Analysis of Some Incremental Variants of Policy Iteration: First Steps Toward Understanding Actor-Critic Learning Systems*

Ronald J. Williams
College of Computer Science
Northeastern University
Boston, MA 02115
rjw@ccs.neu.edu

and

Leemon C. Baird, III
Wright Laboratory
Wright-Patterson Air Force Base, OH 45433-6543
bairdlc@wL.wpafb.af.mil

Northeastern University
College of Computer Science
Technical Report NU-CCS-93-11

September 8, 1993

*We gratefully acknowledge the substantial contributions to this effort provided by Andy Barto, who sparked our original interest in these questions and whose continued encouragement and insightful comments and criticisms have helped us greatly. Recent discussions with Satinder Singh and Vijay Gullapalli have also had a helpful impact on this work. Special thanks also to Rich Sutton, who has influenced our thinking on this subject in numerous ways. This work was supported by Grant IRI-8921275 from the National Science Foundation and by the U. S. Air Force.

Abstract

This paper studies algorithms based on an incremental dynamic programming abstraction of one of the key issues in understanding the behavior of actor-critic learning systems. The prime example of such a learning system is the ASE/ACE architecture introduced by Barto, Sutton, and Anderson (1983). Also related are Witten's adaptive controller (1977) and Holland's bucket brigade algorithm (1986). The key feature of such a system is the presence of separate adaptive components for action selection and state evaluation, and the key issue focused on here is the extent to which their joint adaptation is guaranteed to lead to optimal behavior in the limit. In the incremental dynamic programming point of view taken here, these questions are formulated in terms of the use of separate data structures for the current best choice of policy and current best estimate of state values, with separate operations used to update each at individual states. Particular emphasis here is on the effect of complete asynchrony in the updating of these data structures across states. The main results are that, while convergence to optimal performance is not guaranteed in general, there are a number of situations in which such convergence is assured. Since the algorithms investigated represent a certain idealized abstraction of actor-critic learning systems, these results are not directly applicable to current versions of such learning systems but may be viewed instead as providing a useful first step toward more complete understanding of such systems. Another useful perspective on the algorithms analyzed here is that they represent a broad class of asynchronous dynamic programming procedures based on policy iteration.

Contents

1 Introduction

1.1 Overview

The purpose of this paper is to present some results in the theory of *incremental dynamic programming* (Watkins, 1989), which, in its general form, combines elements of the theory of dynamic programming with features appropriate for on-line learning in the absence of an *a priori* model of the environment. An excellent discussion of this approach is provided by Barto, Sutton, and Watkins (1989), who point out that it may be regarded as a direct method of adaptive control. Another investigator who has emphasized the potential role of dynamic programming-based strategies in biological and artificial learning systems is Werbos (1987), who has used the term *heuristic dynamic programming* for his approach.

The analysis presented here was inspired by the desire to gain an analytic understanding of the behavior of *actor-critic* learning systems, which consist of two jointly adaptive modules. One module, called the *actor*, implements the current policy but also learns to improve its choice of action in order to optimize the reward provided by the other module, which is called the *critic*. The reward provided by the critic is intended to represent an estimate of the expected long-term external reward, but the critic must also adapt in order both to improve the accuracy of its estimate given the current policy and to keep up with the ever-changing policy as the actor learns. The prime example of such a learning system is the ASE/ACE architecture used in the pole-balancing controller of Barto, Sutton, and Anderson (1983). Other algorithms that are related are Witten's (1977) adaptive controller and Holland's (1986) bucket brigade algorithm. Sutton (1984; 1988) appears to be one of the first to call attention to a common basis for these algorithms.

One important question that remains to be answered for such a learning system is the effect of asynchronous adaptation of the actor and critic modules at individual states (or state-action pairs, as appropriate). Here we consider a simplified abstraction designed to address this kind of issue in an idealized sense. In particular, we ask: What happens if the actor and critic are updated state by state in no particular order? Will such joint incremental adaptation always succeed at finding optimal policies, or can it sometimes fail? Ignored here are a number of important additional issues that also must eventually be addressed in order to gain a full understanding of existing actor-critic algorithms, including problems of dealing with unknown stochastic environments through on-line interaction. Here we simply assume that all transition probabilities and expected rewards can be determined exactly from interaction with the initially unknown environment. This includes as a special case the situation when all transitions and rewards are deterministic. Another simplifying assumption we make is to consider arbitrary sequences of application of certain one-state-at-a-time operators to actor-critic systems without regard for how those states come to be visited by the learning system. Alternatively, the analysis presented here can be viewed more as involving a model-based approach, as discussed by Barto, Bradtke, and Singh (1991). In fact, the closest learning algorithm in the literature to which the results to be obtained here potentially apply is the Dyna-PI algorithm of Sutton (1990; 1991).

While the investigation reported here was motivated by the desire to understand certain types of learning system, an entirely different point of view may be taken. In particular, the results obtained here may be viewed as providing alternative off-line algorithms for performing dynamic programming computations on parallel processing machines with asynchronous communication between processors, the main issue being the extent to which the final result obtained is independent of the precise order in which updating occurs among processors.[1]

Many of the results proved here first appeared in summary form in a short workshop paper (Williams & Baird, 1990).

1.2 Incremental Dynamic Programming

The theory of dynamic programming provides a general framework within which to define, and sometimes to solve, *sequential decision tasks*, in which an appropriate sequence of actions must be chosen in order to receive optimal cumulative rewards over time. In general, the dynamic programming approach involves the use of a state-space dynamical system representation together with a space of possible control actions to be taken at each state. The theory is intended to provide computational tools for helping to determine a closed-loop control law, or *policy*, specifying the particular choice of action to be taken as a function of system state in order to perform optimally over time. The cornerstone of the dynamic programming approach is the *optimality principle*, which essentially states that, for any optimal sequence of actions from a given initial state, the subsequence of actions obtained by deleting the first action must be optimal when starting from that next state. This is used to decompose the problem of determining an optimal sequence of decisions into a recursive form involving a one-step optimal decision problem together with another multi-step optimal decision problem. A important ingredient of this computational approach is the use of an *evaluation function*, or *value function*, assigning to each state the real number representing the value of the specified objective function when the system is started in that state and the optimal policy is used. In this paper we will refer to this particular mapping from states to real numbers as the *ideal evaluation function*, reserving the more general term for functions which may differ from the ideal. These more general evaluation functions may be viewed simply as estimates of the ideal, and they play a role similar to that of *static evaluation functions* in artificial intelligence game-tree search algorithms.

The standard dynamic programming formulation assumes the existence of a complete model of the system, including the effect of actions on system state. Combining elements of this theory with issues associated with on-line learning in the absence of an *a priori* system model has led to an approach Watkins (1989) has called *incremental dynamic programming.* As pointed out by Barto, Sutton, and Watkins (1989), this approach can be viewed as representing a *direct* method of adaptive control, in contrast to the more obvious *indirect* method of alternating the fitting of a model to the system in question with the use of a dynamic programming analysis of the most recent model. The use of incremental dynamic programming methods actually allow the creation of on-line learning systems that operate with reasonable efficiency and fall somewhere between these two extremes (Sutton, 1990; 1991; Peng & Williams, 1992; Moore & Atkeson, 1992).

The function of the two modules in an actor-critic learning system can be described quite simply from a dynamic programming point of view: The actor module implements the current policy and the critic module represents the current value function. In this paper we assume that table-lookup representations are used for both modules and we investigate the result of jointly

[1] However, we do not actually treat the completely asynchronous case here since the results actually obtained are based on the assumption that there are never two or more simultaneous updates, We conjecture that our results carry over to this more general asynchronous update case but we have not verified this.

updating these data structures using certain specific incremental algorithms naturally derived from conventional dynamic programming. In particular, we study several incremental approaches based on *policy iteration*.

1.3 Organization of The Remainder of This Paper

Sections 2 and 3 are intended to provide the conceptual foundation on which the rest of this paper rests. In particular, Section 2 contains a brief introduction to the mathematical formulation of Markov decision tasks, including discussion of value functions and the Bellman equation. Section 3 introduces a set of one-state-at-a-time and one-state-action-pair-at-a-time update operations out of which various existing dynamic programming algorithms can be constructed, and which can also be combined in potentially novel ways. In particular, it is observed that certain at least roughly systematic orderings of these operations correspond to asynchronous approaches to policy iteration and value iteration that are known to converge to solutions to the underlying Markov decision task in the limit. This section also introduces a number of additional definitions and proves several fundamental results to set the stage for the original material appearing in subsequent sections.

It is in Sections 4, 5, and 6, where the generally new results are to be found. The primary focus in Section 4 is on convergence and non-convergence results using the same operators introduced in Section 3, while Section 5 introduces some new operators and establishes some additional convergence results for these. Section 6 establishes still more convergence results, one using a combination of operators not considered in the earlier sections and another based on recent work of Singh and Gullapalli (1993).

Finally, in Section 7 the potential significance of these results is discussed and we observe what future directions might be useful for this theory.

2 Mathematical Preliminaries

Here we give the mathematical formulation of the type of problem addressed in this paper and introduce appropriate definitions and notational conventions. We also summarize some standard computational strategies from the theory of dynamic programming and describe additional elementary results relevant to the incremental point of view adopted here.

2.1 Markov Environment

We take as given a *Markov environment*, or *controlled Markov chain*, having a finite set of states X and a finite set of actions A, which, for simplicity, is assumed to be the same for all states. For any finite set S we denote its cardinality by $|S|$. We let $f(x, a)$ denote the randomly chosen successor of state x when action a is chosen. The behavior of this random next-state function is determined by the transition probabilities $p_{xy}^a = Pr\{f(x,a) = y\}$ for $x, y \in X$ and $a \in A$. We also assume that associated with each choice of action a at each state x is a randomly determined immediate reward $r(x, a)$, with $R(x, a) = E\{r(x, a)\}$ denoting its expected value.

In general, a non-randomized policy is a function π assigning to each possible history of states and actions a choice of action to be used at the current time. Of special importance here are *stationary* policies, which select actions according to the current state only. Thus the set of stationary policies can be identified with $A^X = \{\pi : X \to A\}$. Similarly, the set of value functions is $\mathcal{R}^X = \{v : X \to \mathcal{R}\}$, where $\mathcal{R}$ denotes the real numbers.

2.2 Total Expected Discounted Return

The results presented here are all based on the use of a discounting parameter $\gamma \in (0, 1)$, which is used to define a measure of performance on policies as follows. For any policy π, define $v^\pi \in \mathcal{R}^X$, the *total expected discounted return* (or just the *return*) for π, to be that value function assigning to state x the quantity

$$v^\pi(x) = E\left\{\sum_{t=0}^{\infty} \gamma^t r(x(t), a(t)) \mid x_0 = x\right\}$$

where $x(0), x(1), x(2), \ldots$ is a random sequence of states determined by $x(0) = x$ and $x(t+1) = f(x(t), a(t))$ for each $t \geq 0$, and $a(0), a(1), a(2), \ldots$ is the corresponding sequence of actions, each selected according the specified policy π. When π is a stationary policy this simply means that $a(k) = \pi(x(k))$, but this definition also applies to *nonstationary policies*, where $a(t)$ may depend on additional information retained over the history of operation of the controller, such as the past states $x(0), x(1), \ldots, x(t)$ visited, the past actions $a(0), a(1), \ldots, a(t-1)$ taken, and the current time index t. In general, this dependence may even be stochastic, in which case one has a *randomized policy*. The factor γ determines the relative value of short-term and long-term rewards when combining rewards over long time periods.

2.3 Ideal Evaluation Function and Optimal Policies

Define a partial order relation on value functions by $v \leq v'$ if and only if $v(x) \leq v'(x)$ for all $x \in X$. An *optimal* policy is one for which the return is maximal at each state. With v^* denoting the *ideal evaluation function*, or the return from any optimal policy, it follows that $v^\pi \leq v^*$ for any policy π (including nonstationary or randomized policies). Clearly v^* is unique if there are any optimal policies. In fact, it is easy to show that an optimal policy exists and that there are stationary optimal policies (e.g., Ross, 1983; Bertsekas, 1987).

2.4 Markov Decision Task

For purposes of this paper we define a *(stationary, finite) Markov decision task*, to be a 5-tuple (X, A, r, f, γ) consisting of a finite state set X, finite action set A, stationary random reward function r, stationary random transition function f, and discount parameter γ. As discussed earlier, we adopt the assumption throughout that all transition probabilities and expected rewards can be determined exactly when needed. There is no problem with this if we adopt the asynchronous off-line computation point of view. In an on-line learning situation this corresponds to the unrealistic assumption that these probabilities and expected rewards can be determined directly from interaction with the environment, which is reasonable only in the special case that the transition function f and reward function r are deterministic.

2.5 One-Step Lookahead Values

For given $v \in \mathcal{R}^X$, $x \in X$, and $a \in A$, define

$$\begin{aligned} L^v(x,a) &= E\{r(x,a) + \gamma v(f(x,a))\} \\ &= R(x,a) + \gamma \sum_{y \in X} p^a_{xy} v(y). \end{aligned}$$

This quantity represents a *one-step lookahead* value, or the expected reward that would be received if, starting in state x, action a is selected, giving rise to an immediate reward of $r(x,a)$, and then the process terminates with an additional reward of $\gamma v(f(x,a))$.

Some important elementary results from the theory of dynamic programming (see, e.g., Ross, 1983, or Bertsekas, 1987) that we will need are given by the following.

Proposition 2.5.1 *For any stationary policy* π,

$$v^\pi(x) = L^{v^\pi}(x, \pi(x))$$

for all $x \in X$. *Also, a value function* v *satisfies the Bellman equation*

$$v(x) = \max_{\alpha \in A} L^v(x, \alpha) \ \ \forall x \in X$$

if and only if $v = v^*$.

We also note the following.

Lemma 2.5.2 (Lookahead Monotonicity) *For any* $v, v' \in \mathcal{R}^X$, $v \geq v'$ *implies* $L^v(x,a) \geq L^{v'}(x,a)$ *for any* $x \in X$ *and* $a \in A$. *Also, if* $v(x) > v'(x)$ *for all* $x \in X$, *then* $L^v(x,a) > L^{v'}(x,a)$ *for any* $x \in X$ *and* $a \in A$.

Proof.

$$\begin{aligned} L^v(x,a) &= R(x,a) + \gamma \sum_{y \in X} p_x y^a v(y) \\ &\geq R(x,a) + \gamma \sum_{y \in X} p^a_{xy} v'(y) \\ &= L^{v'}(x,a). \end{aligned}$$

This proves the first part. The argument for the second part simply involves making the inequality strict. □

Another elementary result that we will use later is the following.

Lemma 2.5.3 (Less Than v^* Lemma) *If* $v \leq v^*$, *then* $L^v(x,a) \leq v^*(x)$ *for any* $x \in X$ *and* $a \in A$. *Also, if* $v(x) < v^*(x)$ *for all* x, *then* $L^v(x,a) < v^*(x)$ *for any* x *and* a.

Proof. First note that $L^{v^*}(x,a) \leq v^*(x)$ for any x and a. This is true because the left-hand side is the return at x from the (possibly nonstationary) policy consisting of taking action a at the first step and following an optimal policy thereafter, and this return cannot exceed that obtained by following an optimal policy. The two parts then follow from application of the corresponding parts of the Lookahead Monotonicity Lemma (2.5.2). □

2.6 Distance Between Value Functions

Throughout this paper, distances between value functions are based on the supremum norm, so we define

$$\|v - v'\| = \|v - v'\|_\infty = \max_{x \in X} |v(x) - v'(x)|$$

for any v and $v' \in \mathcal{R}^X$.

3 Incremental Dynamic Programming Formulation

We now build on the preceding definitions, notation, and observations and show how incremental dynamic programming algorithms for actor-critic systems can be viewed in terms of application of certain operator sequences. In particular, this section focuses on incremental strategies for performing policy iteration and value iteration.

