A Tighter Analysis of Randomised Policy Iteration

Abstract

Policy Iteration (PI) is a popular family of algorithms to compute an optimal policy for a given Markov Decision Problem (MDP). PI begins with an arbitrary initial policy, and repeatedly performs locally-improving switches until an optimal policy is found. The exact form of the switching rule gives rise to different variants of PI. Two decades ago, Mansour and Singh [1999] provided the first non-trivial upper bound on the number of iterations taken by "Howard's PI" (HPI), a widely-used variant of PI. They also proposed a randomised variant (RPI) with an even tighter upper bound. Their bounds for HPI and RPI have not been improved subsequently, although curiously, these algorithms tend to perform much better in practice than theoretically-superior variants.

We revisit the algorithms and analysis of Mansour and Singh [1999]. Using a new counting argument, we show a substantially tighter upper bound for RPI on n-state, 2-action-MDPs. We also provide an $\Omega(n)$ *lower* bound for RPI, matching a similar bound for HPI. Extending our analysis to k-action MDPs, $k \geq 2$, we first present a structural lemma characterising the resulting policy space. Thereafter, we show that a minor change to RPI can yield a bound of $(O(\sqrt{k \log k}))^n$ iterations, improving significantly upon Mansour and Singh's bound of roughly $(k/2)^n$. Interestingly, a similar analysis of a randomised variant of HPI also yields an upper bound of $(O(\sqrt{k \log k}))^n$ iterations, becoming the first exponential improvement for HPI over the trivial bound of k^n. Finally, we analyse a randomised variant of the "batch-switching" algorithm of Kalyanakrishnan *et al.* [2016] to obtain a bound of 1.6001^n iterations for 2-action MDPs. This bound is the tightest yet for any variant of PI.

1 Introduction

Markov Decision Problems (MDPs) [Bellman, 1957; Puterman, 1994] are an abstraction of sequential decision making, providing a formal basis for problems such as automated planning and reinforcement learning. Applications of MDPs span a variety of domains, and have proliferated in recent times as agents have gained more autonomy.

An MDP describes an agent's *environment*. For each state $s \in S$ in which the agent can be, and for each action $a \in A$ that it can take, the MDP specifies for all $s' \in S$, the probability that taking a from s will reach $s\prime$. Denote this probability$P(s, a, s')$. The transition from s to s' by taking action a yields a numeric reward $R(s, a, s')$.

The agent's behaviour is encoded as a *policy* $\pi : S \to A$, which specifies the action $\pi(s)$ that the agent must take from each state $s \in S$. Suppose indeed that the agent starts from some state $s^0 \in S$ and follows π, it encounters a random "state-reward" sequence $s^0, r^0, s^1, r^1, \ldots$ over time, wherein for $t \geq 0$, s^{t+1} is drawn according to $P(s^t, \pi(s^t), \cdot)$, and $r^t = R(s^t, \pi(s^t), s^{t+1})$. The *value* of s^0 is commonly defined to be $\mathbb{E}[r^0 + \gamma r^1 + \gamma^2 r^2 + \ldots]$, where $\gamma \in [0, 1)$ is a discount factor. An MDP is fully specified by S, A, P, R, and γ.

Our definition of value (above) is as "infinite discounted reward". The analysis provided in our paper can also be extended to alternative definitions such as "average reward" (CITE) and "total reward" (CITE) (which allows the use of $\gamma = 1$). Also, it suffices for our purposes to consider policies that do not vary over time, and which map states to actions (rather than map histories to distributions over actions).

Let (S, A, P, R, γ) be an arbitrary MDP, and Π be the set of all policies for this MDP. It is a well-known result that there is an *optimal policy* $\pi^\star \in \Pi$ such that for all $\pi \in \Pi$, $s \in S$, $V^{\pi^\star}(s) \geq V^\pi(s)$. The problem we consider in this paper is precisely that of computing an optimal policy for a given MDP (which the the MDP planning problem). Over the decades, many families of algorithms have been proposed to solve this problem—varying from formulations involving Linear Programming, to dynamic programming techniques such as Value Iteration, Policy Iteration, and combinations thereof (see the survey by [Littman *et al.*, 1995]).

In this paper, we focus exclusively on the Policy Iteration (PI) family of algorithms (CITE) when applied to finite MDPs. PI provides a conceptually simple, iterative approach to search the space of policies, which works extremely well in practice. However, there has only been limited progress towards formally quantifying its running time. The first breakthrough in this direction was provided by [Mansour and Singh, 1999], who showed non-trivial upper-bounds on the running time of Howard's PI (a commonly-used variant), and proposed a randomised variant of PI with an even tighter upper bound. More recently, even tighter upper bounds have been shown for newer variants of PI—both deterministic () and randomised (). We make several contributions, including (1) improved upper bounds for Mansour and Singh's randomised algorithm and a related variant, (2) a non-trivial lower bound for the same algorithm, (3) a novel upper bound for a randomised variant of Howard's PI, and (4) a new "batching" variant that enjoys the tightest upper bound shown to date across the PI family for 2-action MDPs.

We present our technical contributions in sections X, Y, and Z, after first describing PI in Section 2 and surveying previous analyses in Section 3. In Section X we share some experimental findings to accompany our theoretical results; we conclude with a discussion in Section 8.

2 Policy Iteration

In this section, we formalise Policy Iteration (PI), borrowing notation and definitions from [Mansour and Singh, 1999]. We assume that S and A are finite, with $|S| = n \geq 1$ and $|A| = k \geq 2$. Specifically, we take $A = \{0, 1, \ldots, k-1\}$.

Policy evaluation is a basic step that is used in PI and many other approaches for planning. Given a policy $\pi \in \Pi$, we observe that state values (taken together, the *value function* V^π) satisfy a recursive relation: for $s \in S$,

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

Hence, a given policy π can be evaluated by solving a system of linear equations.

