Operations Research Letters 48 (2020) 543–548

Contents lists available at ScienceDirect

Operations Research Letters

journal homepage: www.elsevier.com/locate/orl

Complexity bounds for approximately solving discounted MDPs by value iterations

Eugene A. Feinberg [a,*], Gaojin He [b]

[a] Department of Applied Mathematics and Statistics, Stony Brook University, United States of America
[b] Department of Mathematics, University of California San Diego, United States of America

ARTICLE INFO

Article history:
Received 27 January 2020
Received in revised form 1 July 2020
Accepted 1 July 2020
Available online 3 July 2020

Keywords:
Markov decision process
Discounting
Algorithm
Complexity
Optimal policy

ABSTRACT

For an infinite-horizon discounted Markov decision process with a finite number of states and actions, this note provides upper bounds on the number of operations required to compute an approximately optimal policy by value iterations in terms of the discount factor, spread of the reward function, and desired closeness to optimality. One of the provided upper bounds on the number of iterations has the property that it is a non-decreasing function of the value of the discount factor.

1. Introduction

Value and policy iteration algorithms are the major tools for solving infinite-horizon discounted Markov decision processes (MDPs). Policy iteration algorithms also can be viewed as implementations of specific versions of the simplex method applied to linear programming problems corresponding to discounted MDPs [6,17]. Ye [17] proved that for a given discount factor the policy iteration algorithm is strongly polynomial as a function of the total number of state–action pairs. Kitahara and Mizuno [7] extended Ye's [17] results by providing sufficient conditions for strong polynomiality of a simplex method for linear programming, and Scherrer [13] improved Ye's [17] bound for MDPs. For deterministic MDPs Post and Ye [10] proved that, for the version of policy iterations improving the policy at one state at each iteration, there is a polynomial upper bound on the number of operations, which does not depend on the value of the discount factor. Feinberg and Huang [4] showed that value iterations are not strongly polynomial for discounted MDPs. Earlier Tseng [16] proved weak polynomiality of value iterations.

In this note we show that the value iteration algorithm computes ϵ-optimal policies in strongly polynomial time for a given ϵ, discount factor, and spread of the reward functions. This is an important observation because value iterations are broadly used in applications, including reinforcement learning [1,2,15], for computing nearly optimal policies.

Let us consider an MDP with a finite state space $\mathbb{X} = \{1, 2, \dots, m\}$ and with a finite nonempty set of actions $A(x)$ available at each state $x \in \mathbb{X}$. Each action set $A(x)$ consists of k_x actions. Thus, the total number of actions is $k = \sum_{x=1}^{m} k_x$, and this number can be interpreted as the total number of state–action pairs. Let $\alpha \in [0, 1)$ be the discount factor. According to [13], Howard's version of the policy iteration algorithm finds an optimal policy within $N_I^{PI}(\alpha)$ iterations, and each iteration requires at most N_O^{PI} operations with

$$N_I^{PI}(\alpha) := (k-m)\left\lceil \frac{1}{1-\alpha}\log\frac{1}{1-\alpha}\right\rceil = O\left(\frac{k}{1-\alpha}\log\frac{1}{1-\alpha}\right),$$
$$N_O^{PI} := O\left(m^3 + mk\right). \tag{1.1}$$

This paper shows that for each $\epsilon > 0$, the value iteration algorithm finds an ϵ-optimal policy within $N_I^{VI}(\epsilon)$ iterations, and each iteration requires at most N_O^{VI} operations, where

$$N_I^{VI(\epsilon)}(\alpha) := \max\left\{\left\lceil \frac{\log\frac{(1-\alpha)\epsilon}{[R+(1+\alpha)V]}}{\log\alpha}\right\rceil, 1\right\}, \quad N_O^{VI} := O(mk), \tag{1.2}$$

where R and V are the constants defined by a one-step reward function r and a function of terminal rewards v^0. In addition, $N_I^{VI(\epsilon)}(\alpha)$ is non-decreasing in $\alpha \in [0, 1)$.

To define the values of R and V, let us denote by $\mathrm{sp}(u)$ the span seminorm of $u \in \mathbb{R}^m$, where $\mathbb{R}^m$ is an m-dimensional Euclidean space,

$$\mathrm{sp}(u) := \max_{x\in\mathbb{X}} u(x) - \min_{x\in\mathbb{X}} u(x).$$

The properties of this seminorm can be found in [11, p. 196].

* Correspondence to: Department of Applied Mathematics and Statistics, Stony Brook University, Stony Brook, NY 11794-3600, United States of America.
E-mail address: eugene.feinberg@stonybrook.edu (E.A. Feinberg).

https://doi.org/10.1016/j.orl.2020.07.001
0167-6377/

Let $r(x, a)$ be the reward collected if the system is at the state $x \in \mathbb{X}$ and the action $a \in A(x)$ is chosen. Let $v^0(x)$ be the reward collected at the final state $x \in \mathbb{X}$. Then

$$V := \mathrm{sp}(v^0). \tag{1.3}$$

We denote by

$$v^1(x) := \max_{a\in A(x)} r(x, a), \quad x \in \mathbb{X}, \tag{1.4}$$

the maximal one-step reward that can be collected at the state x. Then

$$R := \mathrm{sp}(v^1) = \max_{x\in\mathbb{X}} \max_{a\in A(x)} r(x, a) - \min_{x\in\mathbb{X}} \max_{a\in A(x)} r(x, a). \tag{1.5}$$

Observe that

$$R \le \mathrm{sp}(r) := \max_{x\in\mathbb{X}} \max_{a\in A(x)} r(x, a) - \min_{x\in\mathbb{X}} \min_{a\in A(x)} r(x, a).$$

Of course, the total number of operations to find an ϵ-optimal policy is bounded above by $N_I^{VI(\epsilon)}(\alpha)\cdot N_O^{VI}$ for the value iteration algorithm. The total number of operations to find an optimal policy is bounded above by $N_I^{PI}(\alpha)\cdot N_O^{PI}$ for the policy iteration algorithm. Each iteration in the policy iteration algorithm requires solving a system of m linear equations. This can be done by Gaussian elimination within $O(m^3)$ operations. This is the reason the formula for N_O^{PI} in (1.1) depends on the term m^3, which can be reduced to m^ω with $\omega < 3$ by using contemporary methods for solving linear equations. For example, $\omega = 2.807$ for Strassen's algorithm [14]. According to [8], the best currently available $\omega = 2.37286$, but this method of solving linear equations is impractical due to the large value of the constant in O.

