# Lecture 4: Policy Iteration

## Motivation

In the previous lecture, we saw that Value Iteration converges to $V^\star$ at a geometric rate $\gamma^k$. While this is efficient, Value Iteration has a notable property: in exact arithmetic, it *may not exactly reach* $V^\star$ in finite steps—it only approaches it asymptotically. Moreover, the optimal policy might stabilize long before the value function converges.

*Can we work with policies directly and achieve exact convergence in finite time?*

**Policy Iteration Idea:** Instead of iterating on value functions, alternate between two steps:

1. **Policy Evaluation:** Compute $V^\pi$ *exactly* for the current policy $\pi$
2. **Policy Improvement:** Compute the greedy policy with respect to $V^\pi$

Since there are only finitely many deterministic policies ($|\mathcal{A}|^{|\mathcal{S}|}$), and each step either improves the policy or terminates, this process must finish in finite time!

## Policy Evaluation

The first component of Policy Iteration is computing the value function $V^\pi$ for a given policy $\pi$. This is called the *policy evaluation* problem.

**Definition 1** (Policy Evaluation Problem)**.** *Given a stationary policy $\pi : \mathcal{S} \to \mathcal{A}$ (or $\pi : \mathcal{S} \to \Delta(\mathcal{A})$), compute the value function $V^\pi(s)$ for all $s \in \mathcal{S}$.*

Recall that $V^\pi$ satisfies the Bellman equation:

$$V^\pi(s) = R(s, \pi(s)) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}(s'|s, \pi(s)) V^\pi(s').$$

In matrix form: $V^\pi = R^\pi + \gamma \mathcal{P}^\pi V^\pi$, which gives

$$V^\pi = (I - \gamma \mathcal{P}^\pi)^{-1} R^\pi$$

where $R^\pi \in \mathbb{R}^{|\mathcal{S}|}$ has entries $R^\pi(s) = R(s, \pi(s))$ and $\mathcal{P}^\pi \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{S}|}$ has entries $\mathcal{P}^\pi(s, s') = \mathcal{P}(s'|s, \pi(s))$.

## Method 1: Matrix Inversion (Exact)

The most direct approach is to solve the linear system $(I - \gamma \mathcal{P}^\pi)V^\pi = R^\pi$.

**Lemma 1** (Policy Evaluation via Matrix Inversion). *For any stationary policy $\pi$, the value function $V^\pi$ can be computed exactly in $O(|\mathcal{S}|^3)$ time using Gaussian elimination or matrix inversion.*

*Proof.* The matrix $(I - \gamma \mathcal{P}^\pi)$ is invertible (since $\|\gamma \mathcal{P}^\pi\|_\infty = \gamma < 1$, all eigenvalues of $\gamma \mathcal{P}^\pi$ have magnitude less than 1). Solving a linear system with an $n \times n$ matrix takes $O(n^3)$ time, so the total cost is $O(|\mathcal{S}|^3)$. □

## Method 2: Iterative Policy Evaluation

An alternative is to use the Bellman operator $\mathcal{T}^\pi$ iteratively.

**Algorithm 1** Iterative Policy Evaluation

**Require:** Policy $\pi$, tolerance $\varepsilon > 0$
Initialize $V^{(0)}(s) \leftarrow 0$ for all $s \in \mathcal{S}$
**for** $t = 0, 1, 2, \ldots$ **do**
  **for** each $s \in \mathcal{S}$ **do**
    $V^{(t+1)}(s) \leftarrow R(s, \pi(s)) + \gamma \sum_{s'} \mathcal{P}(s'|s, \pi(s)) V^{(t)}(s')$ ▷ $= (\mathcal{T}^\pi V^{(t)})(s)$
  **end for**
  **if** $\|V^{(t+1)} - V^{(t)}\|_\infty < \frac{(1-\gamma)\varepsilon}{\gamma}$ **then**
    **break**
  **end if**
**end for**
**return** $V^{(t+1)}$ ▷ *Approximation to $V^\pi$*

**Lemma 2** (Convergence of Iterative Policy Evaluation). *The iterates $V^{(t+1)} = \mathcal{T}^\pi V^{(t)}$ converge to $V^\pi$ at rate $\gamma^t$:*

$$\|V^{(t)} - V^\pi\|_\infty \leq \gamma^t \|V^{(0)} - V^\pi\|_\infty \leq \frac{\gamma^t R_{\max}}{1 - \gamma},$$

*where the last inequality uses $\|V^\pi\|_\infty \leq R_{\max}/(1-\gamma)$ (recall that rewards lie in $[0, R_{\max}]$).*

*Proof.* Since $\mathcal{T}^\pi$ is a $\gamma$-contraction and $V^\pi$ is its unique fixed point, the result follows from the Banach Fixed Point Theorem (identical to the Value Iteration analysis). □

The stopping criterion $\|V^{(t+1)} - V^{(t)}\|_\infty < \frac{(1-\gamma)\varepsilon}{\gamma}$ guarantees $\|V^{(t+1)} - V^\pi\|_\infty < \varepsilon$, by the same argument as the Stopping Criterion Lemma in Lecture 3.

### Comparison of Methods

| Method | Time Complexity | Space | Exact? |
|---|---|---|---|
| Matrix Inversion | $O(\|\mathcal{S}\|^3)$ | $O(\|\mathcal{S}\|^2)$ | Yes |
| Iterative PE ($\varepsilon$-approx) | $O\left(\frac{\|\mathcal{S}\|^2}{1-\gamma}\log\frac{R_{\max}}{\varepsilon(1-\gamma)}\right)$ | $O(\|\mathcal{S}\|)$ | No |

**Remark 1.** *Matrix inversion is preferred when $|\mathcal{S}|$ is moderate and we need exact evaluation. Iterative PE is preferred when $|\mathcal{S}|$ is very large or when approximate evaluation suffices.*

## Policy Improvement

The second component of Policy Iteration is *policy improvement*: given the value function $V^\pi$ of the current policy, construct a new policy $\pi'$ that is at least as good as $\pi$ (and strictly better unless $\pi$ is already optimal).

**Definition 2** (Policy Improvement)**.** *Given a policy $\pi$ with value function $V^\pi$, define the* improved policy $\pi'$ *by:*

$$\pi'(s) \in \operatorname*{argmax}_{a\in\mathcal{A}} \left(R(s,a) + \gamma \mathbb{E}_{s'\sim\mathcal{P}(\cdot|s,a)}[V^\pi(s')]\right) = \operatorname*{argmax}_{a\in\mathcal{A}} Q^\pi(s,a).$$

*That is, $\pi'$ is greedy with respect to $V^\pi$.*

The following theorem is the key result that makes Policy Iteration work.

