open issue.

We have considered state aggregation based on a prespecified partition. An interesting direction for research concerns how to automate partitioning based on problem data. To this end, adaptive aggregation schemes have been proposed and studied in Bertsekas and Castañon [6] and Moore and Atkeson [33].

Natural extensions to the results of §§4, 5, and 7 can be established for the general-state-space (e.g., continuous-state) bounded-cost model of Bertsekas and Shreve [8]. One notable difference, though, is that solving $\Phi\tilde{r} = \Pi_{\pi_{\tilde{r}}} T\Phi\tilde{r}$ or $\Phi\tilde{r} = \Pi_{\pi_{\tilde{r}}} T^{\epsilon}\Phi\tilde{r}$ requires $\mu_{\tilde{r}}$ to yield an invariant distribution. For a given MDP and partition, there may be no such solution.

There is a literature on discretization methods for MDPs with continuous state-spaces (Fox [23], Bertsekas [4], Whitt [57], Hinderer [27], Chow and Tsitsiklis [18, 19], Kushner and Dupuis [30], Rust [38]). Discretization is closely related to state-aggregation; states are essentially partitioned into subsets, each of which is represented by a point in a grid. An important issue is to understand the granularity required to limit performance loss to some desired level ϵ for a given class of continuous-state MDPs. An interesting research question is whether the ideas in this paper can be used to reduce granularity requirements.

Our results can be extended to accommodate an average cost objective, assuming that the MDP is communicating. With Boltzmann exploration, the equation of interest becomes

$$\Phi\tilde{r} = \Pi_{\pi^{\epsilon}_{\tilde{r}}}(T^{\epsilon}\Phi\tilde{r} - \tilde{\lambda}\mathbf{1}).$$

The variables include an estimate $\tilde{\lambda} \in \Re$ of the minimal average cost $\lambda^* \in \Re$ and an approximation $\Phi\tilde{r}$ of the optimal differential cost function h^*. The discount factor α is set to 1 in computing an ϵ-greedy Boltzmann exploration policy as well as T^{ϵ}. There is an average-cost version of temporal-difference learning for which any limit of convergence $(\tilde{\lambda}, \tilde{r})$ satisfies this equation (Tsitsiklis and Van Roy [49]–[51]). Generalization of Theorem 4.1 does not lead to a useful result because the right-hand side of the bound becomes infinite as α approaches 1. On the other hand, generalization of Theorem 7.1 yields the first performance loss bound for approximate value iteration with an average-cost objective:

THEOREM 9.1. *For any MDP with an average-cost objective, partition, and $\epsilon > 0$, if $\Phi\tilde{r} = \Pi_{\pi^{\epsilon}_{\tilde{r}}}(T^{\epsilon}\Phi\tilde{r} - \tilde{\lambda}\mathbf{1})$, then*

$$\lambda_{\mu^{\epsilon}_{\tilde{r}}} - \lambda^* \leq 2 \min_{r \in \Re^K} \|h^* - \Phi r\|_{\infty} + \epsilon.$$

Here, $\lambda_{\mu^{\epsilon}_{\tilde{r}}} \in \Re$ denotes the average cost under policy $\mu^{\epsilon}_{\tilde{r}}$, which is well defined because the process is irreducible under an ϵ-greedy Boltzmann exploration policy. This theorem can be proved by taking limits on the left- and right-hand sides of the bound of Theorem 7.1. It is easy to see that the limit of the left-hand side is $\lambda_{\mu^{\epsilon}_{\tilde{r}}} - \lambda^*$. The limit of $\min_{r \in \Re^K} \|J^* - \Phi r\|_{\infty}$ on the right-hand side is $\min_{r \in \Re^K} \|h^* - \Phi r\|_{\infty}$. (This follows from the analysis of Blackwell [12].)

Acknowledgments. This material is based upon work supported by the National Science Foundation under Grant ECS-9985229 and by the Office of Naval Research under Grant MURI N00014-00-1-0637. The author's understanding of the topic benefited from collaborations with Dimitri Bertsekas, Daniela de Farias, and John Tsitsiklis. A number of useful comments and suggestions from anonymous reviewers helped to improve the paper.

