# On the Complexity of Policy Iteration

**Yishay Mansour*and Satinder Singh**
AT&T Labs-Research
180 Park Avenue
Florham Park, NJ 07932-0971
{mansour,baveja}@research.att.com

## Abstract

Decision-making problems in uncertain or stochastic domains are often formulated as Markov decision processes (MDPs). Policy iteration (PI) is a popular algorithm for searching over policy-space, the size of which is exponential in the number of states. We are interested in bounds on the complexity of PI that do not depend on the value of the discount factor. In this paper we prove the first such non-trivial, worst-case, upper bounds on the number of iterations required by PI to converge to the optimal policy.

## 1 Introduction

The problem of decision-making in uncertain or stochastic environments is central to artificial intelligence (AI) [7, 6]. The framework of Markov decision processes (MDPs) developed in the operations research community [1] is increasingly used within AI to formulate such problems. In this formulation, the environment is assumed to be in one of a finite-set of states, the decision-making agent has a choice of actions in each state of the environment, executing an action causes a stochastic change in the state of the environment, and the agent receives a stochastic reward in return for executing the action. The agent's goal is to choose actions so as to maximize a cumulative discounted measure of rewards over some time horizon. Here we consider the planning problem in which we are given a full description of the MDP and have to compute the optimal action-selection policy.

*On sabbatical from Tel-Aviv University.

One reason for the popularity of the MDP framework within AI is the availability of a number of well-studied classes of algorithms for planning in MDPs: linear-programming [2], value iteration [1], and policy iteration [3]. Linear programming and value iteration are known to compute the optimal policy in time polynomial in the size of the representation of the MDP and the discount factor [4, 2]. While no direct analysis of policy iteration is available, one can bound the number of steps of "greedy" policy iteration (which greedily accepts all single-state action changes that are improvements) by the number of steps of value iteration. This implies that policy iteration also runs in time polynomial in the size of the representation and discount factor [3, 2].

However, our goal is to derive bounds for solving MDPs that do not depend on the discount factor. For value iteration the dependence on the discount factor is unavoidable. For linear programming, in general, it is a major open problem whether there exists a strongly polynomial algorithm, i.e., runs in time polynomial in the number of parameters and independent of the size of the representation of the parameters. For PI we can bound the number of steps independent from the representation size and discount factor as follows: PI is guaranteed to improve the policy at every step and therefore the total number of steps is trivially upper-bounded by the total number of policies. This bound is of course independent of the discount factor. However, note that the total number of policies is exponential in the number of states.

In this paper, we prove the first non-trivial upper bound on the worst-case number of steps PI can take. For the specific case of $n$ states and two actions the total number of policies is exactly $2^n$. We show that "greedy" PI will take at most $O(\frac{2^n}{n})$ steps. We also define a randomized PI (which accepts each single-state action change that is an improvement with probability 0.5) and prove that in the worst-case it will take at most $O(2^{0.78n})$ steps. Note that these bounds are independent of the size of representation of the specific parameters of the MDP and in particular do not depend on the discount factor. Our analysis also sheds new light on the manner in which PI progresses through the space of policies.

We view our results as the first step towards a better understanding of PI. This is an important issue because there is strong empirical evidence in favor of PI over value iteration and linear programming in solving MDPs [4]. While in practice it is difficult to construct MDPs for which greedy PI takes more than $n$ steps, no general rigorous lower bounds are known. A lower-bound is known for a particular form of PI, called sequential PI (which at each step accepts only one of the single-state action changes that are improvements) — in the worst-case, sequential PI can take $\Omega(2^n)$ steps when the adversary controls which improvements are selected [5, 4]. However it is not clear whether results about sequential PI transfer to other forms of PI, e.g., greedy or random PI. In fact our results show that there is a gap between the worst-case complexities of sequential and both greedy and random PI.

This paper is organized as follows: Section 2 defines the MDP model and its notation. Section 3 introduces the general scheme of policy iteration and proves a few general results concerning it. Section 4 derives an upper-bound on the time complexity of greedy PI, while section 5 derives an upper-bound to the time complexity of random PI. Section 6 concludes with a summary of our contributions and open problems.