**Theorem 3** (Policy Improvement Theorem)**.** *Let $\pi$ be any policy and let $\pi'$ be greedy with respect to $V^\pi$. Then:*

$$V^{\pi'}(s) \geq V^\pi(s) \quad \textit{for all } s \in \mathcal{S},$$

*with equality for all $s$ if and only if $\pi$ is already optimal (i.e., $V^\pi = V^\star$).*

*Proof.* The proof relies on the monotonicity of the Bellman operator $\mathcal{T}^{\pi'}$.

**Step 1: One-step improvement.** By definition of $\pi'$ (greedy w.r.t. $Q^\pi$), for any $s$:

$$Q^\pi(s,\pi'(s)) = \max_{a\in\mathcal{A}} Q^\pi(s,a) \geq Q^\pi(s,\pi(s)) = V^\pi(s).$$

In operator notation, this means $\mathcal{T}^{\pi'}V^\pi \geq \mathcal{T}^\pi V^\pi = V^\pi$.

**Step 2: Apply monotonicity.** The Bellman operator $\mathcal{T}^{\pi'}$ is monotonic: if $U(s) \leq V(s)$ for all $s$, then $(\mathcal{T}^{\pi'}U)(s) \leq (\mathcal{T}^{\pi'}V)(s)$ (since the transition matrix $\mathcal{P}^{\pi'}$ only has non-negative

entries). Applying $\mathcal{T}^{\pi'}$ to both sides of $V^\pi \leq \mathcal{T}^{\pi'} V^\pi$:

$$\mathcal{T}^{\pi'} V^\pi \leq \mathcal{T}^{\pi'}(\mathcal{T}^{\pi'} V^\pi) = (\mathcal{T}^{\pi'})^2 V^\pi.$$

By induction, applying $(\mathcal{T}^{\pi'})^k$ yields a non-decreasing sequence:

$$V^\pi \leq \mathcal{T}^{\pi'} V^\pi \leq (\mathcal{T}^{\pi'})^2 V^\pi \leq \cdots \leq (\mathcal{T}^{\pi'})^k V^\pi.$$

**Step 3: Convergence.** Since $\mathcal{T}^{\pi'}$ is a $\gamma$-contraction, the sequence $(\mathcal{T}^{\pi'})^k V_0$ converges to its unique fixed point $V^{\pi'}$ for any initial $V_0$. Choosing $V_0 = V^\pi$, we have:

$$V^\pi \leq \lim_{k \to \infty} (\mathcal{T}^{\pi'})^k V^\pi = V^{\pi'}.$$

Thus $V^\pi(s) \leq V^{\pi'}(s)$ for all $s$.

**Step 4: Characterize equality.** Equality holds throughout if and only if $V^\pi = \mathcal{T}^{\pi'} V^\pi$, which means $V^\pi(s) = \max_{a \in \mathcal{A}} Q^\pi(s, a)$ for all $s$. This is the Bellman optimality equation $V^\pi = \mathcal{T}^\star V^\pi$. By uniqueness of the fixed point of $\mathcal{T}^\star$, this implies $V^\pi = V^\star$. □

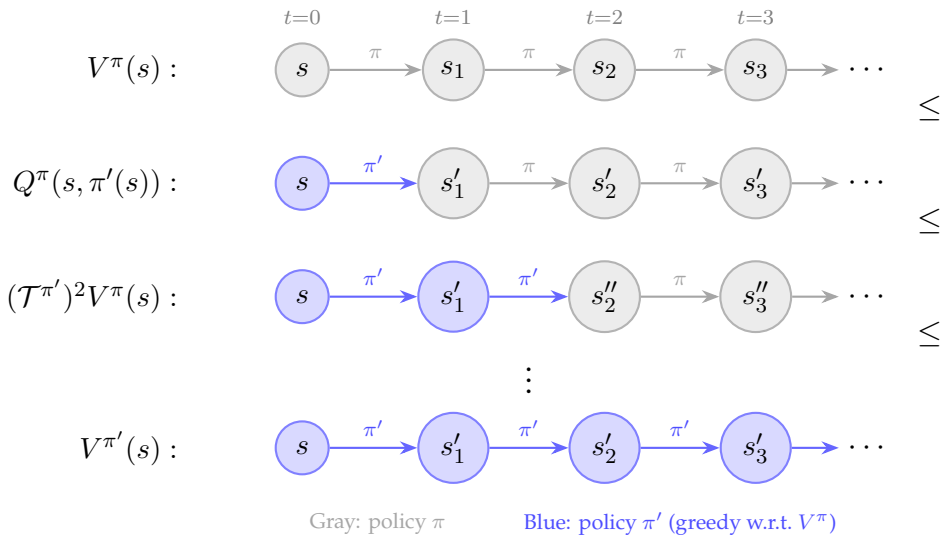


Figure 1: Visualizing the Policy Improvement Theorem (Theorem 3). Starting from $V^\pi$ (top), we gradually replace actions from $\pi$ with $\pi'$ (the greedy policy). Each replacement improves the value because $\pi'(s) = \text{argmax}_a Q^\pi(s, a)$. The state labels $(s_1, s'_1, s''_2, \ldots)$ indicate that different policies lead to different state distributions.

**Corollary 4** (Strict Improvement). *If $\pi'(s) \neq \pi(s)$ for some state $s$, then either:*

1. *$V^{\pi'}(\tilde{s}) > V^\pi(\tilde{s})$ for some state $\tilde{s}$, or*
2. *$V^\pi = V^\star$ already.*

*In other words, if $\pi$ is not optimal, policy improvement makes strict progress.*

*Proof.* If $\pi' \neq \pi$ but $V^{\pi'} = V^{\pi}$ everywhere, then by the equality condition in the Policy Improvement Theorem, $V^{\pi} = V^{\star}$, so $\pi$ was already optimal. □

# The Policy Iteration Algorithm

We now combine policy evaluation and policy improvement into the complete Policy Iteration algorithm.

