

An Improved Lower Bound on the Length of Locally-Improving Policy Sequences in MDPs with Large Action Sets

Pratyush Agarwal*, Mulinti Shaik Wajid*, Shivaram Kalyanakrishnan

Indian Institute of Technology Bombay
 pratyush1019@gmail.com, wajidali.mshaik@gmail.com, shivaram@cse.iitb.ac.in

Abstract

Popular algorithms to solve Markov Decision Problems (MDPs) include policy iteration and the Simplex method (executed on an induced linear program). Each run of these algorithms can be associated with a sequence of “locally-improving” policies for the input MDP. For integers $n \geq 2$, $k \geq 2$, let $f(n, k)$ denote the *longest* possible sequence of locally-improving policies for any MDP with n states and k actions per state. An alternative view of $f(n, k)$ is as a descriptive structural property of the policy space of MDPs: it is the largest possible “c-height” in an induced “LP-digraph” of any n -state, k -action MDP. How large can $f(n, k)$ be?

A trivial *upper* bound on $f(n, k)$ is the total number of (Markovian, deterministic) policies, which is k^n . A construction from Melekopoglou and Condon (1994) shows that $f(n, 2) = 2^n$, implying that the trivial upper bound is tight for $k = 2$. For $k \geq 3$, the tightest lower bound on $f(n, k)$ in the current literature is only $\Omega(\sqrt{k}^n)$ (Ashutosh et al. 2020). In this paper, we propose a family of MDPs to show a lower bound of $\Omega(\lfloor k/2 \rfloor^n)$ on $f(n, k)$ —giving an exponential-in- n tightening for each $k \geq 6$. Our investigation brings out technical challenges that do not arise for $k = 2$. Our result still leaves open the important question of whether $f(n, k)$ is indeed k^n for $n \geq 2, k \geq 2$. We furnish an affirmative answer for the special case of $n = 2, k \geq 2$.

1 Introduction

Markov Decision Problems (Bellman 1957; Howard 1960; Puterman 2014) have been employed for more than half a century as a mathematical basis for sequential decision making in stochastic environments. They are widely used in planning (Mausam and Kolobov 2012), control (Bertsekas 2012), and reinforcement learning (Sutton and Barto 1998).

An MDP specifies an *environment*, with which an *agent* interacts by taking actions according to a *policy*. The policy carries the agent through a random sequence of states, gathering a reward at each step. These rewards can be suitably aggregated to associate a “value” for the policy from each starting state. For an MDP that has $n \geq 2$ states and $k \geq 2$ actions available at each state, the number of possible (time-independent, deterministic) policies is k^n . For many

common definitions of value, it can be shown that there exists an *optimal* policy, whose value cannot be exceeded from any starting state. Planning in MDPs is the problem of computing an optimal policy for any given MDP.

There are two contrasting algorithmic approaches to MDP planning. The first is value iteration (Bellman 1957), which uses dynamic programming to progressively refine an estimate of optimal state values. The second approach is to progressively update to better policies. Examples of this approach are policy iteration (Howard 1960) and its special case the Simplex algorithm (Dantzig 1963) (which is performed on a related linear program). The basis for iterating over policies is provided by the structure of the policy space. Define any two policies to be *neighbours* if their actions are different for exactly one state. It is a fact that for any two neighbouring policies π and π' , one must “locally-improve” the other, or both have equal values at all states. π' is said to locally-improve π if π' has a strictly higher value than π in one or more states, and equal values in all other states. Given a policy, it is relatively straightforward (polynomial-time in n and k) to determine all its locally-improving neighbours.

Starting with an arbitrary initial policy, at each step the Simplex algorithm picks a locally-improving neighbour, continuing so until an optimal policy is found. Thus any run of the algorithm may be associated with a “locally-improving” policy sequence $\pi_0, \pi_1, \dots, \pi_N$, where (1) π_N is an optimal policy, and (2) for $i \in \{0, 1, \dots, N - 1\}$, π_{i+1} is a locally-improving neighbour of π_i . Policy iteration can potentially “jump” from π to a policy π' that is not a neighbour (they can differ on more than one state), but which is still locally-improving. It remains that any run of policy iteration can also be associated with one or more locally-improving policy sequences, since there must be a locally-improving subsequence corresponding to each jump.

Let $f(n, k)$ denote the longest possible locally-improving policy sequence for any n -state, k -action MDP. This quantity is of interest in the analysis of algorithms, since it upper-bounds the number of steps that any algorithm from the Simplex and policy iteration families can possibly take. As illustrated in Section 2, $f(n, k)$ is also an informative structural property of the feasible polytopes of linear programs induced by n -state, k -action MDPs. How does $f(n, k)$ scale with n and k ? We contribute to the current understanding of this fundamental theoretical question.

*These authors contributed equally.

Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Melekopoglou and Condon (1994) have already furnished a definitive answer for the special case of $k = 2$. They provide an n -state, 2-action MDP construction in which all 2^n policies can be arranged as a locally-improving sequence, thus establishing that $f(n, 2) = 2^n$. The picture is less clear for $k \geq 3$. When $k = 2$, each policy has exactly one neighbour along each state—hence selecting a neighbouring policy is equivalent to selecting a state. For $k \geq 3$, each state gets associated with $k - 1 \geq 2$ neighbours, which introduces a new layer of complexity for stringing together lengthy locally-improving sequences. The best efforts yet are due to Ashutosh et al. (2020), who show that $f(n, k) = \Omega(\sqrt{k}^n)$ for $k \geq 3$. Our contribution is an exponentially tighter lower bound on $f(n, k)$ for $k \geq 3$.

- The special case of 2-state MDPs helps us develop intuitions for the general case. We show that indeed all k^2 policies can be visited along a locally-improving sequence on such MDPs. Key to this result is the use of stochasticity; in fact we show that sequences of length k^2 cannot be realised on deterministic MDPs.
- Our lower bound for the general case of $n \geq 2, k \geq 2$ shows that $f(n, k) = \Omega(\lfloor k/2 \rfloor^n)$. This bound is achieved through a construction in which all policies taking only “even-numbered” actions can be visited, enabled by intermediate switches to some “odd-numbered” actions. Our construction, like that of Ashutosh et al. (2020), only uses deterministic transitions. It remains open to improve our lower bound (possibly by using stochastic MDPs), or to tighten the trivial upper bound of k^n (for $k \geq 3$).

We begin with a formal description of MDPs and their policy space in Section 2. We present the related literature on lower bounds in Section 3. In Section 4 we present our k^2 lower bound for 2-state MDPs, and in Section 5 follow with the general result: the $\Omega(\lfloor k/2 \rfloor^n)$ lower bound for n -state MDPs. Section 6 concludes with a summary and outlook.

2 Markov Decision Problems

A Markov Decision Problem (MDP) is defined on a set of states S and a set of actions A . In this paper, we assume $S = \{1, 2, \dots, n\}$ for some $n \geq 2$, and $A = \{0, 1, \dots, k - 1\}$ for some $k \geq 2$: thus $|S| = n$ and $|A| = k$. The effect of taking action $a \in A$ from state $s \in S$ is for the agent to obtain a reward $r \in \mathbb{R}$ and go to a next state s' . The reward $r = R(s, a)$ is determined by the reward function R , while the next state s' is generated at random with probability $T(s, a, s')$, where T is the transition function of the MDP. If s' is *terminal* (from set \bar{S}), no more actions are performed. If $s' \in S$, the agent again takes an action from s' , and this process continues. Observe that the agent goes along a sequence of states, actions, and rewards; its aim is to maximise the expected long-term reward (or *value*) along this sequence.