## 2 Model

In this section we define the Markov decision process (MDP) framework.

**Definition 1** *An MDP is a tuple $(S, A, P, R)$: $S$ is a finite set of states the environment can be in, $A$ is a finite set of actions available to the agent, $P$ is the table of transition probabilities, where $P(s'|s,a)$ is the probability of a transition to state $s'$ upon executing action $a$ in state $s$, and $R$ is the reward function, where $R(s,a)$ is the expected reward received by the agent upon executing action $a$ in state $s$.*

We define the agent's *return* to be the discounted sum of rewards over an infinite horizon, i.e., we use the infinite-horizon discounted framework in this paper. More formally, the agent's return is $\sum_{t=0}^{\infty} \gamma^t r_t$ where $r_t$ is reward received at time step $t$, and $0 \leq \gamma < 1$ is a discount factor that makes future reward less valuable than immediate reward. The agent's goal is to select actions so as to maximize its expected return. In infinite-horizon discounted MDPs the agents expected return is maximized by a policy (a mapping from states to actions), called the optimal policy.

Without loss of generality, we will make the simplifying assumption that there are only two actions, i.e. $|A| = 2$. This implies that the total number of policies is $2^n$, where $n = |S|$. Other useful quantities in MDP-decision-making are value functions: one defined over states and the other defined over state-action pairs.

**Definition 2** *Let $V^\pi(s)$ be the expected return if the start-state is $s$ and the agent executes policy $\pi$ forever. Let $Q^\pi(s,a)$ be the expected return if the start-state is $s$ and the agent executes action $a$ to begin with and thereafter follows policy $\pi$.*

Note that by the definition above $Q^\pi(s,a) = R(s,a) + \gamma \sum_{s'} P(s'|s,a) V^\pi(s')$. The agent's goal, restated in terms of value functions, is that of finding an optimal policy $\pi^*$ that satisfies $\pi^* = argmax_\pi V^\pi$. The optimal value function $V^{\pi *}$ is denoted simply as $V^*$ and the associated $Q$-value function as $Q^*$. Note that there can be more than one optimal policy, however, $V^*$ and $Q^*$ are unique.

## 3 General Policy Iteration

General policy iteration works as follows. At each iteration consider changing the action at each state while keeping the actions for all the other states fixed to the current policy. Some such single-state action changes will improve upon the current policy. Different variants of policy iteration differ in which single-state improvements they accept at each step.

Before we can describe general PI, we must define what it means for one policy to be better than another. We define a partial order between the policies as follows.

**Definition 3** *For two policies, $\pi$ and $\pi'$, we have $\pi \succ \pi'$ if for each state $s$, $V^\pi(s) \geq V^{\pi'}(s)$, and for some state $s$, $V^\pi(s) > V^{\pi'}(s)$. If for every state $s$ we have $V^\pi(s) = V^{\pi'}(s)$ then $\pi \approx \pi'$.*

The partial ordering tells us when a policy is better than another and when they are incomparable. Clearly any optimal policy is better than all suboptimal policies and equivalent ($\approx$) to all other optimal policies. This partial order is central to our analysis.

Given a policy, $\pi$, we can define, using the function $Q^\pi$, the single-state improvements that could improve that policy. The following definition gives the necessary notation that we use later.

**Definition 4** *Given a policy $\pi$, let $T^\pi \subset S \times A$ be a set of all pairs $(s, a)$ such that changing the action of $\pi$ to $a$ in state $s$ improves the return of the policy, i.e. $Q^\pi(s, a) > V^\pi(s)$. We define* $\texttt{states}(T^\pi)$ *to be the states that appear in $T^\pi$, i.e., $\{s : (s, a) \in T^\pi\}$.*

For a given policy $\pi$ and a set $U \subset T^\pi$ let $\texttt{modify}(\pi, U)$ define a policy $\pi'$ whose actions are the same as those of policy $\pi$ on states not in $\texttt{states}(U)$ and $\pi'(s) = a$ for $(s, a) \in U$. (Here we use the assumption that we have only two actions. This guarantees that for each state $s$ there is at most one pair $(s, a)$ in $T^\pi$.)

