Extended version of paper published in Proc. ICAPS 2025, pages 84--92, AAAI Press, 2025.

# Howard's Policy Iteration is Subexponential for Deterministic Markov Decision Problems with Rewards of Fixed Bit-size and Arbitrary Discount Factor

**Dibyangshu Mukherjee, Shivaram Kalyanakrishnan**

Department of Computer Science and Engineering, IIT Bombay, Mumbai, India
{dbnshu, shivaram}@cse.iitb.ac.in

**Abstract**

Howard's Policy Iteration (HPI) is a classic algorithm for solving Markov Decision Problems (MDPs). HPI uses a "greedy" switching rule to update from any non-optimal policy to a dominating one, iterating until an optimal policy is found. Despite its introduction over 60 years ago, the best-known upper bounds on HPI's running time remain exponential in the number of states—indeed even on the restricted class of MDPs with only deterministic transitions (DMDPs). Meanwhile, the tightest lower bound for HPI for MDPs with a constant number of actions per state is only linear. In this paper, we report a significant improvement: a *subexponential* upper bound for HPI on DMDPs, which is parameterised by the bit-size of the rewards, while independent of the discount factor. The same upper bound also applies to DMDPs with only two possible rewards (which may be of arbitrary size).

## 1 Introduction

Markov Decision Problems (MDPs) (Bellman 1967; Puterman 1994) are a well-studied abstraction of sequential decision making, which are widely used as a formal basis for automated control (Bertsekas 2007), planning (Mausam and Kolobov 2012) and reinforcement learning (Sutton and Barto 1998). An MDP models an *environment*, which comprises a set of states $S$ and a set of actions $A$. When action $a \in A$ is executed from state $s \in S$, the MDP transitions to a next state $s' \in S$, while also emitting a numeric reward $r \in \mathbb{R}$. Transitions are in general stochastic: the probability of reaching $s'$ by taking action $a$ from state $s$ is given by $T(s, a, s')$, where $T : S \times A \times S \to [0, 1]$ (satisfying $\sum_{s' \in S} T(s, a, s') = 1$ for $s \in S, a \in A$) is called the transition function of the MDP. Similarly, $r = R(s, a)$, where $R : S \times A \to \mathbb{R}$ is the reward function of the MDP.

An *agent* that interacts with an MDP goes through a sequence $(s^t, a^t, r^t)_{t=0}^{\infty}$, starting with initial state $s^0 \in S$. The choice of which action $a^t \in A$ to execute at each time step $t \geq 0$ lies with the agent. This choice influences both the immediate reward $r^t = R(s^t, a^t)$ and the next state $s^{t+1} \sim T(s^t, a^t, \cdot)$. Hence, to maximise its *long-term reward*, the agent's action choices must appropriately balance immediate and future rewards. Suppose that the agent fixes an action for each state—that is, it executes a *policy* $\pi : S \to A$. In other words, the agent takes action $a^t = \pi(s^t)$ for $t \geq 0$. For tasks with infinite horizons, there are two common definitions of long-term reward.

Under **discounted reward**, future rewards are discounted geometrically by a factor $\gamma \in [0, 1)$: the long-term reward (also called "value") of each state $s \in S$ under $\pi$ is

$$V_\gamma^\pi(s) = \mathbb{E}_\pi \left[ \sum_{t=0}^{\infty} \gamma^t R(s^t, a^t) \,\middle|\, s^0 = s \right]. \quad (1)$$

Observe that the value depends on the discount factor $\gamma$, which is specified as a part of the MDP.

An agent that follows policy $\pi$ also has a well-defined **average reward**, depending on its starting state $s \in S$. This long-term reward, called the *gain* of $\pi$ from $s$, is given by

$$V_{\mathrm{g}}^\pi(s) = \lim_{T \to \infty} \mathbb{E}_\pi \left[ \frac{1}{T} \sum_{t=0}^{T-1} R(s^t, a^t) \,\middle|\, s^0 = s \right]. \quad (2)$$

Accompanying its gain is $\pi$'s *bias*, defined for $s \in S$ as

$$V_{\mathrm{b}}^\pi(s) = \mathbb{E}_\pi \left[ \sum_{t=0}^{\infty} \left( R(s^t, a^t) - V_{\mathrm{g}}^\pi(s) \right) \,\middle|\, s^0 = s \right], \quad (3)$$

which is the cumulative difference over time between the reward and the average reward, starting from $s$.

For any given MDP, then, the natural problem is that of computing an *optimal* policy. In the discounted reward setting, it is established that for each input MDP $(S, A, T, R, \gamma)$, there exists an optimal policy $\pi^\star : S \to A$, which satisfies $V_\gamma^{\pi^\star}(s) \geq V_\gamma^\pi(s)$ for $s \in S$ and all $\pi : S \to A$. Similarly, under average reward, each input MDP $(S, A, T, R)$ has an optimal policy $\pi^\star : S \to A$, which satisfies $V_{\mathrm{g}}^{\pi^\star}(s) \geq V_{\mathrm{g}}^\pi(s)$ for $s \in S$ and all $\pi : S \to A$, and which additionally satisfies certain "optimality" equations (provided in Appendix A)

There are multiple algorithmic approaches to solve MDPs, including value iteration, linear programming, and policy iteration (Littman, Dean, and Kaelbling 1995). In this paper, we restrict our attention to Policy Iteration (PI) (Howard 1960; Puterman 1994), which is widely used. PI is also conceptually simple: initialised at some arbitrary policy, at each iteration PI switches to a dominating policy, unless none exists. The running time of PI depends on the

switching rule used for updating the policy. At one end of a spectrum are "Simplex" variants of PI that change the action at a single state, while the canonical greedy variant, called "Howard's PI" (Howard 1960) (HPI), switches actions at *all* "improvable" states. Yet other variants of PI (Mansour and Singh 1999; Kalyanakrishnan, Mall, and Goyal 2016) switch some subset of improvable states. It is natural to wonder which variant is the "most efficient"—and this question exposes a gap in our current theoretical understanding.

On the one hand, experiments have invariably shown HPI to significantly outperform other PI variants (Kalyanakrishnan, Mall, and Goyal 2016; Taraviya and Kalyanakrishnan 2020). However, theoretical upper bounds on the iterations taken by HPI are still much looser than those for other variants. Consider finite MDPs with $|S| = n \geq 2$ states and $|A| = k \geq 2$ actions. Whereas upper bounds of $(O(\log k))^n$ (Kalyanakrishnan, Misra, and Gopalan 2016) and $O(k^{0.7207n})$ (Gupta and Kalyanakrishnan 2017) have been shown for other variants, the tightest known upper bounds for HPI are $O(\frac{k^n}{n})$ (Mansour and Singh 1999)—only a linear improvement over the trivial bound of $k^n$. A similar trend plays out on Deterministic MDPs (DMDPs), the restricted class of MDPs in which every transition has a fixed next state: in other words, all transition probabilities are either 0 or 1. Whereas DMDPs can be solved in strongly polynomial time (Karp 1978; Madani 2002b), and are in fact done so even by a Simplex variant of PI (Post and Ye 2013), the best upper bound for HPI on DMDPs remains exponential (Goenka et al. 2025).

Over the years, multiple researchers have explicitly earmarked improving the upper bound for HPI as an important target for theoretical research (Schmitz 1985; Scherrer 2013; Post and Ye 2013). This paper presents a significant step in this direction: a *subexponential* upper bound for HPI on DMDPs, parameterised by the "bit-size" of the rewards.

### 1.1 Bit-size of rewards

Consider DMDP $M = (S, A, T, R, \gamma)$ under discounted reward, or $M = (S, A, T, R)$ under average reward. Let $b'(R)$ denote the smallest positive integer such that for each $(s, a) \in S \times A$, $R(s, a)$ belongs to the set $\{0, 1, 2, \ldots, 2^{b'(R)} - 1\}$ (with the convention that $b'(R) = \infty$ if no finite integer satisfies this requirement for $R$). Thus, if the only rewards in $R$ are 2, 3, and 5, we would have $b'(R) = 3$. We note that the sequence of policies visited by HPI on $M$ remains identical if the rewards in $M$ are positively scaled and/or shifted. Define, for $\alpha > 0, \beta \in \mathbb{R}$, and $(s, a) \in S \times A$: $R_{\alpha,\beta}(s, a) \stackrel{\text{def}}{=} \alpha R(s, a) + \beta$. We define the bit-size of $R$ by

$$b(R) \stackrel{\text{def}}{=} \min_{\alpha>0,\beta\in\mathbb{R}} b'(R_{\alpha,\beta}). \tag{4}$$

By this definition, $b(R)$ is 3 not only for the set of rewards $\{2, 3, 5\}$, but among others, also for $\{0.2, 0.3, 0.5\}$, $\{2\sqrt{3}, 3\sqrt{3}, 5\sqrt{3}\}$, and $\{-0.8, -0.7, -0.5\}$. Note as a special case that $b(R) = 1$ for any reward function that takes exactly two values, since those values can be scaled and shifted to $\{0, 1\}$. In theory, $b(R)$ could be infinite (an example is when 1, $\sqrt{2}$, and $\sqrt{3}$ are rewards). However, in applications of MDPs, rewards are invariably encoded in `float` or `double` variables, which are of fixed bit-size (say 32 or 64). In games such as Chess and Go, and tasks where the only rewards are from success and failure, the reward set is typically $\{0, 1\}$ or $\{-1, 0, 1\}$ (Silver et al. 2016, 2018). In yet other tasks, reward is based on the number of time steps elapsed or energy expended, usually discretised to take a few tens or hundreds of contiguous integer values (Crites and Barto 1995). It is therefore reasonable to consider $b(R)$ a *constant* for tasks encountered in practice. Henceforth we shall drop "$(R)$" from "$b(R)$" when the context is clear.

