Operations Research Letters 9 (1990) 287-297
North-Holland

September 1990

SOLVING *H*-HORIZON, STATIONARY MARKOV DECISION PROBLEMS IN TIME PROPORTIONAL TO LOG(*H*)

Paul TSENG

Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139, USA

Received July 1988
Revised October 1989

We consider the H-horizon, stationary Markov decision problem. For the discounted case, we give an ε-approximation algorithm whose time is proportional to $\log(1/\varepsilon)$, $\log(H)$ and $1/(1-\alpha)$, where α is the discount factor. Under an additional stability assumption, we give an exact algorithm whose time is proportional to $\log(H)$ and $1/(1-\alpha)$. For problems where α is bounded away from 1, we obtain, respectively, a fully polynomial approximation scheme and a polynomial-time algorithm. For the undiscounted case, by refining a weighted maximum norm contraction result of Hoffman, we derive analogous results under the assumption that all stationary policies are proper.

computational complexity * dynamic programming * Markov decision process

1. Introduction

Complexity analysis [6,14] has been widely applied in the areas of theoretical science and combinatorial/integer optimization to measure the inherent difficulty of problems. In dynamic programming, such analysis has been less common [12,13,15]. In this article, we make some progress towards filling this gap. In particular, we consider the H-horizon, stationary Markov decision problem [1,4,8], which is not known to be polynomial-time solvable, and show that an ε-optimal solution is computable in time that is proportional to $\log(1/\varepsilon)$ and $\log(H)$. Under an additional stability assumption, we show that an exact solution is computable in time that is proportional to $\log(H)$. For the special case of discounted problems where the discount factor is bounded away from 1, we obtain, respectively, a fully polynomial approximation scheme and a polynomial-time algorithm. Our result in a sense brings us closer to a complete complexity theory for Markov decision problems, for which it is known that the infinite horizon, stationary case is P-complete, and the finite horizon, nonstationary case is in NC [15] (complexity for the infinite horizon, nonstationary case is undefined). (See Appendix A for a brief explanation of the complexity terms used throughout this article.)

We describe the stationary Markov decision problem below. We are given a time horizon $H > 0$ (possibly $H = +\infty$), a finite set of states $S = \{1,\dots, n\}$ and, for each state i, a finite set $D_i = \{1,\dots, m_i\}$ of controls. At each time t ($t = 0, 1,\dots, H-1$), we are in exactly one of the n states (the state at time 0 is given) and, if we are in state i, we choose a control from D_i. If we choose control $k \in D_i$, we incur a cost g_i^k and, with probability p_{ij}^k, we arrive in state j at time $t+1$. If we terminate in state i at time H, we incur a cost c_i. Let $u_i(t)$ denote the control chosen when we are in state i at time t and let $\mu(t) = (u_1(t),\dots, u_n(t))$. Then the Markov decision problem is to choose a *policy* $(\mu(0), \mu(1),\dots, \mu(H-1))$ to minimize the expected total cost

$$\sum_{t=0}^{H-1} \alpha^t \sum_{i\in S} g_i^{u_i(t)} p(i,\ t;\ \mu(0),\dots,\mu(H-1)) + \alpha^H \sum_{i\in S} c_i p(i,\ H;\ \mu(0),\dots,\mu(H-1)),$$

0167-6377/90/$3.50 © 1990 - Elsevier Science Publishers B.V. (North-Holland)

where $\alpha \in (0, 1]$ is the *discount factor* and $p(i, t; \mu(0), \ldots, \mu(H-1))$ denotes the probability of being in state i at time t under the policy $(\mu(0), \mu(1), \ldots, \mu(H-1))$. Such a policy will be called an *optimal policy*. Also a policy $(\mu(0), \mu(1), \ldots, \mu(H-1))$ satisfying $\mu(0) = \mu(1) = \cdots = \mu(H-1)$ will be called *stationary* and will be written as $\mu(0)$. The Markov decision problem is *discounted* (*undiscounted*) if $\alpha < 1$ $(\alpha = 1)$ and has finite (infinite) horizon if $H < +\infty$ $(H = +\infty)$. In what follows, $\|\cdot\|$ will denote the L_∞-norm and $\log(\cdot)$ will denote the logarithm in base 2. For any $\mu = (u_1, \ldots, u_n) \in D_1 \times \cdots \times D_n$, we will denote $P^\mu = [p_{ij}^{u_i}]$ and $g^\mu = (g_1^{u_1}, \ldots, g_n^{u_n})$. We will also denote $c = (c_1, \ldots, c_n)$.