A common definition of values in the AI literature (Sutton and Barto 1998) relies on discounting future rewards geometrically by a factor $\gamma \in (0, 1)$. Hence the tuple $(S, A, T, R, \gamma, \bar{S})$ completely specifies the MDP. Suppose the agent takes actions according to a *policy* $\pi : S \rightarrow A$: that is, from state $s \in S$, the agent takes action $\pi(s)$. Then

the value accrued, starting from state $s \in S$, is defined to be

$$V^\pi(s) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r_t \mid s_0 = s \right], \quad (1)$$

where actions are taken according to π , starting from initial state $s_0 = s$, and the sequence of rewards is r_0, r_1, r_2, \dots . Alternatively, values can be undiscounted (equivalent to having $\gamma = 1$) if sequences are guaranteed to eventually reach some terminal state $\tau \in \bar{S}$: this setting is called “total reward”. By definition terminal states have value 0. For non-terminal states $s \in S$, expanding (1) gives the recursion

$$V^\pi(s) = R(s, \pi(s)) + \gamma \sum_{s' \in S \cup \bar{S}} T(s, \pi(s), s') V^\pi(s'). \quad (2)$$

This linear system of equations, called the Bellman equations for π , can be solved to obtain the state values V^π . We find it convenient to present our lower bound for $n = 2$ in Section 4 using discounted reward, and the one for $n \geq 2$ in Section 5 using total reward. We outline in Section 6, how both our constructions can be made to work in both settings.

2.1 Policy Improvement

Let Π denote the set of all policies $\pi : S \rightarrow A$. An agent’s natural aim is to seek policies in Π with “larger” state values. Formally, consider the relations \succeq (“dominates or equals”) and \succ (“dominates”) on the set of policies Π . For $\pi, \pi' \in \Pi$,

$$\begin{aligned} \pi \succeq \pi' &\text{ iff } \forall s \in S (V^\pi(s) \geq V^{\pi'}(s)); \\ \pi \succ \pi' &\text{ iff } \pi \succeq \pi' \wedge \exists s \in S (V^\pi(s) > V^{\pi'}(s)). \end{aligned}$$

A quantity called “gain” comes handy to identify dominating policies. For state $s \in S$ and action $a \in A$, the “action value” under policy $\pi \in \Pi$, denoted $Q^\pi(s, a)$, is the expected long-term reward from taking a from s , and thereafter acting according to π . The gain $G^\pi(s, a)$ measures the advantage of doing so *relative to* following π (all the time) starting from s . These quantities work out to:

$$Q^\pi(s, a) = R(s, a) + \gamma \sum_{s' \in S \cup \bar{S}} T(s, a, s') V^\pi(s'), \quad (3)$$

$$G^\pi(s, a) = Q^\pi(s, a) - V^\pi(s). \quad (4)$$

For a given policy, gains can be computed in $\text{poly}(n, k)$ time. The well-known “policy improvement theorem” (Puterman 2014, see Section 6.4) shows that switching in one or more states to actions with positive gains (while retaining the actions at other states) must result in a dominating policy.

Theorem 1. *Fix policy $\pi \in \Pi$.*

1. *Suppose for policy $\pi' \in \Pi \setminus \{\pi\}$, for all $s \in S$,*

$$\pi'(s) \neq \pi(s) \implies G^\pi(s, \pi'(s)) > 0.$$

Then $\pi' \succ \pi$. We say that π' “locally improves” π .

2. *Suppose for all $(s, a) \in S \times A$,*

$$G^\pi(s, a) \leq 0.$$

Then for all policies $\pi' \in \Pi$, $\pi \succeq \pi'$. We say that π is an “optimal” policy.

The theorem establishes the existence of an *optimal* policy (which has the largest possible value at each state), and also provides a recipe to compute it. The policy iteration (PI) family of algorithms implement this recipe. A PI algorithm is initialised at an arbitrary policy. In each step, if the current policy π has one or more state-action pairs with positive gain, then π is replaced by a locally-improving policy π' . Since Π is finite and no policy can be visited more than once (due to the increase in value under \succ), we must reach a policy with no positive-gain state-action pairs. By the second part of the theorem, this final policy must be optimal.

Algorithms within the PI family are distinguished by the “switching rule” they employ to update from the current policy π to a locally-improving policy π' . In Howard’s PI (Howard 1960), which is widely used in practice, (1) π' takes the same action as π on states with no positive gains for π ; (2) on *all* other states, π' takes an action with the largest gain for π . In contrast, Simplex algorithms are PI variants that on each step switch the action at exactly one state. For example, in “max-gain Simplex” (Ye 2011) the differing state-action pair in π' has the largest gain among all state-action pairs for π . Switching rules can also be randomised (Mansour and Singh 1999).

2.2 Locally-Improving Policy Sequences

It is apparent that any run of any PI algorithm results in a sequence of policies $\pi_0, \pi_1, \dots, \pi_N$, where for $1 \leq i \leq N$, π_i locally-improves π_{i-1} , and π_N is an optimal policy. We refer to such a sequence as a locally-improving policy sequence for the input MDP. Reflecting the practical need for fast algorithms, the literature has provided *upper bounds* on the lengths of the sequences generated by several PI variants. For example, Howard’s PI and Simplex PI geometrically reduce the distance between values of the current and optimal policy, hence cannot visit more than $O(\frac{nk}{1-\gamma} \log(\frac{n}{1-\gamma}))$ policies (Hansen, Miltersen, and Zwick 2013). Several randomised variants eliminate enough policies on most iterations, so the expected lengths of their sequences becomes exponentially smaller than $|\Pi| = k^n$ (Mansour and Singh 1999; Taraviya and Kalyanakrishnan 2020).

In this paper, we investigate the complementary question: what is the longest locally-improving policy sequence that can possibly exist for any n -state, k -action MDP? Formally, we seek a *lower bound* on the largest integer $f(n, k)$ for which the statement below is true.

“There exist an n -state, k -action MDP M and $f(n, k)$ distinct policies $\pi_0, \pi_1, \dots, \pi_{f(n,k)-1}$ for M that constitute a locally-improving policy sequence for M .”