### 1.2 Upper bounds and proof sketches

The effect of $b$ on the complexity of Policy Iteration is relatively straightforward to characterise under average reward. In this case, the gain-bias pair of every state, under every policy, is observed to belong to a finite set of size $\text{poly}(n, 4^b)$. Regardless of which Policy Iteration variant is used, the strict monotonicity of the algorithm yields the following upper bound.

**Proposition 1.1.** *Let $M = (S, A, T, R)$ be a DMDP using average reward, with $|S| = n$ and $|A| = k$. On $M$, any Policy Iteration algorithm can visit at most $O\left(n^5 \cdot 4^b\right)$ policies.*

A proof of the proposition is given in Appendix A.

The pseudopolynomial dependence on the rewards is a novel result for HPI. However, the simple argument behind this upper bound for average reward does not carry over to discounted reward. With discounted reward, the number of possible values a state could have (under different policies) scales exponentially, rather than polynomially, in $n$. See section 2.4. Our main result is that the number of iterations taken by HPI still only scales *subexponentially* in $n$ if $b$ is a constant, or even if $b = o(n^{1-\epsilon})$ for some $\epsilon > 0$.

**Theorem 1.2.** *Let $M = (S, A, T, R, \gamma)$ be a DMDP using discounted reward, with $|S| = n$ and $|A| = k$. On $M$, HPI can visit at most $nk \cdot u \exp(u)$ policies, where $u = O\left(\sqrt{nb} \log \frac{n}{b} + b\right)$.*

The trajectory followed by HPI on the input MDP depends on the discount factor $\gamma$. First, we show that there exists a threshold discount factor, denoted $\gamma_{\text{Q}}$, beyond which the trajectory of HPI is invariant. In other words, for two MDPs $M_\gamma$ and $M_{\gamma'}$ that differ only in their discount factors $\gamma$ and $\gamma'$, HPI will follow the same trajectory as long as $\gamma, \gamma' \in (\gamma_{\text{Q}}, 1)$. An upper bound parameterised by $\gamma$, of $O\left(\frac{nk}{1-\gamma} \log \frac{1}{1-\gamma}\right)$ iterations, is already known for HPI (Ye 2011; Scherrer 2013). For $\gamma > \gamma_{\text{Q}}$, the invariance of the algorithm's trajectory therefore implies an upper bound of $O\left(\frac{nk}{1-\gamma_{\text{Q}}} \log \frac{1}{1-\gamma_{\text{Q}}}\right)$ iterations. Our second innovation is to obtain the subexponential-in-$n$ upper bound on $\frac{1}{1-\gamma_{\text{Q}}}$. This is achieved by showing that $\gamma_{\text{Q}}$ is the largest root to the left of 1 of a certain polynomial with integer coefficients with magnitude $O(2^b)$. We combine some existing results on root-separation in such polynomials to obtain our upper bound.

The average reward setting provides useful intuitions to reason about discounted rewards. We draw connections between these settings where appropriate, but defer the detailed analysis under average reward to Appendix A. The main text remains focused on discounted rewards; the proof of Theorem 1.2 is given in Section 4. We proceed to formal definitions of PI and DMDPs (Section 2), followed by a survey of related work (Section 3).

# 2 Background

In this section, we provide the necessary background on Policy Iteration under discounted reward, and also on DMDPs, thus laying the foundation for our main result in Section 4.

## 2.1 Policy Iteration

Policy Iteration (PI) builds upon the idea that if a policy $\pi : S \to A$ is non-optimal, then it is possible to efficiently identify a set of policies that are guaranteed to improve upon $\pi$ (in terms of long-term reward). Any policy $\pi' : S \to A$ from this improving set is then made the current policy, and the process continued. Since there are only a finite number of policies ($k^n$), and no policy can repeat due to the monotonic improvement, the process is guaranteed to terminate at an optimal policy after a finite number of iterations.

**Policy evaluation** The first step in each iteration of PI is *policy evaluation*: that is, to compute the value of $\pi$. State values as defined in (1) can be computed as the solution of a system of linear equations called the Bellman equations (Bellman 1967): for $s \in S$,

$$V_\gamma^\pi(s) = R(s, \pi(s)) + \gamma \sum_{s' \in S} T(s, \pi(s), s')\, V_\gamma^\pi(s'). \quad (5)$$

**Policy improvement** The basis for policy improvement is to consider the effect of taking an alternative action $a \in A$ from $s$ only for the first time step, and thereafter acting according to $\pi$ (at time steps $2, 3, 4, \dots$). These long-term rewards are called Q-values, and are defined as follows for each $(s, a) \in S \times A$:

$$Q_\gamma^\pi(s, a) = R(s, a) + \gamma \sum_{s' \in S} T(s, a, s')\, V_\gamma^\pi(s'). \quad (6)$$

The intuition behind policy improvement is that if it is more rewarding to take action $a$ from state $s$ just once and then follow $\pi$ (the long-term reward for this is precisely the Q-value), then it must also be beneficial to take $a$ from $s$ all the time—that is, whenever the agent finds itself in $s$. The latter would amount to following a new policy that takes $a$ from $s$, but which mimics $\pi$ at all other states. Continuing with the same intuition, one could similarly switch to actions with higher Q-values simultaneously in any number of states where such actions are available. The "policy improvement theorem", given below, formalises this intuition.

**Theorem 2.1** (Policy Improvement)**.** *Consider MDP $M = (S, A, T, R, \gamma)$ using discounted reward.*

1. *Suppose policy $\pi : S \to A$ satisfies $Q_\gamma^\pi(s, a) \le V_\gamma^\pi(s)$ for all $s \in S, a \in A$. Then $\pi$ is an optimal policy.*
2. *Suppose policies $\pi : S \to A$ and $\pi' : S \to A$ satisfy $Q_\gamma^\pi(s, \pi'(s)) \ge V_\gamma^\pi(s)$ for all $s \in S$, and moreover, there exists $s \in S$ such that $Q_\gamma^\pi(s, \pi'(s)) > V_\gamma^\pi(s)$. Then $V_\gamma^{\pi'}(s) \ge V_\gamma^\pi(s)$ for all $s \in S$, and there exists $s \in S$ such that $V_\gamma^{\pi'}(s) > V_\gamma^\pi(s)$.*

The proof of Theorem 2.1 relies on defining an operator that becomes a contraction mapping in a Banach space, on account of the discount factor $\gamma$ being smaller than 1 (Bertsekas 2007; Szepesvári 2010).

## 2.2 Howard's PI

Notice from Theorem 2.1 that once $\pi$ is evaluated, in general there could be multiple choices of improving policies $\pi'$. Any action $a \in A$ that has a higher Q-value for state $s \in S$ is called an "improving action", and $s$ itself is an "improvable state". As long as we switch to some improving action in one or more states, and we retain the actions of $\pi$ at the other states, the resulting policy $\pi'$ is guaranteed to strictly dominate $\pi$. HPI is a greedy variant of PI in which *every* state that has some improving action is necessarily switched. If there are multiple improving actions, any one with the highest Q-value is selected. Below is a full specification of HPI.

HPI starting from policy $\pi$ under discounted reward

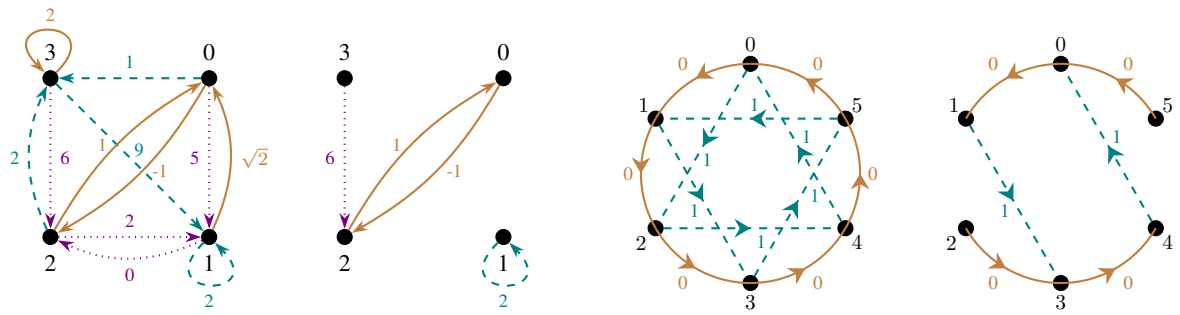
1. For $s \in S$: set $\pi'(s) \leftarrow \arg\max_{a \in A} Q_\gamma^\pi(s, a)$, breaking ties in favour of $\pi(s)$ if possible, else arbitrarily.
2. If $\pi' \neq \pi$, set $\pi \leftarrow \pi'$ and go to 1.
3. Declare $\pi$ to be optimal and terminate.

