Chapter 7

Value and Policy Iteration

[1]For infinite horizon problems, we need to replace our basic computational tool, the DP algorithm, which we used to compute the optimal cost and policy for finite horizon problems. We have already encountered in chapter 6 the value iteration (VI) algorithm, which is similar to the DP algorithm and computes a sequence of functions converging toward the optimal cost, solution of Bellman's equation. In the first part of this chapter, we discuss the VI algorithm and some of its refinements in more details. Then we describe another basic method to solve Bellman's equation, policy iteration (PI), which in contrast to VI generates a sequence of improving *policies*. Throughout this chapter we consider the simple case of discounted cost problems with bounded cost per stage.

7.1 Value Iteration

We consider the infinite horizon discounted cost problem with bounded cost per stage. For this problem, we have seen in chapter 6, lemma 6.3.4 that starting from a bounded cost function J, the iterates $T^k J$ converge uniformly to the optimal cost J^*. This algorithm is called the *value iteration algorithm*, or sometimes *successive approximation*. In this section we discuss this algorithm in more details as well as some of its variations.

Value Iteration with Accelerated Convergence

VI is an iterative method which in general only converges asymptotically to the value function, even if the state space is finite. In practice it is often very useful to have methods to accelerate the convergence of this algorithm.

Proposition 7.1.1 (VI with error bounds)**.** *For every $J \in B(\mathsf{X}, \mathbb{R})$ and index k, we have the lower bounds*

$$J^* \geq T^{k+1} J + \underline{c}_{k+1} e \geq T^k J + \underline{c}_k e \geq \dots$$

[1]This version: October 20 2009

and the upper bounds

$$J^* \leq T^{k+1}J + \bar{c}_{k+1}e \leq T^k J + \bar{c}_k e \leq \dots$$

where

$$\underline{c}_k = \frac{\alpha}{1-\alpha} \min_{x \in \mathsf{X}} [T^k J(x) - T^{k-1}J(x)] \tag{7.1}$$

$$\bar{c}_k = \frac{\alpha}{1-\alpha} \max_{x \in \mathsf{X}} [T^k J(x) - T^{k-1}J(x)]. \tag{7.2}$$

Note that the constants (7.1) and (7.2) can be computed during the standard value iteration algorithm. We still follow the same algorithm computing the iterates $T^k J$, but we use the additional bounds to detect convergence more rapidly.

Proof. We show the inequalities for the lower bounds, the upper bounds being similar. Because $J \in B(\mathsf{X}, \mathbb{R})$ and we assume a bounded cost per state, we have that $TJ \in B(\mathsf{X}, \mathbb{R})$. Next let $\underline{\gamma} = \min_{x \in \mathsf{X}}[TJ(x) - J(x)]$. We have therefore $J + \underline{\gamma}e \leq TJ$. Hence by the offset property and the monotonicity property of lemmas 6.3.1 and 6.3.2, we get $TJ + \alpha\underline{\gamma}e \leq T^2 J$. Combining this inequality with the preceding one, we obtain

$$J + (1+\alpha)\underline{\gamma}e \leq TJ + \alpha\underline{\gamma}e \leq T^2 J. \tag{7.3}$$

Note that $\underline{c}_1 = \alpha\underline{\gamma}/(1-\alpha)$. Repeating this process infinitely many times, we obtain

$$J + \frac{\underline{\gamma}}{1-\alpha}e \leq TJ + \underline{c}_1 e \leq T^2 J + \alpha\underline{c}_1 e \leq \dots \leq J^*. \tag{7.4}$$

Now replace J by $T^k J$ to get the first lower bound (from the second term)

$$T^{k+1}J + \underline{c}_{k+1}e \leq J^*.$$

From the last inequality in (7.3) we have

$$\alpha\underline{\gamma} \leq \min_{x \in \mathsf{X}}[T^2 J(x) - TJ(x)]$$

$$\text{and so} \quad \alpha\underline{c}_1 \leq \underline{c}_2.$$

Hence $TJ + \underline{c}_1 e \leq T^2 J + \underline{c}_2 e$, from the second inequality in (7.4) and the second lower bound follows by replacing J by $T^{k-1}J$. □

