The simplex method is strongly polynomial for deterministic Markov decision processes

Ian Post [*] Yinyu Ye [†]

May 31, 2013

Abstract

We prove that the simplex method with the highest gain/most-negative-reduced cost pivoting rule converges in strongly polynomial time for deterministic Markov decision processes (MDPs) regardless of the discount factor. For a deterministic MDP with n states and m actions, we prove the simplex method runs in $O(n^3 m^2 \log^2 n)$ iterations if the discount factor is uniform and $O(n^5 m^3 \log^2 n)$ iterations if each action has a distinct discount factor. Previously the simplex method was known to run in polynomial time only for discounted MDPs where the discount was bounded away from 1 [Ye11].

Unlike in the discounted case, the algorithm does not greedily converge to the optimum, and we require a more complex measure of progress. We identify a set of layers in which the values of primal variables must lie and show that the simplex method always makes progress optimizing one layer, and when the upper layer is updated the algorithm makes a substantial amount of progress. In the case of nonuniform discounts, we define a polynomial number of "milestone" policies and we prove that, while the objective function may not improve substantially overall, the value of at least one dual variable is always making progress towards some milestone, and the algorithm will reach the next milestone in a polynomial number of steps.

1 Introduction

Markov decision processes (MDPs) are a powerful tool for modeling repeated decision making in stochastic, dynamic environments. An MDP consists of a set of states and a set of actions that one may perform in each state. Based on an agent's actions it receives rewards and affects the future evolution of the process, and the agent attempts to maximize its rewards over time (see Section 2 for a formal definition). MDPs are widely used in machine learning, robotics and control, operations research, economics, and related fields. See the books [Put94] and [Ber96] for a thorough overview.

Solving MDPs is also an important problem theoretically. Optimizing an MDP can be formulated as a linear program (LP), and although these LPs possess extra structure that can be exploited by algorithms like Howard's policy iteration method [How60], they lie just beyond the point at which

[*] Department of Combinatorics and Optimization, University of Waterloo. Research done while at Stanford University. Email: ian@ianpost.org. Research supported by NSF grant 0904325. We also acknowledge financial support from grant #FA9550-12-1-0411 from the U.S. Air Force Office of Scientific Research (AFOSR) and the Defense Advanced Research Projects Agency (DARPA).

[†] Department of Management Science and Engineering, Stanford University. Email: yinyu-ye@stanford.edu. Research supported in part from grant #FA9550-12-1-0396 from the U.S. Air Force Office of Scientific Research (AFOSR).

our ability to solve LPs in strongly-polynomial time ends (and are a natural target for extending this ability), and they have proven to be hard in general for algorithms previously thought to be quite powerful, such as randomized simplex pivoting rules [FHZ11].

In practice [LDK95] MDPs are solved using policy iteration, which may be viewed as a parallel version of the simplex method with multiple simultaneous pivots, or value iteration [Bel57], an inexact approximation to policy iteration that is faster per iteration. If the discount factor γ, which determines the effective time horizon (see Section 2), is small it has long been known that policy and value iteration will find an ϵ-approximation to the optimum [Bel57]. It is also well-known that value iteration may be exponential, but policy iteration resisted worst-case analysis for many years. It was conjectured to be strongly polynomial but except for highly-restricted examples [Mad02] only exponential time bounds were known [MS99]. Building on results for parity games [Fri09], Fearnley recently gave an exponential lower bound [Fea10]. Friedmann, Hansen, and Zwick extended Fearnley's techniques to achieve sub-exponential lower bounds for randomized simplex pivoting rules [FHZ11] using MDPs, and Friedmann gave an exponential lower bound for MDPs using the least-entered pivoting rule [Fri11]. Melekopoglou and Condon proved several other simplex pivoting rules are exponential [MC94].

On the positive side, Ye designed a specialized interior-point method that is strongly polynomial in everything except the discount factor [Ye05]. Ye later proved that for discounted MDPs with n states and m actions, the simplex method with the most-negative-reduced-cost pivoting rule and, by extension, policy iteration, run in time $O(nm/(1-\gamma)\log(n/(1-\gamma)))$ on discounted MDPs, which is polynomial for fixed γ [Ye11]. Hansen, Miltersen, and Zwick improved the policy iteration bound to $O(m/(1-\gamma)\log(n/(1-\gamma)))$ and extended it to both value iteration as well as the strategy iteration algorithm for two player turn-based stochastic games [HMZ11].

But the performance of policy iteration and simplex-style basis-exchange algorithms on MDPs remains poorly understood. Policy iteration, for instance, is conjectured to run in $O(m)$ iterations on deterministic MDPs, but the best upper bounds are exponential, although a lower bound of $O(m)$ is known [HZ10]. Improving our understanding of these algorithms is an important step in designing better ones with polynomial or even strongly-polynomial guarantees.

Motivated by these questions, we analyze the simplex method with the most-negative-reduced-cost pivoting rule on deterministic MDPs. For a deterministic MDP with n states and m actions, we prove that the simplex method terminates in $O(n^3m^2\log^2 n)$ iterations regardless of the discount factor, and if each action has a distinct discount factor, then the algorithm runs in $O(n^5m^3\log^2 n)$ iterations. Our results do not extend to policy iteration, and we leave this as a challenging open question.

Deterministic MDPs were previously known to be solvable in strongly polynomial time using specialized methods not applicable to general MDPs—minimum mean cycle algorithms [PT87] or, in the case of nonuniform discounts, by exploiting the property that the dual LP has only two variables per inequality [HN94]. The fastest known algorithm for uniformly discounted deterministic MDPs runs in time $O(mn)$ [MTZ10]. However, these problems were not known to be solvable in polynomial time with the more-generic simplex method. More generally, we believe that our results help shed some light on how algorithms like simplex and policy iteration function on MDPs.

Our proof techniques, particularly in the case of nonuniform discounts, may be of independent interest. For uniformly discounted MDPs, we show that the values of the primal flux variables must lie within one of two intervals or layers of polynomial size depending on whether an action is on a path or a cycle. Most iterations update variables in the smaller path layer, and we show these converge rapidly to a locally optimal policy for the paths, at which point the algorithm must update the larger cycle layer and makes a large amount of progress towards the optimum. Progress takes the form of many small improvements interspersed with a few much larger ones rather than uniform convergence.

The nonuniform case is harder, and our measure of progress is unusual and, to the best of our knowledge, novel. We again define a set of intervals in which the value of variables on cycles must fall, and these define a collection of intermediate milestone or checkpoint values for each dual variable (the value of a state in the MDP). Whenever a variable enters a cycle layer, we argue that a corresponding dual variable is making progress towards the layer's milestone and will pass this value after enough updates. When each of these checkpoints have been passed, the algorithm must have reached the optimum. We believe some of these ideas may prove useful in other problems as well.

In Section 2 we formally define MDPs and describe a number of well-known properties that we require. In Section 3 we analyze the case of a uniform discount factor, and in Section 4 we extend these results to the nonuniform case.