If each reward in $R$ is scaled by $\alpha > 0$ and shifted by $\beta \in \mathbb{R}$, then each Q-value is scaled by $\alpha$ and shifted by $\frac{\beta}{1-\gamma}$. From the code above, it is clear that positive-scaling and shifting have no effect on the trajectory taken by HPI (except possibly for tie-breaking). This invariance validates our definition of $b(R)$.

## 2.3 DMDPs

In a DMDP $M = (S, A, T, R, \gamma)$, transition probabilities are all 0 or 1. Hence for $s \in S, a \in A$, it becomes convenient to denote as $T(s, a)$ the single next state $s'$ for which $T(s, a, s') = 1$. DMDP $M$ induces the multi-graph $G_M = (\mathcal{V}, \mathcal{E}, w)$, which encodes the states of $M$ as its vertices, the transitions as edges, and the rewards as edge weights. Formally, (1) $\mathcal{V} = S$; (2) for $v_1, v_2 \in \mathcal{V}$ and $a \in A$, $(v_1, v_2, a) \in \mathcal{E}$ iff $T(v_1, a) = v_2$; and (3) $w : \mathcal{E} \to \mathbb{R}$ is such that for $v_1, v_2 \in \mathcal{V}$ and $a \in A$, $w((v_1, v_2, a)) = R(v_1, a)$. Figure 1a shows an example of the induced graph.

**Paths and cycles induced by DMDPs.** To study any fixed policy $\pi : S \to A$, we may consider the subgraph $G_{M,\pi}$ obtained by dropping from $G_M$ all the edges *not* corresponding to actions taken by $\pi$. Thus, $G_{M,\pi}$ has exactly one edge—$(s, T(s, \pi(s)), \pi(s))$—for each state $s \in S$. For each $s \in S$, observe that starting from $s$ and repeatedly taking actions according to $\pi$ yields a (possibly empty) path, which is followed by a cycle. Let $C_s^\pi$ be the sequence of state-action



(a) $G_M$ (b) $G_{M,\pi}$ (c) $G_{M_2}$ (d) $G_{M_2,\pi_1}$

Figure 1: In (a) we see graph $G_M$ induced by DMDP $M$ with $S = \{0, 1, 2, 3\}$, $A = \{0 \text{ (solid)}, 1 \text{ (dashed)}, 2 \text{ (dotted)}\}$. Each transition is annotated with the reward. Subfigure (b) shows the subgraph $G_{M,\pi}$ obtained by fixing policy $\pi$ such that $\pi(0) = 0, \pi(1) = 1, \pi(2) = 0, \pi(3) = 2$. Starting from each state $s$ in $G_{M,\pi}$ is a (possibly null) path $P_s^\pi$ and a cycle $C_s^\pi$ (defined in Section 2.3 as a sequence of state-action pairs). For example: $P_1^\pi = \emptyset, C_1^\pi = \langle(1,1)\rangle$; $P_2^\pi = \emptyset, C_2^\pi = \langle(2,0),(0,0)\rangle$; and $P_3^\pi = \langle(3,2)\rangle, C_3^\pi = \langle(2,0),(0,0)\rangle$. In (c) we see $G_{M_m}$ for $m = 2$, containing 6 states and 2 actions. The family $M_m$ is described in Section 2.4. Subfigure (d) shows the subgraph of $G_M$ induced by some fixed policy $\pi_1$. Notice that state 0 is on a cycle with 4 edges, exactly 2 of which have a reward of 1.

pairs in the cycle that is reached from $s$ by following $\pi$, beginning with the first state in the cycle that is thereby reached. Let $c_s^\pi$ be the length of $C_s^\pi$. Let $P_s^\pi$ be the sequence of state-action pairs in the path emanating from $s$ under $\pi$, with its final element being the state-action pair that reaches the starting state of $C_s^\pi$. Let $p_s^\pi$ be the length of $P_s^\pi$. Figure 1a provides some examples of $P_s^\pi$ and $C_s^\pi$ along with the DMDP shown for illustration. Observe that in general, $0 \le p_s^\pi \le n-1$ and $1 \le c_s^\pi \le n$ for all $\pi : S \to A$ and $s \in S$. For $1 \le i \le p_s^\pi$, we denote by $P_s^\pi[i]$ the $i$-th element of $P_s^\pi$. Similarly, for $1 \le i \le c_s^\pi$, we denote by $C_s^\pi[i]$ the $i$-th element of $C_s^\pi$. If, say, this element is $(\bar{s}, \bar{a}) \in S \times A$, then $R(C_s^\pi[i])$ shall denote $R(\bar{s}, \bar{a})$.

With this notation, we obtain for DMDP $(S, A, T, R, \gamma)$ using discounted reward, that for $s \in S$,

$$V_\gamma^\pi(s) = \sum_{i=1}^{p_s^\pi} \gamma^{i-1} R(P_s^\pi[i]) + \frac{\gamma^{p_s^\pi}}{1-\gamma^{c_s^\pi}} \sum_{i=1}^{c_s^\pi} \gamma^{i-1} R(C_s^\pi[i]). \quad (7)$$

Also, simplifying (6) for DMDPs yields for $s \in S, a \in A$:

$$Q_\gamma^\pi(s,a) = R(s,a) + \gamma V_\gamma^\pi(T(s,a)). \quad (8)$$

### 2.4 Average reward versus discounted reward

Before proceeding, we describe a technical difference between the average reward and discounted reward settings, which illustrates why the latter is more challenging for PI.

Consider the family of DMDPs $M_m$, where for $m \ge 1$, we have $n = 3m$ and $k = 2$. For each state $s \in S \stackrel{\text{def}}{=} \{0, 1, \ldots, n-1\}$, action 0 earns a reward of 0 and action 1 a reward of 1. Also, for $s \in S$, $T(s,0) = (s+1) \mod n$, while $T(s,1) = (s+2) \mod n$. Figure 1c shows the graph induced by $M_2$.

Now, let $S_{2m}$ denote the set of bit-strings of length $2m$ that contain an equal number of 0's and 1's. Clearly, $|S_{2m}| = \binom{2m}{m} = \binom{2n/3}{n/3} = \exp(\Omega(n))$. We observe that each bit-string in $S_{2m}$ can be mapped to a cycle of length $2m$ in $M_m$, which passes through state 0, and with a sequence of rewards the same as in the bit-string. For example, Figure 1d shows a cycle corresponding to bit-string 0101. In the average reward setting, all such cycles will have the *same* gain, of $\frac{1}{2}$. However, the (discounted) value of state 0 in the cycle from Figure 1d is $\frac{\gamma+\gamma^3}{1-\gamma^4}$. For appropriate choices of $\gamma$, each of the $|S_{2m}|$ cycles can yield a different value for state 0, hence resulting in exponentially many values. Roughly speaking, this disparity means that a single iteration of PI under average reward could potentially require an exponential number of iterations to cover under discounted reward. In Section 4 we show that HPI, however, can visit at most a subexponential number of cycles (and policies) as a function of $n$.

## 3 Related Work

Mansour and Singh (1999) established that any run of HPI can take at most $O\left(\frac{k^n}{n}\right)$ iterations, providing the first non-trivial upper bound for the algorithm. Hollanders, Delvenne, and Jungers (2012) later improved this bound by a constant factor. To date, the bound of $O\left(\frac{k^n}{n}\right)$ iterations—only a linear improvement over the trivial bound of $k^n$ iterations—remains the tightest known bound for HPI on general MDPs (among those depending solely on $n$ and $k$).

Experiments suggest that HPI may be much more efficient on MDPs than the current upper bound indicates. Currently-known lower bounds do not rule out this possibility. In an important breakthrough, Fearnley (2010) constructed an MDP with a path of length $\exp(\Omega(n))$ for HPI. More recently, Christ and Yannakakis (2023) have shown that a form of smoothed complexity (Spielman and Teng 2004) for HPI (with an appropriate perturbation model) is

super-polynomial, of the form $\exp(\Omega(n^{\frac{1}{3}}))$. However, these results do not settle the complexity of HPI. Notably, in both constructions, $k$, which is the number of actions per state in the MDP, is *not* a free parameter. Rather, $k$ is set to be $\Theta(n)$, where $n$ is the number of states. As yet, the tightest lower bound for HPI on MDPs with constant $k$ is only $\Omega(n)$ (Hansen and Zwick 2010). MDPs with $k = 2$ actions induce abstract cubes called acyclic unique sink orientations (AUSOs) (Szabó and Welzl 2001). The vertices of the AUSO correspond to policies of the MDP, and oriented edges encode the direction of improvement between neighbours. HPI does have a lower bound of $\exp(\Omega(n))$ iterations when run on AUSOs (Schurr and Szabó 2005). However, not all AUSOs are induced by MDPs; it remains unknown if exponentially-long chains are possible for HPI on 2-action MDPs.