Bounds in (1.1) and (1.2) can be used to compare upper bounds on computational complexities for finding an optimal policy by policy iterations and for finding an ϵ-optimal policy by value iterations. The upper bound for the former is $N_I^{PI}(\alpha)\cdot N_O^{PI} = O(m^3k + mk^2)$, and, if the spread R of one-step rewards does not grow with the size of the problem, then the upper bound for the latter is $N_I^{VI(\epsilon)}(\alpha)\cdot N_O^{VI} = O(mk)$ (the constant V is controlled by the algorithm, and ϵ and α do not depend on m and k). The latter bound is asymptotically better than the former.

The upper bounds $N_I^{VI(\epsilon)}(\alpha)$ and $N_I^{PI}(\alpha)$ on the number of iterations increase in α. Therefore, an upper bound on the number of iterations for some $\alpha^* \in [0, 1)$ is also the bound for all discount factors $\alpha \in [0, \alpha^*)$. In addition to (1.2), more accurate upper bounds on the number of iterations for computing ϵ-optimal policies by value iterations are presented in (3.5), (3.8), and (3.10). However, the bounds in (3.5) and (3.8) depend on the additional constant γ defined in (3.4), and finding γ requires additional computations. Bounds (3.5) and (3.10) may not increase in $\alpha \in [0, 1)$, and, as Example 1 demonstrates, it is possible that these upper bounds for some $\alpha \in [0, 1)$ are not upper bounds on the number of iterations for a smaller discount factor.

As is well-known and clear from (1.1) and (1.2), the number of operations at each step is larger for the policy iteration algorithm than for the value iteration algorithm. If the number of states m is large, then the difference $(N_O^{PI} - N_O^{VI})$ can be significant. In order to accelerate policy iterations, the method of modified policy iterations was introduced in [12]. This method uses value iterations to solve linear equations. As shown in Feinberg et al. [5], modified policy iterations and their versions are not strongly polynomial algorithms for finding optimal policies.

2. Definitions

Let $\mathbb{N}$ and $\mathbb{R}$ be the sets of natural numbers and real numbers respectively. For a finite set E, let $|E|$ denote the number of elements in the set E. We consider an MDP with a finite state space $\mathbb{X} = \{1, 2, \ldots, m\}$, where $m \in \mathbb{N}$ is the number of states, and nonempty finite action sets $A(x)$ available at states $x \in \mathbb{X}$. Let $\mathbb{A} := \bigcup_{x\in\mathbb{X}} A(x)$ be the action set. We recall that $k = \sum_{x\in\mathbb{X}} |A(x)|$ is the total number of actions at all states or, in slightly different terms, the number of all state–action pairs. For each $x \in \mathbb{X}$, if an action $a \in A(x)$ is selected at the state $x \in \mathbb{X}$, then a one-step reward $r(x, a)$ is collected and the process moves to the next state $y \in \mathbb{X}$ with the probability $p(y|x, a)$, where $r(x, a)$ is a real number and $\sum_{y\in\mathbb{X}} p(y|x, a) = 1$. The process continues over a finite or infinite planning horizon. For a finite-horizon problem, the terminal real-valued reward $v^0(x)$ is collected at the final state $x \in \mathbb{X}$.

A *deterministic policy* ϕ is a mapping $\phi : \mathbb{X} \mapsto \mathbb{A}$ such that $\phi(x) \in A(x)$ for each $x \in \mathbb{X}$, and, if the process is at a state $x \in \mathbb{X}$, then the action $\phi(x)$ is selected. An arbitrary policy π can be randomized and history-dependent; see e.g., Puterman [11, p. 154] for definitions of various classes of policies. In particular, a nonrandomized Markov policy φ is defined by a sequence of mappings $(\varphi_t)_{t=0,1,\ldots}$ such that $\varphi_t : \mathbb{X} \mapsto \mathbb{A}$ with $\varphi_t(x) \in \mathbb{A}(x)$ for all $x \in \mathbb{X}$ and $t = 0, 1, \ldots$. We denote by Π, $\mathbb{M}$, and $\mathbb{F}$ the set of all policies, Markov nonrandomized policies, and deterministic policies respectively; $\mathbb{F} \subset \mathbb{M} \subset \Pi$.