2 Preliminaries

Many variations and extensions of MDPs have been defined, but we will study the following problem. A Markov decision process consists of a set of n states S and m actions A. Each action a is associated with a single state s in which it can be performed, a reward $\mathbf{r}_a \in \mathbb{R}$ for performing the action, and a probability distribution P_a over states to which the process will transition when using action a. We denote by $P_{a,s}$ the probability of transitioning to state s when taking action a. There is at least one action usable in each state. Let $\mathbf{r}$ be the vector of rewards indexed by a with entries $\mathbf{r}_a$, $A_s \subset A$ be the set of actions performable in state s, and P be the n by m matrix with columns P_a and entries $P_{a,s}$. We will restrict the distributions P_a to be deterministic for all actions, in which case states may be thought of as nodes in a graph and actions as directed edges. However, the results in this section apply to MDPs with stochastic transitions as well.

At each time step, the MDP starts in some state s and performs an action a admissible in state s, at which point it receives the reward $\mathbf{r}_a$ and transitions to a new state s' according to the probability distribution P_a. We are given a discount factor $\gamma < 1$ as part of the input, and our goal is to choose actions to perform so as to maximize the expected discounted reward we accumulate over an infinite time horizon. The discount can be thought of as a stopping probability—at each time step the process ends with probability $1 - \gamma$. Normally, the discount γ is uniform for the entire MDP, but in Section 4 we will allow each action to have a distinct discount γ_a.

Due to the Markov property—transitions depend only the current state and action—there is an optimal strategy that is memoryless and depends only on the current state. Let π be such a *policy*, a distribution of actions to perform for each state. This defines a Markov chain and a value for each state:

Definition 2.1. *Let π be a policy, P^π be the n by n matrix where $P^\pi_{s,s'}$ is the probability of transitioning from s' to s using π, and $\boldsymbol{r}_\pi$ the vector of expected rewards for each state according to the distribution of actions in π. The* value vector *$\boldsymbol{v}^\pi$ is indexed by states, and $\boldsymbol{v}^\pi_s$ is equal to the expected total discounted reward of starting in state s and following policy π. It is defined as $\boldsymbol{v}^\pi = \sum_{i \geq 0} (\gamma (P^\pi)^T)^i \boldsymbol{r}_\pi = (I - \gamma P^\pi)^{-T} \boldsymbol{r}_\pi$ or equivalently by*

$$\boldsymbol{v}^\pi = \boldsymbol{r}_\pi + \gamma (P^\pi)^T \boldsymbol{v}^\pi. \tag{1}$$

If policy π is randomized and uses two or more actions in some state s, then the value of $\mathbf{v}_s^\pi$ is an average of the values of performing each of the pure actions in s, and one of these is the largest. Therefore we can replace the distribution by a single action and only increase the value of the state. In the remainder of the paper we will restrict ourselves to pure policies in which a single action is taken in each state.

In addition to the value vector, a policy π also has an associated flux vector $\mathbf{x}^\pi$ that will play a critical role in our analysis. It acts as a kind of "discounted flow." Suppose we start with a single unit of "mass" on every state and then run the Markov chain. At each time step we remove $1-\gamma$ fraction of the mass on each state and redistribute the remaining mass according to the policy π. Summing over all time steps, the total amount of mass that passes through each action is its flux. More formally,

Definition 2.2. *Let π be a policy and P^π the n by n transition matrix for π formed by the columns P_a for actions in π. The* flux vector *$\boldsymbol{x}^\pi$ is indexed by actions. If action a is not in π then $\boldsymbol{x}_a^\pi = 0$, and if π uses a in state s, then $\boldsymbol{x}_a^\pi = \boldsymbol{z}_s$, where*

$$\boldsymbol{z} = \sum_{i \geq 0} (\gamma P^\pi)^i \boldsymbol{1} = (I - \gamma P^\pi)^{-1} \boldsymbol{1} \,, \tag{2}$$

and $\boldsymbol{1}$ is the all ones vector of dimension n. The flux is the total discounted number of times we use each action if we start the MDP in all states and run the Markov chain P^π discounting by γ each iteration.

Note that if $a \in \pi$ then $\mathbf{x}_a^\pi \geq 1$, since the initial flux placed on a's state always passes through a. Further note that each bit of flux can be traced back to one of the initial units of mass placed on each state, although the vector $\mathbf{x}^\pi$ sums flux from all states. This will be important in Section 4.

Solving the MDP can be formulated as the following primal/dual pair of LPs, in which the flux and value vectors correspond to primal and (possibly infeasible) dual solutions:

$$\begin{array}{rlrl} \textsc{Primal:} & & & \\ \text{maximize} & \sum_a \mathbf{r}_a \mathbf{x}_a & & \\ \text{subject to} & \forall s \in S, \quad \sum_{a \in A_s} \mathbf{x}_a & = 1 + \gamma \sum_a P_{a,s} \mathbf{x}_a & \\ & \mathbf{x} & \geq 0 & \end{array} \tag{3}$$

$$\begin{array}{rl} \textsc{Dual:} & \\ \text{minimize} & \sum_s \mathbf{v}_s \\ \text{subject to} & \forall s \in S, a \in A_s, \quad \mathbf{v}_s \geq \mathbf{r}_a + \gamma \sum_{s'} P_{a,s'} \mathbf{v}_{s'} \end{array} \tag{4}$$

The constraint matrix of (3) is equal to $M - \gamma P$, where $M_{s,a} = 1$ if action a can be used in state s and 0 otherwise. The dual value LP (4) is often defined as the primal, as it is perhaps more intuitive, and (3) is rarely considered. However, our analysis centers on the flux variables, and algorithms that manipulate policies can more naturally be seen as moving through the polytope (3), since vertices of the polytope represent policies:

Lemma 2.3. *The LP* (3) *is non-degenerate, and there is a bijection between vertices of the polytope and policies of the MDP.*

Proof. Policies have exactly n nonzero variables, and solving for the flux vector in (2) is identical to solving for a basis in the polytope, so policies map to bases. Write the constraints in (3) in the standard matrix form $A\mathbf{x} = \mathbf{b}$. The vector $\mathbf{b}$ is $\mathbf{1}$, and $A = M - \gamma P$. In a row s of A the only positive entries are on actions usable in state s, so if $A\mathbf{x} = \mathbf{b}$, then $\mathbf{x}$ must have a nonzero entry for every state, i.e., a choice of action for every state. Bases of the LP have n variables, so they must include only one action per state.

Finally, as shown above $\mathbf{x}^\pi_a \geq 1$ for all a in a policy/basis, so the LP is not degenerate, and bases correspond to vertices. □

By Lemma 2.3, the simplex method applied to (3) corresponds to a simple, single-switch version of policy iteration: we start with an arbitrary policy, and in each iteration we change a single action that improves the value of some state. Since the LP is not degenerate, the simplex method will find the optimal policy with no cycling. We will use Dantzig's most-negative-reduced-cost pivoting rule to choose the action switched. Since (3) is written as a maximization problem, we will refer to reduced costs as gains and always choose the highest gain action to switch/pivot. For MDPs, the gains have a simple interpretation:

Definition 2.4. *The* gain *(or* reduced cost*) of an action a for state s with respect to a policy π is denoted $\boldsymbol{r}^\pi_a$ and is the improvement in the value of s if s uses action a once and then follows π for all time. Formally, $\boldsymbol{r}^\pi_a = (\boldsymbol{r}_a + \gamma P_a^T \boldsymbol{v}^\pi) - \boldsymbol{v}^\pi_s$, or, in vector form*

$$\boldsymbol{r}^\pi = \boldsymbol{r} - (M - \gamma P)^T \boldsymbol{v}^\pi \ . \tag{5}$$

We denote the optimal policy by π^*, and the optimal flux, values, and gains by $\mathbf{x}^*$, $\mathbf{v}^*$, and $\mathbf{r}^*$. The following are basic properties of the simplex method, and we prove them for completeness.