3.1 Formalization of the Actor-Critic Architecture

We regard the internal state of an actor-critic system as simply an ordered pair $(\pi, v) \in A^X \times \mathcal{R}^X$. Whatever learning algorithm is used to adapt the actor is considered to modify the stationary policy π, while learning in the critic alters the value function v. With the system initialized to some internal state (π_0, v_0), our interest is in analyzing properties of the succession of internal states that result from applying some particular learning algorithm (i.e., sequence of operators). In particular, we are concerned with the question of whether the system will eventually learn to perform optimally.

3.2 Local Backup Operators

For each $x \in X$ and $\pi \in A^X$, define the operator $B_x^\pi : \mathcal{R}^X \to \mathcal{R}^X$ by

$$B_x^\pi v(y) = \begin{cases} L^v(x, \pi(x)) & \text{if } y = x \\ v(y) & \text{otherwise.} \end{cases}$$

That is, $B_x^\pi v$ is the value function obtained from v by replacing $v(x)$ by the one-step lookahead value along π. Also, for each $x \in X$, define the operator $B_x : A^X \times \mathcal{R}^X \to A^X \times \mathcal{R}^X$ by

$$B_x(\pi, v) = (\pi, B_x^\pi v).$$

We refer to operators of either form as *local backup operators.*

3.3 Local Policy Improvement Operators

For each $x \in X$, $a \in A$, and $v \in \mathcal{R}^X$, define the operator $I_{x,a}^v : A^X \to A^X$ by

$$I_{x,a}^v \pi(y) = \begin{cases} a & \text{if } y = x \\ & \text{and } L^v(x, a) \geq L^v(x, \pi(x)) \\ \pi(y) & \text{otherwise.} \end{cases}$$

Also, for each $x \in X$ and $a \in A$, define the operator $I_{x,a} : A^X \times \mathcal{R}^X \rightarrow A^X \times \mathcal{R}^X$ by

$$I_{x,a}(\pi, v) = (I^v_{x,a}\pi, v).$$

In addition, for each $x \in X$ and $v \in \mathcal{R}^X$, define the operator $I^v_x : A^X \rightarrow A^X$ by

$$I^v_x\pi(y) = \begin{cases} a & \text{if } y = x \\ & \text{and } L^v(x, a) = \max_{\alpha \in A} L^v(x, \alpha) \\ \pi(y) & \text{otherwise,} \end{cases}$$

where it is understood that some method is used to choose among several equal candidates for giving the maximum one-step lookahead value.[2] Also, for each $x \in X$, define the operator I_x : $A^X \times \mathcal{R}^X \rightarrow A^X \times \mathcal{R}^X$ by

$$I_x(\pi, v) = (I^v_x\pi, v).$$

We refer to any of these operators as *local policy improvement operators*. The operator $I_{x,a}$ alters the policy at state x by comparing the lookahead values for the current policy and for action a while I_x considers all possible actions. As we discuss below, I_x can be viewed as representing a step in both the value iteration and policy iteration procedures. The reason we also wish to consider the $I_{x,a}$ operators is that they represent an even finer-grained incrementality, thus conforming to the overall spirit of our approach. In some applications, particularly when there are a large number of actions, it may be difficult or computationally infeasible to compare all possible actions. Related to this is the possibility that the need for fast reaction may preclude consideration of all but some small number of alternative actions at any one time. Clearly, I_x can be expressed as the composition of the operators $I_{x,a}$ with a running over A. Also, there is always one particular $I_{x,a}$ operator that gives the same result as I_x on any particular argument.

Note that both the $I_{x,a}$ and I_x operators make changes to a policy based on comparison of the lookahead values of two or more actions. In some situations it may not be realistic to assume that even this is possible. Later we will introduce additional operators which do not require comparing two or more lookahead values in this way but instead involve comparing a lookahead value with the current value for a state.

3.4 Convergence to Optimality Under Asynchronous Operator Application

The results presented in this paper all deal with problems having the following general form. We assume given a finite set $\mathcal{S}$ of operators mapping $A^X \times \mathcal{R}^X \rightarrow A^X \times \mathcal{R}^X$, and we apply a sequence $T_1, T_2, \ldots$ of these operators in succession, beginning with an initial policy-value pair $(\pi_0, v_0) \in A^X \times \mathcal{R}^X$. This gives rise to a sequence of policy-value pairs $(\pi_0, v_0), (\pi_1, v_1), (\pi_2, v_2), \ldots$, where $(\pi_k, v_k) = T_k(\pi_{k-1}, v_{k-1})$ for all $k > 0$. We adhere to this indexing convention throughout the remainder of this paper.

For any such set $\mathcal{S}$, we make the following definitions. Given a finite sequence $\Sigma = (T_1, T_2, \ldots, T_n)$ of operators from a set $\mathcal{S}$, we say that Σ is *exhaustive* (with respect to $\mathcal{S}$) if $\{T_1, T_2, \ldots, T_n\} = \mathcal{S}$, or, in other words, if every operator in $\mathcal{S}$ appears at least once in Σ. In addition, define $c_{\mathcal{S}}(\Sigma)$

[2]The precise method used for resolving ties makes no essential difference.

to be the maximum number of disjoint contiguous exhaustive subsequences into which Σ can be partitioned.

A slight generalization of this idea will also be necessary, but since we intend to apply this more general idea in only one situation, we define it only for this specific case.[3] Given a set of operators indexed by states $x \in X$, we apply the adjective *X-exhaustive* to any finite sequence of these operators in which all possible states are represented. Then we define $c_X(\Sigma)$ to be the maximum number of disjoint contiguous X-exhaustive subsequences into which Σ can be partitioned.

Given any infinite sequence $\Sigma = (T_1, T_2, \ldots)$ of operators from $\mathcal{S}$, we consistently use the notation Σ_n to mean the finite sequence $(T_1, T_2, \ldots, T_n)$ consisting of the first n elements. We then call an infinite sequence Σ of such operators *$\mathcal{S}$-forever* if $c_{\mathcal{S}}(\Sigma_n) \to \infty$ as $n \to \infty$. We have a corresponding of notion of *X-forever* using c_X. Equivalently, such a sequence is $\mathcal{S}$-forever if every operator in $\mathcal{S}$ appears infinitely often, and it is X-forever if there are infinitely many operators having index x for each $x \in X$. Also, when we speak of a *forever sequence of operators from $\mathcal{S}$* we mean an $\mathcal{S}$-forever sequence, and we say that a result holds under *asynchronous application of (the operators from) $\mathcal{S}$* if it holds for any forever sequence of operators from $\mathcal{S}$.[4]

Finally, our primary objective in this paper is to identify situations when the limiting behavior of the system is optimal, which we formulate as follows. Let us say that a sequence of policy-value pairs $(\pi_0, v_0), (\pi_1, v_1), (\pi_2, v_2), \ldots$ *converges to optimality* if $\lim_{i \to \infty} v_i = v^*$ and there exists M such that every π_i is optimal when $i \geq M$.

3.5 Computation of the Return for a Stationary Policy

Since $v^\pi(x) = L^{v^\pi}(x, \pi(x))$ for all states x when π is a stationary policy, v^π can be computed by solving the system of $|X|$ linear equations

$$v^\pi(x) = R(x, \pi(x)) + \gamma \sum_{y \in X} p_{xy}^{\pi(x)} v^\pi(y).$$

This system can be solved directly using standard linear algebra techniques, but there are other approaches of more relevance in incremental dynamic programming which involve starting with an arbitrary value function v and then iteratively updating the values $\{v(x) \mid x \in X\}$ by replacing each by $L^{v^\pi}(x, \pi(x))$. If one complete set of these values is determined in terms of an earlier complete set (i.e., updating occurs *synchronously*), then this amounts to a Jacobi method for solving this system of linear equations. If, instead, these values are updated one at a time in cyclical fashion, always using the most recent values of all other states, this amounts to a Gauss-Seidel method. Here we note that even more asynchrony of updating is allowable.

First, we make a general definition that we will need both here and later. Let $T_x : \mathcal{R}^X \to \mathcal{R}^X$ be an operator indexed by $x \in X$ and let $\gamma < 1$. We call T_x a *coordinatewise γ-contraction* if, for

[3] A general definition would have the following form: Given a mapping $\varphi : \mathcal{S} \to \mathcal{T}$, a finite sequence of operators from $\mathcal{S}$ is *φ-exhaustive* if the image under φ of the elements of the sequence equals $\mathcal{T}$, Specializing to the cases where φ is the identity map or the mapping assigning to each operator its index yields the two forms we use here.

[4] Note that we do not allow two or more operators to be applied simultaneously, so the notion of asynchrony used here is more restrictive than one might use in the sense of parallel computation. We conjecture that all results given here for this more restrictive notion of asynchrony are, in fact, true even if simultaneous updating were allowed, but we have not checked this.

any v and v' in $\mathcal{R}^X$,

$$|T_x v(y) - T_x v'(y)| \leq \|v - v'\| \tag{1}$$

for $y \neq x$ and

$$|T_x v(x) - T_x v'(x)| \leq \gamma \|v - v'\|. \tag{2}$$

Lemma 3.5.1 (Local Backup Lemma) *For any $x \in X$ and $\pi \in A^X$, the local backup operator B_x^π is a coordinatewise γ-contraction.*

Proof. Since B_x^π does not change the value at any state $y \neq x$, (1) is trivially satisfied. To check (2), let $\pi(x) = a$. Then

$$\begin{aligned} B_x^\pi v(x) - B_x^\pi v'(x) &= L^v(x,a) - L^{v'}(x,a) \\ &= R(x,a) + \gamma \sum_{y \in X} p_{xy}^a v(y) - R(x,a) - \gamma \sum_{y \in X} p_{xy}^a v'(y) \\ &= \gamma \sum_{y \in X} p_{xy}^a \left[v(y) - v'(y) \right], \end{aligned}$$

so

$$\begin{aligned} |B_x^\pi v(x) - B_x^\pi v'(x)| &\leq \gamma \sum_{y \in X} p_{xy}^a |v(y) - v'(y)| \\ &\leq \gamma \sum_{y \in X} p_{xy}^a \|v - v'\| \\ &= \gamma \|v - v'\|. \end{aligned}$$

□

Lemma 3.5.2 (Sequential Contraction) *Let $v_0, v_0' \in \mathcal{R}^X$ and let $\Sigma = (T_1, T_2, \ldots)$ be a sequence of operators indexed by X such that each is a coordinatewise γ-contraction. Define $v_k = T_k v_{k-1}$ and $v_k' = T_k v_{k-1}'$ for all $k > 0$. Then*

$$\|v_n - v_n'\| \leq \gamma^{c_X(\Sigma_n)} \|v_0 - v_0'\|,$$

and, if Σ is X-forever,

$$\|v_n - v_n'\| \to 0 \text{ as } n \to \infty.$$

Proof. The second result clearly follows from the first. We show the first by induction on the value of $c_X(\Sigma_n)$. It is trivially true for $c_X(\Sigma_n) = 0$ since each operator cannot increase distance. Now let $c_X(\Sigma_n) = k$. This means that Σ_n can be partioned into a sequence whose c_X-value is $k-1$ followed by an X-exhaustive sequence. The result will thus follow by induction if we prove that application of an X-exhaustive sequence of these operators reduces distance by a factor of γ. That is, it suffices to prove the result for $c_X(\Sigma_n) = 1$.

Therefore, assume that Σ_n is exhaustive. For notational simplicity, introduce the quantities $d_i(x) = |v_i(x) - v_i'(x)|$ and $D_i = \max_x d_i(x) = \|v_i - v_i'\|$ for $i \geq 0$. Now select a state x and let k be the largest index in the sequence such that $1 \leq k \leq n$ and T_k is indexed by x. By the coordinatewise γ-contraction property,

$$d_k(x) = |v_k(x) - v'_k(x)| \leq \gamma \|v_{k-1} - v'_{k-1}\| = \gamma D_{k-1} \leq \gamma D_0.$$

It follows further that $d_{k'}(x) \leq d_k(x) \leq \gamma D_0$ for any k' with $k \leq k' \leq n$ since each $T_{k'}$ is indexed by a state other than x. Thus, in particular, $d_n(x) \leq \gamma D_0$. Since this is true for any state x, it follows that

$$D_n = \max_x d_n(x) \leq \gamma D_0.$$

This establishes the result for the case $c_X(\Sigma_n) = 1$ and completes the proof. $\square$

In light of this result, we will say that any sequence of X-indexed operators satisfying the conditions of this lemma, and hence its conclusions, satisfies the *sequential contraction property.* The following is then an immediate consequence of the Local Backup Lemma (3.5.1).

Lemma 3.5.3 (Sequential Backup Contraction) *Any sequence of operators from* $\{B_x^\pi \mid x \in X, \pi \in A^X\}$ *satisfies the sequential contraction property.* $\square$

Theorem 3.5.4 (Asynchronous Return Computation) *If* $\{B_x \mid x \in X\}$ *is applied asynchronously to any arbitrary* (π_0, v_0)*, then* $v_n \to v^{\pi_0}$ *as* $n \to \infty$.

Proof. Consider the two sequences of value functions arising from applying this sequence of operators to v_0 and v^{π_0}. Since v^{π_0} is fixed under any of these operators, the latter sequence has all elements equal to v^{π_0}. Since the sequence of operators satisfies the sequential contraction property and is X-forever, it follows that $\|v_n - v^{\pi_0}\| \to 0$ as $n \to \infty$. $\square$

3.6 Policy Iteration

The *policy iteration* procedure determines an optimal policy by starting with any initial policy π_0 and repeating the following step: For policy π_k, compute its return v^{π_k} and then define policy π_{k+1} to satisfy

$$L^{v^{\pi_k}}(x, \pi_{k+1}(x)) = \max_\alpha L^{v^{\pi_k}}(x, \alpha),$$

for each x, with the understanding that $\pi_{k+1}(x) = \pi_k(x)$ if $\pi_k(x)$ is one of several candidates for giving the maximum lookahead value.[5] The process terminates as soon as $\pi_k(x) = \pi_{k+1}(x)$ for all x.

The justification for this is the following standard result, which is a consequence of the optimality principle and which we cite without proof.

[5]The only reason we insist on this tie-breaking method is to simplify the termination test.

Proposition 3.6.1 (Policy Improvement) *Let $v \in \mathcal{R}^X$, $\pi, \pi' \in A^X$. Then $L^{v^\pi}(x, \pi'(x)) \geq L^{v^\pi}(x, \pi(x))$ for all $x \in X$ implies $v^{\pi'} \geq v^\pi$. If, in addition, $L^{v^\pi}(x, \pi'(x)) > L^{v^\pi}(x, \pi(x))$ for some x, then $v^{\pi'}(x) > v^\pi(x)$.*

Of special interest to us here is that we can use the Asynchronous Return Computation Theorem (3.5.4) to define a sequence of operators to be applied to a given initial (π_0, v_0) that essentially carries out the policy iteration strategy. To do this, we first observe that it is only necesssary to compute v^π to a sufficiently close approximation in the policy iteration procedure. To this end, we formulate a general definition and then prove a result that will be useful later as well. For any $v \in \mathcal{R}^X$, let

$$\delta_x^v = \min \left[\left\{ \max_{\alpha \in A} L^v(x, \alpha) - L^v(x, a) \mid a \in A \right\} - \{0\} \right],$$

and let

$$\delta^v = \min_{x \in X} \delta_x^v.$$

That is, δ_x^v is the smallest nonzero difference between the lookahead value at state x when using an action giving the largest lookahead value and when using any other action, while δ^v is the smallest such nonzero difference taken across all states.

Lemma 3.6.2 (Close Enough Lemma) *Let $v, v' \in \mathcal{R}^X, x \in X$. Let $a \in A$ be such that $L^v(x, a) = \max_{\alpha \in A} L^v(x, \alpha)$ and let $a' \in A$ satisfy $L^{v'}(x, a') \geq L^{v'}(x, a)$. Then*

$$\|v - v'\| < \frac{\delta_x^v}{2\gamma}$$

implies $L^v(x, a') = L^v(x, a)$.