Figure 1 presents the general *policy iteration* algorithm. In every iteration there are two basic steps: the first, Improvement Selection Step, selects which single-state improvements to make, and the second, Policy Improvement Step, modifies the policy accordingly. Different methods for selecting subsets of $T^\pi$ to modify the policy lead to different PI algorithms.

The following two theorems are well known properties of general policy iteration. The first claims that accepting any non-zero number of single-state improvements can only improve the policy, and the second claims that there always exists at least one single-state improvement that improves the policy, unless the policy is already optimal. (For proofs see, e.g., [2].)

**Theorem 1** *For any $U \subset T^\pi$, let $\pi' = \texttt{modify}(\pi, U)$. If $U \neq \emptyset$ then $\pi' \succ \pi$.*

**Theorem 2** *For any sub-optimal $\pi$, $T^\pi \neq \emptyset$.*

The above two theorems immediately imply that all instantiations of general PI strictly improve the policy at every iteration. Therefore, every iteration from a current to a next policy at least skips, or rules out, all the policies that are equal to or better than the current policy and worse than the next policy. How many such policies are there at every iteration? There is at least one such policy: the current policy itself. This, of course, implies an upper bound of $2^n$ steps. For specific improvement-selection methods, defined in the following sections, we perform a more careful analysis of the number of equal or better policies that get ruled out at each iteration. The more policies we can rule out at each iteration the better the upper-bound will be. Our analysis will be mainly based on properties derived from Theorems 1 and 2.

In the rest of this section, we prove two properties that hold for all instances of PI. The first is actually a property of the partial order itself: in general two policies may be incomparable but if they differ only in one state then they must be comparable.

**Lemma 3** *Let $\pi$ and $\pi'$ be two policies whose actions differ in only one state $s$, i.e., $\pi(u) = \pi'(u)$ for $u \neq s$. Then either $\pi \succ \pi'$, $\pi' \succ \pi$, or $\pi \approx \pi'$.*

**Proof:** If $Q^\pi(s, \pi'(s)) > V^\pi(s)$ then $\pi' \succ \pi$. If $Q^\pi(s, \pi'(s)) < V^{\pi'}(s)$ then $\pi \succ \pi'$. Otherwise $\pi \approx \pi'$. $\Box$

The second property that we would like to show is that PI, at different iterations, considers an improvement over a different subset of the states. This result is general to any PI.

**Lemma 4** *During a run of general policy iteration, there are no $i$ and $j$, $i < j$, such that* $\texttt{states}(T^{\pi_i}) \subseteq \texttt{states}(T^{\pi_j})$.

**Proof:** First, note that for any policy $\pi$, and any policy $\pi'$ that is identical to $\pi$ on states in $\texttt{states}(T^\pi)$, either $\pi \succ \pi'$, or $\pi \approx \pi'$. For a proof, consider an MDP $M'$ such that the only action possible from a state $s \in \texttt{states}(T^\pi)$ is $\pi(s)$. Clearly both $\pi$ and $\pi'$ are valid policies for $M'$. On the other hand in $M'$ there is no local improvement for $\pi$, i.e. $T^\pi_{M'} = \emptyset$. By Theorem 2, $\pi$ is optimal for $M'$. Therefore $\pi \succ \pi'$ (or $\pi \approx \pi'$).

Now we prove the lemma by contradiction. Assume that $i < j$ such that $\texttt{states}(T^{\pi_i}) \subseteq \texttt{states}(T^{\pi_j})$ exists. Let $T = \texttt{states}(T^{\pi_i}) \subseteq \texttt{states}(T^{\pi_j})$. Let $U' = \{(s, a) : a = \pi_i(s) \text{ and } \pi_i(s) \neq \pi_j(s) \text{ and } s \in T\}$. Clearly $U' \subseteq T^{\pi_j}$, since there are only two actions. Then $\pi' = \texttt{modify}(\pi_j, U')$ is identical with $\pi_i$ on the states in $T = \texttt{states}(T^{\pi_i})$. Therefore the general claim above implies that $\pi_i \succ \pi'$. This contradicts the fact that $\pi' \succ \pi_j \succ \pi_i$. $\Box$