Ye (2011) provides an upper bound of poly $\left(n, k, \frac{1}{1-\gamma}\right)$ iterations for HPI on MDPs; this result was later refined by Hansen, Miltersen, and Zwick (2013) and Scherrer (2013). Our analysis, which ultimately yields a $\gamma$-independent bound, relies on the following result of Scherrer (2013).

**Theorem 3.1.** *Let $M = (S, A, T, R, \gamma)$ be an MDP using discounted reward, with $|S| = n$ and $|A| = k$. On $M$, HPI can visit at most $O\left(\frac{nk}{1-\gamma} \log \frac{1}{1-\gamma}\right)$ policies.*

Although DMDPs are simpler than MDPs, they have themselves challenged theoretical research for many years now (Karp 1978; Papadimitriou and Tsitsiklis 1987; Madani 2002b) While Papadimitriou and Tsitsiklis (1987), and also Madani, Thorup, and Zwick (2010), have shown strongly-polynomial algorithms for DMDPs, there is still a gap between upper and lower bounds. Post and Ye (2013) analyse max-gain Simplex specifically on DMDPs, and show a strongly-polynomial upper bound, which was later improved by Hansen, Kaplan, and Zwick (2014).

HPI, typically faster in practice than Simplex, is conspicuous by its absence from the results above. Post and Ye (2013) specifically earmark extending their techniques to HPI as "a difficult but natural next step". To the best of our knowledge, the only advance that has subsequently been made is by Goenka et al. (2025), who have shown an upper bound of roughly poly$(n, k) \cdot \left(\frac{k}{e}\right)^n$ iterations (when $k$ is large) for HPI on DMDPs. Constraints arising from rewards have not been sufficiently utilised in preceding analyses. It is also to note that lower bound constructions for HPI (Hansen and Zwick 2010; Fearnley 2010; Christ and Yannakakis 2023) invariably require rewards whose magnitude increases exponentially in $n$—as is seldom the case in practice. Setting out specifically to examine the role of rewards, our work takes a first step by assuming finite bit-sizes. In practice rewards are set by designers; our assumption is somewhat more natural than assumptions made on the transitions ("ADAG" structure by Madani (2002a); partitioned state space by Scherrer (2013)) to obtain tighter bounds for HPI.

The technical core of our analysis is similar to the method used by Grand-Clément and Petrik (2023), though with a different objective. While these authors aim to establish an upper bound on $\gamma_{\text{bw}}$, the so-called "Blackwell" discount factor of an MDP, we focus on tracking the trajectory of a classical algorithm to demonstrate that $\gamma_{\text{Q}}$ (unknown to the algorithm) constrains its running time nonetheless.

The reader might be curious if an explicit *lower bound* can be shown for HPI in terms of $b$. Recall that with no constraint on $b$ (effectively, $b \to \infty$), the tightest lower bound is currently only $\Omega(n)$ (Hansen and Zwick 2010). This lower bound can be achieved up to a constant factor even with $b = 1$ on a DMDP, as shown in Figure 2. Unless a superlinear lower bound is achieved for unrestricted rewards, pursuing a $b$-dependent lower bound does not appear worthwhile.

## 4 Pseudosubexponential Upper Bound for Discounted Reward

In this section, we derive our subexponential-in-$n$ upper bound for HPI under discounted reward. To begin, in Section 4.1, we define a "threshold" discount factor $\gamma_{\text{Q}}$. In Section 4.2, we consider and establish the properties of the "Q-difference sign polynomial", which is central to our analysis. In Section 4.3, we provide a standalone mathematical result, which we obtain by adapting existing analyses of the roots of polynomials with integer coefficients. This completes the list of ingredients required for our final proof, which we furnish in Section 4.4.

### 4.1 Threshold discount factor

Consider DMDP $M = (S, A, T, R, \gamma)$. For policy $\pi : S \to A$; $s \in S$; $a, a' \in A$, we define

$$\gamma^{\pi}_{s,a,a'} \stackrel{\text{def}}{=} \inf \left\{ \gamma \in [0,1) \,\middle|\, \forall \tau \in (\gamma, 1), \right.$$
$$\left. \left(Q^{\pi}_{\gamma}(s,a) > Q^{\pi}_{\gamma}(s,a') \implies Q^{\pi}_{\tau}(s,a) > Q^{\pi}_{\tau}(s,a')\right) \right\}.$$

In other words, $\gamma^{\pi}_{s,a,a'}$ is the greatest lower bound on $\gamma \in [0,1)$ such that if $a$ has a larger Q-value than $a'$ under $\gamma$-discounting, then it also has a larger Q-value under $\tau$-discounting for all $\tau \in (\gamma, 1)$. Although it may not be apparent from the definition, our upcoming working will show

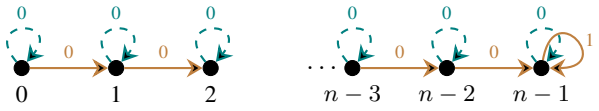



Figure 2: Induced graph of a DMDP with states $S = \{0, 1, \ldots, n-1\}$ and actions 0 (dashed) and 1 (solid). Annotations show the reward on each transition. Discount factor $\gamma$ is any element of $(0, 1)$. Policies are encoded as $n$-bit strings: bit $i$ specifies the action taken at state $i-1$. For starting policy $0^n$, the only improving action is at state $n-1$. Hence every PI algorithm must proceed from $0^n$ to $0^{n-1}1$. Now, for policy $0^{n-1}1$, the only improving action is at state $n-2$. Hence every PI algorithm must proceed from $0^{n-1}1$ to $0^{n-2}1^2$. The same pattern continues until, after $n$ total iterations, the optimal policy $1^n$ is reached.

that $\gamma^{\pi}_{s,a,a'}$ always exists in $[0,1)$. Consequently, our threshold discount factor

$$\gamma_{\mathrm{Q}} \stackrel{\text{def}}{=} \max_{\pi:S\to A; s\in S; a,a'\in A} \gamma^{\pi}_{s,a,a'} \tag{9}$$

is also well-defined and guaranteed to lie in $[0,1)$.

It is insightful to compare $\gamma_{\mathrm{Q}}$ with the "Blackwell discount factor" $\gamma_{\mathrm{bw}}$ that is defined by Grand-Clément and Petrik (2023). "Blackwell-optimality" (Blackwell 1962) is an alternative to discounted reward and average reward for defining optimal policies. Like average reward, Blackwell optimality does not require the specification of a discount factor, and may hence be defined for any MDP $M = (S, A, T, R)$. A policy $\pi : S \to A$ is said to be Blackwell-optimal for $M$ if there exists $\gamma \in [0,1)$ such that $\pi$ is discount-optimal for every discount factor $\gamma' \in (\gamma, 1)$. Denote the set of discount-optimal policies for $M$ with discount factor $\gamma$ by $\Pi^{\star}_{\gamma}$, and the set of Blackwell-optimal policies for $M$ by $\Pi^{\star}_{\mathrm{bw}}$. Grand-Clément and Petrik (2023) define the Blackwell discount factor for $M$ as

$$\gamma_{\mathrm{bw}} \stackrel{\text{def}}{=} \inf\left\{\gamma \in [0,1) \;\middle|\; \forall \gamma' \in (\gamma,1)\left(\Pi^{\star}_{\gamma'} = \Pi^{\star}_{\mathrm{bw}}\right)\right\}. \tag{10}$$

By implication, for every $\gamma \in (\gamma_{\mathrm{bw}}, 1)$, any optimal policy with $\gamma$-discounting is also Blackwell-optimal. The significance of this observation arises from a historical context, wherein the computation of Blackwell-optimal policies has not been straightforward. Many procedures for computing Blackwell-optimal policies are relatively complex to implement. In contrast, Grand-Clément and Petrik (2023) offer a simple computational recipe when the bit-size of the input MDP is bounded. They derive an explicit upper bound on $\gamma_{\mathrm{bw}}$, which depends on $n$, $k$, as well as the number of bits used to represent the transition probabilities and rewards in $M$. By choosing any discount factor $\gamma$ that exceeds this upper bound, one only has to compute a discount-optimal policy with $\gamma$-discounting in order to obtain a Blackwell-optimal policy. Although $\gamma_{\mathrm{Q}}$ is syntactically similar to $\gamma_{\mathrm{bw}}$, its intended purpose in this paper is different. We do not design a new algorithm based on $\gamma_{\mathrm{Q}}$, but rather, highlight its role in determining the running time of an existing, classical algorithm. Correspondingly, $\gamma_{\mathrm{Q}}$ is intimately involved with each step performed by HPI, whereas $\gamma_{\mathrm{bw}}$ is only useful to characterise a subset of Blackwell-optimal policies. It can be seen that $\gamma_{\mathrm{Q}}$ is lower-bounded by $\gamma_{\mathrm{bw}}$.