For policies $\pi, \pi' \in \Pi$, we write $\pi \succeq \pi'$ if for all $s \in S$, $V^\pi(s) \geq V^(\pi')(s)$. If $\pi \succeq \pi'$, and in addition, for some $s \in S$, $V^\pi(s) > V^{\pi'}(s)$, we write $\pi \succ \pi'$. Observe that $\pi \succeq \pi'$ and $\pi' \succeq \pi$ implies that their value functions are identical: in which case we write $\pi \approx \pi'$. We find it convenient to distinguish between such "equally-good" policies by using an arbitrary total order L on Π. Since policies can be represented as n-length k-ary strings from $\{0, 1, \ldots, k-1\}^n$, we take that for $\pi, \pi' \in \Pi$, $\pi L \pi'$ if π' occurs after π in lexicographic ordering. We define $\pi \gtrsim \pi'$ if (1) $\pi \succ \pi'$ or (2) $\pi \approx \pi'$ and $\pi L \pi'$. By this definition, observe that there is a *unique* optimal policy $\pi^\star$ such that for all $\pi \in \Pi$, $\pi^\star \gtrsim \pi$. Our algorithms will be designed to find this policy.

A naïve way to find $\pi^\star$ would be to evaluate and compare each of the k^n policies in Π. The PI family of algorithms exploits an interesting structure of the policy space to find $\pi^\star$ more efficiently. The *action value function* Q^π for a policy $\pi \in \Pi$ provides for each $s \in S$, $a \in A$ the expected long-term reward obtained by taking a from s for just one time step, and thereafter acting according to π. It follows that

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

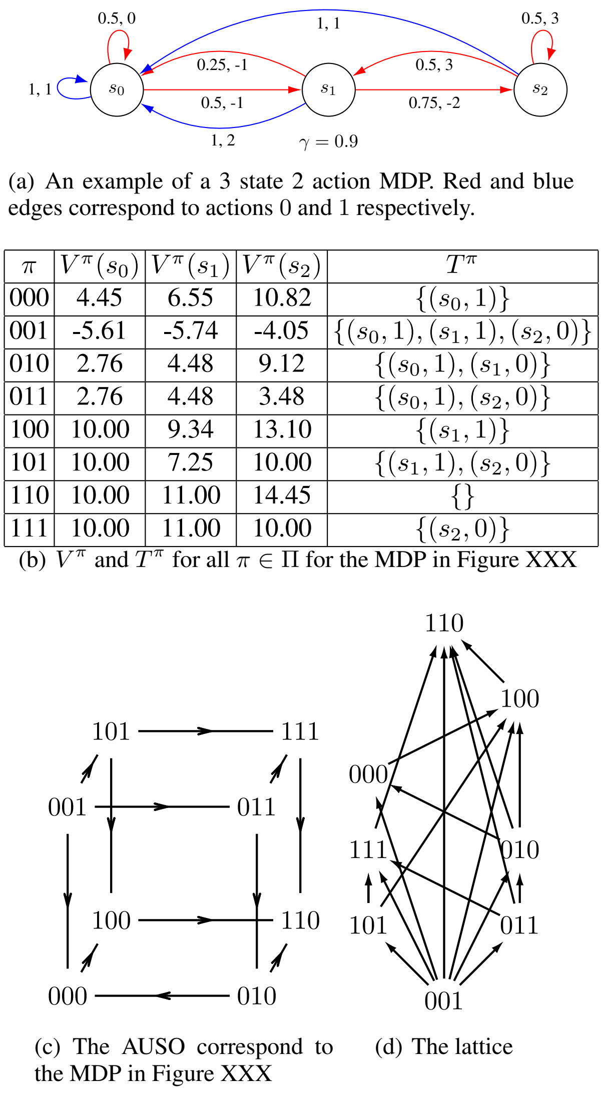
Now, define T^π as follows:

$$\begin{aligned} T^\pi = &\{(s, a) | Q^\pi(s, a) > V^\pi(s)\} \cup \\ &\{(s, a) | Q^\pi(s, a) = V^\pi(s) \text{ and } a < \pi(s)\}. \end{aligned}$$

If $(s, a) \in T^\pi$, then s is termed an "improvable state" and a an "improving action" for s. Let states(T^π) denote the set of all improvable states $s \in S$, and $T^\pi(s)$ denote the set of all improving actions for fixed $s \in S$. Indeed assume $|T^\pi| > 0$. Now, let $U \subseteq T^\pi$ be such that $|U| \geq 1$, and no two elements (s, a) and $(s', a') \in U$ have $s = s'$. In other words, U collects a subset of improvable states—denoted states(U)—and one improving action for each such state. In general, there can be many choices of U that satisfy this property for T^π. For every choice U, let modify(π, U) denote the policy π' such that for all $(s, a) \in U$, $\pi'(s) = a$ and for all $s \in S \setminus$states(U), $\pi'(s) = \pi(s)$. The following result is known as the Policy Improvement Theorem; we refer the reader to CITE for a proof.

Theorem 1. *For $\pi \in \Pi$: (1) if $T^\pi = \emptyset$, then for all $\pi' \in \Pi$, $\pi \gtrsim \pi'$; (2) else if $U \subseteq T^\pi$ satisfies the condition described above and $\pi' =$ modify(π, U), then $\pi' \gtrsim \pi$.*

In essence, the theorem allows us to test if a given policy π is optimal, and if it is not, to update to a dominating policy π'. The PI family of algorithms is based on



(a) An example of a 3 state 2 action MDP. Red and blue edges correspond to actions 0 and 1 respectively.

π	$V^\pi(s_0)$	$V^\pi(s_1)$	$V^\pi(s_2)$	T^π
000	4.45	6.55	10.82	$\{(s_0, 1)\}$
001	-5.61	-5.74	-4.05	$\{(s_0, 1), (s_1, 1), (s_2, 0)\}$
010	2.76	4.48	9.12	$\{(s_0, 1), (s_1, 0)\}$
011	2.76	4.48	3.48	$\{(s_0, 1), (s_2, 0)\}$
100	10.00	9.34	13.10	$\{(s_1, 1)\}$
101	10.00	7.25	10.00	$\{(s_1, 1), (s_2, 0)\}$
110	10.00	11.00	14.45	$\{\}$
111	10.00	11.00	10.00	$\{(s_2, 0)\}$

(b) V^π and T^π for all $\pi \in \Pi$ for the MDP in Figure XXX

(c) The AUSO correspond to the MDP in Figure XXX

(d) The lattice

Figure 1: my caption

repeatedly performing such updates until eventually, an optimal policy is reached. Given $\pi \in \Pi$, observe that V^π and Q^π, and therefore T^π, can be computed efficiently—using poly(n, k) arithmetic and comparison operations. The complexity of PI is therefore primarily determined by the number of iterations performed to reach $\pi^\star$. Algorithms in the PI family are set apart by how they pick $U \subseteq T^\pi$; in other words, which improvable states they decide to "switch" (and to which improving actions). In the next section, we discuss some existing switching rules—both deterministic and randomised—and known bounds on their complexity.