$f(n, k)$ is of theoretical interest as a combinatorial property of the policy space of MDPs. In a common linear programming formulation to solve MDPs (Post and Ye 2015), the vertices of the feasible polytope are in 1-1 correspondence with policies. The “LP digraph” (Avis, Miyata, and Moriyama 2013) is an unweighted directed graph connecting these vertices to neighbours that improve the objective function. A neighbour corresponds to a policy that differs on exactly one state-action pair. Figure 1 shows a small MDP

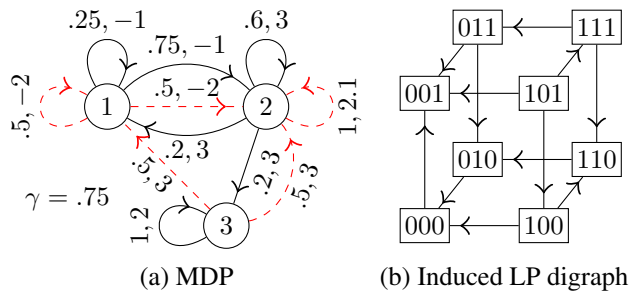


Figure 1: Subfigure (a) shows an MDP with 3 states and 2 actions. Transitions (black, solid for action 0 and red, dashed for action 1) show associated probabilities and rewards. Subfigure (b) shows the corresponding LP digraph. Notice locally-improving policy sequences (101, 111, 011, 010, 000, 001) and (101, 100, 110, 010, 000, 001) of length 6; none are longer.

and its LP digraph. Policies are represented as strings, in which the i -th entry specifies the action at state i .

$f(n, k)$ is the number of vertices on the longest possible path in the LP digraph induced by any n -state, k -action MDP. In the literature on convex polyhedra, such paths are called “ c -monotone” paths, and the length of the longest such path called the “ c -height” (Kalai 1992; Blanchard, Lopera, and Louveaux 2021).

Whereas Simplex algorithms necessarily visit all the policies along some path in the LP digraph, PI algorithms may “jump” from a policy π to a non-neighbour π' . The result below shows that nevertheless, there must be a path in the LP digraph connecting π to π' (assuming no tied policy values).

Proposition 2. *Suppose policies $\pi, \pi' \in \Pi$ are such that (1) π' locally-improves π , and (2) π and π' differ on exactly $m \geq 1$ state(s). Then there exists a sequence of m policies $\pi_1, \pi_2, \dots, \pi_m \in \Pi$ such that (1) $\pi_1 \succ \pi$; (2) for $1 \leq i \leq m-1$, π_{i+1} differs from π_i on exactly 1 state and π_{i+1} locally-improves π_i ; (3) $\pi' = \pi_m$.*

The proposition implies that to generate long trajectories either for Simplex or for general PI, it suffices to constrain consecutive policies to be neighbours. The proof of Proposition 2 relies on the proof of Theorem 1, which is available in textbooks (Bertsekas 2012; Szepesvári 2009). For easy reference, we furnish both proofs in Appendix A.¹

3 Known Lower Bounds on $f(n, k)$

The earliest literature in the spirit of our investigation pertains to the analysis of the Simplex algorithm for linear programming. After Klee and Minty (1972) established an exponential lower bound (in the number of variables) for the number of iterations taken by “Dantzig’s” pivoting rule (Dantzig 1963), similar lower bounds have been shown for other variants (Avis and Friedmann 2017; Disser, Friedmann, and Hopp 2023). These results for general linear programs need not apply to MDPs. On the other hand,

¹Appendices are included in a longer version of the paper linked from SK’s home page: <https://www.cse.iitb.ac.in/~shivaram/>.

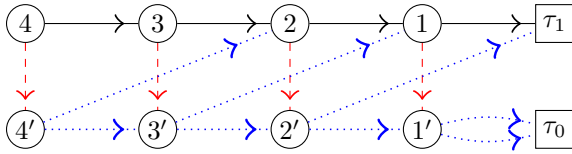
lower bounds have also been shown on the number of iterations taken by specific PI variants on MDPs—for example a bound of $\exp(\Omega(n))$ for Howard’s PI on an MDP with $\Theta(n)$ states and actions (Fearnley 2010).

Any lower bound for a specific PI variant is also a lower bound on $f(n, k)$. Interestingly, a lower bound given by Melekopoglou and Condon (1994) for “Simple PI” settles $f(n, 2)$ to be 2^n . These authors’ analysis is restricted to 2-action MDPs. Assuming a fixed numbering of the states, Simple PI always switches the smallest-numbered state with a positive-gain action. On a particular n -state, 2-action MDP, Simple PI can visit all 2^n policies. This MDP, illustrated in Figure 2a for $n = 4$, is for “total reward”, and has two terminal states. It also has n “dummy” states from which there are no actions to take. These states merely simplify the description; they can be absorbed into the transitions of the n decision-making states. For $n = 3$, the following is a locally-improving policy sequence for the MDP.

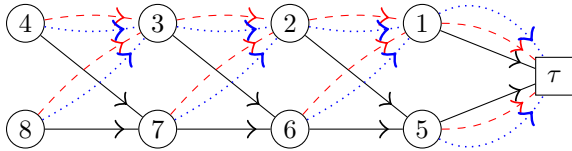
100, 101, 111, 110, 010, 011, 001, 000.

The sequence above is an example of a Gray code (Gray 1953), which covers all 2^n bit strings while switching only 1 bit between consecutive elements. Gray codes can be generalised in many ways to a k -ary alphabet for $k \geq 2$ (Joichi, White, and Williamson 1980; Squire 1996). However, for $k \geq 3$, it becomes challenging to construct MDPs for which the corresponding policy sequences are locally-improving.

The longest locally-improving policy sequences yet shown for $k \geq 3$ are from Ashutosh et al. (2020). An instance of their MDP (for $n = 8, k = 3$) is shown in Figure 2b. The idea is to have some m states take all k^m possi-



(a) Transitions in the MDP of Melekopoglou and Condon (1994), shown for $n = 4$ ($k = 2$). States τ_1 and τ_0 are terminal. All rewards are 0 excepting a 1 for reaching τ_1 . The two actions are 0 (solid, black) and 1 (dashed, red). There are no decisions to make at “states” $1', 2', 3',$ and $4'$, which transition with equal probability to their next states.



(b) Transitions in the MDP of Ashutosh et al. (2020), shown for $n = 8, k = 3$. Assume $n = 2m$. State τ is terminal. All transitions are deterministic. Each state $s \in \{1, 2, \dots, m\}$ has a partner state $s + m$ with the same rewards and transitions for each action. For state $s \in \{1, 2, \dots, m\}$ and action $a \in \{0, 1, \dots, k - 1\}$, the reward is $a \cdot k^{m-s}$.

Figure 2: Instances of existing lower-bound constructions. Both use total reward ($\gamma = 1$).

ble configurations of actions. Unfortunately, the construction uses up another m states to devise subsequences of policies between any consecutive configurations of this k -ary counter. Consequently the largest that m can be made is $\frac{n}{2}$. Correspondingly the lower bound on $f(n, k)$ that is furnished by Ashutosh et al. (2020) is $\Omega(k^{n/2})$.

Our work is motivated precisely by the current gap between lower and upper bounds for $k \geq 3$. Like Ashutosh et al. (2020), we also construct a deterministic MDP, using total reward. However, we devise an m -ary counter for $m = \lfloor \frac{k}{2} \rfloor$. In analogous terms, we use $\lfloor \frac{k}{2} \rfloor$ actions as helpers, rather than $\frac{n}{2}$ states, to get a lower bound of $\Omega(\lfloor \frac{k}{2} \rfloor^n)$.

4 Tight Lower Bound for 2-state MDPs

In this section, we consider the special case of tasks with $n = 2$ states (and for concreteness, assume there are no terminal states). This setup facilitates a 2-dimensional visualisation of the challenge of constructing a lower bound for $k \geq 3$, preparing us for our general result in Section 5.