Lemma 2.5. *Let π and π' be any policies. The gains satisfy the following properties*

- $(\boldsymbol{r}^\pi)^T \boldsymbol{x}^{\pi'} = \boldsymbol{r}^T \boldsymbol{x}^{\pi'} - \boldsymbol{r}^T \boldsymbol{x}^\pi = \boldsymbol{1}^T \boldsymbol{v}^{\pi'} - \boldsymbol{1}^T \boldsymbol{v}^\pi$,
- $\boldsymbol{r}^\pi_a = 0$ *for all $a \in \pi$, and*
- $\boldsymbol{r}^*_a \leq 0$ *for all a.*

Proof. From the definition of gains $(\mathbf{r}^\pi)^T \mathbf{x}^{\pi'} = (\mathbf{r} - (M - \gamma P)^T \mathbf{v}^\pi)^T \mathbf{x}^{\pi'} = \mathbf{r}^T \mathbf{x}^{\pi'} - (\mathbf{v}^\pi)^T (M - \gamma P) \mathbf{x}^{\pi'} = \mathbf{r}^T \mathbf{x}^{\pi'} - (\mathbf{v}^\pi)^T \mathbf{1}$, using that $(M - \gamma P)$ is the constraint matrix of (3). From the definition of value and flux vectors $\mathbf{r}^T \mathbf{x}^\pi = \mathbf{r}_\pi^T (I - \gamma P^\pi)^{-1} \mathbf{1} = (\mathbf{v}^\pi)^T \mathbf{1}$, where $\mathbf{r}_\pi$ is the reward vector restricted to indices π. Combining these two gives the first result.

For the second result, if a is in π, then $\mathbf{v}^\pi_s = \mathbf{r}_a + \gamma P_a^T \mathbf{v}^\pi$, so $\mathbf{r}^\pi_a = 0$. Finally, if $\mathbf{r}^*_a > 0$ for some a, then consider the policy π that is identical to π^* but uses a. Then $(\mathbf{r}^*)^T \mathbf{x}^\pi > 0$, and the first identity proves that π^* is not optimal. □

A key property of the simplex method on MDPs that we will employ repeatedly is that not only is the overall objective improving, but also the values of all states are monotone non-decreasing, and there exists a single policy we denote by π^* that maximizes the values of all states:

Lemma 2.6. *Let π and π' be policies appearing in an execution of the simplex method with π' being used after π. Then $\boldsymbol{v}^{\pi'} \geq \boldsymbol{v}^\pi$. Further, let π^* be the policy when simplex terminates, and π'' be any other policy. Then $\boldsymbol{v}^* \geq \boldsymbol{v}^{\pi''}$.*

Proof. Suppose π and π' are subsequent policies. The gains of all actions in π' with respect to π are equal to $\mathbf{r}_{\pi'} - (I - \gamma P^{\pi'})^T \mathbf{v}^\pi$, all of which are nonnegative. Therefore $\mathbf{0} \leq (I - \gamma P^{\pi'})^{-T}(\mathbf{r}_{\pi'} - (I - \gamma P^{\pi'})^T)\mathbf{v}^\pi = \mathbf{v}^{\pi'} - \mathbf{v}^\pi$, using that $(I - \gamma P^{\pi'})^{-T} = \sum_{i \geq 0}(\gamma (P^\pi)^T)^i \geq \mathbf{0}$. By induction, this holds if π and π' occur further apart. Performing a similar calculation using the gains $\mathbf{r}^*$, which are nonpositive, shows that $\mathbf{v}^* - \mathbf{v}^{\pi''} \geq \mathbf{0}$ for any policy π''. □

3 Uniform discount

As a warmup before delving into our analysis of deterministic MDPs, we briefly review the analysis of [Ye11] for stochastic MDPs with a fixed discount. Consider the flux vector in Definition 2.2. One unit of flux is added to each state, and every step it is discounted by a factor of γ, for a total of $n(1 + \gamma + \gamma^2 + \cdots) = n/(1 - \gamma)$ flux overall. If π is the current policy and Δ is the highest gain, then, by Lemma 2.5 the farthest π can be from π^* is if all $n/(1 - \gamma)$ units of flux in π^* are on the action with gain Δ, so $\mathbf{r}^T\mathbf{x}^* - \mathbf{r}^T\mathbf{x}^\pi \leq n\Delta/(1 - \gamma)$. If we pivot on this action, at least 1 unit of flux is placed on the new action, increasing the objective by at least Δ. Thus we have reduced the gap to π^* by a $1 - (1 - \gamma)/n$ fraction, which is substantial if $1/(1 - \gamma)$ is polynomial.

Now consider $\mathbf{r}^T\mathbf{x}^* - \mathbf{r}^T\mathbf{x}^\pi = -(\mathbf{r}^*)^T\mathbf{x}^\pi$. All the terms $-\mathbf{r}^*_a\mathbf{x}^\pi_a$ are nonnegative, and for some action a in π we have $-\mathbf{r}^*_a\mathbf{x}^\pi_a \geq -(\mathbf{r}^*)^T\mathbf{x}^\pi/n$. The term $-\mathbf{r}^*_a\mathbf{x}^\pi_a$ is at most $-\mathbf{r}^*_a n/(1 - \gamma)$, so $-\mathbf{r}^*_a \geq -(\mathbf{r}^*)^T\mathbf{x}^\pi/(n^2/(1 - \gamma))$. But for any policy π' that includes a, $-(\mathbf{r}^*)^T\mathbf{x}^{\pi'} \geq -\mathbf{r}^*_a\mathbf{x}^{\pi'}_a \geq -\mathbf{r}^*_a$, so after $\mathbf{r}^T\mathbf{x}^* - \mathbf{r}^T\mathbf{x}^\pi$ has shrunk by a factor of $n^2/(1 - \gamma)$, action a cannot appear in any future policy, and this occurs after

$$\log_{1-(1-\gamma)/n} \frac{1 - \gamma}{n^2} = O\left(\frac{n}{1 - \gamma} \log \frac{n}{1 - \gamma}\right)$$

steps. See [Ye11] for the details.

The above result hinged on the fact that the size of all nonzero flux lay within the interval $[1, n/(1 - \gamma)]$, which was assumed to be polynomial but gives a weak bound if γ is very close to 1. However, consider a policy for a deterministic MDP. It can be seen as a graph with a node for each state with a single directed edge leaving each state representing the action, so the graph consists of one or more directed cycles and directed paths leading to these cycles. Starting on a path, the MDP uses each path action once before reaching a cycle, so the flux on paths must be small. Flux on the cycles may be substantially larger, but since the MDP revisits each action after at most n steps, the flux on cycle actions varies by at most a factor of n.