Let $\alpha \in [0, 1)$ be a *discount factor*. For a policy $\pi \in \Pi$ and for an initial state $x_0 = x$, the expected total discounted reward for an n-horizon problem is

$$v_{n,\alpha}^\pi(x) := \mathbb{E}_x^\pi \left[\sum_{t=0}^{n-1} \alpha^t r(x_t, a_t) + \alpha^n v^0(x_n) \right], \quad n \in \mathbb{N},\ x \in \mathbb{X},$$

and for the infinite-horizon problem it is

$$v_\alpha^\pi(x) := \mathbb{E}_x^\pi \sum_{t=0}^{\infty} \alpha^t r(x_t, a_t), \quad x \in \mathbb{X},$$

where $\mathbb{E}_x^\pi$ is the expectation defined by the initial state x and the policy π, and where x_t and a_t are states and actions at epochs $t = 0, 1, \ldots$. The *value functions* are defined for initial states $x \in \mathbb{X}$ as

$$v_{n,\alpha}(x) := \sup_{\pi\in\Pi} v_{n,\alpha}^\pi(x), \quad n \in \mathbb{N},\ x \in \mathbb{X}, \tag{2.1}$$

for n-horizon problems, and

$$v_\alpha(x) := \sup_{\pi\in\Pi} v_\alpha^\pi(x), \quad x \in \mathbb{X}, \tag{2.2}$$

for infinite-horizon problems. Note that $v_{0,\alpha}^\pi = v_{0,\alpha} = v^0$ for all $\alpha \in [0, 1)$ and $\pi \in \Pi$.

A policy π is called *optimal* (*n-horizon optimal* for $n = 1, 2, \ldots$) if $v_\alpha^\pi(x) = v_\alpha(x)$ ($v_{n,\alpha}^\pi(x) = v_{n,\alpha}(x)$) for all $x \in \mathbb{X}$. It is well-known that for discounted MDPs with finite action sets there exist nonrandomized Markov optimal policies for finite-horizon problems and deterministic optimal policies for infinite horizon problems; see [11, p. 154]. Therefore, (2.1) and (2.2) can be rewritten as $v_{n,\alpha}(x) := \sup_{\varphi\in\mathbb{M}} v_{n,\alpha}^\varphi(x)$ and $v_\alpha(x) := \sup_{\phi\in\mathbb{F}} v_\alpha^\phi(x)$ respectively.

A policy π is called ϵ-optimal for $\epsilon \ge 0$ if $v_\alpha^\pi(x) \ge v_\alpha(x) - \epsilon$ for all $x \in \mathbb{X}$. A 0-optimal policy is optimal. The objective of this paper is to estimate the complexity of the value iteration algorithm for finding a deterministic ϵ-optimal policy for $\epsilon > 0$. The rest of this paper deals only with deterministic policies.

3. Main results

For a real-valued function $v : \mathbb{X} \to \mathbb{R}$, let us define

$$T_\alpha^a v(x) := r(x, a) + \alpha \sum_{y\in\mathbb{X}} p(y|x, a)v(y), \quad x \in \mathbb{X},\ a \in A(x). \tag{3.1}$$

We shall use the notation $T_\alpha^\phi v(x) := T_\alpha^{\phi(x)} v(x)$ for a deterministic policy ϕ. For $v : \mathbb{X} \to \mathbb{R}$ we also define the optimality operator T_α,

$$T_\alpha v(x) := \max_{a \in A(x)} T_\alpha^a v(x), \quad x \in \mathbb{X}. \tag{3.2}$$

Every real-valued function v on $\mathbb{X}$ can be identified with a vector $v = (v(1), \dots, v(m))$. Therefore, all real-valued functions on $\mathbb{X}$ form the m-dimensional Euclidean space $\mathbb{R}^m$.

For each $n \in \mathbb{N}$ and all $v^0 \in \mathbb{R}^m$, the expected total discounted rewards $v_{n,\alpha}^\phi$ and v_α^ϕ satisfy the equations

$$v_{n,\alpha}^\phi = T_\alpha^\phi v_{n-1,\alpha}^\phi, \quad v_\alpha^\phi = T_\alpha^\phi v_\alpha^\phi,$$

the value functions $v_{n,\alpha}$, v_α satisfy the *optimality equations*

$$v_{n,\alpha} = T_\alpha v_{n-1,\alpha}, \quad v_\alpha = T_\alpha v_\alpha,$$

and *value iterations* converge to the infinite-horizon expected total rewards and optimal values

$$\begin{aligned} v_\alpha^\phi &:= \lim_{n\to\infty} \left(T_\alpha^\phi\right)^n v^0 = T_\alpha^\phi v_\alpha^\phi, \\ v_\alpha &:= \lim_{n\to\infty} (T_\alpha)^n v^0 = \lim_{n\to\infty} v_{n,\alpha} = T_\alpha v_\alpha, \end{aligned} \tag{3.3}$$

and a deterministic policy ϕ is optimal if and only if $T_\alpha^\phi v_\alpha = T_\alpha v_\alpha$; see, e.g., [3], [11, pp. 146–151]. Therefore, if we consider the nonempty sets

$$A_\alpha(x) = \{a \in A(x) : v_\alpha(x) = T_\alpha^a v_\alpha(x)\},$$

then a deterministic policy ϕ is optimal if and only if $\phi(x) \in A_\alpha(x)$ for all $x \in \mathbb{X}$.

For a given $\epsilon > 0$, the following value iteration algorithm computes a deterministic ϵ-optimal policy. It uses a stopping rule based on the value of the span $\mathrm{sp}(v_{n,\alpha} - v_{n-1,\alpha})$. As mentioned in [11, p. 205], this algorithm generates the same number of iterations as the relative value iteration algorithm originally introduced to accelerate value iterations.