Figure XXX shows a 3-state, 2-action MDP and the value functions of the 8 resulting policies. Figure XXX shows the partial order induced on Π by the $\gtrsim$ relation.

By Theorem XX, for every non-optimal policy π, some subset of dominating policies can be reached by switching actions at improvable states of π. Interestingly, it also follows from Theorem X that any two policies π and π' that differ in only one state must satisfy either $\pi \gtrsim \pi'$ or $\pi' \gtrsim \pi$. As shown in Figure XXX, policies for 2-state MDPs form the vertices of an n-dimensional hypercube with edges directed based on $\gtrsim$. Each face of the hypercube is guaranteed to have a unique sink and no cycles, making the hypercube an Acyclic Unique Sink Orientation (AUSO). Additionally, it can be shown that AUSOs resulting from MDPs satisfy the Holt-Klee conditions (Cite), which are that every d-dimensional face of the hypercube will have at least d vertex-disconnected paths from source to sink. The Holt-Klee condisiotns hold more generally for AUSOs induced by Linear Programs, and can be shown for MDPs by considering their Linear programming formulation (CITE).

Although our primary focus in this paper is the application of PI to MDPs, the upper bounds we show in Section XX also hold for AUSOs. The problem therein is to identify the sink based on vertex-evaluations that reveal the outgoing edges. PI begins with an arbitrary vertex u, and repeatedly updates to a vertex v in the subface formed by the outgoing edges of u. Note that the lower bound we show in Section XX is specific to MDPs, and does not apply to (Holt-Klee) AUSOs.

3 Related Work and Contributions

In this section, we survey results on the complexity of different variants of PI. We also foreshadow our own contributions, which build upon ideas from the literature.

Howard's PI. The earliest variant of PI, introduced by [Howard, 1960], is also the most commonly used. Howard's PI (HPI) dictates that *every* improvable state must be switched; in other words, that for policy $\pi \in \Pi$, *States*(U) = *States*(T^π). For 2-action MDPs, this description fixes the switching rule, since $|T^\pi(s)|$ can at most be 1 for any state s. If $k \geq 3$, there might be multiple improving actions for some state—in which case we may pick an *arbitrary* one for switching. The tightest upper bound on the number of iterations taken by HPI is $O(k^n/n)$, shown by [Mansour and Singh, 1999] (the multiplicative factor was subsequently improved by [Hollanders *et al.*, 2014]). Mansour and Singh's analysis partitions Π into a set of *large-improvement* policies and a set of *small-improvement* policies. For large-improvement policies π, defined as those that for which $|T^\pi|$ is above some threshold m; a structural argument shows that modify(π) will dominate at least m policies that themselves dominate π. Since each large-improvement policy is thereby guaranteed eliminate m policies, and since the set of small-improvement policies is itself small, tuning m yields a bound of $O(k^n/n)$ on the total number of policies visited by HPI.

Randomised PI. [Mansour and Singh, 1999] also propose a randomised variant of PI (RPI), in which for policy π, states(U) is picked uniformly at random from the non-empty subsets of states(T^π). Again, if $k \geq 3$, improving actions can be picked arbitrarily. In this case, it can be seen that visits to large-improvement policies π eliminate $\theta(2^m)$ policies in expectation, leading to an overall bound of $1.7xx^n$ for 2-action MDPs and roughly $(k/2)^n$ for $k \geq 3$.[1]

Batch-switching PI. Among deterministic variants of PI, the tightest bound shown for 2-action MDPs is $1.64hj^n$ CITE and for k-action MDPs is k^{cvbn}. These variants of PI (BSPI) switch only a fixed-size batch of states at each iteration. Assuming that batches are indexed, the batch with the highest index among those have improvable states is picked. The analysis of these algorithms proceeds by numerically bounding the complexity of HPI on small MDPs (currently done up to 7 states), and using a recursive argument to scale up to general (n-state) MDPs. Interestingly, the tightest upper bound shown yet for k-action MDPs is for a randomised variant of BSPI with a batch size of 1 (CITE). The crucial aspect of this algorithm is that an improving action is picked uniformly at random from those available for the chosen improvable state. The resulting upper bound is $(2+\ln(k-1))^n$.

Lower bounds. Interestingly, on 2-action MDPs, BSPI with a batch size of 1 can take as many as $\sqrt{2}^n$ iterations [Melekopoglou and Condon, 1994]. However, only a lower bound of $\Omega(n)$ iterations has been shown on 2-action MDPs for HPI [Hansen and Zwick, 2010]. Exponential bounds have been shown for HPI on MDPs when the number of actions can depend linearly on the number of states [Fearnley, 2010; Hollanders *et al.*, 2012]. [Schurr and Szabó, 2005] also show an exponential lower bound for HPI on AUSOs. ARE THESE HOLT KLEE?

Our contributions. We make four contributions to the analysis of PI. (1) Using the same analysis structure but a tighter counting argument, we show that the RPI algorithm of [Mansour and Singh, 1999] enjoys an upper bound of $1.6fgfg^n$ on 2-action MDPs—significantly tighter than the authors had originally shown. Identifying a key structural property of k-action MDPs, we also propose a natural modification to their algorithm, such that the resulting bound is $(O(\sqrt{k \log k}))^n$. (2) We also realise a randomised variant of HPI (switch *all* improvable states; in each select an improving action uniformly at random) that achieves a bound of $(O(\sqrt{k \log k}))^n$ iterations. This bound marks the first exponential improvement for Howard's PI over the trivial bound of k^n. (3) Using numerical data, we show that the batch-switching approach, if implemented with RPI in place of HPI, enjoys an upper bound of 1.6001^n for 2-action MDPs, which becomes the tightest bound yet for the PI family. (4) We show an $\Omega(n)$ lower bound for RPI on 2-action MDPs.