Lemma 3.1. *Let π be a policy with flux vector $\boldsymbol{x}^\pi$ and a an action in π. If a is on a path in π then $1 \leq \boldsymbol{x}^\pi_a \leq n$, and if a is on a cycle then $1/(1 - \gamma) \leq \boldsymbol{x}^\pi_a \leq n/(1 - \gamma)$. The total flux on paths is at most n^2, and the total flux on cycles is at most $n/(1 - \gamma)$.*

Proof. All actions have at least 1 flux. If a is on a path, then starting from any state we can only use a once and never return, contributing flux at most 1 per state, so $\mathbf{x}^\pi_a \leq n$. Summing over all path actions, the total flux is at most n^2.

If a is on a cycle, each state on the cycle contributes a total of $1/(1 - \gamma)$ flux to the cycle. By symmetry this flux is distributed evenly among actions on the cycle, so $\mathbf{x}^\pi_a \geq 1/(1 - \gamma)$. The total flux in the MDP is $n/(1 - \gamma)$, so $\mathbf{x}^\pi_a \leq n/(1 - \gamma)$. □

The overall range of flux is large, but all values must lie within one of two polynomial layers. We will prove that simplex can essentially optimize each layer separately. If a cycle is not updated, then not much progress is made towards the optimum, but we make a substantial amount of progress in optimizing the paths for the current cycles. When the paths are optimal the algorithm is forced to update a cycle, at which point we make a substantial amount of progress towards the optimum but resets all progress on the paths.

First we analyze progress on the paths:

Lemma 3.2. *Suppose the simplex method pivots from π to π', which does not create a new cycle. Let π'' be the final policy such that cycles in π'' are a subset of those in π (i.e., the final policy before a new cycle is created). Then $\boldsymbol{r}^T(\boldsymbol{x}^{\pi''} - \boldsymbol{x}^{\pi'}) \leq (1 - 1/n^2)\boldsymbol{r}^T(\boldsymbol{x}^{\pi''} - \boldsymbol{x}^{\pi})$.*

Proof. Let $\Delta = \max_a \mathbf{r}^{\pi}_a$ be the highest gain. Consider $(\mathbf{r}^{\pi})^T\mathbf{x}^{\pi''}$. Since cycles in π'' are contained in π, $\mathbf{r}^{\pi}_a = 0$ for any action a on a cycle in π'', and by Lemma 3.1, π'' has at most n^2 units of flux on paths, so $(\mathbf{r}^{\pi})^T\mathbf{x}^{\pi''} = \mathbf{r}^T(\mathbf{x}^{\pi''} - \mathbf{x}^{\pi}) \leq n^2\Delta$.

Policy π' has at least 1 unit of flux on the action with gain Δ, so

$$\mathbf{r}^T(\mathbf{x}^{\pi''} - \mathbf{x}^{\pi'}) \leq \mathbf{r}^T(\mathbf{x}^{\pi''} - \mathbf{x}^{\pi}) - \Delta \leq \left(1 - \frac{1}{n^2}\right)\mathbf{r}^T(\mathbf{x}^{\pi''} - \mathbf{x}^{\pi}) \,. \qquad \square$$

Due to the polynomial contraction in the lemma above, not too many iterations can pass before a new cycle is formed.

Lemma 3.3. *Let π be a policy. After $O(n^2 \log n)$ iterations starting from π, either the algorithm finishes, a new cycle is created, a cycle is broken, or some action in π never appears in a policy again until a new cycle is created.*

Proof. Let π be the policy in some iteration, π' the last policy before a new cycle is created, and π'' an arbitrary policy occurring between π and π' in the algorithm. Policy π differs from π' in actions on paths and possibly in cycles that exist in π but have been broken in π'. By Lemma 2.5 $-(\mathbf{r}^{\pi'})^T\mathbf{x}^{\pi} = \mathbf{r}^T(\mathbf{x}^{\pi'} - \mathbf{x}^{\pi}) = \mathbf{1}^T(\mathbf{v}^{\pi'} - \mathbf{v}^{\pi})$.

We divide the analysis into two cases. First suppose that there exists an action a used in state s on a path such that $-\mathbf{r}^{\pi'}_a\mathbf{x}^{\pi}_a \geq -(\mathbf{r}^{\pi'})^T\mathbf{x}^{\pi}/n$ (note $(\mathbf{r}^{\pi'})^T\mathbf{x}^{\pi} \leq 0$). Since a is on a path $\mathbf{x}^{\pi}_a \leq n$, which implies $-\mathbf{r}^{\pi'}_a n^2 \geq -(\mathbf{r}^{\pi'})^T\mathbf{x}^{\pi}$. Now if policy π'' uses action a, then

$$\begin{aligned} -(\mathbf{r}^{\pi'})^T\mathbf{x}^{\pi''} = \mathbf{1}^T(\mathbf{v}^{\pi'} - \mathbf{v}^{\pi''}) \geq \mathbf{v}^{\pi'}_s - \mathbf{v}^{\pi''}_s &= \mathbf{v}^{\pi'}_s - (\mathbf{r}_a + \gamma P_a \mathbf{v}^{\pi''}) \\ &\geq \mathbf{v}^{\pi'}_s - (\mathbf{r}_a + \gamma P_a \mathbf{v}^{\pi'}) = -\mathbf{r}^{\pi'}_a \geq -\frac{-(\mathbf{r}^{\pi'})^{\pi}\mathbf{x}^{\pi}}{n^2} \,, \end{aligned}$$

using that the values of all states are monotone increasing.

In the second case there is no action a on a path in π satisfying $-\mathbf{r}^{\pi'}_a\mathbf{x}^{\pi}_a \geq -(\mathbf{r}^{\pi'})^T\mathbf{x}^{\pi}/n$. The remaining portion of $-(\mathbf{r}^{\pi'})^T\mathbf{x}^{\pi}$ is due to cycles, so there must be some cycle C consisting of actions $\{a_1, \ldots, a_k\}$ used in states $\{s_1, \ldots, s_k\}$ such that $\sum_{a \in C} -\mathbf{r}^{\pi'}_a\mathbf{x}^{\pi}_a \geq -(\mathbf{r}^{\pi'})^T\mathbf{x}^{\pi}/n$.