**Lemma 4.1.** *For every DMDP $(S, A, T, R)$, $\gamma_{bw} \le \gamma_{Q}$.*

*Proof.* Fix $\gamma_0 \in (\gamma_{\mathrm{Q}}, 1)$ and an arbitrary policy $\overline{\pi} \in \Pi^{\star}_{\gamma_0}$. Since $\overline{\pi}$ is optimal with $\gamma_0$-discounting, we have that for each $s \in S; a \in A$,

$$Q^{\overline{\pi}}_{\gamma_0}(s, \overline{\pi}(s)) \ge Q^{\overline{\pi}}_{\gamma_0}(s, a). \tag{11}$$

Now suppose there exist $s \in S, a \in A$, and $\gamma_1 \in (\gamma_{\mathrm{Q}}, \gamma_0)$ such that $Q^{\overline{\pi}}_{\gamma_1}(s,a) > Q^{\overline{\pi}}_{\gamma_1}(s, \overline{\pi}(s))$. Since $\gamma_1 > \gamma_{\mathrm{Q}}$, and therefore $\gamma_1 > \gamma^{\overline{\pi}}_{s,a,\overline{\pi}(s)}$, and therefore $\gamma_0 > \gamma^{\overline{\pi}}_{s,a,\overline{\pi}(s)}$, it follows that $Q^{\overline{\pi}}_{\gamma_0}(s,a) > Q^{\overline{\pi}}_{\gamma_0}(s,\overline{\pi}(s))$—which contradicts (11). We conclude that for $s \in S, a \in A, \gamma_1 \in (\gamma_{\mathrm{Q}}, \gamma_0)$,

$$Q^{\overline{\pi}}_{\gamma_1}(s, \overline{\pi}(s)) \ge Q^{\overline{\pi}}_{\gamma_1}(s, a). \tag{12}$$

In other words, if $\overline{\pi}$ is optimal with $\gamma_0$-discounting, it must be optimal with $\gamma_1$-discounting for all $\gamma_1 \in (\gamma_{\mathrm{Q}}, \gamma_0)$. From (10), we observe that $\gamma_{\mathrm{bw}}$ cannot exceed $\gamma_{\mathrm{Q}}$. □

## 4.2 Q-difference sign polynomial

As detailed in Section 2.1, the basis for policy improvement is the comparison of Q-values of different actions at each state. For the working below, fix $\pi : S \to A; s \in S$; and $a, a' \in A$. From (7) and (8), we have

$$\begin{aligned}
Q^{\pi}_{\gamma}(s,a) &= R(s,a) + \sum_{i=1}^{p^{\pi}_{T(s,a)}} \gamma^i R(P^{\pi}_{T(s,a)}[i]) \\
&\quad + \frac{\gamma^{p^{\pi}_{T(s,a)}}}{1-\gamma^{c^{\pi}_{T(s,a)}}} \sum_{i=1}^{c^{\pi}_{T(s,a)}} \gamma^i R(C^{\pi}_{T(s,a)}[i]); \\
Q^{\pi}_{\gamma}(s,a') &= R(s,a') + \sum_{i=1}^{p^{\pi}_{T(s,a')}} \gamma^i R(P^{\pi}_{T(s,a')}[i]) \\
&\quad + \frac{\gamma^{p^{\pi}_{T(s,a')}}}{1-\gamma^{c^{\pi}_{T(s,a')}}} \sum_{i=1}^{c^{\pi}_{T(s,a')}} \gamma^i R(C^{\pi}_{T(s,a')}[i]).
\end{aligned}$$

Define

$$\begin{aligned}
f^{\pi}_{s,a,a'}(\gamma) &\stackrel{\text{def}}{=} \left(Q^{\pi}_{\gamma}(s,a) - Q^{\pi}_{\gamma}(s,a')\right) \\
&\quad \times \left(1-\gamma^{c^{\pi}_{T(s,a)}}\right) \cdot \left(1-\gamma^{c^{\pi}_{T(s,a')}}\right).
\end{aligned} \tag{13}$$

We observe that the sign of $Q^{\pi}_{\gamma}(s,a) - Q^{\pi}_{\gamma}(s,a')$ is the same as the sign of $f^{\pi}_{s,a,a'}$. Hence, $a$ has a higher Q-value than $a'$ at $s$ under $\pi$ if and only if $f^{\pi}_{s,a,a'}(\gamma) > 0$. By expanding out $f^{\pi}_{s,a,a'}$, we observe that

$$\begin{aligned}
f^{\pi}_{s,a,a'}(\gamma) &= f_1(\gamma) + f_2(\gamma) + f_3(\gamma) + f_4(\gamma), \text{ where} \\
f_1(\gamma) &= \left(R(s,a) + \sum_{i=1}^{p^{\pi}_{T(s,a)}} \gamma^i R(P^{\pi}_{T(s,a)}[i])\right) \\
&\quad \times \left(1-\gamma^{c^{\pi}_{T(s,a)}}\right)\left(1-\gamma^{c^{\pi}_{T(s,a')}}\right), \\
f_2(\gamma) &= -\left(R(s,a') + \sum_{i=1}^{p^{\pi}_{T(s,a')}} \gamma^i R(P^{\pi}_{T(s,a')}[i])\right) \\
&\quad \times \left(1-\gamma^{c^{\pi}_{T(s,a)}}\right)\left(1-\gamma^{c^{\pi}_{T(s,a')}}\right), \\
f_3(\gamma) &= \left(\gamma^{p^{\pi}_{T(s,a)}} \sum_{i=1}^{c^{\pi}_{T(s,a)}} \gamma^i R(C^{\pi}_{T(s,a)}[i])\right) \\
&\quad \times \left(1-\gamma^{c^{\pi}_{T(s,a')}}\right), \text{ and} \\
f_4(\gamma) &= -\left(\gamma^{p^{\pi}_{T(s,a')}} \sum_{i=1}^{c^{\pi}_{T(s,a')}} \gamma^i R(C^{\pi}_{T(s,a')}[i])\right) \\
&\quad \times \left(1-\gamma^{c^{\pi}_{T(s,a)}}\right).
\end{aligned}$$

We draw the following observations about $f^{\pi}_{s,a,a'}(\gamma)$.

1. When treated as a function of $\gamma$, $f^{\pi}_{s,a,a'}(\gamma)$ is a polynomial with integer coefficients.
2. The degree of $f^{\pi}_{s,a,a'}(\gamma)$ cannot exceed the degree of any of its constituents $f_1(\gamma)$, $f_2(\gamma)$, $f_3(\gamma)$, and $f_4(\gamma)$. The degree of $f_1(\gamma)$ is $\max\{1, p^{\pi}_{T(s,a)}\} + c^{\pi}_{T(s,a)} + c^{\pi}_{T(s,a')}$. Since $p^{\pi}_{T(s,a)} + c^{\pi}_{T(s,a)}$ (and similarly $p^{\pi}_{T(s,a')} + c^{\pi}_{T(s,a')}$) can at most be $n$, the degree of $f_1$ is at most $2n+1$. By a similar argument, the degree of $f_2(\gamma)$ is also upper-bounded by $2n+1$. The degree of $f_3(\gamma)$ is $p^{\pi}_{T(s,a)} + c^{\pi}_{T(s,a)} + c^{\pi}_{T(s,a')}$, which is upper-bounded by $2n$, as also is the degree of $f_4(\gamma)$ by a similar argument. Thus, the degree of $f^{\pi}_{s,a,a'}(\gamma)$ is at most $2n+1$.
3. Let the operator $H(\cdot)$ denote the "height" of a polynomial, which is the largest absolute value of any coefficient in the polynomial. For example, the height of $2x^2 - 7x + 3$ is 7. It follows that $H(f^{\pi}_{s,a,a'}) \leq H(f_1) + H(f_2) + H(f_3) + H(f_4)$. Since each reward lies in $\{0, 1, \ldots, 2^b - 1\}$, we observe that $H(f_1)$ and $H(f_2)$ are at most $4(2^b - 1)$, while $H(f_3)$ and $H(f_4)$ are at most $2(2^b - 1)$. Aggregating, we have $H(f^{\pi}_{s,a,a'}) \leq 12 \cdot 2^b$.

We refer to $f^{\pi}_{s,a,a'}$ as the Q-difference sign polynomial. The three properties established above enable us to upper-bound the roots of $f^{\pi}_{s,a,a'}$ that are smaller than 1, and this step leads to our final bound for HPI.

### 4.3 Distance of certain algebraic numbers from 1

In this self-contained subsection, we provide an upper bound on the largest root in the interval $(0, 1)$ for a general class of polynomials $P$, of degree $n \geq 1$. The notation used in this subsection is aligned with the literature on polynomials ("$P$" for polynomial, "$n$" for degree, "$a_{(\cdot)}$" for coefficients, and so on). Any symbols used in this subsection are not to be confused with quantities defined outside it (such as $n$ for the number of states and $a$ for actions). Recall that $H(P)$ is the height of $P$. Below we state our main theorem; the remainder of the subsection provides the proof.