Algorithm 1 (*Computing a Deterministic ϵ-optimal Policy by Value Iterations*)**.**

For given MDP, discount factor $\alpha \in (0,1)$, and constant $\epsilon > 0$:

1. *select a vector $u := v^0 \in \mathbb{R}^m$ and a constant*
$\Delta > \dfrac{1-\alpha}{\alpha}\epsilon$ *(e.g., choose $\Delta := \epsilon/\alpha$);*
2. *while $\Delta > \dfrac{1-\alpha}{\alpha}\epsilon$ compute $v = T_\alpha u$, $\Delta := \mathrm{sp}(u - v)$ and set*
$u^* := u$, $u := v$ *end while;*
3. *choose a deterministic policy ϕ such that*
$v = T_\alpha^\phi u^*$; *this policy is $\epsilon-$optimal.*

If $\alpha = 0$, then a deterministic policy ϕ is optimal if and only if $r(x, \phi(x)) = \max_{a\in A(x)}\{r(x,a)\}$. As is well-known, Algorithm 1 converges within a finite number of iterations (e.g., this follows from (3.3)) and returns an ϵ-optimal policy ϕ (e.g., this follows from [11, Proposition 6.6.5]).

Following Puterman [11, Theorem 6.6.6], let us define

$$\gamma := \max_{\substack{x,y\in\mathbb{X} \\ a\in A(x), b\in A(y)}} \left[1 - \sum_{z\in\mathbb{X}} \min\{p(z|x,a), p(z|y,b)\}\right]. \tag{3.4}$$

We notice that $0 \le \gamma \le 1$. If $\gamma = 0$, then $p(z|x,a) = p(z|y,b)$ for all $x, y, z \in \mathbb{X}$ and $a \in A(x), b \in A(y)$, which implies that all deterministic policies have the same transition probabilities. Therefore, a deterministic policy ϕ is optimal if and only if it maximizes the one step reward at each state, that is, $r(x, \phi(x)) = \max_{a\in A(x)}\{r(x,a)\}$. If an MDP has deterministic transition probabilities and there are two or more deterministic policies with nonidentical transition matrices, then $\gamma = 1$.

Finding the value of γ requires computing the sum in (3.4) for all couples $\{(x,a),(y,b)\}$ of state–action pairs such that $(x,a) \ne (y,b)$. The total number of such couples is $k(k-1)/2 = O(k^2)$. The number of arithmetic operations in (3.4), which are additions, is m for each couple. Therefore, the straightforward computation of γ requires $O(mk^2)$ operations, which can be significantly larger than the complexity to compute a deterministic ϵ-optimal policy, which is the product of $N_I^{VI(\epsilon)}(\alpha)$ and N_O^{VI} defined in (1.2). Puterman [11, Eq. (6.6.16)] also provides an upper bound $\gamma' \in [\gamma, 1]$, where $\gamma' := 1 - \sum_{z\in\mathbb{X}} \min_{x\in\mathbb{X}, a\in A(x)} p(z|x,a)$, whose computation requires $O(mk)$ operations.

For $\alpha \in (0,1)$, $\gamma \in (0,1]$, $\epsilon > 0$, and $\mathrm{sp}(v_{1,\alpha} - v^0) > 0$, we define

$$n^*(\alpha) := \max\left\{\left\lceil \frac{\log \frac{(1-\alpha)\epsilon\gamma}{\mathrm{sp}(v_{1,\alpha}-v^0)}}{\log(\alpha\gamma)} \right\rceil, 1\right\}. \tag{3.5}$$

In addition to α, the values of $n^*(\alpha)$ depend also on other parameters presented in (3.5). If all the parameters are fixed in (3.5), except $s := \mathrm{sp}(v_{1,\alpha} - v^0)$, then $n^*(\alpha) = 1$ when s is close to 0. So, we set $n^*(\alpha) = 1$ when $\mathrm{sp}(v_{1,\alpha} - v^0) = 0$. If all the parameters are fixed except γ, then the function $n^*(\alpha)$ has the property that it has a right limit in γ at $\gamma = 0$. We set $n^*(\alpha)$ equal to this limit if $\gamma = 0$. It is easy to see that $n^*(\alpha) \in \{1, 2\}$ if $\gamma = 0$. It is easy to see that the functuion $n^*(\alpha)$ is non-decreasing in $\gamma \in [0, 1]$.

Theorem 1. *For $\alpha \in (0,1)$ and $\epsilon > 0$, Algorithm 1 finds a deterministic ϵ-optimal policy within no more than $n^*(\alpha)$ iterations. In addition, each iteration uses at most $O(mk)$ operations.*

Proof. As follows from (3.1) and (3.2), each iteration uses at most $O(mk)$ arithmetic operations. Let n be the actual number of iterations. According to its steps 2 and 3, the algorithm returns the deterministic policy ϕ, for which $v_{n,\alpha} = T_\alpha v_{n-1,\alpha} = T_\alpha^\phi v_{n-1,\alpha}$, and

$$\mathrm{sp}(T_\alpha v_{n-1,\alpha} - v_{n-1,\alpha}) \le \frac{1-\alpha}{\alpha}\epsilon.$$

In view of [11, Proposition 6.6.5, p. 201], this ϕ is ϵ-optimal. By [11, Corollary 6.6.8, p. 204],

$$\mathrm{sp}(v_{n,\alpha} - v_{n-1,\alpha}) \le (\alpha\gamma)^{n-1}\,\mathrm{sp}(v_{1,\alpha} - v^0). \tag{3.6}$$

Therefore, the minimal number $n \in \mathbb{N}$ satisfying

$$(\alpha\gamma)^{n-1}\,\mathrm{sp}(v_{1,\alpha} - v^0) \le \frac{1-\alpha}{\alpha}\epsilon \tag{3.7}$$

leads to the definition of $n^*(\alpha)$ in and below (3.5). □

Given a discount factor α, formula (1.1) provides an upper bound on the number of iterations for a policy iteration algorithm for all discount factors smaller than or equal to α. This is true because this bound is monotone increasing in the discount factor α. Therefore, monotonicity of the bound in the discount factor is a desired property. The following example shows that bound (3.5) may be exact, and it may not be monotone. As shown in Post and Ye [10], the version of policy iterations changing the policy at each iteration at one state has strongly polynomial bounds for deterministic MDPs. Example 1 also shows that this is not true for Algorithm 1.