We make the following standing assumption:

Assumption A. α and the p_{ij}^k's are rational numbers. The g_i^k's and c_i's are integers.

Let δ be the smallest positive integer for which $\delta\alpha$ and the δp_{ij}^k's are all integers and $|g_i^k| \leqslant \delta$, $|c_i| \leqslant \delta$ for all i and k, and let $\bar{m}$ be the number of nonzero p_{ij}^k's. (δ represents the accuracy in the problem data.) Then, the *input size* for the ∞-horizon problem (i.e. the number of binary bits needed to write down α, n, the p_{ij}^k's, the g_i^k's, the m_i's and the c_i's) is at most some constant times

$$L = \bar{m} \log(\delta),$$

and the input size for the H-horizon problem $(H < +\infty)$ is at most some constant times $L + \log(H)$ (since H requires $\log(H)$ binary bits to write down). We will also denote, for each $i \in S$,

$$\bar{m}_i = \max_{k \in D_i}\{\text{number of positive elements of } p_{i1}^k, \ldots, p_{in}^k\}.$$

Notice that $\bar{m}$ takes on value between $\Sigma_i m_i$ and $n\Sigma_i m_i$ and each $\bar{m}_i$ takes on value between 1 and n.

To motivate our results, consider the special case of the Markov decision problem where $H < +\infty$. If we use dynamic programming to solve this problem, the time is $O(\bar{m}H)$ arithmetic operations. If we use linear programming, then since the linear program formulation of the problem can be seen to contain $H\Sigma_i m_i$ constraints with an input size of $O(HL)$, the theoretically fastest linear programming algorithm [9,11] would take a time of $O(H^4(\Sigma_i m_i)^3 L)$ arithmetic operations. Since the input size of the problem is at most some constant times $L + \log(H)$, neither of these solution times is a polynomial in the input size. In fact, for this finite horizon case, the Markov decision problem is only known to be P-hard (i.e. as hard as any problem that is polynomial-time solvable) [15], but is not known to be polynomial-time solvable. (It is not even known to be in NP, although it can be seen to be in the larger class PSPACE.) Hence, as a first step towards achieving polynomial-time solvability, we would like to find algorithms whose time is a polynomial in $\log(H)$. We propose a number of such algorithms, both exact and inexact. These algorithms can be viewed as truncated dynamic programming methods whereby truncation occurs at the moment that an optimal stationary policy for the ∞-horizon problem is identified. For the discounted case, we give an ε-approximation algorithm that has a complexity of $O((\bar{m}\log(1/\varepsilon) + nL')/(1-\alpha) + \min\{n^3 \log(H), H\Sigma_i \bar{m}_i\})$ arithmetic operations, where $L' = L + \bar{m}\log(n)$, and, under an additional stability assumption, an exact algorithm that has a complexity of $O(nL'/(1-\alpha) + \min\{n^3 \log(H), H\Sigma_i \bar{m}_i\})$ arithmetic operations. Analogous algorithms are derived for the undiscounted case under the assumption that all stationary policies are proper.

The main difference between the time complexity of our algorithms and that of algorithms using the linear/dynamic programming approach is that the former depends on α and H through a polynomial of $1/(1-\alpha)$ and $\log(H)$ while the latter depends on α and H through a polynomial of $\log(1/(1-\alpha))$ (by way of L) and H. Hence, our algorithms are interesting primarily when the horizon length H is large and the discount factor α is not very near 1.

This article proceeds as follows: in Section 2 we show that, for the ∞-horizon, discounted problem, an optimal stationary policy can be identified by the Jacobi successive approximation method in time that is a polynomial in L and $(1-\alpha)^{-1}$; in Section 3 we use the preceding fact to derive exact and approximation algorithms for the finite horizon, discounted problem; in Section 4 and 5 we perform an analogous analysis for the undiscounted problem; in Section 6 we present our conclusion and discuss extensions.