**Theorem 4.2.** *Consider the polynomial:*

$$P(x) \stackrel{def}{=} \sum_{i=0}^{n} a_i x^i, \;\; a_i \in \mathbb{Z}.$$

*Suppose $\tau < 1$ is a root of $P$: that is, $P(\tau) = 0$. Then $\tau \leq 1 - \frac{1}{U_P}$, where $U_P = O\left(\frac{e^z n^{z+2}}{z^z} H(P)\right)$ and $z \geq 0$ is the multiplicity of root 1. Further, $z = O\left(\sqrt{n \log H(P)}\right)$.*

If $\tau$ is a root of $P(x)$, then $1-\tau$ must be a root of $P(1-x)$. Thus to find an upper bound on the root closest to 1 (from below) for $P(x)$ is equivalent to finding a lower bound on the positive roots of $P(1-x)$. The general formula for $P(1-x)$ is given below.

$$P(1-x) = P(1) + \sum_{j=1}^{n} \left[ \sum_{i=j}^{n} \binom{i}{j} a_i \right] (-1)^j x^j. \quad (14)$$

Now, computing a lower bound on the absolute value of roots of a polynomial $J$ is equivalent to computing the reciprocal of an upper bound on the "reverse polynomial", formally stated below.

**Proposition 4.3.** *Let:*

$$J(x) = \sum_{i=0}^{n} \alpha_i x^i, \;\; \alpha_i \in \mathbb{C}.$$

*If $\alpha_n \neq 0$ and $U$ is an upper bound on the absolute value of roots of $J$, then $\frac{1}{U}$ is a lower bound on the absolute values of the roots of $J_r$ where:*

$$J_r(x) \stackrel{def}{=} \sum_{i=0}^{n} \alpha_{n-i} \cdot x^i.$$

We proceed to find an upper bound on the absolute values of the roots of $P_r(1-x)$. We use the following result which was originally proposed by Lagrange, but has also been attributed to Zassenhaus by Knuth (Yap 2000).

**Proposition 4.4.** *For a polynomial $J_r$ described above, if $\alpha_0 \neq 0$, an upper bound of the absolute values of the roots of $J_r$ is given by:*

$$U = 2 \cdot \max_{s=1}^{n} \left( \left| \frac{\alpha_s}{\alpha_0} \right|^{1/s} \right). \quad (15)$$

Applying Proposition 4.4 on $P_r(1-x)$ leads to two cases.

*Case 1*: Suppose $P(1) \neq 0$. Since $P$ has integer coefficients, it follows that $|P(1)| \geq 1$. From the proposition, the upper bound is given by:

$$\begin{aligned} U_P &= 2 \cdot \max_{s=1}^{n} \left( \left| \frac{\sum_{i=s}^{n} \binom{i}{s} a_i}{P(1)} \right|^{\frac{1}{s}} \right) \\ &\leq 2 \cdot \max_{s=1}^{n} \left( \left| \sum_{i=s}^{n} \binom{i}{s} a_i \right|^{\frac{1}{s}} \right) \\ &\leq 2 \cdot \max_{s=1}^{n} \left( \left| \sum_{i=s}^{n} \binom{i}{s} \right|^{1/s} (H(P))^{\frac{1}{s}} \right). \end{aligned}$$

Now $\sum_{i=s}^{n} \binom{i}{s} = \binom{n+1}{s+1}$. And since $H(P) \geq 1$, the maximum occurs at $s = 1$, giving

$$U_P \leq 2 \cdot \binom{n+1}{2} H(P) = n(n+1) \cdot H(P). \quad (16)$$

*Case 2:* Suppose $P(1) = 0$: that is, 1 is a root of $P$. If the multiplicity of the root 1 is $z \geq 1$, we have

$$P(x) = (x-1)^z D(x), \text{ where}$$
$$D(x) = \sum_{i=0}^{n-z} d_i x^i, \;\; d_i \in \mathbb{Z}, D(1) \neq 0.$$

We provide the formula for the $i$-th term of $D$ below.

$$d_i = \sum_{j=i}^{n-z} \binom{n-j-1}{n-j-z} a_{n-j+i}.$$

Now, since

$$\sum_{j=0}^{n-z} \binom{n-j-1}{n-j-z} = \binom{n}{z},$$

for all $i$, we get

$$|d_i| \leq \binom{n}{z} H(P).$$

Since $D(1) \neq 0$, we can use (16) (from case 1) to get a root upper bound for $D(x)$:

$$\begin{aligned} U_D &\leq n(n+1)H(D) \leq n(n+1)\binom{n}{z}H(P) \\ &\leq n(n+1)\left(\frac{en}{z}\right)^z H(P) \leq O\left(\frac{e^z n^{z+2}}{z^z} H(P)\right). \end{aligned} \tag{17}$$

Note that since $P(x) = (x-1)^z D(x)$, $U_P = U_D$. Although the bound from (16) (case 1) shows only polynomial growth with $n$, the one from (17) depends on the number of roots $z$ that $P$ can possibly have. The following result from Borwein, Erdélyi, and Kós (1999) lets us upper-bound $z$.

**Theorem 4.5.** *(Borwein, Erdélyi, and Kós 1999) There is an absolute constant $c > 0$ such that every polynomial $p$ of the form*

$$p(x) = \sum_{j=0}^{n} a_j x^j, \;\; |a_j| \leq 1, a_j \in \mathbb{C}$$

*has at most*

$$c(n(1 - \log|a_0|))^{1/2}$$

*zeros at* 1.

By Theorem 4.5, an upper bound on the number of zeros at 1 of the polynomial $\frac{P(x)}{H(P)}$ and therefore $P(x)$ is given by:

$$c(n(1 + \log H(P)))^{1/2} = O\left(\sqrt{n \log H(P)}\right).$$

In summary, we have shown that for every $\tau < 1$ that is a root of $P$:

$$\tau \leq 1 - \frac{1}{U_P}, \text{ where } U_P \leq O\left(\frac{e^z n^{z+2}}{z^z} H(P)\right), \text{ and } z \leq O\left(\sqrt{n \log H(P)}\right).$$

This concludes the proof of Theorem 4.2.

### 4.4 Proof of Theorem 1.2

We are now ready with our concluding arguments for proving Theorem1.2. In Section 4.2, we showed that for every $\pi : S \to A; s \in S; a, a' \in A$, the polynomial $f^{\pi}_{s,a,a'}(\gamma)$ has integer coefficients, degree at most $2n+1 \leq 3n$, and height at most $O(2^b)$. Since $f^{\pi}_{s,a,a'}(\gamma)$ does not change sign in $(\gamma^{\pi}_{s,a,a'}, 1)$, we see that that $\gamma^{\pi}_{s,a,a'}$ is the largest root of $f^{\pi}_{s,a,a'}(\cdot)$ that is smaller than 1 (but clipped at 0). Applying Theorem 4.2, we observe that:

$$\gamma^{\pi}_{s,a,a'} \leq 1 - \frac{1}{U}, \; U = O\left(\frac{e^z (3n)^{z+2}}{z^z} 2^b\right), \; z = O\left(\sqrt{nb}\right).$$

Since $\left(\frac{ne}{z}\right)^z$ is strictly increasing for $z \leq n$, we get:

$$\log U \leq O\left(\sqrt{nb} \log \frac{n}{b} + b\right).$$

Now from (9), we have

$$\gamma_{\mathrm{Q}} \leq 1 - \frac{1}{U}, \text{ or equivalently, } \frac{1}{1-\gamma_{\mathrm{Q}}} \leq U.$$

For $\gamma \in [0, \gamma_{\mathrm{Q}})$, our upper bound in Theorem 1.2 is trivially valid due to the $\gamma$-dependent upper bound of Scherrer (2013), which we have rephrased in Theorem 3.1. Now consider $\gamma \in (\gamma_{\mathrm{Q}}, 1)$. At any iteration of HPI, with policy $\pi$ and state $s$, if $Q^{\pi}_{\gamma}(s,a) \geq Q^{\pi}_{\gamma}(s,a')$ for $a, a' \in A$, then $Q^{\pi}_{\gamma'}(s,a) \geq Q^{\pi}_{\gamma'}(s,a')$ for all $\gamma' \in (\gamma_{\mathrm{Q}}, 1)$. Since HPI always picks an action maximising the Q-value, starting from any policy, it would visit an identical sequence of policies for any $\gamma' \in (\gamma_{\mathrm{Q}}, 1)$. In particular consider $\gamma'$ that is arbitrarily close to $\gamma_{\mathrm{Q}}$. By Theorem 3.1, the number of iterations taken by HPI on $(S, A, T, R, \gamma')$ would be $O\left(\frac{nk}{1-\gamma_{\mathrm{Q}}} \log \frac{1}{1-\gamma_{\mathrm{Q}}}\right)$, which we have just shown to be

$$nk\left(\sqrt{nb}\log\frac{n}{b} + b\right)\exp\left(O\left(\sqrt{nb}\log\frac{n}{b} + b\right)\right),$$

matching the claim in Theorem 1.2.