Example 1. This example shows that the bound in (3.5) can be exact, and it may not be monotone in the discount factor. Let the state space be $\mathbb{X} = \{1, 2, 3\}$, and the action space be $\mathbb{A} = \{b, c\}$. Let $A(1) = \mathbb{A}$, $A(2) = A(3) = \{b\}$ be the sets of actions available

at states 1, 2, and 3 respectively. The transition probabilities are given by $p(3|1,b) = p(2|1,c) = p(2|2,b) = p(3|3,b) = 1$. The one-step rewards are $r(1,b) = r(1,c) = 0, r(2,b) = 1$, and $r(3,b) = -1$; see Fig. 1.

We set $v^0(1) = 1, v^0(2) = 2, v^0(1) = -2$. As discussed above, $\gamma = 1$ for this MDP with deterministic transitions. Straightforward calculations imply that

$$v_{n,\alpha} = \left(\alpha^n + \sum_{k=1}^{n}\alpha^k,\ \alpha^n + \sum_{k=0}^{n}\alpha^k,\ -\alpha^n - \sum_{k=0}^{n}\alpha^k\right),$$
$$v_{n,\alpha} - v_{n-1,\alpha} = \left(2a^n - a^{n-1},\ 2a^n - a^{n-1},\ -2a^n + a^{n-1}\right),$$
$$\mathrm{sp}(v_{n,\alpha} - v_{n-1,\alpha}) = 2\alpha^{n-1}\,|2\alpha - 1| = \alpha^{n-1}\mathrm{sp}(v_{1,\alpha} - v^0),$$

where the ith coordinates of the vectors correspond to the states $i = 1, 2, 3$. The last displayed equality implies that inequality (3.6) holds in the form of an equality for this example. Therefore, the bound in (3.5) is also the actual number of iterations executed by Algorithm 1 for this MDP, which is

$$n^*(\alpha) = \max\left\{\left\lceil \frac{\log\frac{(1-\alpha)\epsilon}{2|2\alpha-1|}}{\log\alpha}\right\rceil, 1\right\}$$

for $\alpha \neq 0.5$ and $\epsilon > 0$. If $\alpha = 0.5$, then Algorithm 1 stops after the first iteration. Let $\epsilon = 0.02$. Then $n^*(0.24) = 3$, $n^*(0.47) = 4$, and $n^*(0.48) = 3$, which shows that $n^*(\alpha)$ is not monotone in α. It is easy to see that $\lim_{\alpha\uparrow 1} n^*(\alpha) = \infty$. □

Let us consider the vector $v^1 \in \mathbb{R}^m$ defined in (1.4). The following theorem presents a bound which is slightly worse than the bound in Theorem 1, but it monotonically increases in the discount factor α.

Theorem 2. *Let $\alpha \in (0,1)$. For fixed $\epsilon > 0$, $\gamma \in (0,1]$, and v^0, $v^1 \in \mathbb{R}^m$ such that $sp(v^1) + sp(v^0) > 0$, Algorithm 1 finds a deterministic ϵ-optimal policy after a finite number of iterations bounded above by*

$$F(\alpha) := \max\left\{\left\lceil \frac{\log\frac{(1-\alpha)\epsilon\gamma}{sp(v^1)+(1+\alpha)sp(v^0)}}{\log(\alpha\gamma)}\right\rceil, 1\right\}. \tag{3.8}$$

Furthermore, the function $F(\alpha)$ defined in (3.8) for $\alpha \in (0,1)$ has the following properties for an arbitrary fixed parameter $\gamma \in (0,1]$:

(a) $\lim_{\alpha\downarrow 0} F(\alpha) = 1$, $\lim_{\alpha\uparrow 1} F(\alpha) = +\infty$;

(b) *$F(\alpha)$ is non-decreasing in α.*

Proof. By (3.1) and (3.2),

$$\begin{aligned} v_{1,\alpha}(x) &= \max_{a\in A(x)}\left\{r(x,a) + \alpha\sum_{y\in\mathbb{X}} p(y|x,a)v^0(y)\right\} \\ &\le \max_{a\in A(x)}\left\{r(x,a) + \alpha\sum_{y\in\mathbb{X}} p(y|x,a)\max_{z\in\mathbb{X}} v^0(z)\right\} \\ &= \max_{a\in A(x)} r(x,a) + \alpha\max_{z\in\mathbb{X}} v^0(z). \end{aligned}$$

Similarly, $v_{1,\alpha}(x) \ge \max_{a\in A(x)} r(x,a) + \alpha\min_{z\in\mathbb{X}} v^0(z)$. Therefore,

$$\begin{aligned} \mathrm{sp}(v_{1,\alpha}) &= \max_{x\in\mathbb{X}} v_{1,\alpha}(x) - \min_{x\in\mathbb{X}} v_{1,\alpha}(x) \\ &\le \max_{x\in\mathbb{X}}\max_{a\in A(x)} r(x,a) + \alpha\max_{z\in\mathbb{X}} v^0(z) \\ &\quad - \min_{x\in\mathbb{X}}\max_{a\in A(x)} r(x,a) - \alpha\min_{z\in\mathbb{X}} v^0(z) \\ &= \mathrm{sp}(v^1) + \alpha\mathrm{sp}(v^0). \end{aligned}$$

By the properties of seminorm provided in [11, p. 196],

$$\mathrm{sp}(v_{1,\alpha} - v^0) \le \mathrm{sp}(v_{1,\alpha}) + \mathrm{sp}(v^0) \le \mathrm{sp}(v^1) + (1+\alpha)\mathrm{sp}(v^0), \tag{3.9}$$

which together with $\alpha\gamma \in [0,1)$ and definitions of $F(\alpha)$, $n^*(\alpha)$ in (3.5), (3.8) implies $F(\alpha) \ge n^*(\alpha)$.