Since $n = 2$, we have the set of states to be $S = \{1, 2\}$, while the set of actions is $A = \{0, 1, \dots, k - 1\}$. We introduce simplifying notation for the transition probabilities and rewards. For $s \in S, a \in A$, let λ_a^s denote the probability of transitioning from s to s by taking action a . Hence, we have $T(s, a, s) = \lambda_a^s$ and $T(s, a, s') = 1 - \lambda_a^s$ for $s' \in S \setminus \{s\}$. Similarly, for $s \in S, a \in A$, let μ_a^s denote the expected reward obtained by taking action a from state s . In other words, $R(s, a) = \mu_a^s$. Figure 3(a) pictorially depicts state transitions in terms of λ and μ . We continue to denote the discount factor γ . For $i, j \in A$, let $\langle i, j \rangle$ denote the policy that takes action i from state 1 and action j from state 2.

4.1 Design Challenge

The Bellman equations in (2) for $\pi = \langle i, j \rangle$ are equivalently:

$$V^\pi(2) = -\frac{\mu_i^1}{\gamma - \gamma\lambda_i^1} + \left(\frac{1 - \gamma\lambda_i^1}{\gamma - \gamma\lambda_i^1}\right) V^\pi(1); \quad (5)$$

$$V^\pi(2) = \frac{\mu_j^2}{1 - \gamma\lambda_j^2} + \left(\frac{\gamma - \gamma\lambda_j^2}{1 - \gamma\lambda_j^2}\right) V^\pi(1). \quad (6)$$

Now consider plotting the point $(V^\pi(1), V^\pi(2))$ on the xy plane with $V^\pi(1)$ as x coordinate and $V^\pi(2)$ as y coordinate. From (5), we observe that every policy taking action i from state 1 lies on a line determined by action i , and similarly, from (6), that every policy taking action j from state 2 lies on a line determined by action j . Moreover, “ i -lines” all have slopes larger than 1, while “ j -lines” have slopes smaller than 1. Figure 3(b) illustrates a locally-improving sequence of neighbouring policies with this visualisation.

Observe that a locally-improving sequence of neighbouring policies can be viewed as a sequence of line segments $(\pi_1, \pi_2), (\pi_2, \pi_3), \dots$. The slope of the line segment is decided by the action shared by its termini. Now suppose that a policy $\pi_1 = \langle i_1, j_1 \rangle$ in a locally-improving sequence is followed (not necessarily consecutively) by $\pi_2 = \langle i_2, j_2 \rangle$, with $i_2 \neq i_1$. Thereafter, suppose $\pi_3 = \langle i_1, j_3 \rangle$ is encountered. Note that π_1 and π_3 share their “ i ” action (which is i_1), and

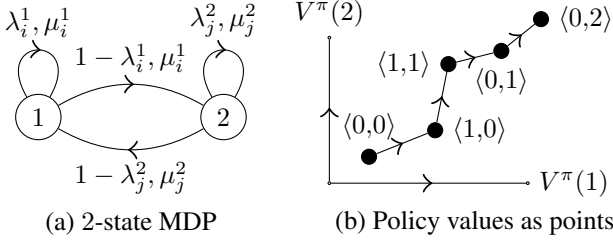


Figure 3: (a) “Probability, reward” pairs for the transitions out of state 1 for action $i \in A$, and out of state 2 for action $j \in A$. There are k such actions for each state (not all displayed). (b) Values of a locally-improving sequence of neighbouring policies plotted as points. Observe that points corresponding to $\langle 0, 0 \rangle$, $\langle 0, 1 \rangle$, and $\langle 0, 2 \rangle$ are collinear.

moreover, there is at least one line segment in between them with a common “ j ” action (hence with a slope smaller than 1). It follows that in between π_1 and π_3 , there must be at least one line segment with a slope larger than that of i_1 . This must be a line segment whose shared action is an “ i ” action (since j actions all have slopes smaller than 1). We infer two symmetric constraints. (Constraint 1) If action i at state 1 is switched out and then it returns in a sequence, there must be another action i' in between the occurrences of i whose slope is larger than that of i . (Constraint 2) If action j at state 2 is switched out and then it returns in a sequence, there must be another action j' in between the occurrences of j whose slope is smaller than that of j .

The first step in designing a lower bound on $f(n, k)$ (here $f(2, k)$) is to propose a sequence of policies, and the next step is to furnish an MDP to facilitate the sequence. However, many sequences that appear “intuitive” and “easy-to-define” are ruled out by the constraints outlined above. Figures 4(a) and 4(b) show two such infeasible sequences for $k = 3$; the reader can verify that any assignment of slopes to the rows and columns will cause both constraints to be violated. However, if $k = 2$, there are only two possible sequences to try out (modulo i - j symmetry), and one of them has been shown to work even for $n \geq 3$ (Melekopoglou and Condon 1994).

For $n \geq 3$, each action corresponds to an $(n - 1)$ -dimensional hyperplane. If additionally $k \geq 3$, we have more complex constraints than the two specified above on sets of policies that can be arranged as locally-improving sequences. This makes it a harder task for designers to come up with tight lower bounds on $f(n, k)$.

4.2 k^2 Lower Bound

Indeed we are able to furnish a locally-improving policy sequence for 2-state, k -action MDPs that visits all k^2 policies on a family of MDPs. The sequence satisfies both constraints given above; it is presented in Figure 4 (c) for $k = 5$. Observe that the sequence traverses the last row and then the last column, thereafter recursing on a matrix with one fewer row and column. For general $k \geq 3$, the first policy in the sequence is $\langle k - 1, 0 \rangle$, and the last policy is $\langle 0, 0 \rangle$. In general,

for each policy $\langle i, j \rangle$, where $i, j \in A$ and $i \neq 0$ or $j \neq 0$, the next policy in the sequence is given by function

$$\text{next}(\langle i, j \rangle) = \begin{cases} \langle i, j + 1 \rangle, & \text{if } i > j, \\ \langle 1, j \rangle, & \text{if } i = j, \\ \langle i + 1, j \rangle, & \text{if } i < j - 1, \\ \langle i, 1 \rangle, & \text{if } i = j - 1. \end{cases} \quad (7)$$

We provide pseudocode in Figure 4 to specify a family of 2-state MDPs on which the sequence constructed above is locally-improving.² Figure 4(d) provides the parameters of an MDP with $k = 5$, which is obtained by executing the pseudocode. Our construction—the policy sequence specified by (7), along with the MDP specified in Figure 4(e)—establishes the following result.

Theorem 3. Fix $k \geq 2$ and $\gamma \in (0, 1)$. There exist

1. MDP $M = (S, A, T, R, \gamma, \emptyset)$ with $|S| = 2$, $|A| = k$, and
2. k^2 distinct policies $\pi_0, \pi_1, \dots, \pi_{k^2-1}$ for M

which constitute a locally-improving policy sequence for M .

The proof of the theorem—essentially the calculations justifying our construction—is provided in Appendix B.

5 Main Result: $\Omega(\lfloor k/2 \rfloor^n)$ Lower Bound

In this section, we prove our main result, stated below for the total reward setting.