So far we have showed that a subset of states can appear at most once in general policy iteration. This still

Set $\pi_0$ to an arbitrary policy.
**WHILE** $T^{\pi_i} \neq \emptyset$ **DO**
    Improvement Selection Step: $U \leftarrow \texttt{select}(T^{\pi_i})$
    Policy Improvement Step: $\pi_{i+1} \leftarrow \texttt{modify}(\pi_i, U)$.
    $i \leftarrow i+1$.
**OUTPUT** $\pi_i$.

Figure 1: General Policy Iteration Algorithm. The only assumption we make about the function `select` is that it returns a non-empty subset of its argument.

leaves open the possibility that all subsets appear in the run of the algorithm, and thus we observe all $2^n$ policies. The next step is to show that each time we perform `modify` on a large subset of the states we rule out many policies.

## 4 Greedy Policy Iteration

Greedy policy iteration is PI with $\texttt{select}(T) = T$, namely, we perform all the possible single-state action improvements at each policy improvement step. The next lemma shows that each time we perform a `modify` operation we rule out a number of policies that is at least the same as the number of changes to the policy.

**Lemma 5** *Let $\pi$ be a policy and $\pi' = \texttt{modify}(\pi, T^\pi)$. Then there are at least $|T^\pi|$ policies $\pi_i$, $1 \leq i \leq |T^\pi|$, such that $\pi' \succeq \pi_i \succ \pi$.*

**Proof:** The proof is by induction on $|T^\pi|$. The base of the induction, $|T^\pi| = 1$ follows from Theorem 1.

Consider all the single state modifications to $\pi$ using $T^\pi$, i.e., consider all $Z_j$, such that $Z_j \subset T^\pi$ and $|Z_j| = 1$. Let $U_1 = Z_j$ such that for any $Z_i$, we have that $\texttt{modify}(\pi, Z_j) \not\prec \texttt{modify}(\pi, Z_i)$. (Note that $Z_j$ is not necessarily unique, since we have a partial order.) Let $\pi_1 = \texttt{modify}(\pi, U_1)$.

With out loss of generality, let $U_1 = \{(s_1, b_1)\}$. For any other pair $(s_i, b_i) \in T^\pi$, for $i > 1$, we show that $(s_i, b_i) \in T^{\pi_1}$. Let $\pi_1^i = \texttt{modify}(\pi_1, \{(s_i, b_i)\})$. By Lemma 3 we know that either $\pi_1 \succ \pi_1^i$ or $\pi_1^i \succeq \pi_1$. We would like to claim that the relation $\pi_1 \succ \pi_1^i$ is not possible.

For contradiction assume that $\pi_1 \succ \pi_1^i$. Consider $q_i = \texttt{modify}(\pi, \{(s_i, b_i)\})$. Note that $\pi_1^i = \texttt{modify}(\pi, \{(s_1, b_1), (s_i, b_i)\})$, and therefore by Lemma 3 we know that either $q_i \succ \pi_1^i$ or $\pi_1^i \succeq q_i$. If $\pi_1^i \succeq q_i$ then $\pi_1 \succ \pi_1^i \succeq q_i$ contradicting the minimality of $\pi_1$. Therefore $q_i \succ \pi_1^i$. Let $\pi(s_1) = a_1$ and $\pi(s_i) = a_i$. Since $\pi_1 \succ \pi_1^i$, this implies that $(s_i, a_i) \in T^{\pi_1^i}$ and similarly, since $q_i \succ \pi_1^i$, this implies that $(s_1, a_1) \in T^{\pi_1^i}$. By Theorem 1 this implies that

$$\pi = \texttt{modify}(\pi_1^i, \{(s_1, a_1), (s_i, a_i)\}) \succ \pi_1^i,$$

contradicting the fact that $\pi_1^i \succ \pi$. Hence, $\pi_1^i \succ \pi_1$.