4 Upper Bounds

In this section, we present new upper bounds on three variants of PI. We begin by noting that the analysis of HPI and RPI provided by [Mansour and Singh, 1999] deals separately with the cases of $k = 2$ and $k \geq 3$. The main reason for this bifurcation is their use of a specific structural property for 2-action MDPs, which is not applicable as is for $k \geq 3$. We begin by generalising the property for all $k \geq 2$, so our analysis does not need separate cases.

Lemma 2. *For policies* $\pi_1, \pi_2 \in \Pi$, *if* $|T^{\pi_1}(s)| = |T^{\pi_2}(s)|$ *for all states* $s \in S$, *then* $\pi_1 = \pi_2$.

Proof. Assume that $|T^{\pi_1}(s)| = |T^{\pi_2}(s)|$ for some policies π_1 and π_2, for all states s. Now, for each state s, note that $A \setminus T^{\pi_1}(s)$ and $T^{\pi_2}(s) \cup \{\pi_2(s)\}$ cannot be disjoint, since that would imply $(T^{\pi_2}(s) \cup \{\pi_2(s)\}) \subseteq T^{\pi_1}(s)$, which cannot be true since $|T^{\pi_2}(s) \cup \{\pi_2(s)\}| = 1 + |T^{\pi_1}(s)|$. Hence, we may construct a policy π_3 such that for each state s, $\pi_3(s) \in (A \setminus T^{\pi_1}(s)) \cap (T^{\pi_2}(s) \cup \{\pi_2(s)\})$. By Theorem X, it follows that $\pi_3 \gtrsim \pi_2$. The proof of the theorem can be adapted (applied with all rewards negated) to show that since for all $s \in S$, $\pi_3(s) \notin T^{\pi_1}(s)$, it must be that $\pi_1 \gtrsim \pi_3$. Hence, $\pi_1 \gtrsim \pi_3 \gtrsim \pi_2$. Now, if we construct a policy π_4 such that for each state s, $\pi_4(s) \in (A \setminus T^{\pi_2}(s)) \cap (T^{\pi_1}(s) \cup \{\pi_1(s)\})$, a similar argument yields $\pi_2 \gtrsim \pi_4 \gtrsim \pi_1$. Since $\pi_2 \gtrsim \pi_1$ and $\pi_1 \gtrsim \pi_2$, we get $\pi_1 = \pi_2$. □

The lemma establishes that for a given policy $\pi \in \Pi$, the sequence $(|T^\pi(s)|)_{s \in S}$ is unique. Since $0 \leq |T^\pi(s)| \leq k-1$, the number of possible sequences is k^n. Since the number of policies is also k^n, we have a bijection between Π and the set of "improvement sequences". This connection was already known for $k = 2$ (CITE CITE),

[1] Upper bounds for randomised algorithms are on the *expected* number of iterations, maximised over MDPs and initial policies.

which is simpler to analyse because $|T^{\pi}(s)| \in \{0, 1\}$ becomes an indicator for whether s is improvable. Our generalisation for all $k \geq 2$ is novel.

4.1 RPI-UIP

It follows from Theorem X that for a given policy π, there is a set of policies $I(\pi)$, with $|I(\pi)| = \prod_{s \in S} |T^{\pi}(s) + 1| - 1$, such that for each $\pi' \in I(\pi)$, $\pi' \gtrsim \pi$, and $\pi' = modify(\pi, U)$ for a suitable choice of U. We propose RPI-UIP, a randomised variant of PI, in which π' is picked uniformly at random from $I(\pi)$, the set of improving policies. RPI-UIP is identical to RPI on 2-action MDPs, but since RPI picks uniformly at random among the improvable *states* (and picks improving actions arbitrarily), the methods do not coincide if $k \geq 3$. Interestingly, Lemma X facilitates a tighter bound for RPI-UIP, albeit with the same analysis structure of [Mansour and Singh, 1999].

Definition 3. *A policy π is called a small-improvement policy if $|I(\pi)| \leq \alpha$ (we shall fix the parameter $\alpha > 0$ subsequently). A policy which is not a small-improvement policy is called a large-improvement policy.*

We present a novel bound on the number of small-improvement policies.

Lemma 4. *For all $\alpha > 0$, there are at most αH_k^{n-1} small improvement policies, where $H_k = \sum_{i=1}^{k} \frac{1}{i}$.*

Proof. Let $X(n, \alpha)$ denote the number of small improvement policies in an n-state, k-action MDP for a particular choice of $\alpha > 0$. We induct on the number of states in the MDP. For a single-state MDP, the improvement-sequences are $0, 1, \ldots, k-1$, and so $X(1, \alpha) = \min(\lfloor \alpha \rfloor, k-1) \leq \alpha$ for all $\alpha > 0$. Now assume that for all $\alpha > 0$ and some $j > 0$, $X(j, \alpha) \leq \alpha H_k^{j-1}$. We are to bound $X(j+1, \alpha)$ for all $\alpha > 0$. For an MDP with $j+1$ states, let s be any fixed state. For each $k' \in \{0, 1, \ldots, k-1\}$, the notice that the number of small-improvement policies π with $|T^{\pi}(s)| = k'$ is exactly $X(j, \frac{\alpha}{k'+1})$. Hence, the total number of small-improvement policies is $X(j, \alpha) + X(j, \frac{\alpha}{2}) + \cdots + X(j, \frac{\alpha}{k}) \leq \sum_{i=1}^{k} \frac{\alpha}{i} H_k^{j-1} = \alpha H_k^{j}$. □

Lemma 5. *If π is a large improvement policies, and π' is obtained from π by applying randomized policy iteration, the expected number of policies π'' such that $\pi \preceq \pi'' \prec \pi'$ is more than $\frac{t}{2}$.*

Proof. The number of available choices for π' is more than t for a large improvement policy. If one is selected from these uniformly at random, the expected number of policies π'' such that $\pi'' \preceq \pi$ and π'' improves upon π is half of the total number. □

Theorem 6. *With probability $1 - e^{-\Omega(k^{0.5n})}$, randomized policy iteration visits at most $8k^{\frac{n}{2}} H_k^{\frac{n-1}{2}}$ iterations.*

Proof. With probability at least $\frac{1}{3}$ we rule out more than $\frac{t}{4}$ policies, at a large improvement policy. (If this occurs with probability strictly less than $\frac{1}{3}$, then the expected number of policies we rule out is less than or equal to $\frac{1}{3} \cdot t + \frac{2}{3} \cdot \frac{t}{4} = \frac{t}{2}$, which contradicts the Lemma 5.)