Proof.

$$\begin{aligned} 0 \leq L^v(x, a) - L^v(x, a') &= L^v(x, a) - L^{v'}(x, a) + L^{v'}(x, a) - L^v(x, a') \\ &\leq L^v(x, a) - L^{v'}(x, a) + L^{v'}(x, a') - L^v(x, a') \\ &\leq |L^v(x, a) - L^{v'}(x, a)| + |L^{v'}(x, a') - L^v(x, a')| \\ &\leq 2\gamma \|v - v'\| \\ &< \delta_x^v. \end{aligned}$$

Since δ_x^v is the minimum nonzero value of $L^v(x, a) - L^v(x, a')$, it follows that $L^v(x, a) - L^v(x, a') = 0$. □

We will say that action a is *greedy for value function v at state x* if $L^v(x, a) = \max_\alpha L^v(x, \alpha)$. The Close Enough Lemma simply says that if v' is sufficiently close to v, any greedy action for v' must also be greedy for v.

Using the Close Enough Lemma, we see that the same policy will result from one complete step of policy iteration if only a sufficiently close approximation to the return is computed. In particular, by the Sequential Backup Contraction Lemma (3.5.3), starting with policy π and value function v, it is sufficient to apply any sequence Σ of local backup operators satisfying

$$\gamma^{cx(\Sigma)}\|v - v^{\pi}\| < \frac{\delta^{v^{\pi}}}{2\gamma}. \tag{3}$$

By the Close Enough Lemma (3.6.2), applying policy improvement will then yield the same policy as if the return had been computed exactly.

Furthermore, improvement of the policy at each state will clearly occur under application of any exhaustive sequence of operators from $\{I_{x,a} \mid x \in X, a \in A\}$. Thus we have established the following result.

Theorem 3.6.3 (Policy Iteration Using Asynchronous Return Computation) *Suppose that a sequence of operators from $\{B_x \mid x \in X\} \cup \{I_{x,a} \mid x \in X, a \in A\}$ of the following form is applied to a given (π_0, v_0). This sequence consists of alternating indefinitely operator sequences of the following two types:*

i. Any finite sequence Σ of operators from $\{B_x \mid x \in X\}$ satisfying (3); and

ii. Any exhaustive finite sequence of operators from $\{I_{x,a} \mid x \in X, a \in A\}$.

Then the resulting sequence of policy-value pairs converges to optimality. □

3.7 Value Iteration

The *value iteration* procedure determines an optimal policy by first computing v^*. Any policy π for which

$$L^{v^*}(x, \pi(x)) = \max_{\alpha} L^{v^*}(x, \alpha)$$

will then be an optimal policy. That is, an optimal policy is determined by choosing for each x an action giving the maximum lookahead value when v^* is used to determine the evaluation of the successor states for x.

The computation of v^* can be carried out using techniques analogous to some of those described above for determining the return v^{π} for a given policy. In this case, what is sought is the unique solution to the system of $|X|$ nonlinear equations

$$v^*(x) = \max_{\alpha \in A} \left\{ R(x, \alpha) + \gamma \sum_{y \in X} p_{xy}^{\alpha} v^*(y) \right\}.$$

The value iteration procedure for solving these equations involves starting with an arbitrary value function v and then iteratively updating the values $\{v(x) \mid x \in X\}$ by replacing each by $\max_{\alpha} L^{v}(x, \alpha)$. It is traditional to carry this out synchronously or by sweeping through the states cyclically, but more arbitrary ordering is also permissible (Bertsekas, 1987; Bertsekas & Tsitsiklis, 1989). This result also serves as the starting point for analysis of the Q-learning algorithm (Watkins, 1989). Here we restate this result in our terms and give a proof.

Before doing this, however, we introduce another local operator on value functions. Since this operator is simply a projection onto the value function coordinate of a certain composition of local operators already defined, it is this composition that we will tend to emphasize later. However, since it can be defined without any explicit reference to a policy, it is useful to have this version available as well.

For each $x \in X$ define the *local max-backup operator* $B_x^* : \mathcal{R}^X \to \mathcal{R}^X$ by

$$B_x^* v(y) = \begin{cases} \max_a L^v(x,a) & \text{if } y = x \\ v(y) & \text{otherwise.} \end{cases}$$

That is, $B_x^* v$ is the value function obtained from v by replacing $v(x)$ by the maximum one-step lookahead value along all actions. Clearly, $B_x I_x(\pi, v) = (\pi', B_x^* v)$ for an appropriate π'.

Lemma 3.7.1 (Local Max-Backup Lemma) *For any $x \in X$, the local max-backup operator B_x^* is a coordinatewise γ-contraction.*

Proof. Since B_x^* does not change the value at any state $y \neq x$, (1) is trivially satisfied. To check (2), choose $a, a' \in A$ so that

$$L^v(x,a) = \max_\alpha L^v(x,\alpha)$$

and

$$L^{v'}(x,a') = \max_\alpha L^{v'}(x,\alpha).$$

Suppose that $L^{v'}(x,a') \leq L^v(x,a)$. Then

$$L^{v'}(x,a) \leq \max_\alpha L^{v'}(x,\alpha) = L^{v'}(x,a'),$$

so

$$\begin{aligned} |B_x^* v(x) - B_x^* v'(x)| &= |L^v(x,a) - L^{v'}(x,a')| \\ &= L^v(x,a) - L^{v'}(x,a') \\ &\leq L^v(x,a) - L^{v'}(x,a) \\ &\leq \gamma \|v - v'\| \end{aligned}$$

by the Local Backup Lemma. A symmetrical argument establishes the result for the case when $L^v(x,a) \leq L^{v'}(x,a')$. □

The following is an immediate consequence.

Lemma 3.7.2 (Sequential Max-Backup Contraction) *Any sequence of operators from $\{B_x^* \mid x \in X\}$ satisfies the sequential contraction property.* □

Theorem 3.7.3 (Asychronous Value Iteration) *Asynchronous application of the set of operators $\{B_x I_x \mid x \in X\}$ to an arbitrary (π_0, v_0) yields convergence to optimality.*

Proof. We first study the behavior of the resulting sequence of value functions. Since $B_x I_x(\pi, v) = (\pi', B_x^* v)$ for some π', it is sufficient to consider a corresponding sequence of operators in which each operator $B_x I_x$ is replaced by B_x^*, with this sequence applied to the initial value function v_0. Consider the effect of applying this sequence of operators to v_0 and to v^*. Since v^* is fixed under application of any B_x^*, it follows from application of the Sequential Max-Backup Contraction Lemma (3.7.2) that $v_n \to v^*$ as $n \to \infty$.

Thus convergence to optimality follows once we can show that there exists M such that all the policies π_n are optimal when $n \geq M$. Select N so that $n \geq N$ implies

$$\|v_n - v^*\| < \frac{\delta^{v^*}}{2\gamma}.$$

This is possible since $v^* = \lim_{n\to\infty}$. Then, for each x, let $N_x > N$ be an index such that $T_{N_x} = B_x I_x$. The existence of such indices is guaranteed by the assumption that each such operator appears infinitely often. Finally, let $M = \max_x N_x$.

Now fix state x and let a^* represent an optimal action at x, so that $L^{v^*}(x, a^*) = \max_{a\in A} L^{v^*}(x, a)$. Suppose that $\pi_n(x) = a'$ for some $n \geq M$. Since $\pi_n(x)$ can only have resulted from application of I_x at some earlier step m, it must be that $L^{v_m}(x, a') = \max_{a\in A} L^{v_m}(x, a) \geq L^{v_m}(x, a^*)$ Furthermore, $N \leq m \leq n$. But then, by the Close Enough Lemma, $L^{v^*}(x, a') = L^{v^*}(x, a^*)$, so, in fact, a' is an optimal action at x. Since this is true for all x, we see that, for this choice of M, all the policies π_n are optimal when $n \geq M$. □

4 Results Involving More General Sequences of Local Backup and Policy Improvement Operators

The last two theorems can be viewed from the point of view of actor-critic systems as describing particular situations in which certain forms of coordination are enforced between the adaptations performed in the actor and critic modules. The asynchronous value iteration theorem corresponds to the situation when the critic is modified via backup for any state only after the actor has been updated to reflect the currently best action for that state, while the policy iteration theorem corresponds to the situation in which the critic is always updated to reflect (a sufficiently close approximation to) the actual return from the current policy in use by the actor. Roughly speaking, for value iteration the actor must always reflect the latest changes to the critic, while for policy iteration the critic must always reflect the latest changes to the actor.

Since these two extremes of coordination between the actor and critic give rise to convergence to optimality, it seems reasonable to ask whether a less tight coupling between these two modules might not also give rise to such convergence. From the point of view of incremental learning, it would certainly be preferable not to have to wait until an accurate approximation of the value function is available before making changes to the policy. Incremental use of the value iteration approach is less problematical when it can be assumed that it is reasonable to always apply a maximization computation over all possible actions every time an action must be selected; indeed, this is the basis of Watkins' *Q-learning* approach (1989). Nevertheless, it might be preferable to allow the actor the option of immediately applying a cached choice of action without first doing any comparison of alternatives, especially in extremely time-critical situations; if the critic is allowed to adapt at these times, however, the value iteration approach is violated and the corresponding convergence theorem is no longer applicable. The purpose of the rest of this paper is to explore the extent to which less tight coupling between the operation of the actor and the critic can also give rise to the desired convergence to optimality. That is, we examine the extent to which other sequences of application of local backup and policy improvement operators also achieve such convergence. In this section we focus specifically on sequences of operators from either the set $\{B_x \mid x \in X\} \cup \{I_{x,a} \mid x \in X, a \in A\}$ or the set $\{B_x \mid x \in X\} \cup \{I_x \mid x \in X\}$. Later in this paper we introduce other similar operators and examine the behavior of sequences of these operators as well. Recall that in all cases we consistently denote the k^{th} operator by T_k and the k^{th} policy-value pair by (π_k, v_k) and adopt the numbering convention that $(\pi_k, v_k) = T_k(\pi_{k-1}, v_{k-1})$ for all $k > 0$.

4.1 A Useful Technical Result

A technical result that we will aid us in several proofs throughout this paper is the following.

Lemma 4.1.1 (Non-v^* Lemma) *Let $v' \in \mathcal{R}^X$ be given, with $v' \neq v^*$. Then there exists $x \in X$ and $\varepsilon > 0$ such that, for any $v \in \mathcal{R}^X$ with $\|v - v'\| < \varepsilon$,*

$$|v^*(x) - L^v(x,a)| < |v^*(x) - v'(x)| - \varepsilon$$

for all $a \in A$ satisfying $L^{v'}(x,a) = \max_\alpha L^{v'}(x,\alpha)$.

Proof. Let

$$x = \arg\max_{y \in X} |v^*(y) - v'(y)|,$$

so that

$$|v^*(x) - v'(x)| = \|v^* - v'\|,$$

and let

$$\varepsilon = \frac{1-\gamma}{1+\gamma}|v^*(x) - v'(x)|,$$

which is positive since $v^*(x) \neq v'(x)$. Then

$$|v^*(x) - v'(x)| - \varepsilon = \frac{2\gamma}{1+\gamma}|v^*(x) - v'(x)|.$$

Now let v be a value function satisfying $\|v - v'\| < \varepsilon$ and let a be an action such that $a = \arg\max_\alpha L^{v'}(x,\alpha)$. By the triangle inequality,

$$|v^*(x) - L^v(x,a)| \leq |v^*(x) - L^{v'}(x,a)| + |L^{v'}(x,a) - L^v(x,a)|.$$

Since $L^{v'}(x,a) = B_x^* v'(x)$ and $B_x^* v^*(x) = v^*(x)$, the Local Max-Backup Lemma (3.7.1) implies that the first term satisfies

$$|v^*(x) - L^{v'}(x,a)| \leq \gamma\|v^* - v'\| = \gamma|v^*(x) - v'(x)|,$$

while the Local Backup Lemma (3.5.1) implies that the second term satisfies

$$|L^{v'}(x,a) - L^v(x,a)| \leq \gamma \|v' - v\| < \gamma\varepsilon.$$

Therefore,

$$\begin{aligned}
|v^*(x) - L^v(x,a)| &< \gamma|v^*(x) - v'(x)| + \gamma\varepsilon \\
&= \gamma|v^*(x) - v'(x)| + \gamma\frac{1-\gamma}{1+\gamma}|v^*(x) - v'(x)| \\
&= \frac{2\gamma}{1+\gamma}|v^*(x) - v'(x)| \\
&= |v^*(x) - v'(x)| - \varepsilon.
\end{aligned}$$

□

4.2 Results For Arbitrary Sequences

Theorem 4.2.1 (Convergence Implies Optimality) *Suppose that a forever sequence of operators from $\{B_x \mid x \in X\} \cup \{I_{x,a} \mid x \in X, a \in A\}$ is applied to some (π_0, v_0). Then, if $\lim_{i\to\infty} v_i(x)$ exists for each $x \in X$, the resulting sequence of policy-value pairs converges to optimality.*

Proof. Let $v_\infty = \lim_{i\to\infty} v_i$. Suppose that $v_\infty \neq v*$. Let x and ε be as guaranteed by the Non-v^* Lemma (4.1.1) applied to v_∞. Pick N so that $n \geq N$ implies

$$\|v_n - v_\infty\| < \min\left(\frac{\delta_x^{v_\infty}}{2\gamma}, \varepsilon\right),$$

and then pick $k \geq N$ so that $T_{k+1} = I_{x,a}$, with a greedy for v_∞. Let $l > k$ be such that $T_{l+1} = B_x$. By the Close Enough Lemma, $\pi_l(x)$ must be a greedy action at x for v_∞ since it must have looked no worse than a using a value function within $\delta_x^{v_\infty}/(2\gamma)$ of v_∞. Thus the Non-v^* Lemma (4.1.1) allows us to conclude that $v_{l+1}(x) = B_x^{\pi_l} v_l(x) = L^{v_l}(x, \pi_l(x))$ satisfies

$$|v^*(x) - v_{l+1}(x)| < |v^*(x) - v_\infty(x)| - \varepsilon,$$

or

$$|v^*(x) - v_\infty(x)| - |v^*(x) - v_{l+1}(x)| > \varepsilon,$$

By the triangle inequality we also have

$$|v^*(x) - v_\infty(x)| \leq |v^*(x) - v_{l+1}(x)| + |v_{l+1}(x) - v_\infty(x)|,$$

so

$$|v_{l+1}(x) - v_\infty(x)| \geq |v^*(x) - v_\infty(x)| - |v^*(x) - v_{l+1}(x)| > \varepsilon.$$

Since $l + 1 \geq N$, this is a contradiction, and we thus conclude that $v_\infty = v*$.

Furthermore, that the policies must eventually all be optimal follows from the Close Enough Lemma, since eventually all v_n will be within $\delta^{v^*}/(2\gamma)$ of v^*. Thus any I_{x,a^*} operator, with a^* optimal at x, applied beyond this point must set the policy at x to an optimal action, and any changes in the policy at x occuring after this can only replace one optimal action with another. $\square$

This result can be used to establish convergence to optimality in a number of cases. Before examining some of these cases, we first observe that it is not true that applying each local backup and policy improvement operator infinitely often guarantees convergence to optimality. The following result is proved by giving examples where the resulting policy-value pairs cycle endlessly.