2. Infinite horizon, discounted case

In this case $H = +\infty$ and $\alpha \in (0, 1)$. Let $T: \mathbb{R}^n \to \mathbb{R}^n$ be the function whose i-th component is given by

$$T_i(x) = \min_{k \in D_i} T_i^k(x), \quad \forall x \in \mathbb{R}^n, \tag{1}$$

where $\mathbb{R}^n$ is the n-dimensional Euclidean space and, for each $k \in D_i$, we define

$$T_i^k(x) = \alpha \sum_j p_{ij}^k x_j + g_i^k. \tag{2}$$

Also, for each $\mu = (u_1, \ldots, u_n) \in D_1 \times \cdots \times D_n$, let $T^\mu: \mathbb{R}^n \to \mathbb{R}^n$ denote the function whose i-th component is $T_i^{u_i}$. It is easily shown using (1)–(2) that T is a *contraction* mapping of modulus α with respect to the L_∞-norm. Hence T has a unique fixed point, which we denote by $x^* = (x_1^*, \ldots, x_n^*)$ (i.e. $x^* = T(x^*)$). Furthermore, there exists at least one optimal policy that is stationary and each stationary policy μ is optimal if and only if $x^* = T^\mu(x^*)$ (see [1, Section 5.3]).

Consider the Jacobi *successive approximation* iterations [1, Section 5.2] for solving this discounted problem:

$$x(t+1) = T(x(t)), \quad t = 0, 1, \ldots, \tag{3a}$$

$$x(0) = c. \tag{3b}$$

Since T is a contraction mapping of modulus α with respect to the L_∞-norm, we have from (3a) that

$$\| x(t+1) - x^* \| \leqslant \alpha \| x(t) - x^* \|, \quad t = 0, 1, \ldots; \tag{4}$$

hence the iterates $x(t)$ converge to x^* at a geometric rate. Furthermore, it is known that an optimal stationary policy is identified after a finite number of iterations [1, p. 236; 4]. Below we refine this result by giving an explicit bound on the number of iterations. This bound will be used in subsequent analysis to derive our main results.

Lemma 1. *Let t^* be the smallest positive integer such that, for all $t \geqslant t^*$, $x(t+1) = T^\mu(x(t))$ implies $x^* = T^\mu(x^*)$. Then $t^* \leqslant \hat{t}$, where*

$$\hat{t} = \left\lceil \log\left(2\delta^{2n+2} n^n \left(\| c \| + \max_{i,k} | g_i^k | / (1 - \alpha) \right)\right) / \log(1/\alpha) \right\rceil.$$

Proof. By (3b) and (4), after $t = \lceil \log(\varepsilon / \| c - x^* \|) / \log(\alpha) \rceil$ iterations, the error $\| x(t) - x^* \|$ is less than ε for any $\varepsilon > 0$. We show below that, for $\varepsilon \leqslant 1/(2\delta^{2n+2}n^n)$, the corresponding policy is optimal. This would then imply that an optimal stationary policy can be identified after $\lceil \log(2\delta^{2n+2}n^n \| c - x^* \|) / \log(1/\alpha) \rceil$ iterations. To obtain a usable bound, notice from (1)–(2) that x^* satisfies

$$(I - \alpha P^\mu) x^* = g^\mu \tag{5}$$

for some $\mu \in D_1 \times \cdots \times D_n$.

Hence

$$\begin{aligned} \| x^* \| &= \| \left(I + (\alpha P^\mu) + (\alpha P^\mu)^2 + \cdots \right) g^\mu \| \\ &\leqslant \| g^\mu \| + \| (\alpha P^\mu) g^\mu \| + \| (\alpha P^\mu)^2 g^\mu \| + \cdots \\ &\leqslant \| g^\mu \| / (1 - \alpha) \\ &\leqslant \max_{i,k} | g_i^k | / (1 - \alpha), \end{aligned}$$

so that $\| c - x^* \|$ is upper bounded by the computable quantity $\| c \| + \max_{i,k} | g_i^k | / (1 - \alpha)$. By plugging the latter quantity into the above bound, we obtain the desired $\hat{t}$.