Termination Issues

Consider a cost function $J : \mathsf{X} \to \mathbb{R}$. Consider a policy μ which achieves the minimum in the equation $T_\mu J = TJ$. Now consider the first bound from lemma 7.1.1, for J^* and J_μ (as usual, results for the latter case are obtained from those for J^* by constraining $\mathsf{U}(x) = \{\mu(x)\}$)

$$\underline{c}_1 e \leq J^* - TJ \leq \bar{c}_1 e$$

$$\underline{c}_1 e \leq J^*_\mu - T_\mu J \leq \bar{c}_1 e.$$

Now since $T_\mu J = TJ$, we get

$$\|J^* - J_\mu\|_\infty \leq \bar{c}_1 - \underline{c}_1. \tag{7.5}$$

In practice, we stop VI once the difference $(\bar{c}_k - \underline{c}_k)$ is sufficiently small. Then we can take as an approximation of the optimal cost the median

$$\tilde{J}_k = T^k J + \frac{\bar{c}_k + \underline{c}_k}{2} e, \tag{7.6}$$

or the average

$$\hat{J}_k = T^k J + \left[\frac{\alpha}{n(1-\alpha)} \sum_{i=1}^{n} \Big(T^k J(i)) - T^{k-1} J(i) \Big) \right] e. \tag{7.7}$$

Then, using (7.5) with c_1 replaced by c_k, we have an estimate on the suboptimality of the policy μ achieving the minimum in the calculation of $T^k J$ (i.e., $T_{\mu_k} T^{k-1} J = T^k J$). Moreover If the state and control spaces are finite, there is a finite number of possible stationary policies. In particular there is some $\epsilon > 0$ such that any stationary policy which verifies $\|J_\mu - J^*\| < \epsilon$ is necessary optimal. The same error bounds (7.5) again then show that after a sufficiently large number of iterations, we obtain an optimal policy.

Computations in Finite State Spaces

Assume that the sets X, U and W are all finite. Without loss of generality, assume that $\mathsf{X} = \{1, \ldots, n\}$, and consider the controlled Markov chain formulation, using the transition probabilities $p_{ij}(u)$ and the cost function $c(i, u, j)$ for the cost of using u in state i and moving to state j. Next define the expected cost per state

$$c(i, u) = \sum_{j \in \mathsf{X}} p_{ij}(u) c(i, u, j).$$

Then the DP operators T and T_μ can be written

$$(TJ)(i) = \min_{u \in \mathsf{U}(i)} \left[c(i, u) + \alpha \sum_{j=1}^{n} p_{ij}(u) J(j) \right], \ i = 1, \ldots, n$$

$$(T_\mu J)(i) = c(i, \mu(i)) + \alpha \sum_{j=1}^{n} p_{ij}(\mu(i)) J(j), \ i = 1, \ldots, n.$$

Note that a function $f : \mathsf{X} \to \mathbb{R}$ can be represented as a vector of length n, with coordinates $\{f(i)\}_{1 \leq i \leq n}$. For the stationary policy μ, define the Markov transition matrix $P_\mu = [p_{ij}(\mu(i))]_{i,j}$, i.e. the transition matrix of the Markov chain $X_0, X_1, \ldots$ obtained once the policy μ is fixed. Let c_μ be the vector with i^{th} coordinate $c(i, \mu(i))$, i.e., the cost of state i under policy μ. Then, representing a cost function J as the vector $[J(1), \ldots J(n)]^T$ we can write Bellman's

equation for T_μ as a *system of n linear equations*, where the unknown is the vector J_μ, cost vector for the stationary policy μ

$$\begin{aligned} J_\mu = T_\mu J_\mu &= c_\mu + \alpha P_\mu J_\mu \\ (I - \alpha P_\mu) J_\mu &= c_\mu \\ J_\mu &= (I - \alpha P_\mu)^{-1} c_\mu. \end{aligned}$$