The formulae in (a) follow directly from (3.8). To prove (b), we recall that $R = \mathrm{sp}(v^1)$ and $V = \mathrm{sp}(v^0)$; see (1.5) and (1.3). By the assumption in the theorem, $R + (1+\alpha)V > 0$. The function $\log\frac{(1-\alpha)\epsilon\gamma}{R+(1+\alpha)V}$ is decreasing in $\alpha \in (0,1)$, and the function $\log(\alpha\gamma)$ is negative and increasing in $\alpha \in (0,1)$. Thus, the function

$$f(\alpha) := \frac{\log\frac{(1-\alpha)\epsilon\gamma}{R+(1+\alpha)V}}{\log(\alpha\gamma)}$$

is increasing when $\log\frac{(1-\alpha)\epsilon\gamma}{R+(1+\alpha)V} < 0$, which is $\alpha > \frac{\epsilon\gamma-R-V}{\epsilon\gamma+V}$. This implies that $F(\alpha)$ is increasing on the interval $(b,1)$, where $b := \max\{\frac{\epsilon\gamma-R-V}{\epsilon\gamma+V}, 0\}$. Thus, if $\frac{\epsilon\gamma-R-V}{\epsilon\gamma+V} \le 0$, then the theorem is proved. Now let $\frac{\epsilon\gamma-R-V}{\epsilon\gamma+V} > 0$. For every $\alpha \in (0, \frac{\epsilon\gamma-R-V}{\epsilon\gamma+V}]$ we have that $\log\frac{(1-\alpha)\epsilon\gamma}{R+(1+\alpha)V} \ge 0$, which implies $f(\alpha) \le 0$. In view of (3.8), $F(\alpha) = \max\{\lceil f(\alpha)\rceil, 1\} = 1$ for all $\alpha \in (0, \frac{\epsilon\gamma-R-V}{\epsilon\gamma+V}]$. Therefore, we conclude that the function $F(\alpha)$ is non-decreasing on $(0,1)$. □

As explained in the paragraph preceding Theorem 1, it may be time-consuming to find the actual value of γ for an MDP. The following corollary provides the bounds that do not use γ.

Corollary 1. *Let $\alpha \in (0,1)$. For a fixed $\epsilon > 0$, if $sp(v^1)+sp(v^0) > 0$, then*

$$n^*(\alpha) \le N^\epsilon(\alpha) := \max\left\{\left\lceil \frac{\log\frac{(1-\alpha)\epsilon}{sp(v_{1,\alpha}-v^0)}}{\log\alpha}\right\rceil, 1\right\}, \tag{3.10}$$

$$F(\alpha) \le N_I^{VI(\epsilon)}(\alpha) := \max\left\{\left\lceil \frac{\log\frac{(1-\alpha)\epsilon}{sp(v^1)+(1+\alpha)sp(v^0)}}{\log\alpha}\right\rceil, 1\right\}, \tag{3.11}$$

and $N^\epsilon(\alpha) \le N_I^{VI(\epsilon)}(\alpha)$.

Proof. We recall that $n^*(\alpha)$ is defined as the smallest $n \in \mathbb{N}$ satisfying (3.7). The right-hand side of (3.10) is the smallest $n \in \mathbb{N}$ satisfying $\alpha^{n-1}\mathrm{sp}(v_{1,\alpha} - v^0) \le \frac{1-\alpha}{\alpha}\epsilon$. Since $0 \le \alpha\gamma \le \alpha$, the inequality in (3.10) holds. The inequality in (3.11) holds because of the similar reasons, where $F(\alpha)$ and $N_I^{VI(\epsilon)}(\alpha)$ are the smallest $n \in \mathbb{N}$ satisfying $(\alpha\gamma)^{n-1}[\mathrm{sp}(v^1) + (1+\alpha)\mathrm{sp}(v^0)] \le \frac{1-\alpha}{\alpha}\epsilon$ and $\alpha^{n-1}[\mathrm{sp}(v^1) + (1+\alpha)\mathrm{sp}(v^0)] \le \frac{1-\alpha}{\alpha}\epsilon$ respectively. The inequality $N^\epsilon(\alpha) \le N_I^{VI(\epsilon)}(\alpha)$ follows from (3.9). □

Remark 1. If $\gamma = 1$ in (3.8), then $N_I^{VI(\epsilon)}(\alpha) = F(\alpha)$. Therefore, the function $N_I^{VI(\epsilon)}$ also satisfies properties (a) and (b) stated in Theorem 2.

We notice that, if the function F from (3.8) is minimized in v^0, then the smallest value is attained when $\mathrm{sp}(v^0) = 0$, that is, $v^0 = const$. The following corollary provides upper bounds for $v^0 = const$ including $v^0 \equiv 0$.