An improvement of a policy is good if it rules out more than $\frac{t}{4}$ policies. The probability that a large improvement policy gets a good improvement is at least $\frac{1}{3}$. A run that visits L large improvement policies is called typical if at least $\frac{L}{4}$ of these L policies cause good improvements. The number of good improvements in a run is bounded by $\frac{4k^n}{t}$. Hence a typical run can visit at most $4 \cdot \frac{4k^n}{t} = \frac{16k^n}{t}$ large improvement policies. Total number of policies visited by a typical run is bounded by $tH_k^{n-1} + \frac{16k^n}{t}$. Setting $t = 4\sqrt{\frac{k^n}{H_k^{n-1}}}$ gives us the bound $N = 8k^{\frac{n}{2}} H_k^{\frac{n-1}{2}}$.

If a run visits more than N policies, it can not be typical. The probability that a run is not typical and visits $L > N$ policies is at most $e^{-2(\frac{1}{3} - \frac{1}{4})^2 L} = e^{-\frac{L}{72}} < e^{-\frac{N}{72}} = e^{-\Omega(k^{0.5n})}$.

□

4.2 Howard's Policy Iteration

We say that a policy π' *strictly* improves upon a policy π if $\forall s \in S : \pi'(s) \in T^{\pi}(s) \wedge (\pi'(s) = \pi(s) \Rightarrow |T^{\pi}(s)| = 1)$. There are $\prod_{s \in S} \max(|T^{\pi}(s)|, 1) \geq \prod_{s \in S} \frac{|T^{\pi}(s)|+1}{2} = \frac{1}{2^n} \cdot \prod_{s \in S}(|T^{\pi}(s)| + 1)$ policies which strictly improve upon the policy π. Howard's policy iteration picks one of them uniformly at random at each iteration.

A large improvement policy has more than $\frac{t}{2^n}$ policies which strictly improve upon it.

Lemma 7. *If π is a large improvement policies, and π' is obtained from π by applying Howard's policy iteration, the expected number of policies π'' such that $\pi \preceq \pi'' \prec \pi'$ is more than $\frac{t}{2^{n+1}}$.*

Proof. The number of available choices for π' is more than $\frac{t}{2^n}$ for a large improvement policy. If one is selected from these uniformly at random, the expected number of policies π'' such that $\pi'' \preceq \pi$ and π'' improves upon π is half of the total number. □

Theorem 8. *With probability $1-e^{-\Omega(k^{0.5n})}$, randomized Howard's policy iteration visits at most $8 \cdot 2^{\frac{n}{2}} k^{\frac{n}{2}} H_k^{\frac{n-1}{2}}$ iterations.*

Proof. With probability at least $\frac{1}{3}$ we rule out more than $\frac{t}{2^{n+2}}$ policies, at a large improvement policy. (If this occurs with probability strictly less than $\frac{1}{3}$, then the expected number of policies we rule out is less than or equal to $\frac{1}{3} \cdot \frac{t}{2^n} + \frac{2}{3} \cdot \frac{t}{2^{n+2}} = \frac{t}{2^{n+1}}$, which contradicts the Lemma 7.)

An improvement of a policy is good if it rules out more than $\frac{t}{2^{n+2}}$ policies. The probability that a large improvement policy gets a good improvement is at least $\frac{1}{3}$. A run that visits L large improvement policies is called typical if at least $\frac{L}{4}$ of these L policies cause good improvements. The number of good improvements in a run is bounded by $4 \cdot \frac{2^n k^n}{t}$. Hence a typical run can visit at most $4 \cdot 4 \cdot \frac{2^n k^n}{t} = 16 \cdot \frac{2^n k^n}{t}$ large improvement policies. Total number of policies visited by a typical run is bounded by $tH_k^{n-1} + 16 \cdot \frac{2^n k^n}{t}$. Setting $t = 4\sqrt{\frac{2^n k^n}{H_k^{n-1}}}$ gives us the bound $N = 8 \cdot 2^{\frac{n}{2}} k^{\frac{n}{2}} H_k^{\frac{n-1}{2}}$.

If a run visits more than N policies, it can not be typical. The probability that a run is not typical and visits $L > N$ policies is at most $e^{-2(\frac{1}{3}-\frac{1}{4})^2 L} = e^{-\frac{L}{72}} < e^{-\frac{N}{72}} = e^{-\Omega(k^{0.5n})}$. □

5 R-BSPI

While knowing a lower upper bound for PRI than for HPI does not mean that RPI performs better than HPI on the worst case, we do have some evidence supporting this statement in AUSOs. Through a computer search, we found out that there are 18 3-AUSOs, 16 of which are Holt-Klee and 12640 4-AUSOs, 6113 of which are Holt-Klee. Figures 4, 5, 6 and 6 show the number of policies visited by RPI and HPI on 3-AUSOs, 4-AUSOs, Holt-Klee 3-AUSOs and Holt-Klee 4-AUSOs respectively. The policy space for MDPs are known to be representable as Holt-Klee AUSOs, which are AUSOs with as many inner vertex disjoint paths from source to sink as their dimension. We ordered the AUSOs in the increasing order of number of policies visited by each PI algorithm, and plotted this number against their index in the order. The starting policies for each AUSO are kept as the policy from which the algorithm visits the maximum number of policies in the AUSO. Thus the right most and the left most points in these plots correspond to the worst case and the best case number of policies visited by these algorithms.