Here I denotes the $n \times n$ matrix. The invertibility of the matrix $I - \alpha P$ follows directly from the fact that we already know that the system $T_\mu J_\mu = J_\mu$ has a unique solution[2]. Thus any method available to solve linear systems of equations can be used to compute the cost function of a given policy μ. To compute the *optimal* value function and policy however, we need to solve a nonlinear system of equations.

Rate of Convergence of VI with Error Bounds

Consider a problem with a finite state and control space, as in the preceding paragraph. Assume for simplicity that there is a unique optimal policy μ^*. Then after a while, the VI

$$\min_\mu T_\mu T^{k-1} J = T^k J$$

always yields μ^* and thus the iterations reduce to

$$J := c_{\mu^*} + \alpha P_{\mu^*} J,$$

which is a linear system of equations. The rate of convergence of the iterations should then be controlled by the maximum eigenvalue modulus of αP_μ, i.e., α since P_μ has a unit eigenvalue. When the error bounds are used, the rate at which $\tilde{J}_k$ or $\hat{J}_k$ in (7.6) and (7.7) approach J^* turns out to be governed by the eigenvalue with the second largest modulus instead, see [Ber07, p.28] and problems B.3.6, B.4.1.

Variants of Value Iteration

Gauss-Seidel VI In standard VI, we use the same current cost estimate J_k to update all states at the next iterate $J_{k+1} = TJ_k$. Instead, we could incorporate earlier in the computations the new values of the cost for the states that have already been treated. More precisely, with a state space $\mathsf{X} = \{1, 2, \ldots, n\}$, assume that the current cost estimate is J. Then we can start the next iteration with state 1, defining the mapping F by

$$FJ(1) = \min_{u \in \mathsf{U}(1)} \left[c(1,u) + \alpha \sum_{j=1}^{n} p_{1j}(u) J(j) \right],$$

[2]or purely algebraically: the eigenvalues of αP_μ are in the interior of the unit disk - see Gershgorin's theorem for example.

and we continue updating the states in order, incorporating the new values $FJ(j)$ already computed

$$FJ(i) = \min_{u \in \mathsf{U}(i)} \left[c(i,u) + \alpha \sum_{j=1}^{i-1} p_{ij}(u) FJ(j) + \alpha \sum_{j=i}^{n} p_{ij}(u) J(j) \right], \text{ for } i = 2, \ldots, n.$$

This corresponds to the Gauss-Seidel method to solve the nonlinear system of equations $J = TJ$. There is some evidence that this method performs roughly as efficiently as VI with bounds, so that it is a good method to use (note that it is as easy to compute as standard VI).

Proposition 7.1.2. *Let $\mathsf{X} = \{1, \ldots, n\}$, and $J, J' : \mathsf{X} \to \mathbb{R}$. For Gauss-Seidel Value Iteration, we have*

$$\|FJ - FJ'\|_\infty \leq \alpha^k \|J - J'\|_\infty.$$

Furthermore $FJ^* = J^*$ and $\lim_{k \to \infty} \|F^k J - J^*\|_\infty = 0$. Finally, if J verifies $J \leq TJ \leq J^*$, then

$$T^k J \leq F^k J \leq J^*.$$

The last result of the preceding proposition says that Gauss-Seidel VI performs always at least as well as VI if the initial choice J verifies $J \leq TJ \leq J^*$.