This implies that $(s_i, b_i) \in T^{\pi_1}$, for $i > 1$. Therefore, we have $|T^{\pi_1}| = |T^\pi| - 1$. The lemma follows from the inductive hypothesis on $\pi_1$. □

We can now state and prove our upper-bound on the number of steps of Greedy policy iteration.

**Theorem 6** *The greedy policy iteration algorithm considers at most $O(2^n/n)$ different policies.*

**Proof:** The analysis has two parts. The first part includes the case where the set $T^\pi$ is small. For this case we simply show that there are very few such policies. The second case will include the cases when $T^\pi$ is large. For this case we show that each iteration eliminates $\Omega(n)$ policies, that have not been eliminated before.

We define a set to be small if $|T^\pi| \leq n/3$. By Lemma 4 we do not consider the same set of states twice. This bounds the number of such modifications by

$$\sum_{k=0}^{n/3} \binom{n}{k} \leq 2\binom{n}{n/3} \leq 3\frac{2^n}{n},$$

where the second inequality holds for $n \geq 3$.

For policies $\pi_i$ such that $|T^{\pi_i}| \geq n/3$, by Lemma 5 we have that at least $n/3$ policies better than or equal to our current policy are ruled out after this iteration. This implies that the total number of policies that we consider is bounded by,

$$3\frac{2^n}{n} + \frac{2^n}{n/3} = 6\frac{2^n}{n},$$

where the first term is the number of policies with small number of improvements and the second term is a bound on the number of policies with a large number of improvements. $\square$

## 5 Random Policy Iteration

Formally, random policy iteration defines `select`$(T)$ as a random subset of $T$ where each subset has probability $2^{-|T|}$. Intuitively, we can think of it as deciding to accept each local improvement with probability half. Even though we allow for the empty subset for convenient proofs, in practice one may ignore such iterations.

The property that we would like to prove is that the number of policies that we rule out after considering each policy is at least $2^{|T^{\pi_i}|-1}$ rather than only $|T^{\pi_i}|$ (as in greedy policy iteration). This enables us to improve our bound on the running time significantly.

We first show another property of general PI: that no policy $\pi'$ incomparable to $\pi_i$ is ever considered after iteration $i$.

**Lemma 7** *At iteration $i$ we have a policy $\pi_i$, and a non-empty set $T^{\pi_i}$. At iteration $i+1$ we select $U =$ `select`$(T^{\pi_i})$ and $\pi_{i+1} =$ `modify`$(\pi_i, U)$. Let $U' \subset T^{\pi_i}$ and $\pi' =$ `modify`$(\pi_i, U')$, such that $\pi' \not\succ \pi_{i+1}$. For any $j > i+1$ we have that $\pi_j \neq \pi'$.*

**Proof:** By Theorem 1 we know that for each $j$ we have that $\pi_j \succ \pi_{j-1}$. By transitivity, we have that $\pi_j \succ \pi_{i+1}$, for $j > i+1$. Since $\pi' \not\succ \pi_{i+1}$, it implies that $\pi' \neq \pi_j$. $\square$

¿From the above lemma we know that the only policies that we can reach after $\pi_i$ are policies that are comparable with $\pi_i$. This implies that any policy which is either strictly inferior to $\pi_i$, or incomparable to $\pi_i$ will never be considered. The next step is to argue that the number of policies that we rule out at phase $i$ has an expected value of at least $2^{|T^{\pi_i}|-1}$. We first prove a general property of selecting a random element in a partial order.

**Lemma 8** *Let $\succ$ be a partial order over $\Pi$. If we chose a random element $r \in \Pi$, with uniform probability, then the expected number of elements $s \in \Pi$ such that $s \succ r$ is at most $|\Pi|/2$.*

**Proof:** For any element $v \in \Pi$ we associate two sets. $\Pi_v^+$ includes all the elements $s$ such that $s \succ v$, and $\Pi_v^-$ includes all the elements $s$ such that $v \succ s$. For every pair of elements $v_1 \succ v_2$ we have that $v_1 \in \Pi_{v_2}^+$ and $v_2 \in \Pi_{v_1}^-$. This implies that

$$\sum_{v\in\Pi} |\Pi_v^+| = \sum_{v\in\Pi} |\Pi_v^-| \leq \frac{|\Pi|^2}{2}.$$

Therefore the expected value of $|\Pi_r^+|$ is at most $|\Pi|/2$. $\square$

The following corollary is immediate from the above lemma.