## 5 Conclusion

HPI (Howard 1960) has, over several decades, cemented its position as a method of choice among practitioners for solving MDPs. However, known upper bounds for HPI that hold independently of the discount factor are exponentially separated from known lower bounds—not only for MDPs, but even for the restricted subclass of DMDPs. In this paper, we give the first subexponential upper bound for HPI on DMDPs with discounted reward, contingent on an assumption about the bit-size of rewards. This assumption is reasonable in practice, as reward bit-sizes are almost always constant. Theoretically, our bound remains valid and significant even when there are exactly two reward values of arbitrary size. En route to this result, we also show a pseudopolynomial upper bound for HPI on DMDPs under average reward.

We employ a novel analytical technique showing that when the discount factor is "large," every choice made by HPI matches the one it would make with a smaller, "threshold" discount factor. We then upper-bound this threshold by proving a result on polynomials with integer coefficients. These steps let us piggyback on a $\gamma$-dependent upper bound from Ye (2011) and Scherrer (2013), by plugging in the threshold discount factor.

From the related work of Grand-Clément and Petrik (2023), it appears unlikely that the same technique can show a subexponential bound for HPI on general MDPs. However, it would be interesting to investigate whether intermediate classes of MDPs (which interpolate in some manner between DMDPs and MDPs) can benefit from our approach. It may also be possible to improve our subexponential upper bound by exploiting more properties of the Q-difference sign polynomial. On a related note, it is worth exploring if the dependence on $b$, which is currently exponential in both our upper bounds, can be improved.

## Acknowledgements

The authors thank Sundar Vishwanathan, Supratik Chakraborty, Akash Kumar, and Rohit Gurjar for providing useful comments.

## References

Bellman, R. 1967. *Dynamic Programming*. Princeton University Press.

Bertsekas, D. P. 2007. *Dynamic Programming and Optimal Control, 3rd Edition*. Athena Scientific.

Blackwell, D. 1962. Discrete dynamic programming. *Annals of Mathematical Statistics*, 33(2): 719–726.

Borwein, P.; Erdélyi, T.; and Kós, G. 1999. Littlewood-Type Problems on [0,1]. *Proceedings of the London Mathematical Society*, 79(1): 22–46.

Christ, M.; and Yannakakis, M. 2023. The Smoothed Complexity of Policy Iteration for Markov Decision Processes. In *Proc. STOC 2023*, 1890–1903. ACM.

Crites, R. H.; and Barto, A. G. 1995. Improving Elevator Performance Using Reinforcement Learning. In *Proc. NeurIPS 1995*, 1017–1023. MIT Press.

Dasdan, A. 2004. Experimental analysis of the fastest optimum cycle ratio and mean algorithms. *ACM Transactions on Design Automation of Electronic Systems*, 9(4): 385–418.

Fearnley, J. 2010. Exponential Lower Bounds for Policy Iteration. In *Proc. ICALP 2010*, 551–562. Springer.

Goenka, R.; Gupta, E.; Khyalia, S.; and Kalyanakrishnan, S. 2025. Upper Bounds for All and Max-Gain Policy Iteration Algorithms on Deterministic MDPs. *Math. of Oper. Res.*

Grand-Clément, J.; and Petrik, M. 2023. Reducing Blackwell and Average Optimality to Discounted MDPs via the Blackwell Discount Factor. In *Proc. NeurIPS 2023*, 52628–52647. Curran Associates, Inc.

Gupta, A.; and Kalyanakrishnan, S. 2017. Improved strong worst-case upper bounds for MDP planning. In *Proc. IJCAI 2017*, 1788–1794.

Hansen, T. D.; Kaplan, H.; and Zwick, U. 2014. Dantzig's pivoting rule for shortest paths, deterministic MDPs, and minimum cost to time ratio cycles. In *Proc. SODA 2014*, 847–860. SIAM.

Hansen, T. D.; Miltersen, P. B.; and Zwick, U. 2013. Strategy Iteration Is Strongly Polynomial for 2-Player Turn-Based Stochastic Games with a Constant Discount Factor. *Journal of the ACM*, 60(1): 1–16.

Hansen, T. D.; and Zwick, U. 2010. Lower Bounds for Howard's Algorithm for Finding Minimum Mean-Cost Cycles. In *Proc. ISAAC 2010*, 415–426. Springer.

Hollanders, R.; Delvenne, J.; and Jungers, R. M. 2012. The complexity of Policy Iteration is exponential for discounted Markov Decision Processes. In *Proc. CDC 2012*, 5997–6002. IEEE.

Howard, R. A. 1960. *Dynamic Programming and Markov Processes.* MIT Press.

Kalyanakrishnan, S.; Mall, U.; and Goyal, R. 2016. Batch-Switching Policy Iteration. In *Proc. IJCAI 2016*, 3147–3153. IJCAI/AAAI Press.

Kalyanakrishnan, S.; Misra, N.; and Gopalan, A. 2016. Randomised Procedures for Initialising and Switching Actions in Policy Iteration. In *Proc. AAAI 2016*. AAAI Press.

Karp, R. M. 1978. A characterization of the minimum cycle mean in a digraph. *Discrete mathematics*, 23(3): 309–311.

Littman, M. L.; Dean, T. L.; and Kaelbling, L. P. 1995. On the Complexity of Solving Markov Decision Problems. In *Proc. UAI 1995*, 394–402. Morgan Kaufmann.

Madani, O. 2002a. On Policy Iteration as a Newton's Method and Polynomial Policy Iteration Algorithms. In *Proc. AAAI/IAAI 2002*, 273–278. AAAI Press / MIT Press.

Madani, O. 2002b. Polynomial Value Iteration Algorithms for Detrerminstic MDPs. In *Proc. UAI 2002*, 311–318. Morgan Kaufmann.

Madani, O.; Thorup, M.; and Zwick, U. 2010. Discounted Deterministic Markov Decision Processes and Discounted All-Pairs Shortest Paths. *ACM Trans. on Alg.*, 6(2): 1–25.

Mansour, Y.; and Singh, S. 1999. On the Complexity of Policy Iteration. In *Proc. UAI 1999*, 401–408. Morgan Kaufmann.

Mausam; and Kolobov, A. 2012. *Planning with Markov Decision Processes: An AI Perspective*. Morgan & Claypool.

Papadimitriou, C. H.; and Tsitsiklis, J. N. 1987. The Complexity of Markov Decision Processes. *Math. of Oper. Res.*, 12(3): 441–450.

Post, I.; and Ye, Y. 2013. The simplex method is strongly polynomial for deterministic Markov decision processes. In *Proc. SODA 2013*, 1465–1473. SIAM.

Puterman, M. L. 1994. *Markov Decision Processes*. Wiley.

Scherrer, B. 2013. Improved and Generalized Upper Bounds on the Complexity of Policy Iteration. In *Proc. NeurIPS 2013*, 386–394. Curran Associates, Inc.

Schmitz, N. 1985. How Good is Howard's Policy Improvement Algorithm? *Zeitschrift für Oper. Res.*, 29: 315–316.

Schurr, I.; and Szabó, T. 2005. Jumping Doesn't Help in Abstract Cubes. In *Proc. IPCO 2005*, 225–235. Springer.

Silver, D.; et al. 2016. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587): 484–489.

Silver, D.; et al. 2018. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*, 362(6419): 1140–1144.

Spielman, D. A.; and Teng, S.-H. 2004. Smoothed Analysis of Algorithms: Why the Simplex Algorithm Usually Takes Polynomial Time. *Journal of the ACM*, 51(3): 385–463.

Sutton, R. S.; and Barto, A. G. 1998. *Reinforcement Learning: An Introduction*. MIT Press.

Szabó, T.; and Welzl, E. 2001. Unique Sink Orientations of Cubes. In *Proc. FOCS 2001*, 547–555. IEEE.

Szepesvári, C. 2010. *Algorithms for Reinforcement Learning.* Morgan & Claypool.

Taraviya, M.; and Kalyanakrishnan, S. 2020. A tighter analysis of randomised policy iteration. In *Proc. UAI 2020*, 519–529. PMLR.

Yap, C.-K. 2000. *Fundamental Problems in Algorithmic Algebra*. Oxford University Press.

Ye, Y. 2011. The Simplex and Policy-Iteration Methods Are Strongly Polynomial for the Markov Decision Problem with a Fixed Discount Rate. *Math. of Oper. Res.*, 36(4): 593–603.

# A Analysis under Average Reward

The "average cost" criterion (cost is the negative of reward) is a commonly used method for optimising stochastic dynamical systems over an infinite time horizon. Solving DMDPs under the average cost criterion is equivalent to finding minimum mean cost cycles (MMCC) in the induced graph. Karp (1978) provided a polynomial time algorithm to the MMCC problem. However, the complexity of Howard's Policy Iteration (HPI) in this context remains an open problem, even though, in practice, HPI is typically more efficient than Karp's algorithm (Dasdan 2004). On the other hand, Hansen and Zwick (2010) established a lower bound of $\Omega(n^2)$ for HPI when the graph has $\Theta(n^2)$ edges. We provide a polynomial upper bound for all policy iteration algorithms under the assumption that the bit-size of the rewards is constant.