All flux in C first enters C either from a path ending at C or from the initial unit of flux placed on some state s in C. If $y_s \geq 1$ units of flux first enter C at state s in policy π, then that flux earns $y_s(\mathbf{v}^{\pi'}_s - \mathbf{v}^{\pi}_s)$ reward with respect to the rewards $-\mathbf{r}^{\pi'}$, so $\sum_{a \in C} -\mathbf{r}^{\pi'}_a\mathbf{x}^{\pi}_a = \sum_{s \in C} y_s(\mathbf{v}^{\pi'}_s - \mathbf{v}^{\pi}_s)$. Moreover, each term $\mathbf{v}^{\pi'}_s - \mathbf{v}^{\pi}_s$ is nonnegative, since the values of all states are nondecreasing. Now note that $\sum_{s \in C}(\mathbf{v}^{\pi'}_s - \mathbf{v}^{\pi}_s) = \sum_{a \in C} -\mathbf{r}^{\pi'}_a/(1-\gamma)$, and at most n units of flux enter each state from outside. Therefore $-n\sum_{a \in C}\mathbf{r}^{\pi'}_a/(1-\gamma) \geq \sum_{a \in C} -\mathbf{r}^{\pi'}_a\mathbf{x}^{\pi}_a$, implying $-n^2\sum_{a \in C}\mathbf{r}^{\pi'}_a/(1-\gamma) \geq -(\mathbf{r}^{\pi'})^T\mathbf{x}^{\pi}$.

As long as cycle C is intact, each $a \in C$ has $1/(1-\gamma)$ flux from states in C (Lemma 3.1), so if C is in policy π'' then

$$-(\mathbf{r}^{\pi'})^T\mathbf{x}^{\pi''} = \mathbf{1}^T(\mathbf{v}^{\pi'} - \mathbf{v}^{\pi''}) \geq \sum_{s \in C} \mathbf{v}_s^{\pi'} - \mathbf{v}_s^{\pi''} = -\frac{\sum_{a \in C} \mathbf{r}_a^{\pi''}}{1-\gamma} \geq -\frac{-(\mathbf{r}^{\pi'})^T\mathbf{x}^{\pi}}{n^2} \ . \tag{6}$$

Now if $\log_{n^2/(n^2-1)} n^2$ iterations occur between π and π'', Lemma 3.2 implies

$$-(\mathbf{r}^{\pi'})^T\mathbf{x}^{\pi''} < -\left(1 - \frac{1}{n^2}\right)^{\log_{n^2/(n^2-1)} n^2} (\mathbf{r}^{\pi'})^T\mathbf{x}^{\pi} \leq -\frac{-(\mathbf{r}^{\pi'})^T\mathbf{x}^{\pi}}{n^2} \ .$$

In the first case action a cannot appear in π'', and in the second case cycle C must be broken in π''. This takes $\log_{n^2/(n^2-1)} n^2 = O(n^2 \log n)$ iterations if no new cycles interrupt the process. □

Lemma 3.4. *Either the algorithm finishes or a new cycle is created after $O(n^2 m \log n)$ iterations.*

Proof. Let π_0 be a policy after a new cycle is created, and consider the policies $\pi_1, \pi_2, \ldots$ each separated by $O(n^2 \log n)$ iterations. If no new cycle is created, then by Lemma 3.3 each of these policies π_i has either broken another cycle in π_0 or contains an action that cannot appear in π_j for all $j > i$. There are at most n cycles in π_0 and at most m actions that can be eliminated, so after $(m+n)O(n^2 \log n) = O(n^2 m \log n)$ iteration, the algorithm must terminate or create a new cycle. □

When a new cycle is formed, the algorithm makes a substantial amount of progress towards the optimum but also resets the path optimality above.

Lemma 3.5. *Let π and π' be subsequent policies such that π' creates a new cycle. Then $\boldsymbol{r}^T(\boldsymbol{x}^* - \boldsymbol{x}^{\pi'}) \leq (1 - 1/n)\boldsymbol{r}^T(\boldsymbol{x}^* - \boldsymbol{x}^{\pi})$.*

Proof. Let $\Delta = \max_{a'} \mathbf{r}_{a'}^{\pi}$ and $a = \operatorname{argmax}_{a'} \mathbf{r}_a^{\pi}$. There is a total of $n/(1-\gamma)$ flux in the MDP, so $\mathbf{r}^T\mathbf{x}^* - \mathbf{r}^T\mathbf{x}^{\pi} = (\mathbf{r}^{\pi})^T\mathbf{x}^* \leq \Delta n/(1-\gamma)$. By Lemma 3.1, pivoting on a and creating a cycle will result in at least $1/(1-\gamma)$ flux through a. Therefore $\mathbf{r}^T\mathbf{x}^{\pi'} \geq \mathbf{r}^T\mathbf{x}^{\pi} + \Delta/(1-\gamma)$, so

$$\mathbf{r}^T(\mathbf{x}^* - \mathbf{x}^{\pi'}) \leq \mathbf{r}^T(\mathbf{x}^* - \mathbf{x}^{\pi}) - \frac{\Delta}{1-\gamma} \leq \left(1 - \frac{1}{n}\right)\mathbf{r}^T(\mathbf{x}^* - \mathbf{x}^{\pi}) \ .$$ □

Lemma 3.6. *Let π be a policy. Starting from π, after $O(n \log n)$ iterations in which a new cycle is created, some action in π is either eliminated from cycles for the remainder of the algorithm or entirely eliminated from policies for the remainder of the algorithm.*

Proof. Consider a policy π with respect to the optimal gains $\mathbf{r}^*$. There is an action a such that $-\mathbf{r}_a^*\mathbf{x}_a^{\pi} \geq -(\mathbf{r}^*)^T\mathbf{x}^{\pi}/n$. If a is on a path in π, then $1 \leq \mathbf{x}_a^{\pi} \leq n$, so $-\mathbf{r}_a^* \geq -(\mathbf{r}^*)^T\mathbf{x}^{\pi}/n^2$, and if a is on a cycle, then $1/(1-\gamma) \leq \mathbf{x}_a^{\pi} \leq n/(1-\gamma)$, so $-\mathbf{r}_a^*/(1-\gamma) \geq -(\mathbf{r}^*)^T\mathbf{x}^{\pi}/n^2$.

Since $\mathbf{r}^*$ are the gains for the optimal policy, $\mathbf{r}_{a'}^* \leq 0$ for all a'. Therefore if π' is any policy containing a, then $-\mathbf{r}_a^* \leq -\mathbf{r}_a^*\mathbf{x}_a^{\pi'} \leq -(\mathbf{r}^*)^T\mathbf{x}^{\pi'}$, and if π' is any policy containing a on a cycle, then $-\mathbf{r}_a^*/(1-\gamma) \leq -\mathbf{r}_a^*\mathbf{x}_a^{\pi'} \leq -(\mathbf{r}^*)^T\mathbf{x}^{\pi'}$. Now by Lemma 3.5, if there are more than $\log_{n/(n-1)} n^2 = O(n \log n)$ new cycles created between policies π and π' then

$$-(\mathbf{r}^*)^T\mathbf{x}^{\pi'} < -\left(1 - \frac{1}{n}\right)^{\log_{n/(n-1)} n^2} (\mathbf{r}^*)^T\mathbf{x}^{\pi} = -\frac{(\mathbf{r}^*)^T\mathbf{x}^{\pi}}{n^2} \ .$$

Therefore if π contained a on a path, then a cannot appear in any policy after π' for the remainder of the algorithm, and if π contained a on a cycle, then a cannot appear in a cycle after π' (but may appear in a path) for the remainder of the algorithm. □

Theorem 3.7. *The simplex method converges in at most $O(n^3 m^2 \log^2 n)$ iterations on deterministic MDPs with uniform discount using the highest gain pivoting rule.*

Proof. Consider the policies $\pi_0, \pi_1, \pi_2, \ldots$ where $O(n \log n)$ new cycles have been created between π_i and π_{i+1}. By Lemma 3.6, each π_i contains an action that is either eliminated entirely in π_j for $j > i$ or eliminated from cycles. Each action can be eliminated from cycles and paths, so after $2m$ such rounds of $O(n \log n)$ new cycles the algorithm has converged. By Lemma 3.4 cycles are created every $O(n^2 m \log n)$ iterations, for a total of $O(n^3 m^2 \log^2 n)$ iterations. □

4 Varying Discounts

In this section we allow each action a to have a distinct discount γ_a. This significantly complicates the proof of convergence since the total flux is no longer fixed. When updating a cycle we can no longer bound the distance to the optimum based solely on the maximum gain, since the optimal policy may employ actions with smaller gain to the current policy but substantially more flux.