**Corollary 9** *Let $\pi_i$ be the policy at iteration $i$, then the expected number of policies $\pi'$, such that $\pi_{i+1} \succ \pi' \succ \pi_i$ is at least $2^{|T^{\pi_i}|-1}$.*

Unfortunately it is not true that at each step we expect to rule out $\Omega(2^{|T^{\pi_i}|})$ policies, with high probability. Rather we can say that there is some constant probability that this will happen, and then claim that in a run with $m$ iterations we should have, with high probability, this occurring $\Omega(m)$ times.

**Theorem 10** *The random greedy policy improvement considers at most $O(2^{0.78n})$ different policies, with probability $1 - 2^{-2^{O(n)}}$.*

**Proof:** As before we consider two cases, that of small sets and that of large sets. We define a set to be small if $|T^{\pi_i}| \leq pn$, where the constant $p > 0$ will be selected later. As before we bound the number of iterations with small sets by $\sum_{i=0}^{pn} \binom{n}{i} \leq 2^{H(p)n+1}$, where $H(p)$ is the binary entropy, i.e. $H(p) = -p\log p - (1-p)\log(1-p)$.

Now we are interested in bounding the number of iterations with large sets. Assume that we have $m$ such iterations. By Lemma 9 the expected number of policies we rule out is at least $2^{pn-1}$ policies, at each such iteration. This implies that with probability $1/3$ we rule out at least $2^{pn-2}$ policies.

An iteration with a large set is *good* if it chooses a set that rules out at least $2^{pn-2}$ policies. From above, the probability that an iteration is good is at least $1/3$. A run is called typical if at least $m/4$ of the $m$ iterations with large sets are good. The number of iterations in a typical run is bounded by $2^{(1-p)n+4}$. The total number of iterations in a typical run is bounded by,

$$2^{H(p)n+1} + 2^{(1-p)n+4} \leq 2^{0.78n},$$

for $p = 0.227$ and sufficiently large $n$.

The probability that a run is not typical is at most $e^{-(1/3-1/4)^2 m}$. We are interested in runs in which $m \geq 2^{(1-p)n+4}$, in which case the probability is bounded by $2^{-2^{O(n)}}$. $\square$

## 6 Conclusion

In this paper we developed a proof technique for deriving upper-bounds on the number of steps required by policy iteration to find an optimal policy. Using our proof technique we are able to establish non-trivial upper-bounds for two important variations of policy iterations.

For greedy policy iteration we proved an upper-bound of $O(\frac{2^n}{n})$, and for random policy iteration we proved an upper-bound of $O(2^{0.78n})$. This should be contrasted with the lower-bound of $\Omega(2^n)$ for sequential policy iteration [5, 4].

Unfortunately not much is known about lower-bounds for general policy iteration other than the trivial lower-bound of $n$. The gap between upper and lower bounds is still very large and is an interesting subject for future research.

## References


[1] D. P. Bertsekas. *Dynamic Programming: Deterministic and Stochastic Models.* Prentice-Hall, Englewood Cliffs, NJ, 1987.

[2] D. P. Bertsekas. *Dynamic Programming and Optimal Control.* Athena Scientific, Belmont, MA, 1995.

[3] R. Howard. *Dynamic Programming and Markov Processes.* MIT Press, Cambridge, MA, 1960.

[4] M. L. Littman. *Algorithms for Sequential Decision Making.* PhD thesis, Brown University, 1996.

[5] M. Melekopoglou and A. Condon. On the complexity of policy iteration for stochastic games. Technical Report CS-TR-90-941, Computer Sciences Department, University of Wisconsin, Madison, 1990.

[6] S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach.* Prentice Hall, Englewood Cliffs, New Jersey, 1995.

[7] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction.* MIT Press, Cambridge, MA, 1998.