References

[1] Axsäter, S. 1983. State aggregation in dynamic programming: An application to scheduling of independent jobs on parallel processors. *Oper. Res. Lett.* **2** 171–176.
[2] Barraquand, J., D. Martineau. 1997. Numerical valuation of high dimensional multivariate American securities. *J. Financial Quant. Anal.* **30**(3) 383–405.
[3] Bean, J. C., J. R. Birge, R. L. Smith. 1987. Aggregation in dynamic programming. *Oper. Res.* **35** 215–220.
[4] Bertsekas, D. P. 1975. Convergence of discretization procedures in dynamic programming. *IEEE Trans. Automat. Control* **20** 415–419.
[5] Bertsekas, D. P. 1994. A counterexample to temporal-difference learning. *Neural Comput.* **7** 270–279.
[6] Bertsekas, D. P., D. A. Castañon. 1989. Adaptive aggregation for infinite horizon dynamic programming. *IEEE Trans. Automat. Control* **34**(6) 589–598.
[7] Bertsekas, D. P., S. Ioffe. 1996. Temporal differences-based policy iteration and applications in neuro-dynamic programming. Technical report LIDS–P–2349, MIT Laboratory for Information and Decision Systems, Cambridge, MA.
[8] Bertsekas, D. P., S. E. Shreve. 1996. *Stochastic Optimal Control: The Discrete Time Case*. Athena Scientific, Belmont, MA.
[9] Bertsekas, D. P., J. N. Tsitsiklis. 1996. *Neuro-Dynamic Programming*. Athena Scientific, Belmont, MA.
[10] Bertsekas, D. P., V. Borkar, A. Nedic. 2004. Improved temporal difference methods with linear function approximation. J. Si, A. G. Barto, W. B. Powell, D. C. Wunsch, II, eds. *Handbook of Learning and Approximate Dynamic Programming*. IEEE Press and John Wiley & Sons, Boston, MA.
[11] Birge, J. R. 1985. Aggregation bounds in stochastic linear programming. *Math. Programming* **31** 25–41.
[12] Blackwell, D. 1962. Discrete dynamic programming. *Ann. Math. Statist.* **33** 719–726.
[13] Boyan, J. A. 1999. Least-squares temporal difference learning. I. Bratko, S. Dzeroski, eds. *Machine Learning: Proc. 16th Internat. Conf.* (ICML), Cambridge, MA.
[14] Boyan, J. A. 2002. Technical update: Least-squares temporal difference learning. *Machine Learning* **49** 2–3.