It only remains to show that if $\| x(t) - x^* \| < 1/(2\delta^{2n+2}n^n)$, then the corresponding policy is optimal. Since $\delta^2(I - \alpha P^\mu)$ and $\delta^2 g^\mu$ are both integers (and the entries of $\delta^2(I - \alpha P^\mu)$ do not exceed δ^2), it follows from (5), Cramer's rule, and the Hadamard determinant inequality [7] that $x^* = w/(\delta^{2n}n^n)$, for some integer vector $w = (w_1, \dots, w_n)$. Consider any i and any $k \in D_i$ such that $x_i^* \neq T_i^k(x^*)$. Since (cf. (2))

$$\begin{aligned} T_i^k(x^*) &= \alpha \sum_j p_{ij}^k x_j^* + g_i^k \\ &= \Big(\delta^2 \alpha \sum_j p_{ij}^k w_j + (\delta^{2n+2} n^n) g_i^k \Big) / (\delta^{2n+2} n^n) \end{aligned}$$

and the numerator is an integer, it must be that x_i^* and $T_i^k(x^*)$ differ by at least $1/(\delta^{2n+2}n^n)$. Hence if $\| x(t) - x^* \| < 1/(2\delta^{2n+2}n^n)$, then

$$\begin{aligned} |T_i^k(x(t)) - x_i^*| &= \Big| \alpha \sum_j p_{ij}^k (x_j(t) - x_j^*) + T_i^k(x^*) - x_i^* \Big| \\ &\geqslant |T_i^k(x^*) - x_i^*| - \Big| \alpha \sum_j p_{ij}^k (x_j(t) - x_j^*) \Big| \\ &\geqslant 1/(\delta^{2n+2}n^n) - \alpha \| x(t) - x^* \| \\ &> 1/(2\delta^{2n+2}n^n) \\ &> \| x(t) - x^* \|. \end{aligned}$$

Since $\| T(x(t)) - x^* \| \leqslant \| x(t) - x^* \|$ (cf. (3a), (4)), this implies that $T_i^k(x(t)) \neq T_i(x(t))$. □

(A slightly different value for $\hat{t}$ is obtained if we use the alternative bound $\| c - T(c) \| / (1 - \alpha)$ on $\| c - x^* \|$.)

Since $\log(\cdot)$ is a concave function and its slope at 1 is 1, we have

$$\begin{aligned} \log(\alpha) &= \log(1 - (1 - \alpha)) \\ &\leqslant -(1 - \alpha). \end{aligned}$$

This, together with the facts (cf. Assumption A) $\| c \| \leqslant \delta$, $\max_{i,k} | g_i^k | / (1 - \alpha) \leqslant \delta^2$, implies that

$$\hat{t} = \mathrm{O}(n \log(n\delta)/(1 - \alpha)), \tag{6}$$

which is a polynomial in L and $1/(1 - \alpha)$.

The a priori estimate $\hat{t}$ on t^* is sometimes too loose to be practical. A tighter estimate of t^* can be obtained by using a more accurate bound on the quantity $\| x^* - x(t) \|$. For example, in [1, p. 190] is described techniques for *generating* a more accurate bound on $\| x^* - x(t) \|$ using the value of $x(t)$ and $x(t-1)$. Then, we can estimate t^* by t whenever this bound is less than $1/(2\delta^{2n+2}n^n)$. Alternatively, we can improve our estimate of t^* by using such a bound to *eliminate* inactive controls. This approach is based on the following lemma:

Lemma 2. *Fix any positive integer $\bar{t}$ and let Δ be any upper bound on $\| x^* - x(\bar{t}) \|$. Then, for any i and any $k \in D_i$, if $T_i^k(x(\bar{t})) > T_i(x(\bar{t})) + 4\alpha\Delta$, then $T_i(x(t)) \neq T_i^k(x(t))$ for all $t \geqslant \bar{t}$.*