**Algorithm 2** Policy Iteration

**Require:** MDP $M = (\mathcal{S}, \mathcal{A}, \mathcal{P}, R, \gamma)$
Initialize $\pi_0$ arbitrarily (e.g., $\pi_0(s) \leftarrow$ any fixed action in $\mathcal{A}$ for all $s$)
**for** $k = 0, 1, 2, \ldots$ **do**
  Compute $V^{\pi_k}$ exactly (e.g., via $V^{\pi_k} = (I - \gamma \mathcal{P}^{\pi_k})^{-1} R^{\pi_k}$) ▷ *Policy Evaluation*
  **for** each $s \in \mathcal{S}$ **do** ▷ *Policy Improvement*
    $\pi_{k+1}(s) \leftarrow \operatorname{argmax}_{a \in \mathcal{A}} \left( R(s,a) + \gamma \sum_{s'} \mathcal{P}(s'|s,a) V^{\pi_k}(s') \right)$
  **end for**
  **if** $\pi_{k+1} = \pi_k$ **then** ▷ *Policy unchanged ⇒ converged*
    **return** $\pi_k$, $V^{\pi_k}$
  **end if**
**end for**

**Remark 2.** *Policy Iteration maintains an explicit* policy *at each iteration, while Value Iteration maintains an explicit* value function*. This seemingly small difference leads to fundamentally different convergence behavior.*

# Convergence Rate Analysis

Beyond the Policy Improvement Theorem, we can establish a stronger result: Policy Iteration converges to $V^{\star}$ at a **geometric rate** $\gamma$, just like Value Iteration, despite working with policies rather than value functions directly.

**Theorem 5** (Convergence Rate of Policy Iteration[1])**.** *Let $\{\pi_k\}_{k \geq 0}$ be the sequence of policies produced by Policy Iteration. Then:*

$$\|V^{\pi_{k+1}} - V^{\star}\|_{\infty} \leq \gamma \|V^{\pi_k} - V^{\star}\|_{\infty}.$$

*Consequently, $\|V^{\pi_k} - V^{\star}\|_{\infty} \leq \gamma^k \|V^{\pi_0} - V^{\star}\|_{\infty}$.*

[1] This theorem and several subsequent results in this lecture follow the presentation in https://rltheory.github.io/.

*Proof.* Fix any state $s \in \mathcal{S}$. We analyze the gap $V^\star(s) - V^{\pi_{k+1}}(s)$.

**Step 1: Expand using Bellman equations.** By the Bellman optimality equation for $V^\star$ and the Bellman equation for $V^{\pi_{k+1}}$:

$$\begin{aligned} V^\star(s) - V^{\pi_{k+1}}(s) = &\max_{a\in\mathcal{A}} \left[R(s,a) + \gamma \mathbb{E}_{s'\sim\mathcal{P}(\cdot|s,a)}[V^\star(s')]\right] \\ &- \left[R(s,\pi_{k+1}(s)) + \gamma \mathbb{E}_{s'\sim\mathcal{P}(\cdot|s,\pi_{k+1}(s))}[V^{\pi_{k+1}}(s')]\right]. \end{aligned}$$

**Step 2: Use monotonicity $V^{\pi_{k+1}} \geq V^{\pi_k}$.** By the Policy Improvement Theorem, $V^{\pi_{k+1}}(s') \geq V^{\pi_k}(s')$ for all $s'$. Therefore:

$$\begin{aligned} V^\star(s) - V^{\pi_{k+1}}(s) \leq &\max_{a\in\mathcal{A}} \left[R(s,a) + \gamma \mathbb{E}_{s'\sim\mathcal{P}(\cdot|s,a)}[V^\star(s')]\right] \\ &- \left[R(s,\pi_{k+1}(s)) + \gamma \mathbb{E}_{s'\sim\mathcal{P}(\cdot|s,\pi_{k+1}(s))}[V^{\pi_k}(s')]\right]. \end{aligned}$$

**Step 3: Use the definition of $\pi_{k+1}$.**
Since $\pi_{k+1}(s) = \operatorname{argmax}_{a\in\mathcal{A}} Q^{\pi_k}(s,a) = \operatorname{argmax}_{a\in\mathcal{A}} \left[R(s,a) + \gamma \mathbb{E}_{s'\sim\mathcal{P}(\cdot|s,a)}[V^{\pi_k}(s')]\right]$, we have:

$$R(s,\pi_{k+1}(s)) + \gamma \mathbb{E}_{s'\sim\mathcal{P}(\cdot|s,\pi_{k+1}(s))}[V^{\pi_k}(s')] = \max_{a\in\mathcal{A}} \left[R(s,a) + \gamma \mathbb{E}_{s'\sim\mathcal{P}(\cdot|s,a)}[V^{\pi_k}(s')]\right].$$

Substituting:

$$V^\star(s) - V^{\pi_{k+1}}(s) \leq \max_{a\in\mathcal{A}} \left[R(s,a) + \gamma \mathbb{E}_{s'\sim\mathcal{P}(\cdot|s,a)}[V^\star(s')]\right] - \max_{a\in\mathcal{A}} \left[R(s,a) + \gamma \mathbb{E}_{s'\sim\mathcal{P}(\cdot|s,a)}[V^{\pi_k}(s')]\right].$$

**Step 4: Apply the max inequality.** Using the inequality $\max_{a\in\mathcal{A}} f(a) - \max_{a\in\mathcal{A}} g(a) \leq \max_{a\in\mathcal{A}}[f(a) - g(a)]$:

$$\begin{aligned} V^\star(s) - V^{\pi_{k+1}}(s) &\leq \max_{a\in\mathcal{A}} \left[\gamma \mathbb{E}_{s'\sim\mathcal{P}(\cdot|s,a)}[V^\star(s') - V^{\pi_k}(s')]\right] \\ &\leq \gamma \max_{a\in\mathcal{A}} \mathbb{E}_{s'\sim\mathcal{P}(\cdot|s,a)}[\|V^\star - V^{\pi_k}\|_\infty] \\ &= \gamma \|V^\star - V^{\pi_k}\|_\infty. \end{aligned}$$

**Step 5: Conclude.** Since $V^\star(s) \geq V^{\pi_{k+1}}(s)$ (optimality of $V^\star$), we have $|V^\star(s) - V^{\pi_{k+1}}(s)| = V^\star(s) - V^{\pi_{k+1}}(s)$.