Theorem 4. Fix $n \geq 2$, $k \geq 2$. There exist

1. MDP $M = (S, A, T, R, \gamma = 1, \bar{S})$ with $|S| = n$, $|A| = k$, and
2. $N + 1$ distinct policies $\pi_0, \pi_1, \dots, \pi_N$ for M , where $N = \Omega(\lfloor \frac{k}{2} \rfloor^n)$,

which constitute a locally-improving policy sequence for M .

Let $\Pi^{n,k}$ denote the set of all policies for n -state, k -action MDPs. In Section 2 we denoted this same set Π , since n and k were clear from the context—but here we shall vary n . Each policy in $\Pi^{n,k}$ is represented as an n -length string over the alphabet $\{1, 2, \dots, k\}$ (we use xy to denote the concatenation of strings x and y). In this section, the j -th entry of the policy string shall represent the action at state $n + 1 - j$. Our first step is to propose a sequence of policies from $\Pi^{n,k}$ that is as long as claimed in Theorem 4. Next, we define a family of MDPs parameterised by n and k . Finally, we prove that the specified sequence of policies is locally-improving on any MDP from the family. For convenient exposition, we assume in this section that k is an even number (for odd k , the construction can proceed using $k - 1$ actions, with one action left unutilised). We also partition the set of actions $A = \{0, 1, \dots, k - 1\}$ into sets

$$A_{\text{even}} \stackrel{\text{def}}{=} \{0, 2, \dots, k - 2\} \text{ and } A_{\text{odd}} \stackrel{\text{def}}{=} \{1, 3, \dots, k - 1\}.$$

5.1 Policy Sequence

For each $u \in A_{\text{odd}}$, we recursively define a sequence of policies $\Pi_u^{n,k} \subset \Pi^{n,k}$ such that the only odd action taken by any policy $\pi \in \Pi_u^{n,k}$ in state n is action u . Let $\ell(n, k)$ be the total

i	j		
	0	1	2
0	0	1	2
1	5	4	3
2	6	7	8

i	j		
	0	1	2
0	0	1	2
1	7	8	3
2	6	5	4

i	j				
	0	1	2	3	4
0	24	23	19	13	5
1	21	22	20	14	6
2	16	17	18	15	7
3	9	10	11	12	8
4	0	1	2	3	4

a	$\lambda_a^1 = \lambda_a^2$	μ_a^1	μ_a^2
0	0.166667	0	0
1	0.333333	-10.6667	-1
2	0.5	-710.741	-93.4444
3	0.666667	-26078.3	-4678.83
4	0.833333	-424201	-119094

$\gamma = 0.9$

(a) Infeasible ($k = 3$) (b) Infeasible ($k = 3$) (c) Feasible ($k = 5$) (d) MDP supporting sequence in (c)

1. Select γ arbitrarily from $(0, 1)$. Select λ_0^1 and λ_0^2 arbitrarily from $(0, 1)$. Set μ_0^1 and μ_0^2 to be arbitrary finite real numbers.
 2. For $a = 0, 1, 2, \dots, k-2$:
 3. Select λ_{a+1}^1 arbitrarily from $(\lambda_a^1, 1)$.
 4. Select λ_{a+1}^2 arbitrarily from $(\lambda_a^2, 1)$.
 5. Set μ_{a+1}^2 to an arbitrary finite real number smaller than
$$\min \left\{ \min_{0 \leq i \leq a-1} \left(\mu_i^1 + \frac{(\mu_{i+1}^1 - \mu_i^1)(1 + \gamma(1 - \lambda_i^1 - \lambda_{a+1}^1))}{\gamma(\lambda_{i+1}^1 - \lambda_i^1)} \right), \frac{\mu_1^2(1 + \gamma(1 - \lambda_1^1 - \lambda_{a+1}^2)) + \gamma(\lambda_{a+1}^2 - \lambda_1^2)\mu_a^1}{1 + \gamma(1 - \lambda_1^1 - \lambda_1^2)} \right\}.$$
 6. Set μ_{a+1}^1 to an arbitrary finite real number smaller than
$$\min \left\{ \min_{0 \leq j \leq a} \left(\mu_j^2 + \frac{(\mu_{j+1}^2 - \mu_j^2)(1 + \gamma(1 - \lambda_{a+1}^1 - \lambda_j^2))}{\gamma(\lambda_{j+1}^2 - \lambda_j^2)} \right), \frac{\mu_1^1(1 + \gamma(1 - \lambda_{a+1}^1 - \lambda_{a+1}^2)) + \gamma(\lambda_{a+1}^1 - \lambda_1^1)\mu_{a+1}^2}{1 + \gamma(1 - \lambda_1^1 - \lambda_{a+1}^2)} \right\}.$$
- (e) Steps to set 2-state MDP parameters $(\lambda_0^1, \lambda_1^1, \dots, \lambda_{k-1}^1), (\lambda_0^2, \lambda_1^2, \dots, \lambda_{k-1}^2), (\mu_0^1, \mu_1^1, \dots, \mu_{k-1}^1), (\mu_0^2, \mu_1^2, \dots, \mu_{k-1}^2), \gamma$.

Figure 4: In (a), (b), and (c), each matrix entry shows the index of a corresponding policy (read as $\langle \text{row}, \text{column} \rangle$) in the visitation sequence. For example, the sequence in (a) is $\langle 0, 0 \rangle, \langle 0, 1 \rangle, \langle 0, 2 \rangle, \langle 1, 2 \rangle, \langle 1, 1 \rangle, \langle 1, 0 \rangle, \langle 2, 0 \rangle, \langle 2, 1 \rangle, \langle 2, 2 \rangle$. The sequences in (a) and (b) cannot be locally-improving on any MDP. The sequence in (c) is locally-improving on the MDP specified in (d). (e) Steps to set 2-state MDP parameters $(\lambda_0^1, \lambda_1^1, \dots, \lambda_{k-1}^1), (\lambda_0^2, \lambda_1^2, \dots, \lambda_{k-1}^2), (\mu_0^1, \mu_1^1, \dots, \mu_{k-1}^1), (\mu_0^2, \mu_1^2, \dots, \mu_{k-1}^2)$, and γ , such that starting from $\langle k-1, 0 \rangle$, the next function defined in (7) yields a locally-improving policy sequence. In the MDP shown in (d), γ is set to 0.9, and for $a \in A$, $\lambda_a^1 = \lambda_a^2 = \frac{a}{k+1}$. The rewards for action 0—that is, μ_0^1 and μ_0^2 —are both set to 0. For $a \in A \setminus \{k-1\}$, μ_{a+1}^1 and μ_{a+1}^2 are set to be 1 less than the smallest quantities on lines 5 and 6, respectively.

number of policies in $\Pi_u^{n,k}$. Also, for $i \in \{1, 2, \dots, \ell(n, k)\}$, let $\Pi_u^{n,k}(i)$ denote the i -th policy in $\Pi_u^{n,k}$.