Proof. Suppose that for some $t \geqslant \bar{t}$ we have $T_i(x(t)) = T_i^k(x(t))$. Since $\| x^* - x(\bar{t}) \| \leqslant \Delta$, by (4) we also have $\| x^* - x(t) \| \leqslant \Delta$, so that $\| x(t) - x(\bar{t}) \| \leqslant 2\Delta$. This, together with (1)–(2), implies

$$\begin{aligned} T_i^k(x(t)) &\leqslant T_i^{\bar{k}}(x(t)) \\ &= \alpha \sum_j p_{ij}^{\bar{k}} x_j(t) + g^{\bar{k}} \\ &\leqslant \alpha \sum_j p_{ij}^{\bar{k}} (x_j(\bar{t}) + 2\Delta) + g_i^{\bar{k}} \\ &= T_i^{\bar{k}}(x(\bar{t})) + 2\alpha\Delta \\ &= T_i(x(\bar{t})) + 2\alpha\Delta, \end{aligned}$$

where $\bar{k}$ is any element of D_i satisfying $T_i^{\bar{k}}(x(\bar{t})) = T_i(x(\bar{t}))$. Similarly, we have

$$T_i^k(x(t)) \geqslant T_i^k(x(\bar{t})) - 2\alpha\Delta.$$

By combining the above two inequalities, we obtain $T_i^k(x(\bar{t})) \leqslant T_i(x(\bar{t})) + 4\alpha\Delta$. □

Lemma 2 provides a test for eliminating controls that are inactive in all future iterations. (Similar tests for eliminating *non-optimal* controls are given in [1, p. 198; 18].) When only those stationary policies that are optimal for the ∞-horizon problem (which can be determined a priori) are left, then the current iteration count is an estimate of t^*. We have emphasized the accurate estimation of t^* because, as we shall see in Section 3, t^* plays a key role in our solution of finite horizon, discounted problems; the more accurately we estimate t^*, the better our solution times will be. (We remark that analogous estimates can be derived for the Gauss–Seidel iteration $x_i(t+1) = T_i(x_1(t+1),\ldots, x_{i-1}(t+1), x_i(t),\ldots, x_n(t))$, but, as we shall see, only the estimates associated with the Jacobi iteration (3a)–(3b) are useful for our subsequent analysis.)

3. Finite horizon, discounted case

In this case, $H < +\infty$ and $\alpha \in (0, 1)$. Consider the following dynamic programming iterations:

$$x(t) = T(x(t+1)), \quad t = H-1,\ldots,1, 0, \tag{7a}$$

$$x(H) = c, \tag{7b}$$

where T is given by (1)–(2) and $x(t)$ denotes the *cost-to-go* vector at time t. A policy $(\mu(0), \mu(1),\ldots, \mu(H-1))$ can be seen to be optimal for the H-horizon discounted problem if and only if $x(t) = T^{\mu(t)}(x(t+1))$ for all $t = H-1,\ldots,1, 0$. The problem is then to compute $x(0)$, which is the optimal expected cost (and perhaps to determine the optimal policy as well).

Since the iteration (7a)–(7b) is identical to (3a)–(3b), except for the reversal in time, Lemma 1 motivates an algorithm for computing $x(0)$ whereby T in the iteration (7a) is switched to T^{μ}, with μ being some optimal stationary policy for the ∞-horizon problem, the moment that such a policy is identified. We state this algorithm below:

Truncated DP (Dynamic Programming) Algorithm

Phase 0. Choose a positive integer $\tilde{t} \leqslant H-1$. Let $\tilde{x}(H) = c$.
Phase 1. Run the recursion $\tilde{x}(t) = T(\tilde{x}(t+1))$ until $t = H - \tilde{t} - 1$.
Phase 2. Let $\tilde{\mu}$ be any stationary policy satisfying $\tilde{x}(H - \tilde{t} - 1) = T^{\tilde{\mu}}(\tilde{x}(H - \tilde{t}))$. Then compute

$$\tilde{x}(0) = (\alpha P^{\tilde{\mu}})^{H-\tilde{t}}\tilde{x}(H-\tilde{t}) + \left[I + (\alpha P^{\tilde{\mu}}) + \cdots + (\alpha P^{\tilde{\mu}})^{H-\tilde{t}-1}\right] g^{\tilde{\mu}}.$$

We have the following complexity and accuracy results:

Proposition 1. *The following hold for the Truncated DP Algorithm*:

(a) *It has a complexity of* $O(\overline{m}\tilde{t} + \min\{n^3 \log(H-\tilde{t}), (H-\tilde{t})\Sigma_i \overline{m}_i\})$ *arithmetic operations.*

(b) *For* $\tilde{t} = \min\{\hat{t}, H-1\}$, *we have* $\| \tilde{x}(0) - x(0) \| \leqslant 4\alpha^H\delta^2$.