Theorem 4.2.2 (Convergence Counterexamples) *For each of the following sets of operators $\mathcal{S}$ and values of γ, there exists a finite Markov decision task with that value of γ, an initial policy-value pair, and a forever sequence of operators from $\mathcal{S}$, such that the resulting sequence of policy-value pairs does not converge to optimality:*

i. $\mathcal{S} = \{B_x \mid x \in X\} \cup \{I_x \mid x \in X\}$, $1/2 < \gamma < 1$

ii. $\mathcal{S} = \{B_x \mid x \in X\} \cup \{B_x I_x \mid x \in X\}$, $(\sqrt{5}-1)/2 < \gamma < 1$

iii. $\mathcal{S} = \{I_x B_x \mid x \in X\}$, $(\sqrt{5}-1)/2 < \gamma < 1$

iv. $\mathcal{S} = \{I_{x,a} B_x \mid x \in X, a \in A\}$, $(\sqrt{5}-1)/2 < \gamma < 1$

v. $\mathcal{S} = \{B_x I_{x,a} \mid x \in X, a \in A\}$, $(\sqrt{5}-1)/2 < \gamma < 1$

Proof. Case (i). Consider the example in Figure 1. Applying B_1 will do the assignment

$$v(1) \leftarrow 1 + \gamma \frac{1}{1-\gamma} = \frac{1}{1-\gamma}.$$

Performing an I_3 operation then compares

$$L^v(3,1) = 1 + \gamma \frac{3}{1-\gamma} = \frac{1+2\gamma}{1-\gamma}$$

with

$$L^v(3,2) = 3 + \gamma \frac{1}{1-\gamma} = \frac{3-2\gamma}{1-\gamma}.$$

Since $\gamma < 1$, both denominators are positive. Since $\gamma > 1/2$, the first numerator is more than 2 and the second is less than 2. Therefore the first action appears better, so $\pi(3) \leftarrow 1$. Similarly, doing a B_4 operation will perform the assignment

$$v(4) \leftarrow 3 + \gamma \frac{3}{1-\gamma} = \frac{3}{1-\gamma}.$$

Applying the I_6 operator then compares

$$L^v(6,1) = 1 + \gamma \frac{1}{1-\gamma} = \frac{1}{1-\gamma}$$

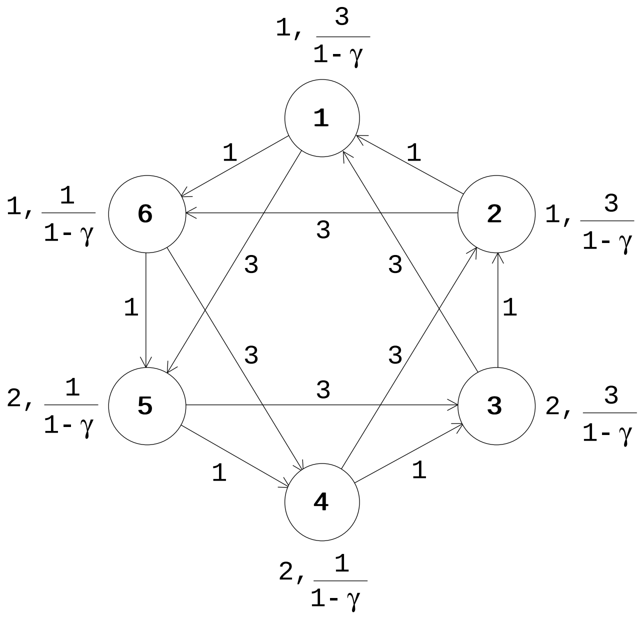

Figure 1: Deterministic Markov environment for the first counterexample. At each state, action 1 causes a transition to the next state in a counterclockwise direction, while action 2 moves two steps in that same direction. Immediate rewards are as indicated, 1 when action 1 is taken in any state and 3 when action 2 is taken. By each state is shown the initial policy and value, respectively, for that state.

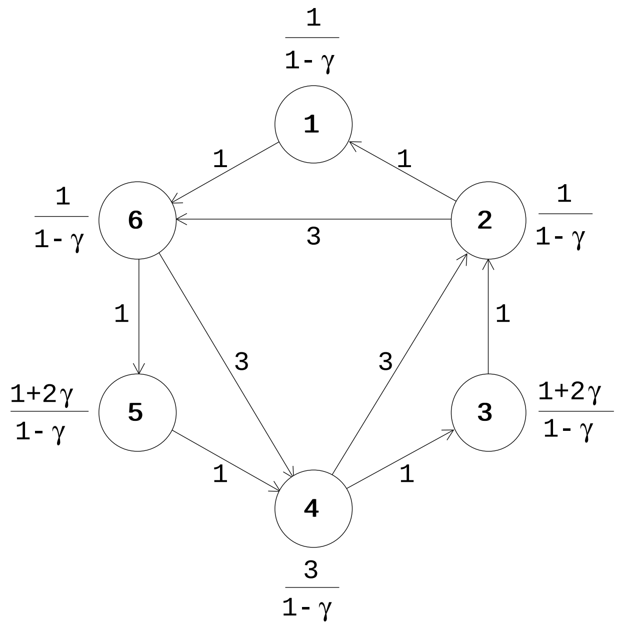

Figure 2: Deterministic Markov environment for the second counterexample. At each state, action 1 causes a transition to the next state in a counterclockwise direction. At even-numbered states, action 2 moves two steps in that same direction, while at odd-numbered states the effect of action 2 is identical to that of action 1. Immediate rewards are as indicated, 1 when action 1 is taken in any state or action 2 is taken in an odd-numbered state, and 3 when action 2 is taken in an even-numbered state. By each state is shown the initial value for that state. For this example the initial policy is arbitrary.

with

$$L^v(6,2) = 3 + \gamma\frac{3}{1-\gamma} = \frac{3}{1-\gamma}.$$

The second choice is better, so $\pi(6) \leftarrow 2$. At this point, the configuration of values and policies has rotated clockwise 1/6 of the way around. We can thus apply corresponding sets of operators to continue rotating the configuration until it returns to its original form. An entire sequence of 24 operators to accomplish this is given by performing twice the following composition:

$$I_2 B_6 I_5 B_3 I_1 B_5 I_4 B_2 I_6 B_4 I_3 B_1$$

Thus endless application of these 12 operators in this order will cause the value function to oscillate between $1/(1-\gamma)$ and $3/(1-\gamma)$ at each state, so it never converges.

Case (ii). Consider the example in Figure 2. First note that at each odd-numbered state the effect of both actions is identical, giving an immediate reward of 1 and causing a transition to the same state. This means that I_n will have at most a trivial effect on $\pi(n)$ when n is odd, so that applying $B_n I_n$ has the same effect on $v(n)$ as simply applying B_n at these states.

Begin by applying $B_6 I_6$. This compares

$$L^v(6,1) = 1 + \gamma\frac{1+2\gamma}{1-\gamma} = \frac{1+2\gamma^2}{1-\gamma}$$

with

$$L^v(6,2) = 3 + \gamma\frac{3}{1-\gamma} = \frac{3}{1-\gamma}$$

to determine $\pi(6)$ and then installs the larger value as $v(6)$. Since $\gamma < 1$, the numerator of the first value is less than 3, so $\pi(6) \leftarrow 2$ and $v(6) \leftarrow 3/(1-\gamma)$. Next, applying $B_4 I_4$ requires the comparison of

$$L^v(4,1) = 1 + \gamma\frac{1+2\gamma}{1-\gamma} = \frac{1+2\gamma^2}{1-\gamma}$$

with

$$L^v(4,2) = 3 + \gamma\frac{1}{1-\gamma} = \frac{3-2\gamma}{1-\gamma}.$$

Since $\gamma > (\sqrt{5}-1)/2$, the first numerator is greater than $4-\sqrt{5}$ and the second one is less than $4-\sqrt{5}$, so $\pi(4) \leftarrow 1$ and $v(4) \leftarrow (1+2\gamma^2)/(1-\gamma)$. After this, applying B_3 makes the assignment

$$v(3) \leftarrow 1 + \gamma\frac{1}{1-\gamma} = \frac{1}{1-\gamma}.$$

Applying $B_3 I_3$ then has no effect. Applying $B_1 I_1$ next sets

$$v(1) \leftarrow 1 + \gamma\frac{3}{1-\gamma} = \frac{1+2\gamma}{1-\gamma}.$$

Finally, applying B_4 leads to

$$v(4) \leftarrow 1 + \gamma\frac{1}{1-\gamma} = \frac{1}{1-\gamma}.$$

At this point the configuration of values has rotated clockwise 1/3 of the way around, and the resulting policy is independent of the initial choice of policy. We can thus apply corresponding sets of operators to continue rotating the configuration of values until it returns to its original form. An entire sequence of 18 operators to accomplish this is given by the following composition:

$$\begin{gathered}B_2(B_5I_5)(B_1I_1)B_1(B_2I_2)(B_4I_4)\\B_6(B_3I_3)(B_5I_5)B_5(B_6I_6)(B_2I_2)\\B_4(B_1I_1)(B_3I_3)B_3(B_4I_4)(B_6I_6)\end{gathered}$$

This includes all possible B_x and B_xI_x operators, and the value function oscillates between $1/(1-\gamma)$ and $(1+2\gamma)/(1-\gamma)$ for odd-numbered states and between $1/(1-\gamma)$ and $3/(1-\gamma)$ for even-numbered states.

Case (iii). Note that in case (ii) the single operators B_n for n odd can be removed without changing the overall effect. Also note that when the single operators B_n for n even were applied, the next operation done on that state was an I_n. If an extra I_n were to be performed immediately after each such B_n, the result would not be affected. Finally, note that the Markov environment used for this example has no direct transitions from any state to itself. In this case the composition $I_xB_xI_xB_x$ has an interesting property. The B_xI_x in the middle finds the best looking action and stores the action and the backed up evaluation along it. The I_x on the left has no effect since none of the other states' values have changed since the first I_x. The B_x on the right also has no effect since it can't influence what the immediately following I_x does and since the next B_x will overwrite $v(x)$ anyway. After deleting unnecessary individual B's, changing the remaining individual B's into IB's, and replacing BI's with $IBIB$'s, the composition of operators used in case (ii) is transformed into:

$$\begin{gathered}(I_2B_2)(I_5B_5I_5B_5)(I_1B_1I_1B_1)(I_2B_2I_2B_2)(I_4B_4I_4B_4)\\(I_6B_6)(I_3B_3I_3B_3)(I_5B_5I_5B_5)(I_6B_6I_6B_6)(I_2B_2I_2B_2)\\(I_4B_4)(I_1B_1I_1B_1)(I_3B_3I_3B_3)(I_4B_4I_4B_4)(I_6B_6I_6B_6)\end{gathered}$$

This composition of operators has the same effect as in case (ii), but it consists entirely of I_xB_x operators, all of which appear at least once.

Case (iv). In general, applying I_x is equivalent to applying the composition $I_{x,a_1}I_{x,a_2}\cdots I_{x,a_m}$. If there are no self loops, then inserting B_x operators won't affect the operation of the I's. Therefore $I_{x,a_1}B_xI_{x,a_2}B_x\cdots I_{x,a_m}B_x$ composed with itself is equivalent to B_xI_x. Also, if the current policy at x is a_k, then I_{x,a_k} does nothing and $I_{x,a_k}B_x$ is equivalent to B_x. If these substitutions are made in the composition of operators given for case (ii), the resulting composition will be constructed entirely out of $I_{x,a}B_x$ operators, with all appearing, and its effect will be the same.

Case (v). Once again, consider the composition of operators from case (ii). If the current policy at any step of the process is a_k, then replace each single B_x with B_xI_{x,a_k} and replace each B_xI_x with $B_xI_{x,a_1}B_xI_{x,a_2}\cdots B_xI_{x,a_m}$. When these substitutions are made for the operators used in case (ii), the resulting composition will represent an exhaustive sequence of $B_xI_{x,a}$ operators, and its effect will be the same. $\square$

The first case gives a counterexample to the most general conjecture one might make concerning convergence to optimality using arbitrary sequences of I_x and B_x operators, while the remaining cases are counterexamples to other more limited conjectures one might have hoped to salvage after discovery of the first counterexample. The second case shows that applying occasional backups along actions not currently believed to be optimal can defeat the guaranteed convergence that would otherwise be obtained when the asynchronous value iteration operators $B_x I_x$ are used. The third case shows that synchronizing the policy updates and the backups in the reverse order from that used in asynchronous value iteration can thwart convergence. That convergence can fail in the fifth case is interesting because one might have considered the use of such operators to represent a reasonable finer-grained approximation to the asynchronous value iteration approach.

Now we return to the question of finding sequences of local backup and policy improvement operators and initial policy-value pairs such that convergence to optimality is guaranteed. The next two results will be useful for this. In order to state these results in complete generality, we define a *policy operator* mapping $A^X \times \mathcal{R}^X \to A^X \times \mathcal{R}^X$ to be any such operator commuting with projection onto $\mathcal{R}^X$. That is, P is a policy operator if $P(\pi, v) = (\pi', v)$ for any v and π and for some π'. Examples are the operators I_x and $I_{x,a}$. Later we consider others.

Lemma 4.2.3 (Boundedness) *Let (π_0, v_0) be any policy-value pair and let $\mathcal{P}$ be any set of policy operators. Let $v^{\max} = \max_x v_0(x)$, $v^{\min} = \min_x v_0(x)$, $R^{\max} = \max_{x,a} R(x,a)$, and $R^{\min} = \min_{x,a} R(x,a)$. Then the sequence of policy-value pairs arising from application of any sequence of operators from $\{B_x \mid x \in X\} \cup \mathcal{P}$ to (π_0, v_0) satisfies*

$$\min\left\{v^{\min}, \frac{R^{\min}}{1-\gamma}\right\} \le v_i(x) \le \max\left\{v^{\max}, \frac{R^{\max}}{1-\gamma}\right\} \tag{4}$$

for all $x \in X$ and for all $i \ge 0$.

Proof. We prove this by induction on i. First, (4) is clearly true for all x when $i = 0$. Now assume that (4) is true for all x when $i = k$. Pick a particular state x. If $v_{k+1}(x) = v_k(x)$, then (4) holds trivially for $i = k+1$. This covers all cases when T_{k+1} is not equal to B_x. To examine the remaining case, assume $T_{k+1} = B_x$.

First consider the upper bound. Note that

$$\begin{aligned} v_{k+1}(x) &= L^{v_k}(x, \pi_k(x)) \\ &\le R^{\max} + \gamma \max_y v_k(y) \\ &\le R^{\max} + \gamma v^{\max}. \end{aligned}$$

If $v^{\max} \le R^{\max}/(1-\gamma)$, it then follows that

$$v_{k+1}(x) \le R^{\max} + \gamma \frac{R^{\max}}{1-\gamma} = \frac{R^{\max}}{1-\gamma},$$

while if $R^{\max}/(1-\gamma) \le v^{\max}$, we conclude that

$$v_{k+1}(x) \le (1-\gamma) v^{\max} + \gamma v^{\max} = v^{\max}.$$

Thus, in either case, $v_{k+1}(x) \le \max\{v^{\max}, R^{\max}/(1-\gamma)\}$.

We have a symmetrical argument for the lower bound, beginning with the observation that

$$\begin{aligned} v_{k+1}(x) &= L^{v_k}(x, \pi_k(x)) \\ &\ge R^{\min} + \gamma \min_y v_k(y) \\ &\ge R^{\min} + \gamma v^{\min}. \end{aligned}$$

When $v^{\min} \ge R^{\min}/(1-\gamma)$, this implies

$$v_{k+1}(x) \ge R^{\min} + \gamma \frac{R^{\min}}{1-\gamma} = \frac{R^{\min}}{1-\gamma},$$

while $R^{\min}/(1-\gamma) \ge v^{\min}$ implies

$$v_{k+1}(x) \ge (1-\gamma)v^{\min} + \gamma v^{\min} = v^{\min},$$

and we conclude that $v_{k+1}(x) \ge \min\{v^{\min}, R^{\min}/(1-\gamma)\}$.

Since x was arbitrary, this suffices to prove that (4) holds for all x when $i = k+1$. Therefore, by induction, it holds for all $i \ge 0$. □

We can establish another interesting set of bounds on the limiting behavior of sequences arising from repeated application of the local backup operators and other operators that affect only the policy if we assume that each backup operator is applied infinitely often. For the proof of this result we will make use a certain construction that is closely related to the *action replay process* introduced by Watkins (1989; Watkins & Dayan, 1992) to analyze Q-learning.