Taking the maximum over all states $s$:

$$\|V^{\pi_{k+1}} - V^\star\|_\infty \leq \gamma \|V^{\pi_k} - V^\star\|_\infty.$$

By induction: $\|V^{\pi_k} - V^\star\|_\infty \le \gamma^k \|V^{\pi_0} - V^\star\|_\infty$. □

**Corollary 6** (Iteration Complexity from Convergence Rate). *To achieve $\|V^{\pi_k} - V^\star\|_\infty \le \varepsilon$, it suffices that*

$$k \ge \frac{1}{1-\gamma} \log\left(\frac{R_{\max}}{\varepsilon(1-\gamma)}\right).$$

*Proof.* Since $\|V^{\pi_0} - V^\star\|_\infty \le R_{\max}/(1-\gamma)$ (using rewards in $[0, R_{\max}]$), we need $\gamma^k \cdot R_{\max}/(1-\gamma) \le \varepsilon$. The result follows by the same calculation as for Value Iteration (Corollary 2 in Lecture 3). □

# Limitation of Value Iteration

Before analyzing the finite convergence of Policy Iteration, we first examine a fundamental limitation of Value Iteration that motivates our interest in stronger convergence guarantees.

While Value Iteration converges to an $\varepsilon$-optimal solution efficiently, it may not find the *exact* optimal policy in finite time. The following example demonstrates this fundamental limitation.

**Proposition 7** (Value Iteration is Not Strongly Polynomial (Gap-Dependent Iterations) (Feinberg et al., 2014)). *There exists a family of MDPs with $|\mathcal{S}| = 3$ states, $|\mathcal{A}| = 2$ actions, and deterministic transitions, such that Value Iteration requires arbitrarily many iterations to find the optimal policy.*

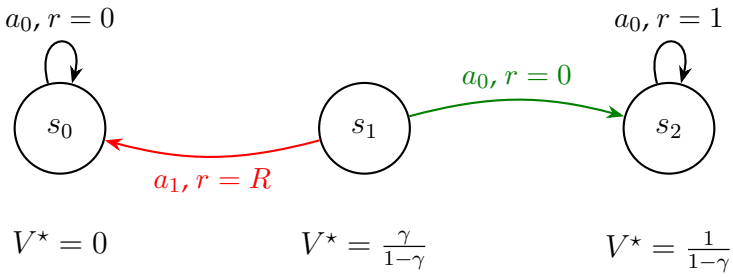


Figure 2: Counterexample MDP showing VI is not strongly polynomial. At $s_1$, the optimal action is $a_0$ (green, go to $s_2$), but VI may choose $a_1$ (red, collect $R$ and go to $s_0$) for arbitrarily many iterations (depending on how close $R$ is to $\frac{\gamma}{1-\gamma}$). **Parameters:** $\gamma \in (0,1)$ is any discount factor; $R = \frac{\gamma}{1-\gamma} - \delta$ for small $\delta > 0$ ensures $a_0$ is optimal by a margin of $\delta$.

*Proof.* Consider the MDP illustrated in Figure 2 with states $\{s_0, s_1, s_2\}$ and actions $\{a_0, a_1\}$:

**Transitions and Rewards:**

- $s_0$: absorbing state with reward 0 (action $a_0$ stays at $s_0$)
- From $s_1$: action $a_0$ goes to $s_2$ (reward 0); action $a_1$ goes to $s_0$ (reward $R = \frac{\gamma}{1-\gamma} - \delta$ for small $\delta > 0$)
- $s_2$: "reward-collecting" state with reward 1 (action $a_0$ stays at $s_2$)

**Optimal Values:**

- $V^\star(s_0) = 0$ (absorbing, no reward)
- $V^\star(s_2) = 1 + \gamma V^\star(s_2) = \frac{1}{1-\gamma}$ (collect reward 1 forever)
- At $s_1$: $Q^\star(s_1, a_0) = 0 + \gamma V^\star(s_2) = \frac{\gamma}{1-\gamma}, \quad Q^\star(s_1, a_1) = R = \frac{\gamma}{1-\gamma} - \delta$

The optimal action at $s_1$ is $a_0$, with $Q^\star(s_1, a_0) - Q^\star(s_1, a_1) = \delta$. That is, **the optimal action is only marginally better by** $\delta$.

**Value Iteration Dynamics:** Starting from $V_0 = 0$:

- $V_k(s_0) = 0$ for all $k$
- $V_k(s_2) = \frac{1-\gamma^k}{1-\gamma} \to \frac{1}{1-\gamma}$ as $k \to \infty$

At iteration $k$, Value Iteration compares at state $s_1$:

- Action $a_0$ (go to $s_2$): $Q_k(s_1, a_0) = 0 + \gamma V_{k-1}(s_2) = \frac{\gamma(1-\gamma^{k-1})}{1-\gamma}$
- Action $a_1$ (collect $R$, go to $s_0$): $Q_k(s_1, a_1) = R = \frac{\gamma}{1-\gamma} - \delta$

VI selects the **correct** action $a_0$ only when $Q_k(s_1, a_0) > Q_k(s_1, a_1)$, i.e., when

$$\frac{\gamma(1-\gamma^{k-1})}{1-\gamma} > \frac{\gamma}{1-\gamma} - \delta \quad \Longleftrightarrow \quad \gamma^{k-1} < \frac{\delta(1-\gamma)}{\gamma}.$$

**Key Observation:** To distinguish the optimal action from the suboptimal one, VI needs $\gamma^{k-1} < O(\delta)$ (ignoring $\gamma$). When $\delta$ is very small (i.e., the two actions are nearly tied), this requires $k$ to be very large:

$$k > 1 + \frac{\log(\gamma/(\delta(1-\gamma)))}{\log(1/\gamma)} = O\left(\frac{1}{1-\gamma} \log \frac{1}{\delta(1-\gamma)}\right).$$

By choosing $\delta$ arbitrarily small, we can make the required number of iterations arbitrarily large. □

**Definition 3** (Strongly Polynomial in MDPs)**.** *An algorithm for discounted MDPs is* strongly polynomial *if it finds an optimal policy using a number of arithmetic operations polynomial in* $|\mathcal{S}|$,

*$|\mathcal{A}|$, and $1/(1-\gamma)$, without dependence on precision parameters (such as $\varepsilon$ or the suboptimality gap $\delta$). We assume rewards are bounded so that all value functions lie in $[0, 1/(1-\gamma)]$.*

**The Key Question:** Value Iteration is *not* strongly polynomial—its iteration count depends on the gap $\delta$. *Can Policy Iteration do better?*

# Runtime Bound for Policy Iteration

The answer is yes. We now show that Policy Iteration not only converges, but terminates in a *finite* number of steps—and in fact, in polynomial time. The key tool for proving this is the Value Difference Identity.