We are able to exhibit a set of layers in which the flux on cycles must lie based on the discount of the actions, and we will show that when a cycle is created in a particular layer we make progress towards the optimum value for the updated state *assuming that it lies within that layer.* These layers will define a set of bounds whose values we must surpass, which serve as milestones or checkpoints to the optimum. When we update a cycle we cannot claim that the overall objective increases substantially but only that the values of individual states make progress towards one of these milestone values. When the values of all states have surpassed each of these intermediate milestones the algorithm will terminate.

We first define some notation. Recall that to calculate flux we place one unit of "mass" in each state and then run the Markov chain, so all flux traces back to some state, but $\mathbf{x}^\pi$ aggregates all of it together. Because we will be concerned with analyzing the values of individual states in this section, it will be useful to separate out the flux originating in a particular state s. Consider the following alternate LP:

$$\begin{array}{llrl}
\text{maximize} & \mathbf{r}^T\mathbf{x} & & \\
\text{subject to} & & \sum_{a\in A_s} \mathbf{x}_a & = 1 + \sum_a \gamma_a P_{a,s}\mathbf{x}_a \\
 & \forall s' \neq s & \sum_{a\in A_{s'}} \mathbf{x}_a & = \sum_a \gamma_a P_{a,s'}\mathbf{x}_a \\
 & & \mathbf{x} & \geq 0
\end{array} \tag{7}$$

The LP (7) is identical to (3), except that initial flux is only added to state s rather than all states, and the dual of (7) matches (4) if the objective in (4) is changed to minimize only $\mathbf{v}_s$. Feasible solutions in (7) measure only flux originating in s and contributing to $\mathbf{v}_s$. For a state s and policy π we use the notation $\mathbf{x}^{\pi,s}$ to denote the corresponding vertex in (7). Note that $\mathbf{x}^\pi = \sum_s \mathbf{x}^{\pi,s}$.

The following lemma is analogous to Lemma 2.5 and has an identical proof:

Lemma 4.1. *For a state s and for policies π and π', $(\boldsymbol{r}^\pi)^T \boldsymbol{x}^{\pi',s} = \boldsymbol{r}^T \boldsymbol{x}^{\pi',s} - \boldsymbol{r}^T \boldsymbol{x}^{\pi,s} = \boldsymbol{v}_s^{\pi'} - \boldsymbol{v}_s^{\pi}$.*

We now define the intervals in which the flux must lie. As in Section 3 flux on paths is in $[1, n]$. Let C be a cycle in some policy, and $\gamma_C = \prod_{a \in C} \gamma_a$ be total discount of C. We will prove that the smallest discount in C determines the rough order of magnitude of the flux through C.

Definition 4.2. *Let C be a cycle and a an action in C, then the discount of a* dominates *the discount of C if $\gamma_a \le \gamma_{a'}$ for all $a' \in C$.*

Lemma 4.3. *Let π be a policy containing the cycle C with discount dominated by γ_a and total discount γ_C. Let s be a state on C, a' the action used in s and a'' an arbitrary action in C, then*

- $\boldsymbol{x}_{a'}^{\pi,s} = 1/(1-\gamma_C)$,
- $\gamma_C/(1-\gamma_C) \le \boldsymbol{x}_{a''}^{\pi,s} \le 1/(1-\gamma_C)$, *and*
- $1/(n(1-\gamma_a)) \le 1/(1-\gamma_C) \le 1/(1-\gamma_a)$.

Proof. For the first equality, all flux originates at s, so the flux through a' (used in state s) either just originated in s or came around the cycle from s, implying $\mathbf{x}_{a'}^{\pi,s} = 1 + \gamma_C \mathbf{x}_{a'}^{\pi,s}$. An analogous equation holds for all other actions a'' on C, but now the initial flow from s may have been discounted by at most γ_C before reaching a'', giving $\gamma_C/(1-\gamma_C) \le \mathbf{x}_{a''}^{\pi,s} \le 1/(1-\gamma_C)$.

The upper bound in the final inequality, $1/(1-\gamma_C) \le 1/(1-\gamma_a)$ holds since $a \in C$ (γ_a dominates the discount of C). For the lower bound, let $\ell = 1 - \gamma_a$. Then $\gamma_C \ge \gamma_a^n = (1-\ell)^n \ge 1 - n\ell = 1 - n(1-\gamma_a)$, implying $1/(1-\gamma_C) \ge 1/(n(1-\gamma_a))$. □

Flux on paths still falls in $[1, n]$, so the algorithm behaves the same on paths as it did in the uniform case:

Lemma 4.4. *Either the algorithm finishes or a new cycle is created after $O(n^2 m \log n)$ iterations.*

Proof. This is identical to the proof of Lemma 3.4, which depends on Lemmas 3.2 and 3.3. Lemma 3.2 holds for nonuniform discounts, and Lemma 3.3 holds after adjusting Equation (6) as follows

$$-(\mathbf{r}^{\pi'})^T \mathbf{x}^{\pi''} \ge \sum_{s \in C} \mathbf{v}_s^{\pi'} - \mathbf{v}_s^{\pi''} \ge -\frac{\sum_{a \in C} \mathbf{r}_a^{\pi''}}{1-\gamma_C} \ge -\frac{-(\mathbf{r}^{\pi'})^T \mathbf{x}^{\pi}}{n^2} \; ,$$

using that $\sum_{a \in C} \mathbf{r}_a^{\pi''} n/(1-\gamma_C) \ge -(\mathbf{r}^{\pi'})^T \mathbf{x}^{\pi}/n$ and Lemma 4.3. □

Now suppose the simplex method updates the action for state s in policy π and creates a cycle dominated by γ_a. Again, $\mathbf{v}_s$ may not improve much, since there may be a cycle with discount much larger than γ_a. However, in any policy π' where s is on a cycle dominated by γ_a and s uses some action a', $1/(n(1-\gamma_a)) \le \mathbf{x}_{a'}^{\pi',s} \le 1/(1-\gamma_a)$, which allows us to argue $\mathbf{v}_s$ has made progress towards the highest value achievable when it is on a cycle dominated by γ_a, and after enough such progress has made, $\mathbf{v}_s$ will beat this value and never again appear on any cycle dominated by γ_a. The optimal values achievable for each state on a cycle dominated by each γ_a serve as the above-mentioned milestones. Since all cycles are dominated by some γ_a, there are m milestones per state.