Corollary 2. *Let $\alpha \in (0,1)$, $\gamma > 0$ and let $v^0 = const$. If $sp(v^1) > 0$, then Algorithm 1 finds a deterministic ϵ-optimal policy after a finite number of iterations bounded above by*

$$F^*(\alpha) := \max\left\{\left\lceil \frac{\log\frac{(1-\alpha)\epsilon\gamma}{sp(v^1)}}{\log(\alpha\gamma)}\right\rceil, 1\right\} \le \max\left\{\left\lceil \frac{\log\frac{(1-\alpha)\epsilon}{sp(v^1)}}{\log\alpha}\right\rceil, 1\right\}.$$

Proof. This corollary follows from (3.11). □

Example 2. This example illustrates the monotonicity of the upper bound $F(\alpha)$ for computing ϵ-optimal policies and non-monotonicity of the number of calculations to find an optimal

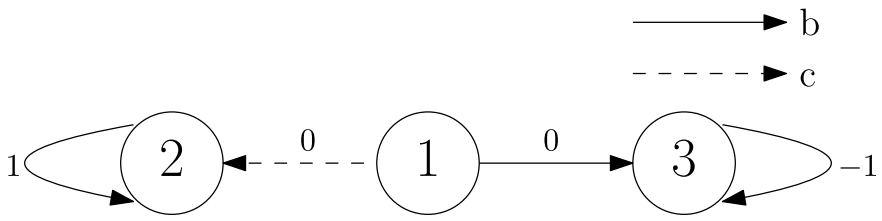

Fig. 1. MDP diagram for Example 1.

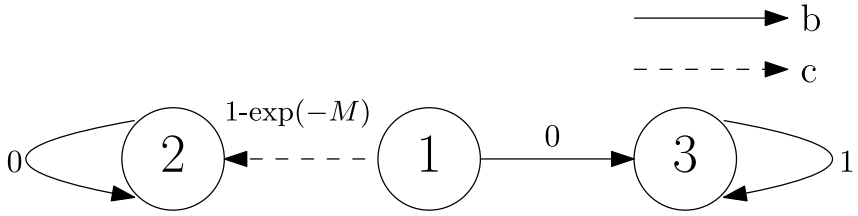

Fig. 2. MDP diagram for Example 2.

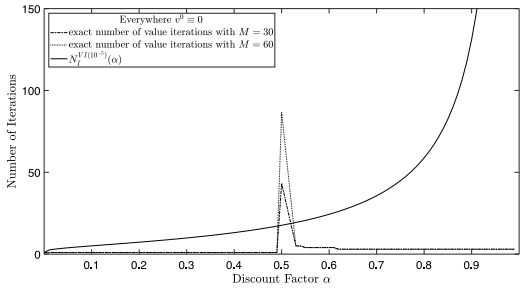

Fig. 3. Exact numbers of iterations for finding optimal policies and $N_l^{VI(10^{-5})}$ in Example 2. (Numbers of iterations are integers, and the graphs display step functions. This figure is generated by the Matlab, and the graphs are automatically smoothened to clarify comparisons.)

policy by value iterations. The following MDP is taken from [4]. Let the state space be $\mathbb{X} = \{1, 2, 3\}$ and the action space be $\mathbb{A} = \{b, c\}$. Let $A(1) = \mathbb{A}$, $A(2) = A(3) = \{b\}$ be the sets of actions available at states 1, 2, 3 respectively; see Fig. 2. The transition probabilities are given by $p(2|1, c) = p(3|1, b) = p(2|2, b) = p(3|3, b) = 1$. The one-step rewards are $r(1, b) = r(2, b) = 0, r(3, b) = 1$, and $r(1, c) = 1 - \exp(-M)$ where $M > 0$. We set $v^0(x) = 0$ for $x \in \mathbb{X}$.

As shown in [4], for $\alpha = 0.5$ the number of value iterations required to find an optimal policy increases to infinity as M increases to infinity. This shows that the value iteration algorithm for computing the optimal policy is not strongly polynomial. However, $\mathrm{sp}(v^1) = 1$ does not change with the increasing M in this example. As follows from Corollary 2, for fixed $\epsilon > 0$ and $\alpha \in (0, \alpha^*]$ with $\alpha^* \in (0, 1)$, the number of required iterations $N_l^{VI(\epsilon)}$ for Algorithm 1 is uniformly bounded no matter how large M is; see Fig. 3. □

Let $N(\alpha)$ be the exact number of iterations required for computing an optimal policy by value iterations with discount factor α. Lewis and Paul [9] provide examples of MDPs for which $N(\alpha)$ could be unbounded for discount factor bounded away from 1. In other words, there may exist $\tilde{\alpha} \in (0, 1)$ and a sequence of discount factors $\{\alpha_n\}_{n=0,1,\ldots}$ such that $\lim_{n\to\infty} \alpha_n = \tilde{\alpha}$ and $\lim_{n\to\infty} N(\alpha_n) = \infty$. Here we provide a significantly simpler example.

Example 3. This example shows that exact number of value iterations to compute an optimal policy may be unbounded on any neighborhood of some discount factor $\tilde{\alpha} \in (0, 1)$. Consider an MDP with the state space $\mathbb{X} = \{1, 2, 3\}$, with the action space $\mathbb{A} = \{b, c\}$, and with the sets of actions $A(1) = \mathbb{A}$ and $A(2) = A(3) = \{b\}$ available at states 1, 2, and 3 respectively; see Fig. 4. The transition probabilities are given by $p(2|1, c) = p(3|1, b) = p(2|2, b) = p(3|3, b) = 1$. The one-step rewards are $r(1, c) = r(2, b) = 1, r(1, b) = 2$, and $r(3, b) = 0$.