## Value Difference Identity

A powerful tool for analyzing Policy Iteration is the following identity that relates the value functions of two policies.

**Lemma 8** (Value Difference Identity)**.** *For any two stationary policies $\pi, \pi'$:*

$$V^{\pi'} - V^{\pi} = (I - \gamma \mathcal{P}^{\pi'})^{-1} g(\pi', \pi)$$

*where $g(\pi', \pi) \coloneqq \mathcal{T}^{\pi'} V^{\pi} - V^{\pi}$ is the* advantage *of $\pi'$ relative to $\pi$.*

*Proof.* From the explicit solution of the Bellman equation, $V^{\pi'} = (I - \gamma \mathcal{P}^{\pi'})^{-1} R^{\pi'}$. Therefore:

$$\begin{aligned} V^{\pi'} - V^{\pi} &= (I - \gamma \mathcal{P}^{\pi'})^{-1} R^{\pi'} - V^{\pi} \\ &= (I - \gamma \mathcal{P}^{\pi'})^{-1} \left[ R^{\pi'} - (I - \gamma \mathcal{P}^{\pi'}) V^{\pi} \right] \\ &= (I - \gamma \mathcal{P}^{\pi'})^{-1} \left[ R^{\pi'} + \gamma \mathcal{P}^{\pi'} V^{\pi} - V^{\pi} \right] \\ &= (I - \gamma \mathcal{P}^{\pi'})^{-1} \left[ \mathcal{T}^{\pi'} V^{\pi} - V^{\pi} \right] \\ &= (I - \gamma \mathcal{P}^{\pi'})^{-1} g(\pi', \pi). \end{aligned}$$

□

**Interpretation:** The advantage $g(\pi', \pi)(s)$ measures the immediate benefit (or loss) of switching from $\pi$ to $\pi'$ at state $s$, evaluated under the value function $V^{\pi}$. When $\pi'$ is greedy with respect to $V^{\pi}$, we have $g(\pi', \pi) \geq 0$, which explains why Policy Improvement works. When $\pi^{\star}$ is optimal, $g(\pi, \pi^{\star}) \leq 0$ for any policy $\pi$ (no policy can gain relative to the optimal policy).

## Finite Convergence

**Theorem 9** (Finite Convergence of Policy Iteration)**.** *Policy Iteration terminates in at most $|\mathcal{A}|^{|\mathcal{S}|}$ iterations, outputting an optimal policy $\pi^\star$ and optimal value function $V^\star$.*

*Proof.* The proof uses a counting argument combined with the Policy Improvement Theorem.

**Step 1: Monotonicity.** By the Policy Improvement Theorem (Theorem 3):

$$V^{\pi_{k+1}}(s) \geq V^{\pi_k}(s) \quad \text{for all } s \in \mathcal{S}.$$

The sequence of value functions is monotonically non-decreasing (componentwise).

**Step 2: Strict improvement or termination.** If $\pi_{k+1} \neq \pi_k$, then by Corollary 4, either:

- $V^{\pi_{k+1}}(s) > V^{\pi_k}(s)$ for at least one state $s$, or
- $V^{\pi_k} = V^\star$ already.

In the second case, we assume *consistent tie-breaking*: when multiple actions achieve the maximum in the policy improvement step, we keep the current action $\pi_k(s)$ if it is among them. This ensures $\pi_{k+1} = \pi_k$ when $\pi_k$ is already optimal.

**Step 3: Counting argument.** There are exactly $|\mathcal{A}|^{|\mathcal{S}|}$ deterministic stationary policies (each of $|\mathcal{S}|$ states can choose from $|\mathcal{A}|$ actions). Since each non-terminal iteration moves to a *strictly better* policy, and value functions are real-valued, we cannot revisit any policy. Therefore, the algorithm terminates in at most $|\mathcal{A}|^{|\mathcal{S}|}$ iterations.

**Step 4: Optimality at termination.** When $\pi_{k+1} = \pi_k$, the greedy policy w.r.t. $V^{\pi_k}$ is $\pi_k$ itself. This means:

$$\pi_k(s) = \operatorname*{argmax}_{a \in \mathcal{A}} Q^{\pi_k}(s, a) \quad \text{for all } s.$$

Therefore $V^{\pi_k}(s) = \max_{a \in \mathcal{A}} Q^{\pi_k}(s, a)$, which is precisely $(\mathcal{T}^\star V^{\pi_k})(s)$. So $V^{\pi_k}$ is a fixed point of $\mathcal{T}^\star$. By uniqueness, $V^{\pi_k} = V^\star$, and hence $\pi_k = \pi^\star$. □

**Can We Do Better?** The bound $|\mathcal{A}|^{|\mathcal{S}|}$ is exponential in $|\mathcal{S}|$, which seems pessimistic. Note that there are only $|\mathcal{S}||\mathcal{A}| - |\mathcal{S}|$ suboptimal (state, action) pairs to eliminate (since at every state, at least one action must be optimal). Can we show that Policy Iteration eliminates these pairs efficiently?

The answer is yes. The following lemma provides a key insight: not only does Policy Iteration make progress, but it makes *measurable* progress every $O\left(\frac{1}{1-\gamma} \log \frac{1}{1-\gamma}\right)$ iterations.

**Lemma 10** (Strict Progress (Scherrer, 2016)). *Fix a suboptimal policy $\pi_0$, and let $\{\pi_k\}_{k\geq 0}$ be the sequence produced by Policy Iteration. Define*

$$k^\star := \left\lceil \log_{1/\gamma}\left(\frac{1}{1-\gamma}\right)\right\rceil + 1 = O\left(\frac{1}{1-\gamma}\log\frac{1}{1-\gamma}\right).$$

*Then there exists a state $s_0 \in \mathcal{S}$ such that:*

*(i) $\pi_0(s_0)$ is strictly suboptimal at $s_0$, i.e., $\pi_0(s_0) \notin \operatorname{argmax}_{a\in\mathcal{A}} Q^\star(s_0, a)$;*

*(ii) For all $k \geq k^\star$: $\pi_k(s_0) \neq \pi_0(s_0)$.*

*Proof.* Fix $k \geq 0$ and let $\pi^\star$ be an optimal policy. By the Value Difference Identity (Lemma 8):

$$V^{\pi_k} - V^{\pi^\star} = (I - \gamma\mathcal{P}^{\pi_k})^{-1} g(\pi_k, \pi^\star).$$

Recall that $g(\pi_k, \pi^\star) := \mathcal{T}^{\pi_k} V^{\pi^\star} - V^{\pi^\star}$ is the advantage of $\pi_k$ relative to $\pi^\star$. Since $\pi^\star$ is optimal, $g(\pi_k, \pi^\star) \leq 0$ (no policy can gain relative to the optimal policy).