Lemma 4.5. *Suppose the simplex method moves from π to π' by updating the action for state s, creating a new cycle C with discount dominated by γ_a for some a in π'. Let π'' be the final policy used by the simplex method in which s is in a cycle dominated by γ_a. Then $\boldsymbol{v}_s^{\pi''} - \boldsymbol{v}_s^{\pi'} \le (1-1/n^2)(\boldsymbol{v}_s^{\pi''} - \boldsymbol{v}_s^{\pi})$.*

Proof. Let $\Delta = \max_{a'} \mathbf{r}^{\pi}_{a'}$ be the value of the highest gain with respect to π. Any cycle contains at most n actions, each of which has gain at most Δ in $\mathbf{r}^{\pi}$, so if s is on a cycle dominated by γ_a in π'' then by Lemma 4.3 and Lemma 4.1, $\mathbf{v}^{\pi''}_s - \mathbf{v}^{\pi}_s \le n\Delta/(1-\gamma_a)$, and since π' creates a cycle dominated by γ_a, by the same lemmas $\mathbf{v}^{\pi'}_s \ge \mathbf{v}^{\pi}_s + \Delta/(n(1-\gamma_a))$. Combining the two,

$$\mathbf{v}^{\pi''}_s - \mathbf{v}^{\pi'}_s = (\mathbf{v}^{\pi''}_s - \mathbf{v}^{\pi}_s) - (\mathbf{v}^{\pi'}_s - \mathbf{v}^{\pi}_s) \le (\mathbf{v}^{\pi''}_s - \mathbf{v}^{\pi}_s) - \frac{\Delta}{n(1-\gamma_a)} \le \left(1 - \frac{1}{n^2}\right)(\mathbf{v}^{\pi''}_s - \mathbf{v}^{\pi}_s)\ . \qquad \square$$

The following lemma is the crux of our analysis and allows us to eliminate actions when we get close to a milestone value. This occurs because the positive gains must shrink or else the algorithm would surpass the milestone, and as the positive gains shrink they can no longer balance larger negative gains, forcing such actions out of the cycle.

Lemma 4.6. *Suppose policy π contains a cycle C with discount dominated by γ_a and s is a state in C. There is some action a' in C (depending on s) such that after $O(n^2 \log n)$ iterations that change the action for s and create a cycle with discount dominated by γ_a, action a' will never again appear in a cycle dominated by γ_a.*

Proof. Let π be a policy containing a cycle C with discount dominated by γ_a and s a state in C. Let π' be another policy where s is on a cycle dominated by γ_a after at least $1 + \log_{n^2/(n^2-1)} n^5 = O(n^2 \log n)$ iterations that create such a cycle by changing the action for s and π'' the final policy used by the algorithm in which s is on a cycle dominated by γ_a.

Consider the policy $\hat{\pi}$ in the iteration immediately preceding π'. By Lemma 4.5, and the choice of π',

$$\mathbf{v}^{\pi''}_s - \mathbf{v}^{\hat{\pi}}_s \le \left(1 - \frac{1}{n^2}\right)^{\log_{n^2/(n^2-1)} n^5} (\mathbf{v}^{\pi''}_s - \mathbf{v}^{\pi}_s) = \frac{1}{n^5}(\mathbf{v}^{\pi''}_s - \mathbf{v}^{\pi}_s)\ ,$$

or equivalently $\mathbf{v}^{\pi}_s - \mathbf{v}^{\pi''}_s \le -n^5(\mathbf{v}^{\pi''}_s - \mathbf{v}^{\hat{\pi}}_s)$, implying

$$\mathbf{v}^{\pi}_s - \mathbf{v}^{\hat{\pi}}_s = (\mathbf{v}^{\pi}_s - \mathbf{v}^{\pi''}_s) + (\mathbf{v}^{\pi''}_s - \mathbf{v}^{\hat{\pi}}_s) \le (-n^5 + 1)(\mathbf{v}^{\pi''}_s - \mathbf{v}^{\hat{\pi}}_s)\ . \tag{8}$$

Since the gap $\mathbf{v}^{\pi}_s - \mathbf{v}^{\hat{\pi}}_s$ is large and negative, there must be highly negative gains in $\mathbf{r}^{\hat{\pi}}$. By Lemma 4.1 $\mathbf{v}^{\pi}_s - \mathbf{v}^{\hat{\pi}}_s = (\mathbf{r}^{\hat{\pi}})^T \mathbf{x}^{\pi,s}$. Let $\mathbf{r}^{\hat{\pi}}_{a'} = \min_{a \in C} \mathbf{r}^{\hat{\pi}}_a$ and s' be the state using a'. By Lemma 4.3, $\mathbf{x}^{\pi,s} \le 1/(1-\gamma_a)$, and C has at most n states, so applying Equation (8)

$$\frac{\mathbf{r}^{\hat{\pi}}_{a'}}{1-\gamma_a} \le \frac{1}{n}(\mathbf{v}^{\pi}_s - \mathbf{v}^{\hat{\pi}}_s) \le \left(-n^4 + \frac{1}{n}\right)(\mathbf{v}^{\pi''}_s - \mathbf{v}^{\hat{\pi}}_s)\ . \tag{9}$$

The positive entries in $\mathbf{r}^{\hat{\pi}}$ must all be small, since there is only a small increase in the value of s. Let $\Delta = \max \mathbf{r}^{\hat{\pi}}$. The algorithm pivots on the highest gain, and by assumption it updates the action for s and creates a cycle dominated by γ_a. By Lemma 4.3, the new action is used at least $1/(n(1-\gamma_a))$ times by flux from s, since it is the first action in the cycle, so

$$\frac{\Delta}{n(1-\gamma_a)} \le \mathbf{v}^{\pi'}_s - \mathbf{v}^{\hat{\pi}}_s \le \mathbf{v}^{\pi''}_s - \mathbf{v}^{\hat{\pi}}_s\ . \tag{10}$$

We prove that the highly negative $\mathbf{r}^{\hat{\pi}}_{a'}$ cannot coexist with only small positive gains bounded by Δ. Consider any policy in which s' is on a cycle C' containing a' (but not necessarily containing s) with total gain $\gamma_{C'}$ dominated by γ_a. By Lemma 4.3, there is at least $1/(1-\gamma_{C'}) \ge 1/(n(1-\gamma_a))$

flux from s going through a', and in the rest of the cycle there are at most $n-1$ other actions with at most $1/(1-\gamma_{C'}) \le 1/(1-\gamma_a)$ flux. The highest gain with respect to $\hat{\pi}$ is Δ, so the value of $\mathbf{v}_{s'}$ relative to $\mathbf{r}^{\hat{\pi}}$ is at most

$$\begin{aligned}\frac{\mathbf{r}^{\hat{\pi}}_{a'}}{n(1-\gamma_a)} + \frac{n\Delta}{1-\gamma_a} &\le \left(-n^3 + \frac{1}{n^2}\right)(\mathbf{v}^{\pi''}_s - \mathbf{v}^{\hat{\pi}}_s) + n^2(\mathbf{v}^{\pi''}_s - \mathbf{v}^{\hat{\pi}}_s) \\ &= \left(-n^3 + \frac{1}{n^2} + n^2\right)(\mathbf{v}^{\pi''}_s - \mathbf{v}^{\hat{\pi}}_s) < 0\end{aligned}$$

using Equations (9) and (10). But $\mathbf{v}^{\hat{\pi}}_{s'} = 0$ relative to $\mathbf{r}^{\hat{\pi}}$, and it only increases in future iterations, so a' cannot appear again in a cycle dominated by γ_a. □