Let there be given an initial stationary policy π_0, a sequence of states $x_1, x_2, x_3, \ldots$, and a sequence of actions $a_1, a_2, a_3, \ldots$. For any n we construct a nonstationary policy making use of the first n states and actions in these sequences and maintaining an indexing variable that accesses these sequences and decreases at each time step. Let K denote the current value of this indexing variable. Initially, $K = n$. Then repeat indefinitely the following steps: When state x is entered, find the largest value $L \le K$ for which $x_L = x$. If there is no such value, set $K \leftarrow 0$ and apply action $\pi_0(x)$. Otherwise, apply action a_L and set $K \leftarrow L - 1$. Using this construction for each $n \ge 0$ we obtain a sequence of policies $\pi_0, \pi_1, \pi_2, \ldots$.

For ease in stating the following result we use the notation B_x^a, with $a \in A$, to mean the same thing as B_x^π where π is any stationary policy such that $\pi(x) = a$.

Lemma 4.2.4 *For the above construction, let $v_0 = v^{\pi_0}$ and define $v_n = B_{x_n}^{a_n} v_{n-1}$ for all $n > 0$. Then $v_n = v^{\pi_n}$ for all $n \ge 0$.*

Proof. We argue by induction. The base case is given, so we now suppose that $v_k = v^{\pi_k}$ for some k. Then $v_{k+1} = B_{x_{k+1}}^{a_{k+1}} v_k$, so we want to show that $v^{\pi_{k+1}} = B_{x_{k+1}}^{a_{k+1}} v_k$. For $x \ne x_{k+1}$, it is clear that

$$v^{\pi_{k+1}}(x) = v_k(x) = B_{x_{k+1}}^{a_{k+1}} v_k$$

since π_{k+1} and π_k represent the same policy for all time steps as long as the initial state is not x_{k+1}. Thus it remains to show that $v^{\pi_{k+1}}(x_{k+1}) = B^{a_{k+1}}_{x_{k+1}} v_k(x_{k+1})$. But

$$\begin{aligned} v^{\pi_{k+1}}(x_{k+1}) &= E\left\{\sum_{t=0}^{\infty} \gamma^t r(x(t), a(t)) \mid x_0 = x_{k+1}\right\} \\ &= R(x_{k+1}, a_{k+1}) + E\left\{\sum_{t=1}^{\infty} \gamma^t r(x(t), a(t)) \mid x_0 = x_{k+1}\right\} \\ &= R(x_{k+1}, a_{k+1}) + \gamma \sum_{y \in X} p^{a_{k+1}}_{xy} E\left\{\sum_{t=1}^{\infty} \gamma^{t-1} r(x(t), a(t)) \mid x_1 = y\right\} \end{aligned}$$

Now since π_{k+1} is identical to π_k after the first time step, the expected value appearing in the sum in this last equation is simply $v^{\pi_k}(y)$, so we have

$$\begin{aligned} v^{\pi_{k+1}}(x_{k+1}) &= L^{v^{\pi_k}}(x_{k+1}, a_{k+1}) \\ &= L^{v_k}(x_{k+1}, a_{k+1}) \\ &= B^{\pi_{k+1}}_{x_{k+1}} v_k(x_{k+1}). \end{aligned}$$

□

We now use this lemma to establish our next result, which requires one further observation and definition. Just as there are optimal policies, there are also *pessimal* policies, which give the minimum total expected discounted return at every state. Define $v^{\#}$ to be the return from any such pessimal policy, so that $v^{\#} \leq v^{\pi}$ for all policies π (including nonstationary and randomized policies).

Theorem 4.2.5 (Limit Bounds) *Let $\mathcal{P}$ be any set of policy operators and let $\mathcal{B} = \{B_x \mid x \in X\}$ Suppose that a $\mathcal{B}$-forever sequence of operators from $\{B_x \mid x \in X\} \cup \mathcal{P}$ is applied to an arbitrary (π_0, v_0). Then the resulting sequence of value functions satisfies*

$$v^{\#}(x) \leq \liminf_{n\to\infty} v_n(x) \leq \limsup_{n\to\infty} v_n(x) \leq v^*(x)$$

for all states x.

Proof. First note that, since $B_x(\pi, v) = (\pi, B^{\pi}_x v)$ and the policies arising in the sequence are obtained from the application of arbitrary policy operators, it is equivalent to establish the conclusion when one applies to any v_0 an X-forever sequence of operators from $\left\{B^{\pi}_x \mid x \in X, \pi \in A^X\right\}$.

Thus we confine attention to sequences of this latter form, in which each operator is some B^{π}_x, and we adopt a corresponding indexing convention,[6] with the i^{th} operator being $B^{\pi_i}_{x_i}$, with $v_i = B^{\pi_i}_{x_i} v_{i-1}$ for $i > 0$.

[6]Thus these indexes differ from those in the original sequence because the policy operators no longer occupy positions in this new sequence.

The proof is based on the observation that, for each n, v_n approximates the return for a particular nonstationary policy, with the approximation error approaching zero as $n \to \infty$. Each such nonstationary policy arises from applying the construction given above, as follows. Given the sequence of operators $B_{x_1}^{\pi_1}, B_{x_2}^{\pi_2}, B_{x_3}^{\pi_3}, \ldots$, extract the sequence of states $x_1, x_2, x_3, \ldots$ and the sequence of actions $a_1, a_2, a_3, \ldots$, where $a_i = \pi_i(x_i)$ for all i. We then use π'_n to denote the nth nonstationary policy resulting from this construction, and we arbitrarily set the initial stationary policy π'_0 for this construction to be π_0.

Now let $v'_0 = v^{\pi'_0}$ and define $v'_n = B_{x_n}^{\pi_n} v'_{n-1}$ for all $n > 0$. The sequence $v'_0, v'_1, v'_2, \ldots$ then has two important properties. First, both it and the original sequence $v_0, v_1, v_2, \ldots$ arise from application of the same X-forever sequence of operators from $\left\{B_x^\pi \mid x \in X, \pi \in A^X\right\}$, so it follows from the Sequential Backup Contraction Lemma (3.5.3) that

$$\|v_n - v'_n\| \to 0 \text{ as } n \to \infty.$$

Second, since $v^{\pi_0} = v^{\pi'_0}$ and $B_{x_n}^{\pi_n} v'_{n-1} = B_{x_n}^{a_n} v'_{n-1}$ for all $n > 0$, the previous lemma allows us to conclude that $v'_n = v^{\pi'_n}$ for all n. But this means that each v'_n is the return from some policy, so $v^\# \le v'_n \le v^*$ for all n. The desired conclusion then follows from taking the limit superior of both sides of the inequality $v'_n - v_n \le v^* - v_n$ and the limit inferior of both sides of the inequality $v^\# - v_n \le v'_n - v_n$, in each case making use of the fact that $\lim_{n\to\infty} \|v'_n - v_n\| = 0$. □

It is interesting to note that the example used in part (i) of the Convergence Counterexample Theorem (4.2.2) has the property that the value for each state oscillates between the optimal and pessimal values. This shows that the bounds given in the Limit Bounds Theorem (4.2.5) are tight even when the policy operators used are the local policy improvement operators.

Theorem 4.2.6 *If $L^{v_0}(x, \pi_0(x)) \ge v_0(x)$ for all $x \in X$, then asynchronous application of $\{B_x \mid x \in X\} \cup \{I_{x,a} \mid x \in X, a \in A\}$ to (π_0, v_0) yields convergence to optimality.*

Proof. We use the Boundedness Lemma (4.2.3) and the Nondecreasing Lemma (4.2.7), given below. From these, it follows that the sequence $v_0(x), v_1(x), v_2(x), \ldots$ is nondecreasing and bounded above for each x. Therefore, the sequence $v_0, v_1, v_2, \ldots$ has a limit and the Convergence Implies Optimality Theorem (4.2.1) gives us the desired result. □

Lemma 4.2.7 (Nondecreasing) *Suppose that a sequence of operators from $\{B_x \mid x \in X\} \cup \{I_{x,a} \mid x \in X, a \in A\}$ is applied to an initial (π_0, v_0) satisfying $L^{v_0}(x, \pi_0(x)) \ge v_0(x)$ for all $x \in X$. Then, for all x and i,*

i. $L^{v_i}(x, \pi_i(x)) \ge v_i(x)$ *and*

ii. $v_{i+1}(x) \ge v_i(x)$.

Proof. We first show that for any fixed i, (ii) holds for all x whenever (i) does. If T_{i+1} is not a local backup operator, then $v_{i+1}(x) = v_i(x)$ for all x and (ii) holds trivially. If $T_{i+1} = B_y$ for some state y, then $v_{i+1}(x) = v_i(x)$ for $x \ne y$ while $v_{i+1}(y) = L^{v_i}(y, \pi_i(y)) \ge v_i(y)$ by assumption.

Now we prove that (i) holds for all i by induction. Assume it's true for $i = k$. Consider the case $T_{k+1} = B_y$ for some y. Then $\pi_{k+1} = \pi_k$, so, for any x,

$$\begin{aligned} L^{v_{k+1}}(x, \pi_{k+1}(x)) &= L^{v_{k+1}}(x, \pi_k(x)) \\ &\geq L^{v_k}(x, \pi_k(x)) \\ &\geq v_k(x), \end{aligned}$$

where the second inequality holds by the induction hypothesis and the first inequality follows from the fact that $v_{k+1} \geq v_k$, which was shown above to be a consequence of this same induction hypothesis. Since

$$v_{k+1}(x) = \begin{cases} L^{v_k}(y, \pi_k(y)) & \text{if } x = y. \\ v_k(x) & \text{otherwise,} \end{cases}$$

it follows that $L^{v_{k+1}}(x, \pi_{k+1}(x)) \geq v_{k+1}(x)$ for all x whenever $T_{k+1} = B_y$ for some y.

Now consider the case when $T_{k+1} = I_{y,a}$ for some state y and action a. Since $\pi_{k+1}(x) = \pi_k(x)$ when $x \neq y$ and

$$\begin{aligned} L^{v_k}(y, \pi_{k+1}(y)) &= L^{v_k}(y, I_{y,a}^{v_k}\pi_k(y)) \\ &\geq L^{v_k}(y, \pi_k(y)), \end{aligned}$$

we see that $L^{v_k}(x, \pi_{k+1}(x)) \geq L^{v_k}(x, \pi_k(x))$ for all x in this case. Since $v_{k+1} = v_k$, it follows that, for any x,

$$\begin{aligned} L^{v_{k+1}}(x, \pi_{k+1}(x)) &= L^{v_k}(x, \pi_{k+1}(x)) \\ &\geq L^{v_k}(x, \pi_k(x)) \\ &\geq v_k(x) \\ &= v_{k+1}(x), \end{aligned}$$

where the second inequality holds by the induction hypothesis. $\square$

We single out the following noteworthy corollaries to Theorem 4.2.6.

Corollary 4.2.8 *If all the expected rewards $R(x, a)$ are nonnegative and $v_0(x) = 0$ for all x, then asynchronous application of $\{B_x \mid x \in X\} \cup \{I_{x,a} \mid x \in X, a \in A\}$ yields convergence to optimality.*

Proof. For any x,

$$\begin{aligned} L^{v_0}(x, \pi_0(x)) &= R(x, a) + \gamma \sum_{y \in X} p_{xy}^a v_0(y) \\ &= R(x, a) \\ &\geq 0 \\ &= v_0(x). \end{aligned}$$

$\square$

The following result provides an interesting generalization of policy iteration. It represents an asynchronous extension of a similar result obtained earlier by Puterman and Shin (1978) based on the use of global backup and policy improvement steps.

Corollary 4.2.9 *If $v_0 = v^{\pi_0}$, then asynchronous application of $\{B_x \mid x \in X\} \cup \{I_{x,a} \mid x \in X, a \in A\}$ yields convergence to optimality.*

Proof. For any x,

$$L^{v_0}(x, \pi_0(x)) = v^{\pi_0}(x) = v_0(x).$$

□

Now we observe that when certain conditions are imposed on the discount parameter γ, convergence of the value function can be guaranteed, which leads to other interesting ways that convergence to optimality may be assured.

Theorem 4.2.10 *Asynchronous application of the operators from $\mathcal{S}$ to an arbitrary (π_0, v_0) yields convergence to optimality whenever $\mathcal{S}$ and γ satisfy:*

i. $\mathcal{S} = \{B_x \mid x \in X\} \cup \{I_x \mid x \in X\}$ *and* $\gamma < 1/2$; *or*

ii. $\mathcal{S} = \{B_x \mid x \in X\} \cup \{I_{x,a} \mid x \in X, a \in A\}$ *and* $\gamma < 1/|A|$.

Proof. In light of the Convergence Implies Optimality Theorem (4.2.1), it suffices to prove convergence of the resulting sequence of value functions $v_0, v_1, v_2, \ldots$. For either case, define

$$L(x) = \limsup_{i\to\infty} v_i(x) - \liminf_{i\to\infty} v_i(x)$$

for each state x, and let

$$L^* = \max_{x\in X} L(x),$$

with x^* chosen to be any state such that $L(x^*) = L^*$. Note that by the Boundedness Lemma (4.2.3), (or by the Limit Bounds Theorem), these limits inferior and superior are finite, so the $L(x)$ values are well defined. Furthermore, L^* is well-defined since there are finitely many states, and, of course, it is nonnegative since each $L(x)$ is. For each x, $L(x)$ is a measure of the limiting extent of variation in the values $\{v_k(x)\}$, with x^* being a state undergoing the widest such variation. We now proceed to show that $L^* = 0$ for either case.

Momentarily fix $\epsilon > 0$ and pick N so that $k \geq N$ implies

$$\liminf_{i\to\infty} v_i(x) - \epsilon < v_k(x) < \limsup_{i\to\infty} v_i(x) + \epsilon. \tag{5}$$

For each $a \in A$ define

$$\begin{aligned} b_0(a) &= \inf_{k\geq N} L^{v_k}(x^*, a) \\ &= R(x^*, a) + \gamma \inf_{k\geq N} \sum_{y\in X} p^a_{x^*y} v_k(y) \end{aligned} \tag{6}$$

and

$$\begin{aligned} b_1(a) &= \sup_{k \geq N} L^{v_k}(x^*, a) \\ &= R(x^*, a) + \gamma \sup_{k \geq N} \sum_{y \in X} p^a_{x^* y} v_k(y). \end{aligned} \tag{7}$$

These represent the lower and upper limits, respectively, on the possible values that could be obtained if one were to apply backup along action a at state x^* at any point beyond step N. From (6) and (5) it follows that

$$b_0(a) < R(x^*, a) + \gamma \sum_{y \in X} p^a_{x^* y} \limsup_{i \to \infty} v_i(x) + \epsilon.$$

Likewise, from (7) and (5) it follows that

$$b_1(a) > R(x^*, a) + \gamma \sum_{y \in X} p^a_{x^* y} \liminf_{i \to \infty} v_i(x) - \epsilon.$$

Thus for each a the lookahead value at x^* along a is always contained in the interval $[b_0(a), b_1(a)]$ after step N. Thus, letting $d(a) = b_1(a) - b_0(a)$ denote the size of this interval we have that

$$\begin{aligned} d(a) &< \gamma \sum_{y \in X} p^a_{x^* y} L(y) + 2\gamma\epsilon \\ &\leq \gamma L^* + 2\gamma\epsilon \end{aligned}$$

for all $a \in A$.

Note that if the sequence of operators contains no policy improvement operators beyond step N, then all subsequent backups at x^* will always be along the same particular action a, so the entire range of variation in the value for x^* beyond step N can be no larger than the size of the interval $[b_0(a), b_1(a)]$, so that

$$L^* \leq d(a) < \gamma L^* + 2\gamma\epsilon$$

in this case, which implies

$$L^* < \frac{2\gamma\epsilon}{1 - \gamma}$$

for any $\gamma < 1$. Since this is true for all $\epsilon > 0$, it follows that $L^* = 0$ in this case.