**Step 1: Bound the advantage norm.** Taking the operator $(I - \gamma\mathcal{P}^{\pi_k})$ on both sides:

$$-g(\pi_k, \pi^\star) = (I - \gamma\mathcal{P}^{\pi_k})(V^\star - V^{\pi_k}) = (V^\star - V^{\pi_k}) - \gamma\mathcal{P}^{\pi_k}(V^\star - V^{\pi_k}).$$

Let $\Delta = V^\star - V^{\pi_k} \geq 0$. Since $\mathcal{P}^{\pi_k}$ is a stochastic matrix (nonnegative entries) and $\Delta \geq 0$, we have $\mathcal{P}^{\pi_k}\Delta \geq 0$. Thus for each state $s$:

$$-g(\pi_k, \pi^\star)(s) = \Delta(s) - \gamma(\mathcal{P}^{\pi_k}\Delta)(s) \leq \Delta(s) \leq \|\Delta\|_\infty.$$

Taking the maximum over $s$ gives $\|g(\pi_k, \pi^\star)\|_\infty \leq \|V^\star - V^{\pi_k}\|_\infty$.

**Step 2: Apply geometric convergence.** By Theorem 5: $\|V^\star - V^{\pi_k}\|_\infty \leq \gamma^k \|V^\star - V^{\pi_0}\|_\infty$.

By the Value Difference Identity (applying with $\pi' = \pi_0$ and $\pi = \pi^\star$):

$$V^{\pi_0} - V^{\pi^\star} = (I - \gamma\mathcal{P}^{\pi_0})^{-1} g(\pi_0, \pi^\star).$$

Therefore:

$$V^{\pi^\star} - V^{\pi_0} = (I - \gamma\mathcal{P}^{\pi_0})^{-1}(-g(\pi_0, \pi^\star)).$$

Since $g(\pi_0, \pi^\star) \leq 0$, we have $-g(\pi_0, \pi^\star) \geq 0$. Taking norms and using $\|(I - \gamma\mathcal{P}^{\pi_0})^{-1}\|_\infty = \|\sum_{t=0}^\infty \gamma^t (\mathcal{P}^{\pi_0})^t\|_\infty \leq \sum_{t=0}^\infty \gamma^t = \frac{1}{1-\gamma}$ (Neumann series expansion; see Lecture 2):

$$\|V^\star - V^{\pi_0}\|_\infty = \|(I - \gamma\mathcal{P}^{\pi_0})^{-1}(-g(\pi_0, \pi^\star))\|_\infty \leq \frac{1}{1-\gamma}\|g(\pi_0, \pi^\star)\|_\infty.$$

Combining: $\|g(\pi_k, \pi^\star)\|_\infty \leq \frac{\gamma^k}{1-\gamma}\|g(\pi_0, \pi^\star)\|_\infty$.

**Step 3: Identify the changing state.** Let $s_0 \in \mathcal{S}$ be a state where $-g(\pi_0, \pi^\star)(s_0) = \|g(\pi_0, \pi^\star)\|_\infty > 0$ (which exists since $\pi_0$ is suboptimal). This implies $V^\star(s_0) - \mathcal{T}^{\pi_0}V^\star(s_0) = \max_{a\in\mathcal{A}} Q^\star(s_0, a) - Q^\star(s_0, \pi_0(s_0)) > 0$, so $\pi_0(s_0)$ is strictly suboptimal at $s_0$, proving conclusion (i).

For conclusion (ii), we show that for large enough $k$, the suboptimality gap of $\pi_k$ at $s_0$ is strictly smaller than that of $\pi_0$, which forces the action to change. By our choice of $s_0$ and the bound from Step 2:

$$-g(\pi_k, \pi^\star)(s_0) \leq \|g(\pi_k, \pi^\star)\|_\infty \leq \frac{\gamma^k}{1-\gamma}\|g(\pi_0, \pi^\star)\|_\infty = \frac{\gamma^k}{1-\gamma}\big(-g(\pi_0, \pi^\star)(s_0)\big).$$

When $k \geq k^\star$, the prefactor satisfies $\frac{\gamma^k}{1-\gamma} < 1$ (this is precisely what $k^\star$ is chosen to ensure), giving $-g(\pi_k, \pi^\star)(s_0) < -g(\pi_0, \pi^\star)(s_0)$. Since $-g(\pi, \pi^\star)(s) = V^\star(s) - Q^\star(s, \pi(s))$ is the suboptimality gap of action $\pi(s)$, this means $Q^\star(s_0, \pi_k(s_0)) > Q^\star(s_0, \pi_0(s_0))$, so $\pi_k(s_0) \neq \pi_0(s_0)$. Crucially, this holds for *all* $k \geq k^\star$ since $V^\star/Q^\star$ is a fixed property of the MDP, so the elimination is permanent, proving conclusion (ii). □

**Theorem 11** (Improved Iteration Bound[2])**.** *Policy Iteration terminates in at most*

$$k^\star(|\mathcal{S}||\mathcal{A}| - |\mathcal{S}|) = O\left(\frac{|\mathcal{S}||\mathcal{A}|}{1-\gamma}\log\frac{1}{1-\gamma}\right)$$

*iterations, where* $k^\star = O\left(\frac{1}{1-\gamma}\log\frac{1}{1-\gamma}\right)$ *is as defined in Lemma 10.*

*Proof.* The key insight is that the Strict Progress Lemma provides a stronger guarantee than simple monotonicity: every $k^\star$ iterations, at least one *strictly suboptimal* (state, action) pair is permanently eliminated from consideration.

**Step 1: Counting suboptimal actions.** For each state $s$, there is at least one optimal action $a^\star(s) \in \operatorname{argmax}_{a\in\mathcal{A}} Q^\star(s, a)$. Therefore, at most $|\mathcal{A}| - 1$ actions at each state are strictly suboptimal, and the total number of strictly suboptimal (state, action) pairs is at most $|\mathcal{S}|(|\mathcal{A}| - 1) = |\mathcal{S}||\mathcal{A}| - |\mathcal{S}|$.