The recursion is on n . As base case, we define

$$\Pi_u^{1,k} \stackrel{\text{def}}{=} (0, 2, 4, \dots, k-4, k-2, u).$$

Hence, observe that $\ell(1, k) = \frac{k+2}{2}$. To construct $\Pi_u^{n,k}$ for $n \geq 2$, we start from the $k/2$ sequences $\Pi_{u'}^{n-1,k}$ for all $u' \in A_{\text{odd}}$. The recursive definition of $\Pi_u^{n,k}$ is conveniently visualised as the row-major traversal of Table 1. The table has $k/2$ rows and $\ell(n-1, k) + 1$ fully-filled columns, and one additional policy $u0^{n-1}$ (u followed by $(n-1)$ 0's). For $n \geq 2$, we observe that the sequence lengths satisfy $\ell(n, k) = \frac{k}{2}(\ell(n-1, k) + 1) + 1$, which implies

$$\ell(n, k) = \frac{k+2}{k-2} \left(\binom{k}{2} - 1 \right). \quad (8)$$

The essential idea in the recursive construction of $\Pi_u^{n,k}$ is to ensure that all $(k/2)^n$ “even-only” policies (whose n actions are all from A_{even}) find place in it. Policies that take

²C++ code for our constructions (sections 4, 5) is provided at <https://github.com/pratyush1019/Improved-Lower-Bound-MDP/>.

odd actions facilitate transitions between even-only policies. Correspondingly, notice that entries in $\Pi_u^{n,k}$ (other than the final one) are obtained by prefixing policies in the sequences $\Pi_{u'}^{n-1,k}$ either with actions from A_{even} or with the odd action u . Formally, for $i \in \{1, 2, \dots, \ell(n, k) - 1\}$, define

$$\text{row}^{n,k}(i) = \left\lceil \frac{i}{\ell(n-1, k) + 1} \right\rceil;$$

$$\text{col}^{n,k}(i) = i - (\text{row}^{n,k}(i) - 1)(\ell(n-1, k) + 1).$$

For $i \in \{1, 2, \dots, \ell(n, k)\}$, $\Pi_u^{n,k}(i)$ is defined as follows.

- If $i = \ell(n, k)$, then $\Pi_u^{n,k}(i) \stackrel{\text{def}}{=} u0^{n-1}$,
- else if $\text{col}^{n,k}(i) \leq \ell(n-1, k)$, then

$$\Pi_u^{n,k}(i) \stackrel{\text{def}}{=} (\text{row}^{n,k}(i) - 1) \Pi_{\text{row}^{n,k}(i)}^{n-1,k}(\text{col}^{n,k}(i)),$$

- else $(\text{col}^{n,k}(i) \text{ must be equal to } \ell(n-1, k) + 1)$

$$\Pi_u^{n,k}(i) \stackrel{\text{def}}{=} (\text{row}^{n,k}(i) + 1) \Pi_{\text{row}^{n,k}(i)}^{n-1,k}(\ell(n-1, k)).$$

For illustration, we enumerate $\Pi_1^{3,6}$ (of length 52) in Appendix C. Our repository includes C++ code to generate

$0\Pi_1^{n-1,k}(1)$	$0\Pi_1^{n-1,k}(2)$	\dots	$0\Pi_1^{n-1,k}(\ell_0)$	$2\Pi_1^{n-1,k}(\ell_0)$	
$2\Pi_3^{n-1,k}(1)$	$2\Pi_3^{n-1,k}(2)$	\dots	$2\Pi_3^{n-1,k}(\ell_0)$	$4\Pi_3^{n-1,k}(\ell_0)$	
$4\Pi_5^{n-1,k}(1)$	$4\Pi_5^{n-1,k}(2)$	\dots	$4\Pi_5^{n-1,k}(\ell_0)$	$6\Pi_5^{n-1,k}(\ell_0)$	
\vdots	\vdots	\vdots	\vdots	\vdots	
$(k-4)\Pi_{k-3}^{n-1,k}(1)$	$(k-4)\Pi_{k-3}^{n-1,k}(2)$	\dots	$(k-4)\Pi_{k-3}^{n-1,k}(\ell_0)$	$(k-2)\Pi_{k-3}^{n-1,k}(\ell_0)$	
$(k-2)\Pi_{k-1}^{n-1,k}(1)$	$(k-2)\Pi_{k-1}^{n-1,k}(2)$	\dots	$(k-2)\Pi_{k-1}^{n-1,k}(\ell_0)$	$u\Pi_{k-1}^{n-1,k}(\ell_0)$	$u0^{n-1}$

Table 1: The sequence of policies $\Pi_u^{n,k}$ shown in row major order. Here $\ell_0 \stackrel{\text{def}}{=} \ell(n-1, k)$.

$\Pi_u^{n,k}$ for arbitrary n, k, u , and validate that it is locally-improving on the family of MDPs we define next.

5.2 Family of MDPs

For each $n \geq 1$ and even $k \geq 2$, we define a deterministic MDP $M^{n,k}$. The transition function T is such that even actions $v \in A_{\text{even}}$ increment the state index by 1 (except at state n , which transitions into a terminal state τ). On the other hand, odd actions increment the state by 2 (except for states $n-1$ and n , which transition to τ). Figure 5 illustrates this transition structure using $M^{4,k}$ as an example.

Notice that every policy must terminate in at most n steps. Leaving the MDP undiscounted, we administer rewards so that “even-only” policies (whose actions are all from A_{even}) function as states of a counter, while the remaining policies facilitate transitions among the even-only policies. Below are formal definitions of T and R .

For $s, s' \in S, v \in A_{\text{even}}$,

$$T(s, v, s') = \begin{cases} 1 & \text{if } s \leq n-1 \text{ and } s' = s+1, \\ 1 & \text{if } s = n \text{ and } s' = \tau, \\ 0 & \text{otherwise;} \end{cases}$$

$$R(s, v) = vk^{s-1}.$$

For $s, s' \in S, u \in A_{\text{odd}}$,

$$T(s, u, s') = \begin{cases} 1 & \text{if } s \leq n-2 \text{ and } s' = s+2, \\ 1 & \text{if } s \in \{n-1, n\} \text{ and } s' = \tau, \\ 0 & \text{otherwise;} \end{cases}$$

$$R(s, u) = (k-1)k^{s-1} + (u-1)k^s.$$

It is convenient to visualise the connection between even-only policies and their values for $k=10$, since this would implement the common decimal system. For example, notice that on $M^{4,10}$ (whose transitions are shown in Figure 5), the policy 6424 would have values of 6000, 6400, 6420, and 6424 for states 4, 3, 2, and 1, respectively.

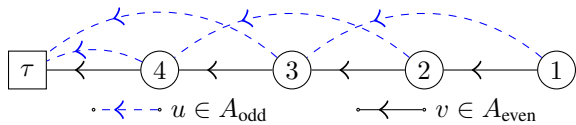


Figure 5: Transitions for even (solid, black) and odd (dashed, blue) actions in $M^{4,k}$. State τ is terminal. Rewards are action-dependent, and specified in the text.

5.3 Proof of Theorem 4

We have already constructed the sequence $\Pi_u^{n,k}$ of length $\ell(n, k) = \Omega(\binom{k}{2}^n)$. For proving Theorem 4, we show that the sequence is locally-improving on $M^{n,k}$.

Our proof is by induction on n . As base case, consider $\Pi_u^{1,k} = (0, 2, 4, \dots, k-2, u)$ for $u \in A_{\text{odd}}$, as defined in Section 5.1. On $M^{1,k}$ the single state 1 has values $0, 2, 4, \dots, (k-2), (ku-1)$, respectively, for these policies. Clearly the sequence of values is increasing for all $u \in A_{\text{odd}}$.