Now suppose that a policy improvement operator $I_{x^*,a}$ or I_{x^*} is applied after step N. Note that when a and a' are two actions such that $b_0(a) > b_1(a')$, action a' will never be preferred over action a. Thus when the operator I_{x^*} is applied after step N, the only possible actions it might recommend are those whose $[b_0, b_1]$ intervals intersect that for any a^* such that $b_0(a^*) = \max_a b_0(a)$. But this implies that the entire range of variation in the value for x^* beyond step N can be no larger than the extent to which these $[b_0, b_1]$ intervals can overlap any having maximum b_0. Thus

$$L^* < 2(\gamma L^* + 2\gamma\epsilon)$$

so that

$$(1 - 2\gamma)L^* < 4\gamma\epsilon.$$

When $\gamma < 1/2$, this implies

$$L^* < \frac{4\gamma\epsilon}{1-2\gamma}.$$

Since this is true for all $\epsilon > 0$, it follows that $L^* = 0$ in this case.

Now suppose that the policy improvement operator $I_{x^*,a}$ is applied after step N, with the policy at that moment being a' at the state x. If the interval $[b_0(a), b_1(a)]$ intersects the interval $[b_0(a'), b_1(a')]$, then it is possible for the subsequent policy to be either a or a'. In this case it is not intersection with the topmost interval that determines whether an action might be recommended as the policy, but intersection with the interval for the current policy. Thus the maximum range of variation in the value for state x^* is equal to the sum of the ranges of variation for the lookahead values along all actions. That is,

$$L^* \leq \sum_{a \in A} d(a) < |A|(\gamma L^* + 2\gamma\epsilon),$$

so that

$$(1 - |A|\gamma)L^* < 2|A|\gamma\epsilon.$$

When $\gamma < 1/|A|$, this implies

$$L^* < \frac{2|A|\gamma\epsilon}{1-|A|\gamma}.$$

Since this is true for all $\epsilon > 0$, it follows that $L^* = 0$ in this case as well. □

Since it is common to use a discount parameter having a value reasonably close to 1, this result would not appear to be particularly useful. Nevertheless, it can be shown to have interesting consequences for more practical values of γ when this theory is extended to include the use of multi-step backup operators. We omit further discussion of this idea here.

4.3 Results For Random Sequences

Up to this point we have focused on application of arbitrary sequences of the local backup and policy improvement operators, noting conditions under which such sequences necessarily result in convergence to optimality and also finding specific sequences where such convergence fails. Now we consider randomly generated sequences of these operators and obtain results that are more generally favorable.

Given a finite set of operators $\mathcal{S}$ on $A^X \times \mathcal{R}^X$, and a random process generating an infinite sequence $T_1, T_2, \ldots$ of operators from $\mathcal{S}$, define

$$p_n = \min_{(T_1, T_2, \ldots, T_n) \in \mathcal{S}^n} Pr\{T_n \mid T_1, T_2, \ldots, T_{n-1}\}.$$

We will say that this process is *stochastically always* if $p_n \geq p > 0$ for all n, or, in other words, if, at any step, any operator in $\mathcal{S}$ has conditional probability at least $p > 0$ of being selected, regardless of the previous $n-1$ operator selections. This includes the case when the sequence is obtained by independent random draws from a stationary distribution over $\mathcal{S}$ such that every operator has nonzero probability.

Lemma 4.3.1 (Stochastically Always Lemma) *Let Σ be an infinite sequence of operators from $\mathcal{S}$ generated by a stochastically always process and let Σ' be a given finite sequence of operators from $\mathcal{S}$. Then*

i. Σ is $\mathcal{S}$-forever with probability 1;

ii. Σ' appears as a contiguous subsequence of Σ with probability 1; and

iii. Σ' appears infinitely often as a contiguous subsequence of Σ with probability 1.

Proof. Clearly, (iii) implies (ii). Also, (iii) implies (i) since we can pick as Σ' any single operator $T \in \mathcal{S}$. Therefore, we prove (iii). Let $E_{m,l}$ represent the event that Σ' fails to occur as a contiguous subsequence aligned to lie along positions $m + kN + 1$ through $m + (k+1)N$ in Σ for some k such that $0 \leq k < l$, with $P_{m,l}$ the corresponding probability. Then let

$$E_m = \bigcap_{l=0}^{\infty} E_{m,l},$$

with P_m its corresponding probability. Since the event that Σ' *does* occur in a specific location within Σ is just the intersection of the events that the specific operators within Σ' occur in their respective positions, it is easy to see by induction that

$$P_{m,l} \leq (1 - p^N)^l.$$

But then

$$P_m = \lim_{l \to \infty} (1 - p^N)^l = 0,$$

since $p > 0$. Now let E be the event that there exists some point n beyond which Σ' fails to occur as a contiguous subsequence within Σ. Then

$$E \subset \bigcup_{m=1}^{\infty} E_m,$$

but the countable union of probability zero events has probability zero. Therefore, the probability that no such n exists is 1. $\square$

Theorem 4.3.2 *Suppose a sequence of operators from $\{B_x \mid x \in X\} \cup \{I_{x,a} \mid x \in X, a \in A\}$ is generated by a stochastically always process. Then for any initial policy-value pair (π_0, v_0) the resulting sequence of policy-value pairs converges to optimality with probability 1.*

Proof. By the Boundedness Lemma, if we start with any (π_0, v_0) and apply any finite sequence of these backup and policy improvement operators. the resulting value function can never be outside a particular bounded region $\mathcal{V}$ in $\mathcal{R}^X$. There are many possible finite sequences of local backup and policy improvement operators that map any (π, v) with $v \in \mathcal{V}$ into what we might call the *optimality capture region*

$$\left\{ (\pi, v) \mid \|v - v^*\| < \frac{\delta^{v^*}}{2\gamma} \text{and } \pi \text{ optimal} \right\}.$$

For concreteness, we show how to construct a particular such finite sequence Σ'. Let d equal the maximum distance from v^* to any $v \in \mathcal{V}$. Take Σ' to be a concatenation of m copies of a sequence formed by cycling through the value iteration operators

$$\{B_x I_{x,a_1} I_{x,a_2} \ldots I_{x,a_{|A|}} \mid x \in X\}$$

in some fixed order. Since each such cycle contracts the distance to v^* by a factor of γ, the number m of such cycles required is then determined by the requirement that

$$\gamma^m d < \frac{\delta^{v^*}}{2\gamma}$$

Now, once the optimality capture region is reached, the policies are always optimal, by the Close Enough Lemma, and the v's must necessarily converge to v^* no matter what operators are applied from that point on, as long as all backup operators continue to appear infinitely often. Thus the only way the sequence of operators could fail to give rise to convergence to optimality is for it to fail to contain the finite sequence Σ as a contiguous subsequence anywhere within it or to fail to be forever in the backup operators. But the Stochastically Always Lemma (4.3.1) implies that these are both probability zero events. $\square$

The particular argument given here is sufficient to prove the theorem but may appear to suggest that it might take an astronomically long time before such convergence occurs, since it involves waiting until a particular long finite string of operators is generated, which has an extremely small probability of occurence. It would be useful to establish some better bounds on the expected number of steps of this process before only optimal policies are generated. It seems reasonable to conjecture that this expected number of steps is not astronomically large since it ought to be the case that a large proportion of strings of these operators of a certain length could do an equivalent job.

5 Use of Policy Operators That Examine Single Actions

Now we consider alternative policy improvement operators that do not require comparing two or more lookahead values in order to make changes to a policy. Such operators can be viewed as conforming more directly to the spirit of the actor-critic algorithms that motivated this study, while the policy improvement operators used up to this point can be seen to be more closely related to steps used in standard dynamic programming algorithms. These new operators involve comparing the lookahead value along some candidate action with the current value for the state in question.

For each $x \in X$, $a \in A$, and $v \in \mathcal{R}^X$, define the operator $J^v_{x,a} : A^X \rightarrow A^X$ by

$$J^v_{x,a}\pi(y) = \begin{cases} a & \text{if } y = x \text{ and } L^v(x,a) \geq v(x) \\ \pi(y) & \text{otherwise.} \end{cases}$$

In other words, $J^v_{x,a}\pi$ is the policy obtained from π by replacing $\pi(x)$ by a if a gives at least as high[7] a lookahead value as the current value of the state x. A consequence of this definition is that

[7]Once again, the precise manner of resolving ties turns out to make essentially no difference.

$L^v(x, J^v_{x,a}\pi(x)) \geq v(x)$. Also, for each $x \in X$ and $a \in A$, define the operator $J_{x,a} : A^X \times \mathcal{R}^X \to A^X \times \mathcal{R}^X$ by

$$J_{x,a}(\pi, v) = (J^v_{x,a}\pi, v).$$

5.1 Results For Arbitrary Sequences

Now we consider applying a sequence of operators $T_1, T_2, \ldots$ from $\{B_x \mid x \in X\} \cup \{J_{x,a} \mid x \in X, a \in A\}$ to an initial policy-value pair (π_0, v_0), obtaining a sequence of policy-value pairs where $(\pi_k, v_k) = T_k(\pi_{k-1}, v_{k-1})$ for all $k > 0$. One result concerning such sequences is the following.

Theorem 5.1.1 *If $L^{v_0}(x, \pi_0(x)) \geq v_0(x)$ for all $x \in X$, then asynchronous application of $\{B_x \mid x \in X\} \cup \{B_x J_{x,a} \mid x \in X, a \in A\}$ to (π_0, v_0) yields convergence to optimality.*

Proof. That the resulting sequence of value functions is nondecreasing is a consequence of Lemma 5.1.2, stated and proved below, which is a variant of the Nondecreasing Lemma (4.2.7) in which the $J_{x,a}$ operators appear in place of the $I_{x,a}$ operators. Furthermore, this sequence is bounded, so it must have a limit v_∞, which cannot exceed v^* by the Limit Bounds Theorem (4.2.5). It therefore follows that $v_n \leq v^*$ for all n. We now show that $v_\infty = v^*$.

Suppose that $v_\infty \neq v^*$. Then we can apply the Non-v^* Lemma (4.1.1) to v_∞ to obtain an $x \in X$ and $\varepsilon > 0$ satisfying certain properties to be spelled out below. Pick N so that $n \geq N$ implies

$$\|v_n - v_\infty\| < \varepsilon,$$

and then pick $k \geq N$ so that $T_{k+1} = B_x J_{x,a}$, with a greedy for v_∞ at x. By the Non-v^* Lemma (4.1.1),

$$v^*(x) - L^{v_k}(x,a) \leq |v^*(x) - L^{v_k}(x,a)| < v^*(x) - v_\infty(x) - \varepsilon,$$

so

$$v_k(x) \leq v_\infty(x) < L^{v_k}(x,a).$$

But this implies that the policy at x will have been set to a by $J_{x,a}$, so that the following B_x will be along a. Thus

$$v_{k+1}(x) = L^{v_k}(x,a) > v_\infty(x),$$

contradicting the fact that $v_{k+1}(x)$ is an element of a nondecreasing sequence converging to $v_\infty(x)$. Therefore $v_\infty = v^*$.

All that remains is to show that eventually all policies are optimal. Pick M so that $n \geq M$ implies

$$0 \leq v^*(x) - v_n(x) < \delta^{v^*}_x$$

for all x. Consider the effect of any backup operator applied after this point. By the Suboptimal Lemma (5.1.3), also given below, if this backup is along a suboptimal action, the resulting updated value at x would fail to lie in this interval, contradicting the fact that the sequence converges to v^*. Therefore, we conclude that all backups after this point must be along optimal actions, so all policies must eventually be optimal. □

Lemma 5.1.2 *Suppose that a sequence of operators from* $\{B_x \mid x \in X\} \cup \{J_{x,a} \mid x \in X, a \in A\}$ *is applied to an initial* (π_0, v_0) *satisfying* $L^{v_0}(x, \pi_0(x)) \geq v_0(x)$ *for all* $x \in X$*. Then, for all* x *and* i*,*

i. $L^{v_i}(x, \pi_i(x)) \geq v_i(x)$ *and*

ii. $v_{i+1}(x) \geq v_i(x)$*.*

Proof. The only difference between this lemma and the Nondecreasing Lemma (4.2.7) is the presence of the $J_{x,a}$ operators in place of the $I_{x,a}$ operators, and most of the proof is the same in both cases so we do not repeat the identical portions of the argument here. The only difference is in the proof of the inductive step for (i) for the case when $T_{k+1} = J_{y,a}$ for some y and a, which we now consider.

If $x = y$ and $\pi_{k+1}(x) \neq \pi_k(x)$, it must be because $L^{v_k}(x, \pi_{k+1}(x)) \geq v_k(x)$. In all other cases, $\pi_{k+1}(x) = \pi_k(x)$, so

$$L^{v_k}(x, \pi_{k+1}(x)) = L^{v_k}(x, \pi_k(x)) \geq v_k(x)$$

by the induction hypothesis. Therefore, in all cases,

$$\begin{aligned} L^{v_{k+1}}(x, \pi_{k+1}(x)) &= L^{v_k}(x, \pi_{k+1}(x)) \\ &\geq v_k(x) \\ &= v_{k+1}(x), \end{aligned}$$

which establishes the induction step for (i) when the $J_{x,a}$ operator is applied. □

Lemma 5.1.3 (Suboptimal Lemma) *If* $v \leq v^*$ *and* a *is not optimal at* x*, then* $L^v(x,a) \leq v^* - \delta_x^{v^*}$*.*

Proof. Let a^* denote an optimal action at x. By the monotonicity of lookahead and the definition of $\delta_x^{v^*}$,

$$L^v(x,a) \leq L^{v^*}(x,a) \leq L^{v^*}(x,a^*) - \delta_x^{v^*} = v^*(x) - \delta_x^{v^*}.$$

□

It turns out that Theorem 5.1.1 is false if $\{B_x \mid x \in X\} \cup \{B_x J_{x,a} \mid x \in X, a \in A\}$ is replaced by $\{B_x \mid x \in X\} \cup \{J_{x,a} \mid x \in X, a \in A\}$. This is because we can arrange a sequence of these operators in which the backup B_x is performed only after applying an operator $J_{x,a}$ with a suboptimal, even though an earlier J_{x,a^*} was applied, with a^* optimal. This will yield an nondecreasing sequence of value functions converging to a limit not equal to v^*. The problem is that momentary changes of policy need not be accompanied by corresponding changes in the value function.

It is also interesting to consider certain conditional backup operators that correspond to the J operators in a certain sense. We first define $C_{x,a} : \mathcal{R}^X \rightarrow \mathcal{R}^X$ by

$$C_{x,a}v(y) = \begin{cases} \max\{v(x), L^v(x,a)\} & \text{if } y = x \\ v(y) & \text{otherwise,} \end{cases}$$

and then, in a harmless abuse of notation, we further define $C_{x,a} : A^X \times \mathcal{R}^X \to A^X \times \mathcal{R}^X$ by

$$C_{x,a}(\pi, v) = (C_{x,a}\pi, v).$$

It should always be clear from the context which operator is intended in any particular reference to $C_{x,a}$. Note that, unlike the local backup operators, these operators have no dependency on the current choice of policy.

We first note the following result.

Lemma 5.1.4 (C Convergence) *Any sequence of value functions obtained by asynchronous application of the set of operators $\{C_{x,a} \mid x \in X, a \in A\}$ to an initial $v_0 \le v^*$ converges to v^*.*

Proof. This sequence is clearly nondecreasing and bounded, so it has a limit v_∞. Furthermore, by the Limit Bounds Theorem (4.2.5), this limit cannot exceed v^*. It therefore follows that $v_n \le v^*$ for all n. We now show that $v_\infty = v^*$.