(c) *If the* ∞*-horizon problem has a unique optimal stationary policy*, *then*, *for* $\tilde{t} = \min\{\hat{t}, H-1\}$, *we have* $\tilde{x}(0) = x(0)$.

Proof. (a) It is easily seen that Phases 0 and 1 require $O(\overline{m}\tilde{t})$ arithmetic operations. Since A^k and $I + A + \cdots + A^k$ can be computed using binary powering and factoring (see Appendix B) in $O(n^3 \log(k))$ arithmetic operations for any $n \times n$ matrix A and $k \geqslant 1$, we can perform Phase 2 in $O(n^3 \log(H-\tilde{t}))$ arithmetic operations. Alternatively, we can perform Phase 2 by multiplying $g^{\tilde{\mu}}$ by $\alpha P^{\tilde{\mu}}$ a total of $H - \tilde{t} - 1$ times and summing all the vectors thus obtained. Since $P^{\tilde{\mu}}$ has at most $\Sigma_i \overline{m}_i$ nonzero entries, this takes $O((H-\tilde{t})\Sigma_i \overline{m}_i)$ arithmetic operations.

(b) If $\hat{t} \geqslant H-1$, then $\tilde{t} = H-1$ so that $\tilde{x}(0) = x(0)$. If $\hat{t} < H-1$, then $\tilde{t} = \hat{t}$ and from (7a)–(7b) and the fact that T is an L_∞-norm contraction mapping of modulus α, we have

$$\| x(0) - x^* \| \leqslant \alpha^H \| c - x^* \|, \qquad \| \tilde{x}(H-\tilde{t}) - x^* \| \leqslant \alpha^{\tilde{t}} \| c - x^* \|. \tag{8}$$

Also, by Lemma 1, $\tilde{\mu}$ is an optimal stationary policy for the ∞-horizon problem, so that $x^* = T^{\tilde{\mu}}(x^*)$. This, together with the observation that $\tilde{x}(0)$ equals $H - \tilde{T}$ successive applications of $T^{\tilde{\mu}}$ to $\tilde{x}(H-\tilde{t})$, implies $\| \tilde{x}(0) - x^* \| \leqslant \alpha^{H-\tilde{t}} \| \tilde{x}(H-\tilde{t}) - x^* \|$. By combining this with (8), we obtain

$$\| \tilde{x}(0) - x(0) \| \leqslant \| \tilde{x}(0) - x^* \| + \| x^* - x(0) \| \leqslant 2\alpha^H \| c - x^* \|.$$

Since $\| c - x^* \| \leqslant \| c \| + \| x^* \| \leqslant 2\delta^2$, this proves (b).

(c) If $\hat{t} \geqslant H-1$, then $\tilde{x}(0) = x(0)$ trivially. If $\hat{t} < H-1$, then $\tilde{t} = \hat{t}$ and, by Lemma 1, every stationary policy μ satisfying $x(H-\tilde{t}-1) = T^{\mu}(x(H-\tilde{t}))$ is optimal for the ∞-horizon problem. Since the ∞-horizon problem by assumption has a unique optimal stationary policy, it follows that $x(t) = T^{\tilde{\mu}}(x(t+1))$ for all $t = H-\tilde{t}-1, \ldots, 1, 0$. This, together with the observations that $\tilde{x}(H-\tilde{t}) = x(H-\tilde{t})$ and $\tilde{x}(0)$ equals $H-\tilde{t}$ successive applications of $T^{\tilde{\mu}}$ to $\tilde{x}(H-\tilde{t})$, implies $\tilde{x}(0) = x(0)$. □