Lemma 4.7. *For any action a, there are at most $O(n^3 m \log n)$ iterations that create a cycle with discount dominated by γ_a.*

Proof. After $O(n^3 \log n)$ iterations that create a cycle dominated by γ_a, some state must have been updated in $O(n^2 \log n)$ of those iterations, so by Lemma 4.6 some action will never appear again in a cycle dominated by γ_a. After m repetitions of this process all actions have been eliminated. □

Theorem 4.8. *Simplex terminates in at most $O(n^5 m^3 \log^2 n)$ iterations on deterministic MDPs with nonuniform discounts using the highest gain pivoting rule.*

Proof. There are $O(m)$ possible discounts γ_a that can dominate a cycle, and by Lemma 4.7 there are at most $O(n^3 m \log n)$ iterations creating a cycle dominated by any particular γ_a, for a total of $O(n^3 m^2 \log n)$ iterations that create a cycle. By Lemma 4.4 a new cycle is created every $O(n^2 m \log n)$ iterations, for a total of $O(n^5 m^3 \log^2 n)$ iterations overall. □

5 Open problems

A difficult but natural next step would be to try to extend these techniques to handle policy iteration on deterministic MDPs. The main problem encountered is that the multiple simultaneous pivots used in policy iteration can interfere with each other in such a way that the algorithm effectively pivots on the *smallest* improving switch rather than the largest. See [HZ10] for such an example. Another challenging open question is to design a strongly polynomial algorithm for general MDPs. Finally, we believe the technique of dividing variable values into polynomial sized layers may be helpful for entirely different problems.

Acknowledgments.

The authors would like to thank Kazuhisa Makino for pointing out an error in Lemma 3.3.

References

[Bel57] Richard E. Bellman. *Dynamic Programming.* Princeton University Press, 1957. 2

[Ber96] Dimitri P. Bertsekas. *Dynamic programming and optimal control.* Athena Scientific, 1996. 1

[Fea10] John Fearnley. Exponential lower bounds for policy iteration. In *Automata, Languages and Programming*, volume 6199 of *Lecture Notes in Computer Science*, pages 551–562. Springer Berlin / Heidelberg, 2010. `arXiv:1003.3418v1`, `doi:10.1007/978-3-642-14162-1_46`. 2

[FHZ11] Oliver Friedmann, Thomas Dueholm Hansen, and Uri Zwick. Subexponential lower bounds for randomized pivoting rules for the simplex algorithm. In *Proc. 43rd Symposium on Theory of Computing*, STOC '11, pages 283–292. ACM, 2011. `doi:10.1145/1993636.1993675`. 2

[Fri09] Oliver Friedmann. An exponential lower bound for the parity game strategy improvement algorithm as we know it. In *Proc. 24th Logic In Computer Science*, LICS '09, pages 145 –156, 2009. `arXiv:0901.2731v1`, `doi:10.1109/LICS.2009.27`. 2

[Fri11] Oliver Friedmann. A subexponential lower bound for zadeh's pivoting rule for solving linear programs and games. In *Integer Programming and Combinatoral Optimization*, volume 6655 of *Lecture Notes in Computer Science*, pages 192–206. Springer Berlin / Heidelberg, 2011. `doi:10.1007/978-3-642-20807-2_16`. 2

[HMZ11] Thomas Dueholm Hansen, Peter Bro Miltersen, and Uri Zwick. Strategy iteration is strongly polynomial for 2-player turn-based stochastic games with a constant discount factor. In *ICS*, pages 253–263, 2011. `arXiv:1008.0530v1`. 2

[HN94] Dorit S. Hochbaum and Joseph (Seffi) Naor. Simple and fast algorithms for linear and integer programs with two variables per inequality. *SIAM Journal on Computing*, 23:1179, 1994. `doi:10.1137/S0097539793251876`. 2

[How60] Ronald Howard. Dynamic programming and markov decision processes. *MIT, Cambridge*, 1960. 1

[HZ10] Thomas Hansen and Uri Zwick. Lower bounds for howard's algorithm for finding minimum mean-cost cycles. In Otfried Cheong, Kyung-Yong Chwa, and Kunsoo Park, editors, *Algorithms and Computation*, volume 6506 of *Lecture Notes in Computer Science*, pages 415–426. Springer Berlin / Heidelberg, 2010. `doi:10.1007/978-3-642-17517-6_37`. 2, 12

[LDK95] Michael L. Littman, Thomas L. Dean, and Leslie Pack Kaelbling. On the complexity of solving markov decision problems. In *Proc. 11th Uncertainty in Artificial Intelligence*, UAI'95, pages 394–402, 1995. Available from: `http://dl.acm.org/citation.cfm?id=2074203`. 2

[Mad02] Omid Madani. On policy iteration as a newton's method and polynomial policy iteration algorithms. In *Proc. 18th National Conference on Artificial intelligence*, pages 273–278, 2002. Available from: `http://www.aaai.org/Papers/AAAI/2002/AAAI02-042.pdf`. 2

[MC94] Mary Melekopoglou and Anne Condon. On the complexity of the policy improvement algorithm for markov decision processes. *ORSA Journal on Computing*, 6(2):188–192, 1994. `doi:10.1287/ijoc.6.2.188`. 2

[MS99] Yishay Mansour and Satinder Singh. On the complexity of policy iteration. In *Proc. 15th Uncertainty in Artificial Intelligence*, UAI'99, pages 401–408, 1999. Available from: `http://dl.acm.org/citation.cfm?id=2073842`. 2

[MTZ10] Omid Madani, Mikkel Thorup, and Uri Zwick. Discounted deterministic markov decision processes and discounted all-pairs shortest paths. *ACM Transactions on Algorithms (TALG)*, 6(2):33:1–33:25, 2010. `doi:10.1145/1721837.1721849`. 2

[PT87] Christos Papadimitriou and John N. Tsitsiklis. The complexity of markov decision processes. *Mathematics of Operations Research*, 12(3):441–450, August 1987. `doi:10.1287/moor.12.3.441`. 2

[Put94] Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* Wiley, New York, NY, USA, 1994. 1

[Ye05] Yinyu Ye. A new complexity result on solving the markov decision problem. *Mathematics of Operations Research*, 30(3):733–749, August 2005. `doi:10.1287/moor.1050.0149`. 2

[Ye11] Yinyu Ye. The simplex and policy-iteration methods are strongly polynomial for the markov decision problem with a fixed discount rate. *Mathematics of Operations Research*, 36(4):593–603, November 2011. `doi:10.1287/moor.1110.0516`. 1, 2, 6