**Step 2: Iterative elimination argument.** Partition iterations into phases of length $k^\star$: phase $i$ covers iterations $(i-1)k^\star$ to $ik^\star - 1$. If the starting policy $\tilde{\pi} = \pi_{(i-1)k^\star}$ of phase $i$ is suboptimal, apply Lemma 10 with $\tilde{\pi}$ as the initial policy. The lemma guarantees a state $\tilde{s}$ where (i) $\tilde{\pi}(\tilde{s})$ is strictly suboptimal at $\tilde{s}$, and (ii) $\pi_k(\tilde{s}) \neq \tilde{\pi}(\tilde{s})$ for all $k \geq ik^\star$.

[2]The first polynomial bound was proved by Ye (2011). The proof here follows the simpler approach of Scherrer (2016).

The eliminated pairs across phases are distinct: if phases $i < j$ both eliminated $(s, a)$, then the permanent elimination from phase $i$ would imply $\pi_{(j-1)k^\star}(s) \neq a$, contradicting that phase $j$ starts with action $a$ at state $s$.

Since each phase eliminates a distinct strictly suboptimal pair, and there are at most $|\mathcal{S}|(|\mathcal{A}| - 1)$ such pairs, Policy Iteration terminates in at most $k^\star(|\mathcal{S}||\mathcal{A}| - |\mathcal{S}|)$ iterations.

Substituting $k^\star = O\left(\frac{1}{1-\gamma} \log \frac{1}{1-\gamma}\right)$:

$$k^\star(|\mathcal{S}||\mathcal{A}| - |\mathcal{S}|) = O\left(\frac{|\mathcal{S}||\mathcal{A}|}{1-\gamma} \log \frac{1}{1-\gamma}\right). \qquad \square$$

**Strongly Polynomial:** Unlike the naive bound of $|\mathcal{A}|^{|\mathcal{S}|}$ (exponential in $|\mathcal{S}|$), this bound is *polynomial* in $|\mathcal{S}|$, $|\mathcal{A}|$, and $1/(1-\gamma)$, with no dependence on precision. This confirms that Policy Iteration is strongly polynomial, answering the question raised in the previous section.

**Why can PI bypass VI's limitation?** The key difference is that PI computes $V^\pi$ *exactly* via matrix inversion, rather than approximating it iteratively. In the counterexample from the previous section, even if PI starts with $\pi_0(s_1) = a_1$ (the wrong action), the first policy evaluation gives $V^{\pi_0}(s_2) = \frac{1}{1-\gamma}$ exactly. Then policy improvement compares:

$$Q^{\pi_0}(s_1, a_0) = \frac{\gamma}{1-\gamma} \quad \text{vs.} \quad Q^{\pi_0}(s_1, a_1) = \frac{\gamma}{1-\gamma} - \delta.$$

No matter how small $\delta$ is, PI immediately identifies $a_0$ as optimal. VI, using the approximation $V_k(s_2) = \frac{1-\gamma^k}{1-\gamma}$, cannot distinguish the two actions until $\gamma^k < O(\delta)$.
**What if PI uses Iterative PE?** If Policy Iteration uses Iterative Policy Evaluation (Algorithm 1) instead of matrix inversion, it loses this advantage. This explains the unified view in Section "Variants of Policy Iteration": only exact PE ($m = \infty$) guarantees finite-time convergence to $\pi^\star$; any finite $m$ (including $m = 1$, which is VI) cannot.

# Computational Complexity

We now analyze the computational cost of Policy Iteration.

**Lemma 12** (Per-Iteration Complexity of Policy Iteration)**.** *Each iteration of Policy Iteration requires:*

- ***Policy Evaluation:*** $O(|\mathcal{S}|^3)$ *via matrix inversion*
- ***Policy Improvement:*** $O(|\mathcal{S}|^2|\mathcal{A}|)$ *to compute the greedy policy*

*Total per iteration:* $O(|\mathcal{S}|^3 + |\mathcal{S}|^2|\mathcal{A}|) = O(|\mathcal{S}|^2(|\mathcal{S}| + |\mathcal{A}|))$.

*Proof.* Policy evaluation requires solving $(I - \gamma \mathcal{P}^\pi)V = R^\pi$, which costs $O(|\mathcal{S}|^3)$.

Policy improvement requires, for each state $s$, computing $\operatorname{argmax}_{a \in \mathcal{A}}[R(s,a) + \gamma \sum_{s'} \mathcal{P}(s'|s,a)V(s')]$. This is $O(|\mathcal{A}|)$ actions times $O(|\mathcal{S}|)$ terms in the sum, for each of $|\mathcal{S}|$ states: $O(|\mathcal{S}|^2|\mathcal{A}|)$. □

**Theorem 13** (Total Complexity of Policy Iteration)**.** *Using exact policy evaluation, Policy Iteration runs in time*

$$O\left(K \cdot |\mathcal{S}|^2(|\mathcal{S}| + |\mathcal{A}|)\right)$$

*where $K$ is the number of iterations. With the improved bound (Theorem 11), $K = O\left(\frac{|\mathcal{S}||\mathcal{A}|}{1-\gamma} \log \frac{1}{1-\gamma}\right)$, giving total complexity*

$$O\left(\frac{|\mathcal{S}|^3|\mathcal{A}|(|\mathcal{S}| + |\mathcal{A}|)}{1-\gamma} \log \frac{1}{1-\gamma}\right).$$

*Proof.* Combine Theorem 11 (iteration bound $K = O\left(\frac{|\mathcal{S}||\mathcal{A}|}{1-\gamma} \log \frac{1}{1-\gamma}\right)$) with Lemma 12 (per-iteration cost $O(|\mathcal{S}|^2(|\mathcal{S}| + |\mathcal{A}|))$). □

### Comparison: Value Iteration vs. Policy Iteration

| Aspect | Value Iteration | Policy Iteration |
|---|---|---|
| Iterates on | Value function $V$ | Policy $\pi$ |
| Per-iteration cost | $O(\lvert\mathcal{S}\rvert^2\lvert\mathcal{A}\rvert)$ | $O(\lvert\mathcal{S}\rvert^3 + \lvert\mathcal{S}\rvert^2\lvert\mathcal{A}\rvert)$ |
| Convergence rate | $\gamma^k$ (geometric) | $\gamma^k$ (geometric) |
| Convergence type | Asymptotic only | Finite (exact termination) |
| $\varepsilon$-optimal iterations | $O\left(\frac{1}{1-\gamma} \log \frac{R_{\max}}{\varepsilon(1-\gamma)}\right)$ | Same bound (but terminates exactly) |
| Memory | $O(\lvert\mathcal{S}\rvert)$ | $O(\lvert\mathcal{S}\rvert^2)$ for matrix inversion |

**Same Rate, Different Behavior:** Both algorithms converge to $V^\star$ at the same geometric rate $\gamma^k$. The key difference is that Policy Iteration terminates *exactly* when it finds $\pi^\star$, while Value Iteration converges asymptotically.