Parts (a) and (c) of Proposition 1 imply that if the ∞-horizon problem has a unique optimal stationary policy, then $x(0)$ (and the corresponding optimal policy) can be computed exactly in $\mathrm{O}(\bar{m}\hat{t} + \min\{n^3 \log(H),\ H\Sigma_i \bar{m}_i\})$ arithmetic operations, which by (6) is at most

$$\mathrm{O}\left(\bar{m}n \log(n\delta)/(1-\alpha) + \min\left\{ n^3 \log(H),\ H \sum_i \bar{m}_i \right\} \right) \tag{9}$$

arithmetic operations. If α is bounded away from one, then this time is a polynomial in the input size. (The term $H\Sigma_i \bar{m}_i$, which is *not* a polynomial in the input size, has been included to reduce the solution time on sparse problems where $\Sigma_i \bar{m}_i$ is much smaller than n^3.) To verify in polynomial time the uniqueness assumption, we can first compute x^*, the fixed point of T. (This can be done either by using recursion (7a)–(7b) to identify an optimal stationary policy μ in time $\hat{t}$ (cf. Lemma 1) and then solving (5), or by solving the linear programming formulation of the ∞-horizon problem [1, p. 206].) Then we check to see if, for some i, there exist two distinct k and k' in D_i satisfying $x_i^* = T_i^k(x^*) = T_i^{k'}(x^*)$. This requires additional $\mathrm{O}(\bar{m})$ arithmetic operations.

If the ∞-horizon problem does not have a unique optimal stationary policy, then the optimal policy may oscillate with time (see Appendix C for an example; also see [4, p. 30]). In this case, it is not even known if the optimal policy has a polynomial-sized description. Nonetheless, we have the following ε-approximation algorithm for solving this problem:

ε-Approximation Algorithm ($\varepsilon > 0$). If $H \leqslant (\log(1/\varepsilon) + 2\log\delta + 2)/\log(1/\alpha)$, then run the Truncated DP Algorithm with $\tilde{t} = H-1$; otherwise run the Truncated DP Algorithm with $\tilde{t} = \min\{\hat{t},\ H-1\}$.

The complexity of this ε-approximation algorithm is given below:

Proposition 2. *For any $\varepsilon > 0$, the ε-Approximation Algorithm computes an $\tilde{x}(0)$ satisfying $\| \tilde{x}(0) - x(0) \| \leqslant \varepsilon$ in $\mathrm{O}(\bar{m}(\log(1/\varepsilon) + n\log(n\delta))/(1-\alpha) + \min\{n^3\log(H),\ H\Sigma_i\bar{m}_i\})$ arithmetic operations.*

Proof. If $H \leqslant (\log(1/\varepsilon) + 2\log\delta + 2)/\log(1/\alpha)$, then clearly the algorithm computes $x(0)$ and the time complexity is $\mathrm{O}(\bar{m}H)$, which is at most $\mathrm{O}(\bar{m}(\log(1/\varepsilon) + \log\delta)/(1-\alpha))$. Otherwise we have from part (b) of Proposition 1 that the algorithm computes an $\tilde{x}(0)$ satisfying $\| \tilde{x}(0) - x(0) \| \leqslant 4\alpha^H\delta^2 \leqslant \varepsilon$, where the second inequality follows from the hypothesis on H. The time complexity then follows from part (a) of Proposition 1 and expression (9). □

Notice that if α is bounded away from one, then the ε-approximation algorithm is, in the terminology of [6], a *fully polynomial approximation scheme*. In this special case, the H-horizon, discounted Markov decision problem remains P-hard [15], but is not known to be in NP.

4. Infinite horizon, undiscounted case

In this case $H = +\infty$ and $\alpha = 1$. This problem is of interest primarily when there is a cost-free state, say state 1, which is absorbing. The objective then is to reach this state at minimum expected cost (see [2, section 4.3.2]). More precisely, we say that a stationary policy μ is *proper* if every entry in the first column of $(P^{\mu})^t \to 1$ as $t \to +\infty$. We make the following assumption in addition to Assumption A:

Assumption B. $p_{11}^k = 1$ and $g_1^k = 0$ for all $k \in D_1$. Furthermore, all stationary policies are proper.