Our induction hypothesis is that $\Pi_u^{n-1,k}$ is a locally-increasing policy sequence for $M^{n-1,k}$, for some $n \geq 2$. To show that $\Pi_u^{n-1,k}$ is a locally-increasing policy sequence for $M^{n,k}$, we consider every pair of consecutive policies π, π' in the sequence $\Pi_u^{n,k}$. We find it instructive to return to Table 1, which guides us to divide our proof that $\pi' \succ \pi$ on M into five cases. Denote $M^{n,k}$ by M and $M^{n-1,k}$ by M' . To achieve clarity in the steps below, we suffix relevant objects with the MDP on which they are defined or applied.

Case 1. $\pi = (x-1)\Pi_x^{n-1,k}(i)$ and $\pi' = (x-1)\Pi_x^{n-1,k}(i+1)$ for some $1 \leq i < \ell(n-1, k)$ and $x \in A_{\text{odd}}$. In this case, π is in row $\frac{x+1}{2}$ and column i of Table 1, while π' is in row $\frac{x+1}{2}$ and column $i+1$. For brevity denote $\Pi_x^{n-1,k}(i)$ by π_i and $\Pi_x^{n-1,k}(i+1)$ by π_{i+1} : thus, $\pi = (x-1)\pi_i$ and $\pi' = (x-1)\pi_{i+1}$. It is clear that $V_M^{\pi'}(n) = V_M^\pi(n) = R_M(n, x-1)$. For arbitrary state $n-1 \geq s \geq 1$, consider the following sub-cases. In a given MDP, we say state s reaches state s' under policy π if starting from s and following π takes the agent to s' .

(1a) If s reaches n on M both under π and π' , then $V_M^\pi(s) = V_{M'}^{\pi_i}(s) + R_M(n, x-1)$, and $V_M^{\pi'}(s) = V_{M'}^{\pi_{i+1}}(s) + R_M(n, x-1)$. By the induction hypothesis, $\pi_{i+1} \succ_{M'} \pi_i$ —and hence $\pi' \succ_M \pi$.

(1b) If s does not reach n on M both under π and π' , then it must reach $n-1$ under both these policies. It can only be that $\pi(n-1) = \pi'(n-1) = x$ (which is the only odd action possible at state $n-1$ in $\Pi_x^{n-1,k}$). Since x transitions from $n-1$ to τ in both policies, the value at $n-1$ does not depend on the action at n for both policies. Indeed $V_M^{\pi'}(s) = V_{M'}^{\pi_{i+1}}(s)$ and $V_M^\pi(s) = V_{M'}^{\pi_i}(s)$. By the induction hypothesis, $\pi' \succ_M \pi$.

(1c) Suppose s reaches n on M under π , but not under π' . In turn this must mean s reaches $n-1$ under π' , and $\pi'(n-1)$

1) = x . Clearly $V_M^{\pi'}(s)$ is at least $R_M(n-1, x)$. On the other hand, $V_M^\pi(s)$ is at most the sum of rewards on any path that reaches $n-2$ and then n . The onward reward from n is $R_M(n, x-1)$. By our definition of R_M , the maximum cumulative reward that can be accrued up to n is by taking odd action $k-1$ at states $n-2, n-4, n-6, \dots$, and if n is even, by taking even action $k-2$ at state 1. Regardless, we can upper-bound $V_M^\pi(s)$ as

$$\begin{aligned} V_M^\pi(s) &\leq \sum_{r=0}^{\infty} ((k-1)k^{n-3-2r} + (k-2)k^{n-2-2r}) + R_M(n, x-1) \\ &= \frac{k^2 - k - 1}{k^2 - 1} k^{n-1} + (x-1)k^{n-1} \\ &< (k-1)k^{n-2} + (x-1)k^{n-1} = R_M(n-1, x) \leq V_M^{\pi'}(s). \end{aligned}$$

The working shown above to establish that $V_M^{\pi'}(s) > V_M^\pi(s)$ in this case is the most critical aspect in our design of $M^{n,k}$.

(1d) We cannot have the case that s does not reach n on M under π , but it reaches it under π' . On the contrary, if this happens, then by the argument in (1c), we would have $V_M^\pi(s) > V_M^{\pi'}(s)$. However, since $V_M^\pi(s) = V_{M'}^\pi(s)$, and $V_M^{\pi'}(s) > V_{M'}^{\pi'}(s)$, this would imply $V_{M'}^\pi(s) > V_{M'}^{\pi'}(s)$, which contradicts the induction hypothesis.

Case 2. $\pi = (x-1)\Pi_x^{n-1,k}(\ell(n-1, k))$ and $\pi' = (x+1)\Pi_x^{n-1,k}(\ell(n-1, k))$ for some $x \in A_{\text{odd}} \setminus \{k-1\}$. Here π and π' are columns $\ell(n-1, k)$ and $\ell(n-1, k) + 1$, respectively, in row $\frac{x+1}{2}$ of the table.

From each state, the sequence of states visited is identical under π and π' . The rewards are also identical at all states except n , where π' has a higher reward. Hence $\pi' \succ_M \pi$.

Case 3. $\pi = (x+1)\Pi_x^{n-1,k}(\ell(n-1, k))$ and $\pi' = (x+1)\Pi_{x+2}^{n-1,k}(1)$ for some $x \in A_{\text{odd}} \setminus \{k-1\}$. Here π is the last entry in row $\frac{x+1}{2}$, and π' is the first entry in row $\frac{x+3}{2}$.

It can be worked out that $\pi = (x+1)x0^{n-2}$, while $\pi' = (x+1)0^{n-1}$. Clearly $V_M^\pi(n) = V_M^{\pi'}(n) = R_M(n, x+1)$. Meanwhile, $V_M^\pi(n-1) = R_M(n-1, x) = (k-1)k^{n-2} + (x-1)k^{n-1}$ and $V_M^{\pi'}(n-1) = (x+1)k^{n-1}$. Hence, $V_M^{\pi'}(n-1) > V_M^\pi(n-1)$. Under both policies, states $n-2, n-3, \dots, 1$ reach $n-1$ with 0 reward, establishing that $\pi' \succ_M \pi$.

Case 4. $\pi = (k-2)\Pi_{k-1}^{n-1,k}(\ell(n-1, k))$ and $\pi' = u\Pi_{k-1}^{n-1,k}(\ell(n-1, k))$. Here π is the last-but-two entry and π' the last-but-one entry in the table.

These policies only differ on state n , from which π' has a higher reward than π : $R_M(n, u) = (k-1)k^{n-1} + (u-1)k^n > (k-2)k^{n-1} = R_M(n, k-2)$. Clearly $\pi' \succ_M \pi$.

Case 5. $\pi = u\Pi_{k-1}^{n-1,k}(\ell(n-1, k))$ and $\pi' = u0^{n-1}$. Here π is the last-but-one entry and π' the last entry in the table.

It can be worked out that $\pi = u^2 0^{n-2}$. We observe $V_M^\pi(n) = V_M^{\pi'}(n) = R_M(n, u)$. Also, $V_M^\pi(n-1) = R_M(n-1, u)$ while $V_M^{\pi'}(n-1) = R_M(n, u)$. Hence, $V_M^{\pi'}(n-1) > V_M^\pi(n-1)$. Like in case 3, under both policies, states $n-2, n-3, \dots, 1$ reach $n-1$ with 0 reward, establishing that $\pi' \succ_M \pi$.