Suppose that $v_\infty \ne v^*$. Then we can apply the Non-v^* Lemma (4.1.1) to v_∞ to obtain an $x \in X$ and $\varepsilon > 0$ satisfying certain properties to be spelled out below. Pick N so that $n \ge N$ implies

$$\|v_n - v_\infty\| < \varepsilon,$$

and then pick $k \ge N$ so that $T_{k+1} = C_{x,a}$, with a greedy for v_∞ at x. By the Non-v^* Lemma (4.1.1),

$$v^*(x) - L^{v_k}(x,a) \le |v^*(x) - L^{v_k}(x,a)| < v^*(x) - v_\infty(x) - \varepsilon,$$

so

$$v_k(x) \le v_\infty(x) < L^{v_k}(x,a).$$

But this implies that

$$v_{k+1}(x) = L^{v_k}(x,a) > v_\infty(x),$$

contradicting the fact that $v_{k+1}(x)$ is an element of a nondecreasing sequence converging to $v_\infty(x)$. Therefore $v_\infty = v^*$. □

We can construct an algorithm roughly analogous to asynchronous value iteration but requiring no comparisons of multiple actions by using the C and J operators in place of the B and I operators. For this algorithm we use compositions of the form $C_{x,a}J_{x,a}$. Note that the effect at state x of applying this composition can be summarized succinctly as follows: If $L^v(x,a) \ge v(x)$ then set the value at x to $L^v(x,a)$ and set the policy at x to a; otherwise, leave them unchanged.

Theorem 5.1.5 *If $v_0(x) < v^*(x)$ for all states x, then asynchronous application of $\{C_{x,a}J_{x,a} \mid x \in X, a \in A\}$ to (π_0, v_0) yields convergence to optimality.*

Proof. By the C Convergence Lemma (5.1.4), the value functions must converge to v^*, so it is only necessary to show that eventually all policies are optimal. First we note that $v_k(x) < v^*(x)$ for all k and x. This follows easily by induction on k, using the Less Than v^* Lemma (2.5.3). Consider a fixed state x. Since $v_n(x) \uparrow v^*(x)$, there are an infinite number of indices k where $v_{k+1}(x) > v_k(x)$, so for each of these it must be the case that $L^{v_k}(x,a) > v_k(x)$ for some a, so the policy at x will have been set to some such a at each such step. Since $v_n(x) \to v^*(x)$ there exists M_x such that $k \geq M_x$ implies $v_k(x) > v^* - \delta_x^{v^*}$. Then set $M = \max_x M_x$. By the Suboptimal Lemma (5.1.3), the policy cannot be changed to a suboptimal action after this point since the resulting value function would then satisfy $v(x) \leq v^* - \delta_x^{v^*}$, a contradiction. Since the operator J_{x,a^*}, where a^* is optimal at x, must appear beyond this point, all subsequent policy actions for x must then be optimal, and thus all policies are eventually optimal. $\square$

What about asynchronous application of $\{C_{x,a} \mid x \in X, a \in A\} \cup \{J_{x,a} \mid x \in X, a \in A\}$? It is possible to construct counterexamples to show that optimal policies need not arise in all cases, but there are interesting cases when convergence to optimality is guaranteed for these operators, as given by the following result.

Theorem 5.1.6 *If $v_0(x) < v^*(x)$ for all states x and there is a unique optimal action at each state, then asynchronous application of $\{C_{x,a} \mid x \in X, a \in A\} \cup \{J_{x,a} \mid x \in X, a \in A\}$ to (π_0, v_0) yields convergence to optimality.*

Proof. We already know that the value functions must converge to v^*, so we now check that eventually all policies are optimal. First we note that $v_k(x) < v^*(x)$ for all k and x. This follows easily by induction on k, using the Less Than v^* Lemma (2.5.3). Consider a fixed state x and let a^* be the unique optimal action at x. Since $v_n(x) \uparrow v^*(x)$, there are an infinite number of indices k where $v_{k+1}(x) > v_k(x)$. Furthermore, there exists M_x such that $k \geq M_x$ implies $v_k(x) > v^* - \delta_x^{v^*}$. Thus, there are infinitely many indices $k \geq M_x$ where $v_{k+1}(x) = L^{v_k}(x,a) > v_k(x)$ for some a, and it follows from the Suboptimal Lemma (5.1.3) that a must be optimal for x, so $a = a^*$. In other words, if $k \geq M_x$ and $v_{k+1}(x) > v_k(x)$, then $T_{k+1} = C_{x,a^*}$ and $v_{k+1}(x) = L^{v_k}(x,a^*)$. Also, the existence of such a k is guaranteed.

Given such a k, pick $m > k$ so that $T_{m+1} = J_{x,a^*}$ and then let l be the largest index less than m such that $v_{l+1}(x) > v_l(x)$. Clearly $l \geq k \geq M_x$, so $v_{l+1} = L^{v_l}(x,a^*)$. Also, it follows from the definition of l that $v_m(x) = v_{l+1}(x)$. Therefore,

$$L^{v_m}(x,a^*) = L^{v_{l+1}}(x,a^*) > L^{v_l}(x,a^*) = v_{l+1}(x) = v_m(x),$$

which implies that $\pi_{m+1}(x) = a^*$. Furthermore, by the Suboptimal Lemma (5.1.3), for any suboptimal action a and any $k \geq M_x$, we have

$$L^{v_k}(x,a) \leq v^* - \delta_x^{v^*} < v_k,$$

so application of $J_{x,a}$ for suboptimal a cannot alter the policy beyond this point. Since x was arbitrary, there is a point far enough out in the sequence when the entire policy becomes and remains optimal. $\square$

Note that in these proofs involving the C and J operators, the key issue is to insure that for each x an operator J_{x,a^*}, with a^* optimal, is triggered (i.e., leads to a policy change) at least once after the value function gets close enough to v^* that the Suboptimal Lemma (5.1.3) applies. It is possible to construct a counterexample to optimal convergence under asynchronous application of the C and J operators (still assuming that $v_0(x) < v^*(x)$ for all x) where the initial choice of suboptimal policy action is never changed because an appropriate J operator is never triggered. As the previous theorem shows, this requires two or more optimal actions at a state. To generate such a counterexample, consider a Markov environment in which a particular state x has exactly two optimal actions, a_1^* and a_2^*, and let the initial policy be set to a third, suboptimal action. The idea is to always precede the application of J_{x,a_i^*} by updating at other states followed by C_{x,a_j^*}, with $i \neq j$, so that the value at x after application of C_{x,a_j^*} is strictly larger than the lookahead along a_i^*, thus guaranteeing that J_{x,a_i^*} is never triggered.

5.2 Results For Random Sequences

Just as with the B_x and $I_{x,a}$ operators, we can obtain probability one convergence under less restrictive conditions if we assume that the operators are selected randomly in such a way that every operator has a probability bounded away from zero of being selected on any step. Recall that earlier we introduced the term *stochastically always* to describe such a process of operator selection.

Before considering the main results of this subsection, we define quantities analogous to the δ_x^v and δ^v used earlier in the Close Enough Lemma (3.6.2). Then we prove a somewhat analogous lemma involving these new quantities that will be used to prove our first main result.

For any $v \in \mathcal{R}^X$, let

$$\nu_x^v = \min\left[\{|L^v(x,a) - L^v(x,a')| \mid a, a' \in A\} - \{0\}\right],$$

and let

$$\nu^v = \min_{x \in X} \nu_x^v.$$

That is, ν_x^v is the smallest nonzero absolute difference between the lookahead value at state x along any two actions, while ν^v is the smallest such nonzero difference taken across all states.

Lemma 5.2.1 (Close To v^π Lemma) *Let $\pi \in A^X$, $x \in X, a \in A$. Let $v \in \mathcal{R}^X$ with $\|v - v^\pi\| < \nu_x^{v^\pi}/(1+\gamma)$. Then $L^v(x,a) \geq v(x)$ implies $L^{v^\pi}(x,a) \geq v^\pi(x)$.*

Proof.

$$\begin{aligned}
v^\pi(x) - L^{v^\pi}(x,a) &= v^\pi(x) - v(x) + v(x) - L^{v^\pi}(x,a) \\
&\leq v^\pi(x) - v(x) + L^v(x,a) - L^{v^\pi}(x,a) \\
&\leq |v^\pi(x) - v(x)| + |L^v(x,a) - L^{v^\pi}(x,a)| \\
&< \frac{\nu_x^{v^\pi}}{1+\gamma} + \frac{\gamma \nu_x^{v^\pi}}{1+\gamma} \\
&= \nu_x^{v^\pi}.
\end{aligned}$$

Since $v^\pi(x) = L^{v^\pi}(x, \pi(x))$, it follows that

$$L^{v^\pi}(x,\pi(x)) - L^{v^\pi}(x,a) < \nu_x^{v^\pi}.$$

But then, if $L^{v^\pi}(x, \pi(x)) > L^{v^\pi}(x, a)$, we would have

$$|L^{v^\pi}(x, \pi(x)) - L^{v^\pi}(x, a)| = L^{v^\pi}(x, \pi(x)) - L^{v^\pi}(x, a) < \nu_x^{v^\pi},$$

and since $\nu_x^{v^\pi}$ is the minimum nonzero value of $|L^{v^\pi}(x, a) - L^{v^\pi}(x, a')|$ for all a and a', this would imply that that $L^{v^\pi}(x, \pi(x)) - L^{v^\pi}(x, a) = 0$, a contradiction. Therefore

$$v^\pi(x) = L^{v^\pi}(x, \pi(x)) \le L^{v^\pi}(x, a).$$

□

Theorem 5.2.2 *Suppose a sequence of operators from* $\{B_x \mid x \in X\} \cup \{J_{x,a} \mid x \in X, a \in A\}$ *is generated by a stochastically always process. Then for any initial policy-value pair* (π_0, v_0) *the resulting sequence of policy-value pairs converges to optimality with probability 1.*

Proof. The argument here has some general similarities to that used earlier to prove the corresponding result with the B_x and $I_{x,a}$ operators. Here we use something resembling policy iteration in that it involves alternately applying a (finite) composition of backup operators to compute a close approximation to the return from the current policy and then applying a certain composition of $J_{x,a}$ operators to improve the policy.

To describe this a little more precisely, let the states be enumerated in some order $x_1, x_2, \ldots, x_{|X|}$, and then let B denote the composition $B_{x_1} B_{x_2} \cdots B_{x_{|X|}}$. Also, for each action a let J_a denote the composition $J_{x_1,a} J_{x_2,a} \cdots J_{x_{|X|},a}$. Finally, let the actions be enumerated in some order $a_1, a_2, \ldots, a_{|A|}$ and let T_N denote the composition $J_{a_1} B^N J_{a_2} B^N \cdots J_{a_{|A|}} B^N$ for any nonegative integer N. Then we show below that for any given (π_0, v_0) there are nonnegative integers M and N such that T_N^M, the M-fold composition of T_N with itself, has the property that the appearance of the corresponding finite sequence of operators anywhere within the infinite sequence will drive the resulting policy and value functions into the optimality capture region

$$\left\{(\pi, v) \mid \|v - v^*\| < \delta^{v^*} \text{and } \pi \text{ optimal}\right\}.$$

Once this happens, it follows from the Suboptimal Lemma (5.1.3) that no $J_{x,a}$ operator can ever make the policy suboptimal, and as long as every backup operator continues to appear infinitely often, the value functions must then converge to v^*. Since the Stochastically Always Lemma (4.3.1) guarantees with probability 1 both the appearance of the desired finite subsequence within the infinite sequence and the infinitely often appearance of every backup operator, the desired result then follows.

Thus it remains to demonstrate the finite composition T_N^M that maps into the optimality capture region given above. Note that for each suboptimal stationary policy $\pi \in A^X$ there is a corresponding $\varepsilon_\pi > 0$ given by the Non-v^* Lemma (4.1.1) applied to $v^\pi \neq v^*$. For any stationary policy π, we then define

$$\eta_\pi = \begin{cases} \delta^{v^*} & \text{if } \pi \text{ is optimal} \\ \min\left(\varepsilon_\pi, \frac{\nu^{v^\pi}}{1+\gamma}\right) & \text{otherwise,} \end{cases}$$

Now by the Boundedness Lemma (4.2.3), if we start with any (π_0, v_0) and apply any finite sequence of B_x and $J_{x,a}$ operators. the resulting value function can never be outside a particular bounded region $\mathcal{V}$ in $\mathcal{R}^X$. Thus, consider any (π, v) such that $v \in \mathcal{V}$. By the Sequential Backup Contraction Lemma (3.5.3) there is an integer N_π such that, for $k \geq N$, applying B^{N_π} to (π, v) results in a value function whose distance to v^π is less than η_π. We thus let $N = \max_{\pi \in A^X} N_\pi$, which is well-defined since A^X is finite, and it follows that applying B^N to any (π, v) with $v \in \mathcal{V}$ results in a value function whose distance to v^π is less than η_π.

Thus any time B^N is applied we are guaranteed that the resulting value function is within an η_π-neighborhood of v^π, for the current policy π. Consider the effect of now applying J_a for some a. If the policy π is already optimal, then by the Suboptimal Lemma (5.1.3) it must remain optimal. For the remainder of the proof, assume π is not optimal. Then the application of J_a either results in the same policy or there is at least one x for which the policy is changed to a because $L^v(x,a) \geq v(x)$. In this latter case, the Close To v^π Lemma (5.2.1) implies that $L^{v^\pi}(x,a) \geq v^\pi(x)$. But then the Policy Improvement Proposition (Proposition 3.6.1) implies that any such change must result in a policy π' such that $v^{\pi'}(x) \geq v^\pi(x)$. Thus application of the composition $J_a B^N$ will always result in a policy no worse than the current policy. In addition, we now show that applying T_N, the composition of all the $J_a B^N$ operators, to a suboptimal policy must result in a strict policy improvement. To see this, suppose that it does not. The only way this can happen is for all the value functions resulting from application of each $J_a B^N$ in turn to be close to the same v^π, since the policies cannot get worse. Thus just before any J_a is applied, the value function v is in an ε_π-neighborhood of $v^\pi \neq v^*$. Therefore the Non-v^* Lemma (4.1.1) implies that there exists x and a such that

$$|v^*(x) - L^v(x,a)| < |v^*(x) - v^\pi(x)| - \varepsilon_\pi.$$

Since $v^\pi(x) \leq v^*(x)$, this means that

$$v^*(x) - L^v(x,a) \leq |v^*(x) - L^v(x,a)| < v^*(x) - v^\pi(x) - \varepsilon_\pi,$$

so

$$L^v(x,a) > v^\pi(x) + \varepsilon_\pi.$$

But it is also true that

$$L^v(x,a) - L^{v^\pi}(x,a) \leq |L^v(x,a) - L^{v^\pi}(x,a)| < \gamma\varepsilon_\pi,$$

so

$$\begin{aligned} L^{v^\pi}(x,a) &> L^v(x,a) - \gamma\varepsilon_\pi \\ &> v^\pi(x) + \varepsilon_\pi - \gamma\varepsilon_\pi \\ &> v^\pi(x). \end{aligned}$$

But by the Policy Improvement Proposition (3.6.1) this means that the policy resulting from application of $J_{x,a}$ for this particular x and a is indeed a strict improvement, contradicting our assumption that no strict improvement occurs under application of T_N. Therefore, each application of T_N results in a strict policy improvement. Since there are $|A|^{|X|}$ distinct policies, it follows that applying T_N^M for $M = |A|^{|X|}$ takes any (π, v) with $v \in \mathcal{V}$ into the desired optimality capture region. $\square$

Theorem 5.2.3 *Suppose a sequence of operators from $\{C_{x,a} \mid x \in X, a \in A\} \cup \{J_{x,a} \mid x \in X, a \in A\}$ is generated by a stochastically always process. Then for any initial policy-value pair (π_0, v_0) such that $v_0(x) < v^*(x)$ for all states x, the resulting sequence of policy-value pairs converges to optimality with probability 1.*

Proof. By the Stochastically Always Lemma (4.3.1) we know that with probability 1 every $C_{x,a}$ operator will appear infinitely often, in which case the C Convergence Lemma (5.1.4) guarantees that the value functions will approach v^*. Thus it remains to show that, with probability 1, eventually all policies are optimal.