Proof. Using the result for T, we know that

$$|FJ(1) - FJ'(1)| \leq \alpha \|J - J'\|_\infty.$$

Then, again using the result for T we have

$$\begin{aligned} |FJ(i) - FJ'(i)| \leq & \alpha \max\{|FJ(1) - FJ'(1)|, \ldots, |FJ(i-1) - FJ'(i-1)|, \\ & |J(i) - J'(i)|, \ldots, |J(n) - J'(n)|\}. \end{aligned}$$

The result then follows by a straightforward induction. The equation $FJ^* = J^*$ follows from the definition of J and the result for T. The convergence property follows from there and the inequalities above. The last result uses the definition of F, the monotonicity property of T, and a straightforward induction. □

Asynchronous VI This algorithm is like Gauss-Seidel value iteration but the states are not updated in order. Instead, the algorithm picks out an infinite sequence of states $x_0, x_1, x_2, \ldots$ such that every state in X occurs infinitely often. This is useful in an implementation of VI using parallel computations, where each processor can work on updating a component of the value function and transmits its result to the other processors when it is done. The process can be asynchronous and will still converge as long as all processors keep updating their values. Convergence of asynchronous VI is proved by showing that the iterations between two time periods at which all states have been updated at least once more form a contraction.

Real-Time VI This algorithm is like asynchronous value iteration, but the sequence of states is generated by simulating the system. A difference with asynchronous VI however is that we are not guaranteed to update each state an infinite number of times. The algorithm is as follows

1. Start with x_0 arbitrary and some initial cost vector J_0.
2. At stage k in state x_k and with current cost approximation J_k, compute the one-step lookahead control
$$u_k \in \arg\min_{u \in \mathsf{U}(x_k)} \left\{ \sum_{y \in \mathsf{X}} p_{x_k y}(u)(c(x_k, u, y) + \alpha J_k(y)) \right\}.$$
3. Update J for the state x_k according to
$$J_{k+1}(x) = \begin{cases} \sum_{y \in \mathsf{X}} p_{xy}(u_k)(c(x, u_k, y) + \alpha J_k(y)), & \text{if } x = x_k \\ J_k(x), & \text{otherwise.} \end{cases}$$
4. Perform a transition to the next state $x_{k+1} = f(x_k, u_k, w_k)$. Repeat from 2.

Theorem 7.1.3. *Assume $|\mathsf{X}|$ is finite. If $J_0 \leq J^*$ then there exists a time $\bar{k}$ (random, but with $\bar{k} < \infty$ almost surely) such that the controls u_k are optimal for all $k \geq \bar{k}$.*

Note that for RTVI we do not necessarily have $J_k \to J^*$. The result concerns sample paths of the algorithm, and the theorem only says that eventually we will be making optimal decisions along each path. The process does not necessarily explore the whole state space, so at the end we do not necessarily obtain a globally defined optimal policy, unless more assumptions are made regarding communication between states. For very large state space, the theorem suggests that RTVI can be used to find an optimal policy for a small subset of the total state space. A few more remarks on the assumption $J_0 \leq J^*$. This assumption means that we are optimistic about the cost-to-go function at each state, so we have incentives to go to each state. Also, from the monotonicity property of T and $TJ^* = J^*$, we get that all iterates verify $J_k \leq J^*$.

Proof. Consider a sample path followed by the RTVI algorithm. Since the state space is finite, there exists a finite time k after which all the states visited by this path are visited infinitely often. Denote this set by $\tilde{\mathsf{X}}$. On this set, we are essentially doing asynchronous value iteration so there is some vector $J^*_{\tilde{\mathsf{X}}}$ such that
$$J_k(x) \to J^*_{\tilde{\mathsf{X}}}(x), \ \forall x \in \tilde{\mathsf{X}}.$$
Moreover there is an actual policy π (defined on all X) such that $J_\pi(x) = J^*_{\tilde{\mathsf{X}}}(x)$ for all $x \in \tilde{\mathsf{X}}$. Hence we have
$$J^*_{\tilde{\mathsf{X}}}(x) \geq J^*(x) \geq J_k(x), \ \forall x \in \tilde{\mathsf{X}},$$

where the last inequality follows from our discussion after the statement of the theorem. Together with the convergence result, we get $J^*_{\tilde{\mathsf{X}}}(x) = J_\pi(x)$ for all $x \in \tilde{\mathsf{X}}$. Overall, the iterates J_k converge to a function $\tilde{J}$ such that

$$\tilde{J}(x) = J^*(x) \; \forall x \in \tilde{\mathsf{X}}$$
$$\tilde{J}(x) \leq J^*(x) \; \forall x \in \tilde{\mathsf{X}}^c.$$