[15] Bradtke, S. J., A. G. Barto. 1996. Linear least-squares algorithms for temporal-difference learning. *Machine Learning* 33–57.
[16] Choi, D. S., B. Van Roy. 2001. A generalized Kalman filter for fixed point approximation and efficient temporal-difference learning. *Machine Learning: Proc. 18th Internat. Conf.* (ICML), Palo Alto, CA.
[17] Choi, D. S., B. Van Roy. 2006. A generalized Kalman filter for fixed point approximation and efficient temporal-difference learning. *Discrete Event Dynamic Systems* **16**(2).
[18] Chow, C. S., J. N. Tsitsiklis. 1989. The complexity of dynamic programming. *J. Complexity* **5** 466–488.
[19] Chow, C. S., J. N. Tsitsiklis. 1991. An optimal one-way multigrid algorithm for discrete-time stochastic control. *IEEE Trans. Automat. Control* **36**(8) 898–914.
[20] de Farias, D. P., B. Van Roy. 2000. On the existence of fixed points for approximate value iteration and temporal-difference learning. *J. Optim. Theory Appl.* **105**(3) 589–608.
[21] de Farias, D. P., B. Van Roy. 2002. Approximate dynamic programming via linear programming. *Advances in Neural Information Processing Systems*, Vol. 14. MIT Press, Cambridge, MA.
[22] de Farias, D. P., B. Van Roy. 2003. The linear programming approach to approximate dynamic programming. *Oper. Res.* **51**(6) 850–865.
[23] Fox, B. L. 1973. Discretizing dynamic programs. *J. Optim. Theory Appl.* **11** 228–234.
[24] Gordon, G. J. 1995. Stable function approximation in dynamic programming. Technical report CMU-CS-95-103, Carnegie Mellon University, Pittsburgh, PA.
[25] Gordon, G. J. 1995. Stable function approximation in dynamic programming. *Machine Learning: Proc. 12th Internat. Conf.* (ICML), San Francisco, CA.
[26] Gordon, G. J. 1999. Approximate solutions to Markov decision processes. Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA.
[27] Hinderer, K. 1979. On approximate solutions of finite-stage dynamic programs. M. Puterman, ed. *Dynamic Programming and Its Applications*. Academic Press, New York.
[28] Jaakkola, T., M. I. Jordan, S. P. Singh. 1994. On the convergence of stochastic iterative dynamic programming algorithms. *Neural Comput.* **6** 1185–1201.
[29] Jaakkola, T., S. P. Singh, M. I. Jordan. 1995. Reinforcement learning algorithms for partially observable Markov decision problems. *Advances in Neural Information Processing Systems*, Vol. 7, 345–352.
[30] Kushner, H. J., P. G. Dupuis. 1992. *Numerical Methods for Stochastic Control Problems in Continuous Time*. Springer-Verlag, New York.
[31] Lambert, III, T., M. A. Epelman, R. L. Smith. 2004. Aggregation in stochastic dynamic programming. Technical report 04-07, Department of Industrial and Operations Engineering, Ann Arbor, MI.
[32] Mendelssohn, R. 1982. An iterative aggregation procedure for Markov decision processes. *Oper. Res.* **30** 62–73.
[33] Moore, Andrew, Chris Atkeson. 1995. The parti-game algorithm for variable resolution reinforcement learning in multidimensional state-spaces. *Machine Learning* **21**(3) 199–233.
[34] Morin, T. 1979. Computational advances in dynamic programming. M. Puterman, ed. *Dynamic Programming and Its Applications.* Academic Press, New York, 202–207.
[35] Nedic, A., D. P. Bertsekas. 2003. Least-squares policy evaluation algorithms with linear function approximation. *Discrete Event Dynam. Systems* **13** 79–110.
[36] Rummery, G. A. 1995. Problem solving with reinforcement learning. Ph.D. thesis, Cambridge University, Cambridge, UK.
[37] Rummery, G. A., M. Niranjan. 1994. On-line Q-learning using connectionist systems. Technical report CUED/F-INFENG/TR 166, Engineering Department, Cambridge University.
[38] Rust, J. 1997. Using randomization to break the curse of dimensionality. *Econometrica* **65**(3) 487–516.
[39] Singh, S. P., R. C. Yee. 1994. An upper-bound on the loss from approximate optimal-value functions. *Machine Learning* **16**(3) 227–233.
[40] Sutton, R. S. 1984. Temporal credit assignment in reinforcement learning. Ph.D. thesis, University of Massachusetts Amherst, Amherst, MA.
[41] Sutton, R. S. 1988. Learning to predict by the methods of temporal differences. *Machine Learning* **3** 9–44.
[42] Sutton, R. S. 1995. On the virtues of linear learning and trajectory distributions. *Proc. Workshop Value Function Approximation, Machine Learning Conf.*
[43] Sutton, R. S. 1996. Generalization in reinforcement learning: Successful examples using sparse coarse coding. *Advances in Neural Information Processing Systems*, Vol. 8. MIT Press, Cambridge, MA.
[44] Sutton, R. S., A. G. Barto. 1998. *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, MA.
[45] Tsitsiklis, J. N. 1994. Asynchronous stochastic approximation and Q-learning. *Machine Learning* **16** 185–202.
[46] Tsitsiklis, J. N., B. Van Roy. 1996. Feature-based methods for large scale dynamic programming. *Machine Learning* **22** 59–94.
[47] Tsitsiklis, J. N., B. Van Roy. 1997. An analysis of temporal-difference learning with function approximation. *IEEE Trans. Automat. Control* **42**(5) 674–690.
[48] Tsitsiklis, J. N., B. Van Roy. 1997. Analysis of temporal-difference learning with function approximation. *Advances in Neural Information Processing Systems*, Vol. 9. MIT Press, Cambridge, MA.
[49] Tsitsiklis, J. N., B. Van Roy. 1997. Average cost temporal-difference learning. *Proc. IEEE Conf. Decision Control*, IEEE, San Diego, CA.
[50] Tsitsiklis, J. N., B. Van Roy. 1999. Average cost temporal-difference learning. *Automatica* **35**(11) 1799–1808.
[51] Tsitsiklis, J. N., B. Van Roy. 2002. On average versus discounted reward temporal-difference learning. *Machine Learning* **49**(2–3) 179–191.
[52] Van Roy, B. 1998. Learning and value function approximation in complex decision processes. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA.
[53] Watkins, C. J. C. H. 1989. Learning from delayed rewards. Ph.D. thesis, Cambridge University, Cambridge, UK.
[54] Watkins, C. J. C. H., P. Dayan. 1992. Q-learning. *Machine Learning* **8** 279–292.
[55] Werbos, P. J. 1977. Advanced forecasting methods for global crisis warning and models of intelligence. *General Systems Yearbook* **22** 25–38.
[56] Werbos, P. J. 1990. Consistency of HDP applied to a simple reinforcement learning problem. *Neural Networks* **3** 179–189.
[57] Whitt, W. 1978. Approximations of dynamic programs I. *Math. Oper. Res.* **3**(3) 231–243.