This exhaustive set of five cases completes the proof that $\Pi^{n,k}$ is a locally-improving policy sequence for $M^{n,k}$.

We note that our construction generalises to MDPs with a *non-uniform* number of actions per state. Suppose state i has k_i actions, where k_i is even. Without loss of generality, let the states be numbered such that their action counts are in non-increasing order. Table 1 would now still look the same, except that to obtain i -length policies, the table would have $\frac{k_i}{2}$ rows. The recurrence on the size of the sequences for $n \geq 2$ becomes:

$$\ell(n) = \frac{k_n}{2} (\ell(n-1) + 1) + 1, \text{ with } \ell(1) = \frac{k_1 + 1}{2}.$$

Thus $\ell(n) \geq \prod_{i=1}^n \frac{k_i}{2}$, which generalises our lower bound of $(k/2)^n$. We again notice that this sequence visits all the ‘‘even-only’’ policies, while using the odd actions to transition between them. Our code incorporates the generalisation to a non-uniform number of actions per state.

6 Summary and Outlook

In this paper, we have improved the lower bound of Ashutosh et al. (2020) on $f(n, k)$, making it exponentially tighter for $k \geq 6$. Our construction using total reward can be adapted to work for discount factor $\gamma \in (\gamma_0, 1)$, where γ_0 could depend on n and k . This is because values vary continuously with γ . We have also shown by construction that $f(2, k) = k^2$. It is straightforward to translate this discounted MDP to become undiscounted and use total reward, by introducing a terminal state into which existing transitions are routed with probability γ .

Our investigation leaves it open if $f(n, k) = k^n$ for $k \geq 3$. A related question that emerges is whether stochasticity is strictly necessary to support a locally-improving policy sequence of length k^n . The answer is affirmative. Taking $n = 2$ again as a special case, we prove the following *upper bound* for deterministic MDPs in Appendix D.

Proposition 5. Fix MDP $M = (S, A, T, R, \gamma, \bar{S})$ with $|S| = 2$, $|A| = k$, where the range of T is $\{0, 1\}$ (all transitions are deterministic). Let $\pi_0, \pi_1, \dots, \pi_N$ be a locally-improving policy sequence for M . Then $N \leq \frac{k^2}{2} + 2k - 1$.

Interestingly, Goenka et al. (2025) have used the structural properties of deterministic MDPs to show that locally-improving policy sequences on them are at most of length $(k-0.18)^n$. These results set the stage for exploring constructions that use stochasticity. As yet, it even remains unresolved whether all $3^3 = 27$ policies can form a locally-improving sequence on some 3-state, 3-action MDP.

Ethical/Societal Impact

The authors do not assess this work to have notable ethical or societal implications.

Acknowledgements

The authors thank Ritesh Goenka, Eashan Gupta, and Sushil Khyalia for reviewing preliminary drafts of this paper.

References

- Ashutosh, K.; Consul, S.; Dedhia, B.; Khirwadkar, P.; Shah, S.; and Kalyanakrishnan, S. 2020. Lower bounds for policy iteration on multi-action MDPs. In *Proceedings of the 59th IEEE Conference on Decision and Control*, 1744–1749. IEEE.
- Avis, D.; and Friedmann, O. 2017. An exponential lower bound for Cunningham’s rule. *Mathematical Programming*, 161(1-2): 271–305.
- Avis, D.; Miyata, H.; and Moriyama, S. 2013. Families of polytopal digraphs that do not satisfy the shelling property. *Computational Geometry*, 46(3): 382–393.
- Bellman, R. 1957. *Dynamic Programming*. Princeton, NJ, USA: Princeton University Press, 1st edition.
- Bertsekas, D. 2012. *Dynamic programming and optimal control: Volume I*, volume 4. Athena scientific.
- Blanchard, M.; Loera, J. A. D.; and Louveaux, Q. 2021. On the Length of Monotone Paths in Polyhedra. *SIAM J. Discret. Math.*, 35(3): 1746–1768.
- Dantzig, G. B. 1963. *Linear Programming and Extensions*. Princeton University Press.
- Disser, Y.; Friedmann, O.; and Hopp, A. V. 2023. An exponential lower bound for Zadeh’s pivot rule. *Mathematical Programming*, 199(1): 865–936.
- Fearnley, J. 2010. Exponential Lower Bounds For Policy Iteration. In *Proceedings of the 37th International Colloquium on Automata, Languages and Programming (ICALP 2010)*, 551–562. Springer.
- Goenka, R.; Gupta, E.; Khyalia, S.; Agarwal, P.; Wajid, M. S.; and Kalyanakrishnan, S. 2025. Upper Bounds for All and Max-Gain Policy Iteration Algorithms on Deterministic MDPs. *Mathematics of Operations Research*. Published on-line April 2025.
- Gray, F. 1953. Pulse code communication. U.S. patent no. 2,632,058.
- 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.
- Howard, R. A. 1960. *Dynamic Programming and Markov Processes*. MIT Press.
- Joichi, J. T.; White, D. E.; and Williamson, S. G. 1980. Combinatorial Gray Codes. *SIAM Journal on Computing*, 9(1).
- Kalai, G. 1992. Upper Bounds for the Diameter and Height of Graphs of Convex Polyhedra. *Discret. Comput. Geom.*, 8: 363–372.
- Klee, V.; and Minty, G. J. 1972. How good is the simplex algorithm? In *Inequalities III: Proceedings of the third Symposium, University of California, Los Angeles*, 159–175. Academic Press.
- Mansour, Y.; and Singh, S. 1999. On the complexity of policy iteration. In *Proceedings of the Fifteenth conference on Uncertainty in Artificial Intelligence*, 401–408.
- Mausam; and Kolobov, A. 2012. *Planning with Markov Decision Processes: An AI Perspective*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers.
- Melekopoglou, M.; and Condon, A. 1994. On the Complexity of the Policy Improvement Algorithm for Markov Decision Processes. *ORSA Journal on Computing*, 6(2): 188–192.
- Post, I.; and Ye, Y. 2015. The simplex method is strongly polynomial for deterministic Markov decision processes. *Mathematics of Operations Research*, 40(4): 859–868.
- Puterman, M. L. 2014. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.
- Squire, M. B. 1996. Gray Codes for A-Free Strings. *Electronic Journal of Combinatorics*, 3(1).
- Sutton, R. S.; and Barto, A. G. 1998. *Reinforcement Learning: An Introduction*. MIT Press.
- Szepesvári, C. 2009. *Algorithms for Reinforcement Learning*. Morgan & Claypool.
- Taraviya, M.; and Kalyanakrishnan, S. 2020. A Tighter Analysis of Randomised Policy Iteration. In *Proceedings of the 35th Uncertainty in Artificial Intelligence Conference*, volume 115 of *Proceedings of Machine Learning Research*, 519–529.
- Ye, Y. 2011. The simplex and policy-iteration methods are strongly polynomial for the Markov decision problem with a fixed discount rate. *Mathematics of Operations Research*, 36(4): 593–603.