Hence the controls chosen by the RTVI algorithm become optimal among policies that do not leave $\forall x \in \tilde{\mathsf{X}}$. Now consider a policy which in some state $x \in \tilde{\mathsf{X}}$ choses a control u with $p_{xy}(u) > 0$ for some $y \in \tilde{\mathsf{X}}^c$. The optimal cost-to-go from y verifies $J^*(y) \geq \tilde{J}(x)$ and so the RTVI policy also performs at least as well as this policy.

□

Using Value Iteration to Prove Properties of the Value Function

For finite horizon problems, we have used backward induction reasoning extensively to prove various properties of the value function and optimal policy. Value iteration can be used to replace the DP algorithm in this context as well, i.e., for the theoretical investigation of the properties of the solution. A typical example of this method is provided in [Ber07, vol. II p. 17], where value iteration is used to show that an optimal policy in a machine replacement problem has the form of a threshold policy.

Q-Value Iteration

If J^* is known, then an optimal policy can be computed as

$$\mu^*(x) = \arg\min_{u \in \mathsf{U}(x)} E\Big[c(x,u,w) + \alpha J^*(f(x,u,w))\Big|x\Big].$$

This motivates the definition of the *Q-factors*

$$Q^*(x,u) := E\Big[c(x,u,w) + \alpha J^*(f(x,u,w))\Big|x\Big], \; \forall (x,u). \tag{7.8}$$

Then if the Q-factors are known, an optimal control at state x is obtained by computing $\min_u Q^*(x,u)$. Minimize over u in (7.8) and recall Bellman's equation to see that

$$J^*(x) = \min_{u \in \mathsf{U}(x)} Q^*(x,u)$$

and so the Q-factors satisfy the fixed point equation

$$Q^*(x,u) = E\Big[c(x,u,w) + \alpha \min_{\bar{u} \in \mathsf{U}(f(x,u,w))} Q^*(f(x,u,w),\bar{u})\Big|x\Big]. \tag{7.9}$$

Q^* is the unique solution of this equation. We can define an operator F similar to T

$$(FQ)(x,u) = E\Big[c(x,u,w) + \alpha \min_{\bar{u} \in \mathsf{U}(f(x,u,w))} Q^*(f(x,u,w),\bar{u})\Big|x\Big].$$

In the case of countable state spaces, this gives

$$(FQ)(x,u) = \sum_{y \in \mathsf{X}} p_{xy}(u)\Big(c(x,u,y) + \alpha \min_{\bar{u} \mathsf{U}(y)} Q(y,\bar{u})\Big).$$

Then (7.9) can be rewritten more succinctly $Q^* = FQ^*$, the equivalent of Bellman's equation. It turns out that F is also an α-contraction for $\|\cdot\|_\infty$, it has a unique fixed point which is Q^*, and various value iterations algorithms converge to Q^*: Q-Value Iteration, i.e., $Q_{k+1} = FQ_k$, Gauss-Seidel Q-VI and asynchronous Q-VI. Also we can apply Real-Time Q-Value Iteration.