We set $v^0(x) = 0$ for $x \in \mathbb{X}$. There are only two deterministic policies denoted by ϕ and ψ, which differ only at state 1 with $\phi(1) = c$ and $\psi(1) = b$. Observe that $v_\alpha^\phi(x) = v_\alpha^\psi(x) = v_\alpha(x)$ for $x = 2, 3$. Hence, to compare ϕ and ψ, we only need to consider the value function of initial state 1. In addition,

$$v_\alpha^\phi(1) = \sum_{k=0}^{\infty} \alpha^k = \frac{1}{1-\alpha}, \quad v_\alpha^\psi(1) = 2,$$

which shows that ψ is optimal for $\alpha \in [0, 0.5]$, and ϕ is optimal for $\alpha \in [0.5, 1)$. Now let us see which policy value iterations pick at the nth iteration. Clearly $v_{n,\alpha}(3) = 0$ and $v_{n,\alpha}(2) = \sum_{i=0}^{n-1} \alpha^i$ for all $n \in \mathbb{N}$. If $n = 1, 2$, then $v_{n,\alpha}(1) = 2$, and value iterations always select policy ψ. For $n \geq 3$, let $\beta_n \in (0, 1)$ such that $\sum_{i=0}^{n-1} (\beta_n)^i = 2$. Notice that $0.5 < \beta_n < 1$, β_n strictly decreases in n, and $\lim_{n\to\infty} \beta_n = 0.5$. Thus, by (3.2) we have

$$v_{n,\alpha}(1) = \max\left\{1 + \alpha \sum_{i=0}^{n-2} \alpha^i, 2 + 0\right\} = \begin{cases} 2, & \text{if } \alpha \in (0, \beta_n]; \\ \sum_{i=0}^{n-1} \alpha^i, & \text{if } \alpha \in [\beta_n, 1), \end{cases} \quad \text{for } n \geq 3.$$

If $\alpha \in (0, 1)$, then at the nth value iteration policy ϕ is selected if $v_{n,\alpha}(1) = \sum_{i=0}^{n-1} \alpha^i > 2$. Therefore, the definition of β_n implies that for each $\alpha \in (\beta_{n+1}, \beta_n)$ value iterations select the optimal policy ϕ for the first time at the $(n + 1)$th iteration. Hence

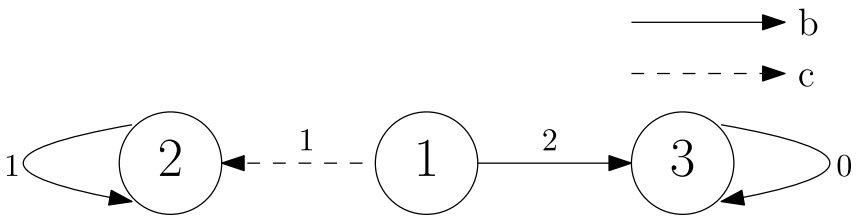

Fig. 4. MDP diagram for Example 3.

$N(\alpha_n) = n + 1$ for $\alpha_n := \frac{1}{2}(\beta_{n+1} + \beta_n) \to 0.5$ as $n \to \infty$. Thus $\lim_{n\to\infty} N(\alpha_n) = \infty$, and $\tilde{\alpha} := \lim_{n\to\infty} \alpha_n = 0.5$. □

Examples 2 and 3 represent the main difficulties of running value iterations for computing optimal policies. Nevertheless, the results of this paper show that these difficulties can be easily overcome by using value iterations for computing ϵ-optimal policies.

Acknowledgment

This research was partially supported by the National Science Foundation grant CMMI-1636193.

References

[1] D.P. Bertsekas, Reinforcement Learning and Optimal Control, Athena Scientific, Belmont, MA, 2019.
[2] D.P. Bertsekas, J.N. Tsitsiklis, Neuro-Dynamic Programming, Athena Scientific, 1996, Belmont, MA.
[3] E.A. Feinberg, Total reward criteria, in: E.A. Feinberg, A. Schwartz (Eds.), Handbook of Markov Decision Processes, Kluwer, Boston, MA, 2002, pp. 173–207.
[4] E.A. Feinberg, J. Huang, The value iteration algorithm is not strongly polynomial for discounted dynamic programming, Oper. Res. Lett. 42 (2014) 130–131.
[5] E.A. Feinberg, J. Huang, B. Scherrer, Modified policy iteration algorithms are not strongly polynomial for discounted dynamic programming, Oper. Res. Lett. 42 (2014) 429–431.
[6] L.C.M. Kallenberg, Finite state and action MDPs, in: E.A. Feinberg, A. Schwartz (Eds.), Handbook of Markov Decision Processes, Kluwer, Boston, MA, 2002, pp. 21–87.
[7] T. Kitahara, S. Mizuno, A bound for the number of different basic solutions generated by the simplex method, Math. Program. 137 (2013) 579–586.
[8] J.F. Le Gall, Powers of tensors and fast matrix multiplication, in: Proc. of 39-th Int. Symp. Symb. Alge. Comput. 2014, pp. 296-303.
[9] M.E. Lewis, A. Paul, Uniform turnpike theorems for finite Markov decision processes, Math. Oper. Res. 44 (2019) 1145–1160.
[10] I. Post, Y. Ye, The simplex method is strongly polynomial for deterministic Markov decision processes, Math. Oper. Res. 40 (2015) 859–868.
[11] M.L. Puterman, Markov Decision Processes: Discrete Stochastic Dynamic Programming, John Wiley & Sons, Inc., New York, 1994.
[12] M.L. Puterman, M.C. Shin, Modified policy iteration algorithms for discounted Markov decision problems, Manag. Sci. 24 (1978) 1127–1137.
[13] B. Scherrer, Improved and generalized upper bounds on the complexity of policy iteration, Math. Oper. Res. 41 (2016) 758–774.
[14] V. Strassen, Gussian elimination is not optimal, Numer. Math. 13 (1969) 354–356.
[15] R.S. Sutton, A.G. Barto, Reinforcement Learning: An Introduction, second ed., MIT Press, Cambridge, 2018.
[16] P. Tseng, Solving h-horizon, stationary Markov decision problems in time proportional to log(h), Oper. Res. Lett. 9 (1990) 287–297.
[17] Y. Ye, The simplex and policy-iteration methods are strongly polynomial for the Markov decision problem with a fixed discount rate, Math. Oper. Res. 36 (2011) 593–603.