In this section we describe the notion of average reward and present the proof of Proposition 1.1.

## A.1 Preliminaries

The average reward values defined in equations (2) and (3) satisfy the pair of Bellman Equations given by:

$$V_{\mathrm{g}}^{\pi}(s) = \sum_{s' \in S} T(s, \pi(s), s') V_{\mathrm{g}}^{\pi}(s'), \tag{18}$$

$$V_{\mathrm{b}}^{\pi}(s) = R(s, \pi(s)) - V_{\mathrm{g}}^{\pi}(s) + \sum_{s' \in S} T(s, \pi(s), s') V_{\mathrm{b}}^{\pi}(s'). \tag{19}$$

Note that the system of equations above is underdetermined, meaning that solutions to equations (18) and (19) are not unique. A standard procedure for ensuring a unique bias function is to require that the long-run average bias over time is zero, known as the canonical bias (see Puterman (1994)), which is consistent with the definition based on expectations. Another common approach—which we adopt—is to select a state $s$ in each recurrent class and set $V_b(s) = 0$. Additionally, in multichain average reward MDPs, computing the optimal gain and identifying optimal policies require two optimality equations, given by:

$$\max_{a \in A} \left\{ V_{\mathrm{g}}(s) - \sum_{s' \in S} T(s, a, s') V_{\mathrm{g}}(s') \right\} = 0, \tag{20}$$

$$\max_{a \in A'} \left\{ -V_{\mathrm{b}}(s) + R(s, a) - V_{\mathrm{g}}(s) + \sum_{s' \in S} T(s, a, s') V_{\mathrm{b}}(s') \right\} = 0, \tag{21}$$

where $A' = \left\{ a' \in A \,:\, V_{\mathrm{g}}(s) - \sum_{s' \in S} T(s, a', s') V_{\mathrm{g}}(s') = 0 \right\}$. A solution to the optimality equations always exists for finite MDPs (Puterman 1994).

**Policy iteration** is analogous to the discounted case, except that it maintains and updates a pair of value functions $(V_{\mathrm{g}}, V_{\mathrm{b}})$, in a lexicographical order. The following pseudocode outlines this process.

PI starting from policy $\pi$ under average reward

*Policy Evaluation:*

1. Obtain $V_{\mathrm{g}}^{\pi}$ and $V_{\mathrm{b}}^{\pi}$ which satisfy (18), (19).

*Policy Improvement:*

2. Let:

$$J_{\mathrm{g}}^{\pi} = \left\{ (s,a) \;\middle|\; s \in S, a \in A,\ \sum_{s' \in S} T(s,a,s') V_{\mathrm{g}}^{\pi}(s') > V_{\mathrm{g}}^{\pi}(s) \right\}.$$

3. If $J_{\mathrm{g}}^{\pi} \neq \emptyset$, pick $I_{\mathrm{g}} \subseteq J_{\mathrm{g}}^{\pi}$ $(I_{\mathrm{g}} \neq \emptyset)$ and let $\pi'(\overline{s}) \leftarrow \overline{a}$ for $(\overline{s}, \overline{a}) \in I_{\mathrm{g}}$ and $\pi'(s) \leftarrow \pi(s)$ for the remaining states $s$. Set $\pi \leftarrow \pi'$ and go to 1. Else go to 4.

4. Let

$$J_{\mathrm{b}}^{\pi} = \left\{ (s,a) \;\middle|\; s \in S, a \in A,\ R(s,a) + \sum_{s' \in S} T(s,a,s') V_{\mathrm{b}}^{\pi}(s') > V_{\mathrm{b}}^{\pi}(s) \right\}.$$

5. If $J_{\mathrm{b}}^{\pi} \neq \emptyset$, pick $I_{\mathrm{b}} \subseteq J_{\mathrm{b}}^{\pi}$ $(I_{\mathrm{b}} \neq \emptyset)$ and let $\pi'(\overline{s}) \leftarrow \overline{a}$ for $(\overline{s}, \overline{a}) \in I_{\mathrm{b}}$ and $\pi'(s) \leftarrow \pi(s)$ for the remaining states $s$. Set $\pi \leftarrow \pi'$ and go to 1. Else declare $\pi$ to be optimal and terminate.

A subset of states $S' \subseteq S$ is called a *recurrent class* under $\pi$ if for each pair $s, s' \in S$, there is a non-zero probability of reaching $s'$ from $s$ in fewer than $n$ steps by following $\pi$, and moreover, no state in $S \setminus S'$ is reachable from any state in $S'$ under $\pi$. The system of equations (18) and (19) becomes uniquely determined if any one state in each recurrent class of $\pi$ is allotted any arbitrary bias. Assuming an indexing of states—concretely, take $S = \{0, 1, 2, \ldots, n-1\}$—we adopt the convention of setting $V_{\mathrm{b}}^{\pi}(s) = 0$ for every state $s$ that is in a recurrent class and has the smallest index in that class (Hansen, Miltersen, and Zwick 2013). Since every bias function must satisfy the linear equation (19), any two valid bias functions within a recurrent class $S' \subseteq S$ differ only by a constant offset—i.e., $V_{b_1}(s) = V_{b_2}(s) + C$ for all $s \in S'$. As a result, the analysis we present below holds uniformly for all such solutions.

**DMDPs.** For DMDP $M = (S, A, T, R)$ using average reward and policy $\pi : S \to A$, the gain of each state $s \in S$ becomes the average of the rewards on $C_s^{\pi}$, which indeed constitutes a recurrent class.

$$V_{\mathrm{g}}^{\pi}(s) = \frac{1}{c_s^{\pi}} \sum_{i=1}^{c_s^{\pi}} R(c_s^{\pi}[i]). \tag{22}$$

The bias $V_{\mathrm{b}}^{\pi}(s)$ distinguishes between states that reach the same recurrent class, and is given by

$$V_{\mathrm{b}}^{\pi}(s) = \begin{cases} 0 & \text{if } s \text{ has the smallest index in } C_s^{\pi}, \\ R(s, \pi(s)) - V_{\mathrm{g}}^{\pi}(s) + V_{\mathrm{b}}^{\pi}(T(s, \pi(s))) & \text{otherwise.} \end{cases} \tag{23}$$

## A.2 Pseudopolynomial Upper Bound for Average Reward

We now present a proof of Proposition 1.1 which yields the pseudopolynomial bound.

**Proof of Proposition 1.1** Recall from (22) that the gain of state $s \in S$ under policy $\pi : S \to A$ is equal to

$$V_{\mathrm{g}}^{\pi}(s) = \frac{\sum_{i=1}^{c_s^{\pi}} R(c_s^{\pi}[i])}{c_s^{\pi}}.$$

The numerator is a sum of $c_s^{\pi}$ rewards, each an integer in $\{0, 1, \ldots, n \cdot (2^b - 1)\}$, while the denominator is an integer in $\{1, 2, \ldots, n\}$. Thus, $V_{\mathrm{g}}^{\pi}(s)$ can take at most $T_1 \stackrel{\text{def}}{=} n^2 \cdot 2^b$ values.

Similarly, the bias $V_{\mathrm{b}}^{\pi}(s)$ is seen from (23) to be the sum of some $i$ terms of the form $R(s_i, \pi(s_i)) - V_{\mathrm{g}}^{\pi}(s_i)$, where $0 \leq i \leq n-1$, and $s_i$ is the $i$-th state visited after starting from $s$. The sum of the $i$ reward terms has to be an integer in $\{0, 1, \ldots, (n-1) \cdot (2^b - 1)\}$. Since the states $s_i$ that are visited from $s$ by following $\pi$ all have the same gain $V_{\mathrm{g}}^{\pi}(s)$, the amount subtracted is

from the set $\{0, V_{\mathrm{g}}^{\pi}(s), 2V_{\mathrm{g}}^{\pi}(s), \ldots, (n-1)V_{\mathrm{g}}^{\pi}(s)\}$. Hence, if $V_{\mathrm{g}}^{\pi}(s)$ is fixed, the number of possible values $V_{\mathrm{b}}^{\pi}(s)$ can take is at most $T_2 \stackrel{\text{def}}{=} n \cdot 2^b \cdot n$. Therefore, there are at most $T_1 \cdot T_2$ unique gain-bias pairs possible for any state $s$.

By the Policy Improvement Theorem (Howard 1960), in each PI improvement step: (1) the gain of each state either remains the same or increases; (2) if gains are unchanged, the bias either remains the same or increases; and (3) at least one state's gain or bias strictly increases. Thus, any PI algorithm, regardless of the variant, follows a strictly monotonic trajectory starting from the initial policy, with each iteration requiring at least one state to adopt a new gain-bias pair. Consequently, the total number of iterations is bounded by $n \cdot T_1 \cdot T_2 = O(n^5 \cdot 4^b)$.