With probability 1, there exists N such that $n \geq N$ implies

$$v_n \in \{v \mid \|v - v^*\| < \delta^{v^*}\}.$$

Furthermore, we know that even beyond this point the value functions must continue to increase (with probability 1) since $v_i(x) < v^*(x)$ for all i by the Less Than v^* Lemma (2.5.3). By the Suboptimal Lemma (5.1.3), such an increase in the value function at x must be caused by application of some $C_{x,a}$ with a optimal. If for each x we can guarantee that there is at least one $k > N$ such that $v_k(x) > v_{k-1}(x)$ and this change in value function is accompanied by setting the policy to the corresponding action, we are done.

One way for this change in value function to be accompanied by a corresponding setting of the policy is to apply $J_{x,a}$ immediately after applying $C_{x,a}$ (rather than before as considered earlier). This is because if $v' = C_{x,a}v$ and $v'(x) > v(x)$, then

$$v'(x) = L^v(x,a) \leq L^{v'}(x,a)$$

by the monotonicity of lookahead, so immediate application of $J_{x,a}$ at this point will cause the policy at x to be set to a.

Therefore, consider the set $\mathcal{K}$ of all the indices k such that $k > N$ and $v_k(x) > v_{k-1}(x)$ and disregard the probability zero event that $\mathcal{K}$ is finite. The corresponding operator T_k for each of the infinitely many such k values is then some $C_{x,a}$ with a optimal. Consider for each such k the next operator T_{k+1}. If at least one of these is the corresponding $J_{x,a}$, with a the same action as in the preceding operator $C_{x,a}$, then the policy at x will become and remain optimal. For any one of these values of k the conditional probability that the next operator is not this particular $J_{x,a}$ is less than or equal to $1-p < 1$, where $p > 0$ is the lower bound guaranteed by the stochastically always condition. Therefore, the probability that for all $k \in \mathcal{K}$ the next operator is not the corresponding J operator is less than or equal to

$$\prod_{k \in \mathcal{K}} (1-p) = 0,$$

since K is an infinite set. Thus we conclude that with probability 1 there will be at least one composition $J_{x,a}C_{x,a}$ appearing beyond step N at which the value function at x undergoes a strict increase, and at which, therefore, the policy will be set to the optimal action a. Since this is true for all x, it follows that all policies are eventually optimal with probability 1. $\square$

6 Other Results

There are certainly many other results one might try to obtain involving the operators defined in this paper, and there may be other related operators that are useful to study as well. One particular combination of operators already defined here that we have not considered up to this point are the $C_{x,a}$ operators in conjunction with the $I_{x,a}$ operators, for which the following result holds.

Theorem 6.0.4 *If $v_0 \le v^*$, then asynchronous application of $\{C_{x,a} \mid x \in X, a \in A\} \cup \{I_{x,a} \mid x \in X, a \in A\}$ to (π_0, v_0) yields convergence to optimality.*

Proof. We have already shown that the value functions must converge to v^*. Thus beyond some point in this sequence, all the value functions will be sufficiently close to v^* that the Close Enough Lemma (3.6.2) applies. Beyond this point, application of the operator I_{x,a^*}, where a^* is optimal at x will either change the policy to a^* or leave it unchanged, in which case the Close Enough Lemma implies that it was already optimal. Furthermore, the Close Enough Lemma (3.6.2) also implies that any subsequent application of $I_{x,a}$, with a suboptimal, cannot change the policy. Therefore, eventually all policies become and remain optimal. $\square$

Singh and Gullapalli (1993) have recently introduced another operator distinct from those we have considered here and established an interesting result involving its use in conjunction with the $I_{x,a}$ operators. For completeness, we describe this operator here and include a proof of this result.

For each $x \in X$ and $\pi \in A^X$, define the operator $D_x^\pi : \mathcal{R}^X \to \mathcal{R}^X$ by

$$D_x^\pi v(y) = \begin{cases} \max(v(x), L^v(x, \pi(x))) & \text{if } y = x \\ v(y) & \text{otherwise.} \end{cases}$$

That is, $D_x^\pi v$ is the value function obtained from v by replacing $v(x)$ by the one-step lookahead value along π, but only if this does not decrease $v(x)$. Also, for each $x \in X$, define the operator $D_x : A^X \times \mathcal{R}^X \to A^X \times \mathcal{R}^X$ by

$$D_x(\pi, v) = (\pi, D_x^\pi v).$$

Singh and Gullapalli have called D_x a *single-sided policy evaluation operator*, but we can think of it as a *local backup operator with a ratchet.*

Singh and Gullapalli have proved the following result.

Theorem 6.0.5 *If $v_0 \le v^*$, then asynchronous application of $\{D_x \mid x \in X\} \cup \{I_{x,a} \mid x \in X, a \in A\}$ to (π_0, v_0) yields convergence to optimality.*

Proof. The sequence of value functions is clearly nondecreasing and bounded, so it has a limit v_∞, which cannot exceed v^*. Suppose that $v_\infty \ne v^*$. Then we can apply the Non-v^* Lemma (4.1.1) to v_∞ to obtain an $x \in X$ and $\varepsilon > 0$ satisfying certain properties to be spelled out below. Pick N so that $n \ge N$ implies

$$\|v_n - v_\infty\| < \min\left(\frac{\delta_x^{v_\infty}}{2\gamma}, \varepsilon\right),$$

and then pick $k \geq N$ so that $T_{k+1} = I_{x,a}$, with a greedy for v_∞. Let $l > k$ be such that $T_{l+1} = D_x$. By the Close Enough Lemma (3.6.2), $\pi_l(x)$ must be a greedy action at x for v_∞ since it must have looked no worse than a using a value function within $\delta_x^{v_\infty}/(2\gamma)$ of v_∞. Thus the Non-v^* Lemma (4.1.1) allows us to conclude that

$$v^*(x) - L^{v_l}(x, \pi_l(x)) \leq |v^*(x) - L^{v_l}(x, \pi_l(x))| < v^*(x) - v_\infty(x) - \varepsilon,$$

so

$$L^{v_l}(x, \pi_l(x)) > v_\infty(x) + \varepsilon > v_\infty(x) \geq v_l(x).$$

But then

$$v_{l+1}(x) = L^{v_l}(x, \pi_l(x)) > v_\infty(x),$$

a contradiction. Therefore $v_\infty = v^*$.

Furthermore, that the policies must eventually all be optimal follows from the Close Enough Lemma (3.6.2), since eventually all v_n will be within $\delta^{v^*}/(2\gamma)$ of v^*. Thus any I_{x,a^*} operator, with a^* optimal at x, applied beyond this point must set the policy at x to an optimal action, and any changes in the policy at x occuring after this can only replace one optimal action with another. $\square$

7 Summary and Discussion

We have presented results here that address questions of convergence to optimality when the policy and value function are updated completely asynchronously across individual states (or, in some cases, state-action pairs). The main results are that, while there are cases when such convergence may fail, there are a number of reasonably general situations in which such convergence is guaranteed. While these results provide immediately applicable alternative strategies for conventional dynamic programming, they are not as directly applicable to the understanding of actor-critic learning systems because of the simplifying assumptions made in their derivation. However, it is interesting to note that there is at least one interesting class of algorithms to which these results at least come close to being applicable in their current form—namely, a certain version of the Dyna-PI form of Sutton's (1990) *relaxation planning* algorithms, in which learning occurs while arbitrarily selected state-action pairs are explored in a model that is gradually acquired on line. For this algorithm, it must be assumed that what Singh (1993) has termed *full backups* are used, requiring the explicit use of the transition and reward probabilities in applying any B_x operator.

In fact, although we omit the details here, it is not hard to extend the theory contained in this paper to the case when the underlying model (i.e., transition and reward probabilities) is itself being estimated. In particular, consider the case when all such reward and transition probability estimates converge to their true values with probability 1. For example, in estimating transition probabilities, it is natural to maintain the obvious counters and form probability estimates by using the appropriate ratios. As long as all transitions are sampled infinitely often, the strong law of large numbers applies and probability 1 convergence to the true transition probabilities is guaranteed. While the convergence results given in this paper assume that the true model is used, it is not hard to show[8] that application of operator sequences yielding convergence to optimality under this assumption may be intermixed freely with a model estimation process that converges to the true model with probability 1, and the overall result is convergence to optimality (for the true model) with probability 1.

An important next step in this theory, especially for questions of on-line learning in stochastic environments, is to perform a corresponding analysis when sample backups, rather than full backups, are used. In this case, the backup operation should take the form

$$v(x) \leftarrow (1-\alpha)v(x) + \alpha(r + \gamma v(y)),$$

where r is the random reward received and y the random next state following a particular visit to state x. This differs from the form of backup considered throughout this paper both in the random nature of the immediate reward and next state actually used and in the presence of the underrelaxation parameter $\alpha \in (0,1)$. Furthermore, it would seem that a corresponding way of more smoothly transitioning between policies would also be necessary. Since the space of actions is assumed to be discrete, this can be done by considering randomized policies, so that the policy can be identified with a point in a certain simplex. In this case policy updates can correspond to small steps in this simplex, as in stochastic learning automata (Narendra & Thathatchar, 1989). This is actually the way the actor works in the ASE/ACE architecture of Barto, Sutton, and Anderson (1983). Perhaps the theoretical tools recently developed by Jaakola, Jordan, and Singh (1993) can be used or extended to provide an analysis of such algorithms.

A separate question, interesting both from the learning system perspective and also from a more general asynchronous dynamic programming perspective, is the rate of convergence of algorithms of the type investigated here.

8 References

Barto, A. G., Bradtke, S. J., & Singh, S. P. (1991). *Real-time learning and control using asynchronous dynamic programming* (COINS Technical Report 91-57). Department of Computer Science, University of Massachusetts, Amherst, MA.

Barto, A. G., Sutton, R. S., & Anderson, C. W. (1983). Neuronlike elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man, and Cybernetics, 13*, 835-846.

Barto, A. G., Sutton, R. S., & Watkins, C. J. C. H. (1989). *Learning and sequential decision making* (COINS Technical Report 89-95). Amherst, MA: University of Massachusetts, Department of Computer and Information Science.

Bertsekas, D. P. (1987). *Dynamic Programming: Deterministic and Stochastic Models.* Englewood Cliffs, NJ: Prentice Hall.

[8] I thank Vijay Gullapalli for first bringing this issue, along with some limited results along these lines, to my attention. Discussions with Peter Dayan have helped convince me of the the validity of the simpler result described here.

Bertsekas, D. P. & Tsitsiklis, J. N. (1989). *Parallel and Distributed Computation: Numerical Methods*. Englewood Cliffs, NJ: Prentice Hall.

Dayan, P. (1992). The convergence of TD(λ) for general λ. *Machine Learning, 8*, 341-362.

Holland, J. H. (1986). Escaping brittleness: The possibility of general-purpose learning algorithms applied to rule-based systems. In: R. S. Michalski, J. G. Carbonell, & T. M. Mitchell (Eds.) *Machine Learning: An Artificial Intelligence Approach, Volume II.* Los Altos, CA: Morgan Kaufmann.

Jaakola, T., Jordan, M. I., & Singh, S. P. (1993). *On the convergence of stochastic iterative dynamic programming algorithms* (Working Paper). Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology.

Moore, A. W. & Atkeson, C. G., (1992) Memory-based reinforcement learning: Converging with less data and less real time, In: S. J. Hanson, J. D. Cowan, & C. L. Giles (Eds.) *Advances in Neural Information Processing Systems 5.* San Mateo, CA: Morgan Kaufmann.

Narendra, K. S. & Thathatchar, M. A. L. (1989). *Learning Automata: An Introduction.* Englewood Cliffs, NJ: Prentice Hall.

Peng, J. & Williams, R. J. (1992). Efficient learning and planning within the Dyna framework, *Proceedings of the Second International Conference on Simulation of Adaptive Behavior*, Honolulu, HI.

Puterman, M. L. & Shin, M. C. (1978). Modified policy iteration algorithms for discounted Markov decision problems. *Management Science, 24*, 1127-1137.

Ross, S. 1983). *Introduction to Stochastic Dynamic Programming.* New York: Academic Press.

Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development, 3*, 210-229. Reprinted in: E. A. Feigenbaum and J. Feldman (Eds.) (1963). *Computers and Thought.* New York: McGraw-Hill.

Singh, S. P. (1993). *Learning Control in Dynamic Environments.* Ph.D. Dissertation, Department of Computer Science, University of Massachusetts, Amherst, MA.

Singh, S. P. & Gullapalli, V. (1993). Asynchronous policy iteration with single-sided updates (Working Paper). Department of Computer Science, University of Massachusetts, Amherst, MA.

Sutton, R. S. (1984). *Temporal credit assignment in reinforcement learning.* Ph.D. Dissertation, Department of Computer and Information Science, University of Massachusetts, Amherst (also COINS Technical Report 84-02).

Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. *Machine Learning, 3*, 9-44.

Sutton, R. S. (1990). Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. *Proceedings of the Seventh International Conference in*

Machine Learning, 216-224.

Sutton, R. S. (1991). Planning by incremental dynamic programming. *Proceedings of the 8th International Machine Learning Workshop.*

Watkins, C. J. C. H. (1989). *Learning from delayed rewards.* Ph.D. Dissertation, Cambridge University, Cambridge, England.

Watkins, C. J. C. H. & Dayan, P. (1992). Q-learning. *Machine Learning, 8*, 279-292.

Werbos, P. J. (1987). Building and understanding adaptive systems: A statistical/numerical approach to factory automation and brain research. *IEEE Transactions on Systems, Man, and Cybernetics, 17*, 7-20.

Williams, R. J. & Baird, L. C., III (1990). A mathematical analysis of actor-critic architectures for learning optimal controls through incremental dynamic programming. *Proceedings of the Sixth Yale Workshop on Adaptive and Learning Systems*, August 15-17, New Haven, CT, 96-101.

Witten, I. H. (1977). An adaptive optimal controller for discrete-time Markov environments. *Information and Control, 34*, 286-295.

A Appendix

A.1 List of Symbols

The following list gives the main symbols used in this paper, the consistent way each is used, and the page number where each first appears or is defined.

Symbol	Meaning	Page
a, a', a^*, α	action	3
A	action set	3
B_x^π	local backup operator	6
B_x	local backup operator	6
B_x^a	local backup operator	22
$c_{\mathcal{S}}(\Sigma)$	number of disjoint $\mathcal{S}$-exhaustive contiguous subsequences in Σ	7
$c_X(\Sigma)$	number of disjoint X-exhaustive contiguous subsequences in Σ	8
$C_{x,a}$	local conditional backup operator	32
γ	discount factor	4
D_x^π	local backup operator with ratchet	39
D_x	local backup operator with ratchet	39
δ_x^v	smallest nonzero difference between largest lookahead value at state x and lookahead for any other action	11
δ^v	minimum of δ_x^v across all states	11
ε	a positive quantity arising from the Non-v^* Lemma	15
$f(x, a)$	random successor of state x under action a	3
$I_{x,a}^v$	local policy improvement operator	6
$I_{x,a}$	local policy improvement operator	7
I_x^v	local policy improvement operator	7
I_x	local policy improvement operator	7
$J_{x,a}$	local conditional policy improvement operator	30
$L^v(x, a)$	one-step lookahead value	5
ν_x^v	smallest nonzero absolute difference between lookahead values along any two actions at state x	35
ν^v	minimum of ν_x^v across all states	35
p_{xy}^a	probability of transition from state x to state y under action a	3
π	policy	3
$r(x, a)$	random immediate reward for action a at state x	3
$R(x, a)$	expected value of $r(x, a)$	3
$\mathcal{R}$	real numbers	4
$\mathcal{S}$	a set of operators	7
Σ	a sequence of operators	7
Σ_n	finite subsequence of Σ consisting of first n operators	8
T_k	kth member of a sequence of operators	7
v, v'	value function	4
v^π	return for policy π	4
v^*	return for optimal policy	4
$v^\#$	return for pessimal policy	23
X	state set	3
x, y	state	3
$\leq, \geq$	partial order relation on value functions	4
$\Vert - \Vert$	max norm	6

A.2 List of Nonstandard Terms

The following is a list of terms used in this paper that are not necessarily in common use in this field, each with the page number where it is defined.

A.3 List of Mathematical Results