## Worked Example: Navigation MDP Revisited

We apply Policy Iteration to the same 3-state navigation problem from Lecture 3.

**Setup.** Recall: $\mathcal{S} = \{\texttt{L}, \texttt{C}, \texttt{R}\}$, $\mathcal{A} = \{\texttt{go-left}, \texttt{go-right}\}$, $\gamma = 0.9$. State R is absorbing with reward 1; all other rewards are 0.

**Iteration 0: Initial Policy.** Let $\pi_0(s) = \texttt{go-left}$ for all $s$ (a deliberately bad initial policy).

**Policy Evaluation:** Under "always go left":

- From L: stay at L. Reward = 0.
- From C: go to L with prob 0.9, stay at C with prob 0.1. Reward = 0.
- From R: absorbing, reward = 1.

Solving $(I - \gamma \mathcal{P}^{\pi_0}) V^{\pi_0} = R^{\pi_0}$:

$$V^{\pi_0}(\texttt{L}) = 0, \quad V^{\pi_0}(\texttt{C}) = 0, \quad V^{\pi_0}(\texttt{R}) = 10.$$

**Policy Improvement:** For each state, compute $\operatorname{argmax}_{a \in \mathcal{A}} Q^{\pi_0}(s, a)$:

- At L: $Q(\texttt{L}, \texttt{go-left}) = 0$, $Q(\texttt{L}, \texttt{go-right}) = 0 + \gamma(0.9 \cdot 0 + 0.1 \cdot 0) = 0$. Tie, choose $\texttt{go-right}$.
- At C: $Q(\texttt{C}, \texttt{go-left}) = 0$, $Q(\texttt{C}, \texttt{go-right}) = 0 + \gamma(0.9 \cdot 10 + 0.1 \cdot 0) = 8.1$. Choose $\texttt{go-right}$.
- At R: Any action keeps us at R. Choose $\texttt{go-right}$.

New policy: $\pi_1(s) = \texttt{go-right}$ for all $s$.

**Iteration 1. Policy Evaluation:** Under "always go right", this is the optimal policy. Solving gives:

$$V^{\pi_1}(\texttt{L}) \approx 7.92, \quad V^{\pi_1}(\texttt{C}) \approx 8.90, \quad V^{\pi_1}(\texttt{R}) = 10.$$

**Policy Improvement:** The greedy policy w.r.t. $V^{\pi_1}$ is still $\pi_1$ (going right is optimal everywhere).

Since $\pi_2 = \pi_1$, the algorithm terminates.

**Comparison:** Policy Iteration found the optimal policy in just **2 iterations**, while Value Iteration required many iterations to converge to the same values. This efficiency gain is typical in practice.

# Variants of Policy Iteration

Several variants of Policy Iteration trade off computation per iteration against the number of iterations.

## Modified Policy Iteration

Instead of solving for $V^\pi$ exactly, we can approximate it with a fixed number of Bellman backups.

**Definition 4** (Modified Policy Iteration)**.** *Modified Policy Iteration with parameter $m$ performs:*

1. ***Partial Policy Evaluation:*** *Starting from the previous value estimate, apply the Bellman operator $\mathcal{T}^{\pi_k}$ exactly $m$ times:*

$$V^{(t+1)} = \mathcal{T}^{\pi_k} V^{(t)}, \quad t = 0, 1, \ldots, m-1.$$

2. ***Policy Improvement:*** *Compute the greedy policy w.r.t. $V^{(m)}$.*

**Unified View of Dynamic Programming:**
- $m = 1$: **Value Iteration** (one Bellman backup, then greedy improvement)
- $m = \infty$: **Policy Iteration** (exact evaluation before improvement)
- $1 < m < \infty$: **Modified Policy Iteration** (partial evaluation)

All three converge to $V^\star$, with different trade-offs between per-iteration cost and number of iterations.

# Summary: Policy Iteration

**Policy Iteration Summary:**
- **Two phases:** Policy Evaluation (compute $V^\pi$) + Policy Improvement (greedy $\pi'$)
- **Key theorems:**
  - Policy Improvement: $V^{\pi'} \geq V^\pi$ (monotonic improvement)
  - Convergence Rate: $\|V^{\pi_{k+1}} - V^\star\|_\infty \leq \gamma \|V^{\pi_k} - V^\star\|_\infty$ (geometric convergence)
- **Convergence:** Finite termination, at most $|\mathcal{A}|^{|\mathcal{S}|}$ iterations; geometric rate $\gamma^k$ toward $V^\star$
- **Per-iteration cost:** $O(|\mathcal{S}|^3)$ for exact evaluation
- **Iterations to $\varepsilon$-optimality:** $O\left(\frac{1}{1-\gamma} \log \frac{R_{\max}}{\varepsilon(1-\gamma)}\right)$
- **Practical performance:** Often converges in $O(|\mathcal{S}|)$ iterations

# Looking Ahead

Both Value Iteration and Policy Iteration assume we have complete knowledge of the MDP: the transition probabilities $\mathcal{P}$ and reward function $R$. This is called the *model-based* or *planning* setting.

In many real-world applications, we don't know the MDP model. Instead, we must *learn* from experience—by interacting with the environment and observing the resulting states and rewards. This leads to:

- **Model-free RL:** Learn value functions or policies directly from samples (Q-learning, SARSA, policy gradient methods)
- **Model-based RL:** Learn the MDP model from data, then apply planning algorithms
- **Exploration-exploitation:** Balance between trying new actions (to learn) and exploiting known good actions (to maximize reward)

These topics will be the focus of upcoming lectures.

# References

Eugene A Feinberg, Jefferson Huang, and Bruno Scherrer. Modified policy iteration algorithms are not strongly polynomial for discounted dynamic programming. *Operations Research Letters*, 42(6-7):429–431, 2014.

Bruno Scherrer. Improved and generalized upper bounds on the complexity of policy iteration. *Mathematics of Operations Research*, 41(3):758–774, 2016.

Yinyu Ye. The simplex and policy-iteration methods are strongly polynomial for the markov decision problem with a fixed discount rate. *Mathematics of Operations Research*, 36(4): 593–603, 2011.