6 Lower Bound

Melekopoglou and Condon [1994] derived an exponential lower bound for Simple PI by demonstrating its runs on a family of MDPs. We will use the same family of MDPs $\mathcal{M} = \{M_n | n \geq 2\}$, as shown in Figure 2 to derive a linear lower bound on the expected number of policies evaluated by Randomized Policy Iteration.

The MDP M_n is defined as $M_n = (S, A, R, T, \gamma)$ with $S = \{1, 2, ..., n\} \cup \{0, 1, ..., n\} \cup \{\tilde{0}, \tilde{1}\}$, $A = \{0, 1\}$. There are 3 different types of states. The states labelled i are deterministic decision states. Depending on the policy, the next state is either $i-1$ or i' for $i \geq 2$. The states labelled i' are stochastic states, and the agent's policy does not matter on these states. Regardless of the action taken, the next state is picked uniformly at random from $(i-1)'$ and $(i-2)$ for $i \geq 3$. States $\tilde{0}$ and $\tilde{1}$ are sinks and the agent stays in them forever once they are reached. There is a reward of -1 on entering $\tilde{1}$ from $0'$ or $1'$. All other rewards are zero. The remaining transitions are demonstrated in Figure 2. We keep $\gamma = 1$ for all MDPs in the family $\mathcal{M}$. However, this family of MDPs has the property that, for any policy π and any state s, $Q^\pi s, 0 \neq Q^\pi s, 1$. Since the update at state s at policy π depends only on the relative order of $Q^\pi(s, a)$ for $a \in A$, which vary continuously with respect to γ, we can change γ by a small amount without changing $I(\pi)$ for any policy π. Formally, the bound we present holds true even if we take $1 > \gamma_n \geq \frac{2^n}{2^n+1}$ for the MDP M_n, keeping the other components same.

Lemma 9. *For a policy π for M_n, a state s is switchable if and only if*

$$\sum_{s' \leq s} \pi(s) \equiv 0 \mod 2$$

Proof. It is easy to see from *Corollary 2.3* in [Melekopoglou and Condon, 1994] that

$$a(s+1) = \frac{(-1)^{\sum_{s' \leq s} \pi(s)}}{2^s} a(1) = \frac{(-1)^{\sum_{s' \leq s} \pi(s)}}{2^s} \frac{-1}{2}$$

Thus, $a(s+1)$ is negative if and only if

$$\sum_{s' \leq s} \pi(s) \equiv 0 \mod 2$$

Since $a(s+1) = a(s)(\frac{1}{2} - \pi(s))$, $a(s+1)$ is negative if and only if $\pi(s) = 0$ and $a(s) < 0$, or $\pi(s) = 1$ and $a(s) > 0$. Combined with *Corollary 2.4* in [Melekopoglou and Condon, 1994], the proof is complete. □

Definition 10. *For a policy π for M_n, we define $f(\pi) = \min$ states$(T^\pi) \cup \{n+1\}$.*

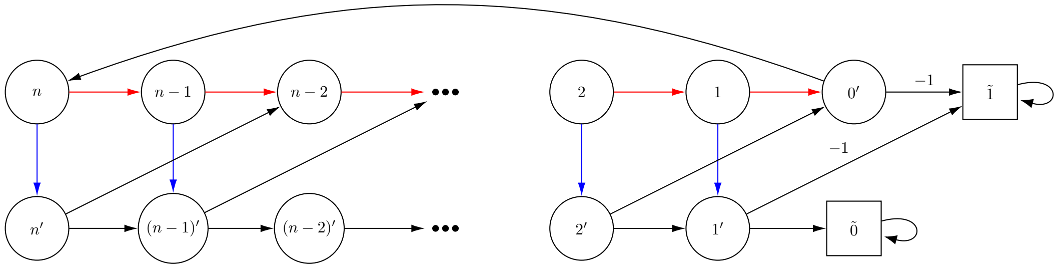


Figure 2: Family of MDPs used to prove the lower bound. Red and blue edges correspond to actions 0 and 1 respectively.

Lemma 11. *If a policy iteration algorithm visits the policies $\pi^0, \pi^1, ..., \pi^m$ then for $1 \le i \le m$, $f(\pi^{i-1}) \le f(\pi^i)$.*

Proof. Let $f(\pi^{i-1}) = s$. Since we stop when there are no improvable states, $f(\pi^{m-1}) < f(\pi^m) = n+1$. Otherwise assume that $i < m$. Since vertex s is the smallest switchable state, any state s' must not be switchable for $1 \le s' < s$. From Lemma 9, we have $\sum_{s'' \le s} \pi^{i-1}(s'') \equiv 1 \mod 2$ for $1 \le s' < s$. Since states $1, 2, ..., s-1$ are not switchable, we have $\pi^i(s') = \pi^{i-1}(s')$ for $1 \le s' < s$. Hence vertices $1, 2, ..., s-1$ should not be switchable in π^i. So $f(\pi^i) \ge s = f(\pi^{s-1})$. □

Corollary 12. *For policies π and π', if $\pi \preceq \pi'$, $f(\pi) \le f(\pi')$.*

Proof. We have a policy iteration algorithm that visits the policies $\pi^0, \pi^1, ..., \pi^m$ where $\pi^0 = \pi$ and $\pi^m = \pi'$, where the switching rule is to only switch improvable states where the policy differs from π'. Applying Lemma 11 gives us the above result. □

Lemma 13. *If Randomized Policy Iteration visits the policies $\pi^0, \pi^1, ..., \pi^m$ then for $1 \le i \le m$, $Pr[f(\pi^i) - f(\pi^{i-1}) \ge t] \le \frac{1}{2^t}$.*