Approximate VI in Infinite State Spaces

The value iteration algorithm converges in problems with infinite state and control spaces. However in these problems as well as in problems with a finite but very large state space, VI might be implementable only approximately. That is, for a function J, we might only be able to compute a function $\tilde{J}$ such that $\|\tilde{J} - TJ\|_\infty \leq \epsilon$, for some $\epsilon > 0$. For example, we might compute $TJ(x)$ for some state x and obtain $\tilde{J}$ on X by interpolation or least-squares fit. The approximate VI algorithm under the condition

$$\|J_{k+1} - TJ_k\|_\infty \leq \epsilon, \ \forall k \geq 0 \tag{7.10}$$

approaches J^* within $\epsilon/(1-\alpha)$. Indeed we have

$$TJ_0 - \epsilon e \leq J_1 \leq TJ_0 + \epsilon e.$$

Now assume we have proved

$$T^k J_0 - \epsilon(1 + \alpha + \ldots + \alpha^{k-1})e \leq J_k \leq T^k J_0 + \epsilon(1 + \alpha + \ldots + \alpha^{k-1})e$$

Applying T to this relation

$$T^{k+1} - \alpha\epsilon(1 + \alpha + \ldots + \alpha^{k-1})e \leq TJ_k \leq T^{k+1} J_0 + \alpha\epsilon(1 + \alpha + \ldots + \alpha^{k-1})e.$$

Now using (7.10), we get

$$T^{k+1} - \epsilon(1 + \alpha + \ldots + \alpha^{k})e \leq J_{k+1} \leq T^{k+1} J_0 + \epsilon(1 + \alpha + \ldots + \alpha^{k})e,$$

and so this relation is true for all k by induction. Taking limits

$$J^* - \frac{\epsilon}{1-\alpha}e \leq \liminf_{k\to\infty} J_k \leq \limsup_{k\to\infty} J_k \leq J^* + \frac{\epsilon}{1-\alpha}e.$$

7.2 Policy Iteration

The policy iteration algorithm generates a sequence of improving stationary policies. The algorithm is as follows.

1. (Initialization) Start with an initial stationary policy μ^0.

2. (Policy Evaluation) Given the stationary policy μ^k, compute its cost J_{μ^k} by solving the linear system of equations

$$(I - \alpha P_{\mu^k}) J_{\mu^k} = c_{\mu^k}.$$

3. (Policy Improvement) Obtain a new stationary policy μ^{k+1} satisfying

$$T_{\mu^{k+1}} J_{\mu^k} = T J_{\mu^k}.$$

   In other words, we have

$$\mu^{k+1}(x) \in \arg \min_{u \in \mathsf{U}(x)} E[c(x,u,w) + \alpha J_{\mu^k}(f(x,u,w))|x].$$

   If the policy does not change, then stop. Otherwise repeat the process from step 2.

Note that when the algorithm stops, we have $\mu^{k+1} = \mu^k$ and so $J_{\mu^k} = T_{\mu^k} J_{\mu^k} = T_{\mu^{k+1}} J_{\mu^k} = T J_{\mu^k}$, hence the policy μ^k is optimal by theorem 6.5.1. The convergence of value iteration is based on the following result.

Proposition 7.2.1. *Let μ and $\bar{\mu}$ be stationary policies such that $T_{\bar{\mu}} J_\mu = T J_\mu$, that is, for all $x \in \mathsf{X}$*

$$E[c(x, \bar{\mu}(x), w) + \alpha J_\mu(f(x, \bar{\mu}(x), w))] = \min_{u \in \mathsf{U}(x)} E[c(x,u,w) + \alpha J_\mu(f(x,u,w))].$$

Then $J_{\bar{\mu}} \leq J_\mu$. Moreover if μ is not optimal, strict inequality holds in this inequality for at least one state.