Proof. Assume $f(\pi^{i-1}) = s$. Let $s' > s$ be the vertex with the smallest index after s which is not switchable. If all vertices after s are switchable, we let $s' = n+1$. Application of Lemma 9 shows that $\pi^{i-1}(s'') = 0$ for $s < s'' < s'$ and $\pi^{i-1}(s') = 1$ if $s' \le n$. Since s' is not switched, $\pi^i_{s'} = \pi^{i-1}(s') = 1$ and hence either s' or $s'-1$ should be switchable in π^i. So $f(S^i) \le k'$.
For $f(\pi^{i-1}) < f(\pi^i)$, s should not be switchable in π^i. Since states less than s are not switched, we require s to be switched. This happens with probability $\frac{1}{2}$. So $Pr[f(\pi^i) - f(\pi^{i-1}) \ge 1] \le \frac{1}{2}$.
For $f(\pi^i) = s+t$ where $t < s'-s$, we require $s+t$ to be switchable, s'' to be not switchable for $s \le s'' < s+t$. For this states s and $s+t$ must be switched in π^{i-1} and $s+t'$ must not be switched for $1 \le t' < t$. Since each switchable state is switched independently with probability $\frac{1}{2}$, the probability of this event is $\frac{1}{2^{t+1}}$. Thus $Pr[f(\pi^i) - f(\pi^{i-1}) = t] = \frac{1}{2^{t+1}}$ for $t < s'-s$. So $Pr[f(\pi^i) - f(\pi^{i-1}) \ge t] = \sum_{t'=t}^{t'=s'-s-1} Pr[f(\pi^i) - f(\pi^{i-1}) = t'] = \sum_{t'=t}^{t'=s'-s-1} \frac{1}{2^{t'+1}} \le \frac{1}{2^t}$. □

Definition 14. *We define $N : [n+1] \to \mathbb{R}_{\ge 0}$, where $N(i)$ is minimum number of iterations RPI takes in expectation to find an optimal policy, starting from any policy π^0 such that $f(\pi^0) = i$.*

Theorem 15. *The expected number of policies RPI evaluates before teaching the optimal policy on M_n is at least $n+1$.*

Proof. If $f(\pi^0) = n+1$, π^0 is already optimal and RPI halts. Hence $N(n+1) = 0$.
If $f(\pi^{i-1}) = s$, Lemma 13 shows that $Pr[f(\pi^i) \ge s+t] \le \frac{1}{2^t}$. Hence $N(s) \ge 1 + \sum_{t=0}^{n+1-s} \frac{N(s+t)}{2^{t+1}} = 1 + \sum_{t=0}^{n-s} \frac{N(s+t)}{2^{t+1}}$. For $s = n$, we have $N(s) = 2$. Assume $N(s') \ge n+2-s'$ for $n \ge s' > s$. Thus $N(s) \ge 1 + \frac{N(s)}{2} + \sum_{t=1}^{n+1-s} \frac{n+2-s-t}{2^{t+1}} \ge 1 + \frac{N(s)}{2} + \sum_{t=1}^{\infty} \frac{n+2-s-t}{2^{t+1}} = 1 + \frac{N(s)+n-s}{2}$. Thus $N(s) \ge n+2-s$. □

7 Data

While knowing a lower upper bound for PRI than for HPI does not mean that RPI performs better than HPI on the worst case, we do have some evidence supporting this statement in AUSOs. Through a computer search, we found out that there are 18 3-AUSOs, 16 of which are Holt-Klee and 12640 4-AUSOs, 6113 of which are Holt-Klee. Figures 4, 5, 6 and 6 show the number of policies visited by RPI and HPI on 3-AUSOs, 4-AUSOs, Holt-Klee 3-AUSOs and Holt-Klee 4-AUSOs respectively. The policy space for MDPs are known to be representable as Holt-Klee AUSOs, which are AUSOs with as many inner vertex disjoint paths from source to sink as their dimension. We ordered the AUSOs in the

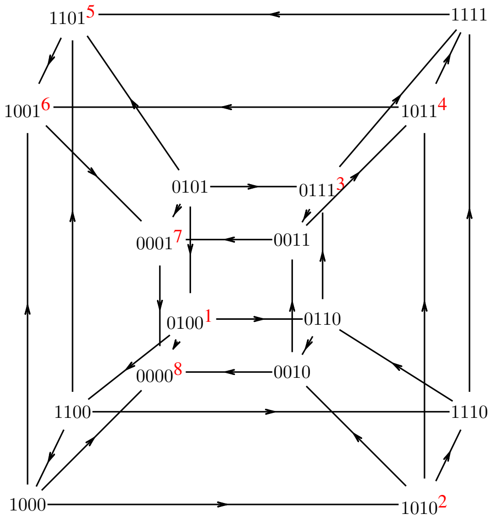


Figure 3: The only 4-AUSO with a 8-length run for Howard's PI. The run is marked with superscripts 1 through 8 on the policies in the run.

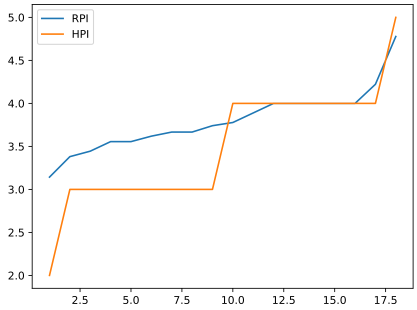


Figure 4: Expected number of policies visited by PI variants on 3-AUSOs

increasing order of number of policies visited by each PI algorithm, and plotted this number against their index in the order. The starting policies for each AUSO are kept as the policy from which the algorithm visits the maximum number of policies in the AUSO. Thus the right most and the left most points in these plots correspond to the worst case and the best case number of policies visited by these algorithms.

8 Conclusion

strong bounds, howard better than vi, best strong bounds not due to pi but lp route and specialised algorithms for ausos deterministic.

The tightest strong upper bounds currently known for MDP planning are summarised in Table **??**. The tightest bound for $k = 2$ is of the form $\text{poly}(n) \cdot 1.6059^n$. This bound is shown for the Fibonacci Seesaw algorithm, proposed by Szabó and Welzl [2001] for solving Unique Sink Orientations (USOs). It is known that the policies for 2-action MDPs can be interpreted as vertices of an Acyclic USO (AUSO), whose sink corresponds to an optimal policy. For $k \geq 3$, Kalyanakrishnan and Gupta [2017] introduced Recursive AUSOs (RAUSOs), which represent the policies of k-action MDPs and extended the Fibonacci Seesaw algorithm to these objects to give a bound of $k^{0.6834n}$.

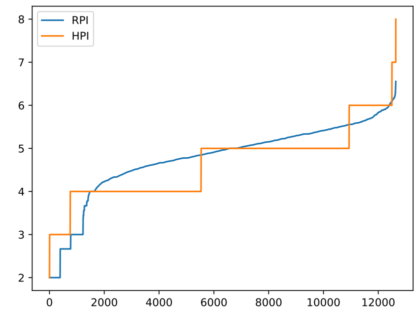


Figure 5: Expected number of policies visited by PI variants on 4-AUSOs

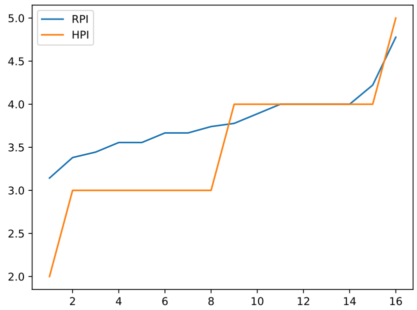


Figure 6: Expected number of policies visited by PI variants on Holt-Klee 3-AUSOs

If we look beyond PI, we find even *subexponential* bounds on the expected running time of MDP planning. Bounds of the form $\text{poly}(n, k) \cdot \exp(O(\sqrt{n \log(n)}))$ [Matoušek *et al.*, 1996] follow directly from posing MDP planning as a linear program with n variables and nk constraints [Littman *et al.*, 1995]. The special structure that results when $k = 2$ admits an even tighter bound of $\text{poly}(n) \cdot \exp(2\sqrt{n})$ [Gärtner, 2002].

Another common choice of picking rule is to pick uniformly at random from all available choices. We prove a

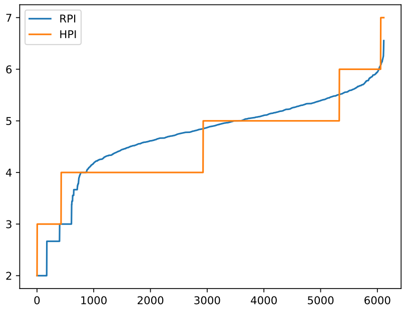


Figure 7: Expected number of policies visited by PI variants on Holt-Klee 4-AUSOs

$(k \log k)^{0.5n}$ upper bound on the expected running time of this PI variant. If we restrict ourselves to only picking policies wherein each improvable state is improved, we get a randomized version of Howard's PI. With a randomized version of the picking rule, wherein we select an improving action uniformly at random from all improving actions, we prove that Howard's PI takes at most $2(k \log k)^{0.5n}$ iterations in expectation.

Possibly make it 5 contributions in sections 1 and 3 if experiments turn out as expected.

References

References

[Bellman, 1957] Richard Bellman. *Dynamic Programming*. Princeton University Press, 1^{st} edition, 1957.

[Fearnley, 2010] John Fearnley. Exponential lower bounds for policy iteration. In *Proceedings of the Thirty-seventh International Colloquium on Automata, Languages and Programming (ICALP 2010)*, pages 551–562. Springer, 2010.

[Gärtner, 2002] Bernd Gärtner. The random-facet simplex algorithm on combinatorial cubes. *Random Structures and Algorithms*, 20(3):353–381, 2002.

[Gupta and Kalyanakrishnan, 2017] Anchit Gupta and Shivaram Kalyanakrishnan. Improved strong worst-case upper bounds for mdp planning. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 1788–1794, 2017.

[Hansen and Zwick, 2010] Thomas Dueholm Hansen and Uri Zwick. Lower bounds for Howard's algorithm for finding minimum mean-cost cycles. In *Proceedings of the Twenty-second International Symposium on Algorithms and Computation (ISAAC 2011)*, pages 425–426. Springer, 2010.

[Hollanders *et al.*, 2012] Romain Hollanders, Balázs Gerencsér, and Jean-Charles Delvenne. The complexity of policy iteration is exponential for discounted Markov decision processes. In *Proceedings of the Fifty-first IEEE Conference on Decision and Control (CDC 2012)*, pages 5997–6002. IEEE, 2012.

[Hollanders *et al.*, 2014] Romain Hollanders, Balázs Gerencsér, Jean-Charles Delvenne, and Raphaël M. Jungers. Improved bound on the worst case complexity of policy iteration, 2014. URL: http://arxiv.org/pdf/1410.7583v1.pdf.

[Howard, 1960] Ronald A. Howard. *Dynamic programming and Markov processes*. MIT Press, 1960.

[Kalyanakrishnan *et al.*, 2016] Shivaram Kalyanakrishnan, Utkarsh Mall, and Ritish Goyal. Batch-switching policy iteration. In *Proceedings of the Twenty-fifth International Joint Conference on Artificial Intelligence (IJCAI 2016)*, pages 3147–3153. AAAI Press, 2016.

[Littman *et al.*, 1995] Michael L. Littman, Thomas L. Dean, and Leslie Pack Kaelbling. On the complexity of solving Markov decision problems. In *Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence (UAI 1995)*, pages 394–402. Morgan Kaufmann, 1995.

[Mansour and Singh, 1999] Yishay Mansour and Satinder Singh. On the complexity of policy iteration. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence (UAI 1999)*, pages 401–408. Morgan Kaufmann, 1999.

[Matoušek *et al.*, 1996] Jiří Matoušek, Micha Sharir, and Emo Welzl. A subexponential bound for linear programming. *Algorithmica*, 16(4/5):498–516, 1996.

[Melekopoglou and Condon, 1994] Mary Melekopoglou and Anne Condon. On the complexity of the policy improvement algorithm for Markov decision processes. *INFORMS Journal on Computing*, 6(2):188–192, 1994.

[Puterman, 1994] Martin L. Puterman. *Markov Decision Processes*. Wiley, 1994.

[Schurr and Szabó, 2005] Ingo Schurr and Tibor Szabó. Jumping doesn't help in abstract cubes. In Michael Jünger and Volker Kaibel, editors, *Integer Programming and Combinatorial Optimization*, pages 225–235, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[Szabó and Welzl, 2001] Tibor Szabó and Emo Welzl. Unique sink orientations of cubes. In *Proceedings of the Forty-second Annual Symposium on Foundations of Computer Science (FOCS 2001)*, pages 547–555. IEEE, 2001.