Proof. We have $T_\mu J_\mu = J_\mu$ by theorem 6.5.1, and $T_{\bar{\mu}} J_\mu = T J_\mu$ by hypothesis. We also clearly have $T_\mu J_\mu \geq T J_\mu$, so finally $J_\mu \geq T_{\bar{\mu}} J_\mu$. Apply $T_{\bar{\mu}}$ on both sides of the inequality repeatedly and use the monotonicity of $T_{\bar{\mu}}$ and the convergence of value iteration to get

$$J_\mu \geq T_{\bar{\mu}} J_\mu \geq T_{\bar{\mu}}^2 J_\mu \geq \ldots \geq J_{\bar{\mu}}.$$

If $J_\mu = J_{\bar{\mu}}$ then $J_\mu = J_{\bar{\mu}} = T_{\bar{\mu}} J_{\bar{\mu}} = T_{\bar{\mu}} J_\mu = T J_\mu$. So $J_\mu = J^*$ by unicity of the solution of Bellman's equation (theorem 6.5.1). Thus μ must be optimal. Hence if μ is not optimal, we must have $J_{\bar{\mu}}(x) < J_\mu(x)$ for some state x. □

Corollary 7.2.2. *If $|\mathsf{X}|$ and $|\mathsf{U}(x)|$ are finite for all x, the policy iteration algorithm finds an optimal stationary policy in a finite number of steps.*

Proof. Under the assumptions, there is only a finite number of possible stationary policies, and the PI algorithm is strictly improving as long as an optimal policy is not found. □

The main advantage of PI over VI is this finite time convergence (for finite spaces). On the other hand, it requires finding the exact value of J_{μ^k} in each policy evaluation step, which is not attractive when the state space is large (we need to solve at each step a large linear system). In some problems, we can improve the computational properties of PI by showing that it generates a sequence of policies with attractive features, for example threshold policies (ee problem B.4.3). As a first variant of PI, we can replace the exact computation in the policy evaluation step by an iterative approach using the value iteration algorithm $T_{\mu^k} J \to J_{\mu^k}$. This is called the *modified policy iteration algorithm*. Error bounds and Gauss-Seidel VI iteration can be used at this step. In general, at the k^{th} policy evaluation step, we have a policy μ^k whose cost J_{μ^k} is only approximated using m_k value iteration steps, to obtain an approximation J_k. Then the next policy μ^{k+1} is defined as $T_{\mu^{k+1}} J_k = T J_k$. If $m_k = 0$ for all k, we obtain the value iteration method (in this case, start with some arbitrary function J_0). If $m_k = \infty$ for all k, this is the policy iteration algorithm. Experience suggests to take $m_k \geq 1$. Note that the iterations $J := T_\mu J$ are still much less computationally expensive thant $J := TJ$ if the number of controls available at each state is large. The convergence of the modified policy iteration, including an asynchronous version of this algorithm, can be found in [Ber07, vol. II, p.44].

Approximate Policy Iteration

For infinite or very large state spaces, the policy evaluation step and the policy improvement steps might only be implementable approximately, similarly to the situation for value iteration. Hence we are led to consider algorithms generating a sequence of stationary policies $\{\mu^k\}$ and approximate cost functions $\{J_k\}$ such that the policy evaluation step produces J_k with

$$\|J_k - J_{\mu_k}\|_\infty \leq \delta,\ k \geq 0,$$

and the policy improvement step produces $\mu k + 1$ such that

$$\|T_{\mu^{k+1}} J_k - T J_k\| \leq \epsilon,\ k \geq 0,$$

for some $\epsilon, \delta > 0$, and starting from an arbitrary stationary policy μ^0. It turns out then that the sequence of policies generated by this algorithm satisfies the following performance bound

$$\limsup_{k \to \infty} \|J_{\mu^k} - J^*\| \leq \frac{\epsilon + 2\alpha\delta}{(1-\alpha)^2}.$$

This bound is proved in [Ber07, vol. II, p.48]. At some point, the iterates J_{μ^k} typically oscillate in a neighborhood of the optimum J^*. Unfortunately the

fact that the size of this neighborhood is proportional to $(1-\alpha)^{-2}$ is not very good for α close to 1. Yet it turns out that this bound cannot be improved in the general case.