Assumption B essentially requires that all states other than state 1 be transient and that state 1 incurs zero cost. Under Assumption B, it can be shown (see Proposition 3.3 in [2, Section 4.3.2]) that an optimal stationary policy for this problem exists. Moreover, a stationary policy μ is optimal if and only if $x^* = T^{\mu}(x^*)$, where T is given by (1)–(2) with $\alpha = 1$ and x^* is the unique fixed point of T restricted to the subspace $X = \{x \in \mathbb{R}^n \mid x_1 = 0\}$.

An important fact is that T restricted to X is a contraction with respect to some *weighted* L_{∞}-norm, which then allows us to apply an argument similar to that used in Section 2. This fact, attributed to Hoffman, is discussed in [2, Section 4.3] (see Example 3.3 therein) and, in a more general context, in [17, Lemma 3]. We give a short proof of this fact below. The proof differs from previous ones in that it gives an explicit expression for the weights and the modulus of contraction.

Under Assumption B, $\{2,\dots,n\}$ can be partitioned into nonempty subsets $S_1,\dots,S_r$ such that

$$\max\left\{p_{ij}^k \mid j \in \{1\} \cup S_1 \cup \cdots \cup S_{s-1}\right\} > 0, \quad \forall k \in D_i, \quad \forall i \in S_s, \quad \forall s = 1,\dots,r$$

(see [2, p. 325]). Roughly speaking, this says that from any state in S_s, there is a positive probability of entering a state in $\{1\} \cup S_1 \cup \cdots \cup S_{s-1}$ in one transition, regardless of the control used. Define weights $\omega_2,\dots,\omega_n$ as follows:

$$\omega_i = 1 - \eta^{2s}, \quad \forall i \in S_s, \quad \forall s = 1,\dots,r, \tag{10}$$

where

$$\eta = \min\left\{p_{ij}^k \mid i,\ j,\ k \text{ such that } p_{ij}^k > 0\right\}.$$

We have the following lemma:

Lemma 3. *$\sum_{j \neq 1} p_{ij}^k \omega_j / \omega_i \leqslant \gamma$ and all $i \neq 1$, where*

$$\gamma = (1 - \eta^{2r-1})/(1 - \eta^{2r}). \tag{11}$$

Proof. Since $\eta \in (0, 1)$, we have from (10) that

$$0 < \omega_i < 1, \quad \forall i \neq 1. \tag{12}$$

Fix any $s \geqslant 1$, $i \in S_s$, and $k \in D_i$. Let j' be an element of $\{1\} \cup S_1 \cup \cdots \cup S_{s-1}$ such that $p_{ij'}^k > 0$. If $j' \neq 1$, then we have from (10) and (12) that

$$\begin{aligned}\left(\sum_{j \neq 1} p_{ij}^k \omega_j\right)/\omega_i &\leqslant \left(\sum_{j \neq j'} p_{ij}^k + p_{ij'}^k \omega_{j'}\right)/\omega_i \\ &= \left(1 + p_{ij'}^k(\omega_{j'} - 1)\right)/\omega_i \\ &\leqslant \left(1 + \eta(\omega_{j'} - 1)\right)/\omega_i \\ &\leqslant (1 - \eta^{2s-1})/\omega_i \\ &= (1 - \eta^{2s-1})/(1 - \eta^{2s}),\end{aligned}$$

where the second inequality follows from $p_{ij'}^k \geqslant \eta$, $\omega_{j'} - 1 < 0$ and the third inequality follows from the fact (cf. (10)) $\omega_{j'} \leqslant 1 - \eta^{2s-2}$. If $j' = 1$, then we have by a similar argument that

$$\Big(\sum_{j \neq 1} p_{ij}^k \omega_j \Big) / \omega_i \leqslant \Big(\sum_{j \neq 1} p_{ij}^k \Big) / \omega_i \leqslant (1-\eta)/\omega_i = (1-\eta)/(1-\eta^{2s}). \quad \square$$

Lemma 3 implies that (cf. (1)–(2)) for any $i \neq 1$, any $x = (x_1, \ldots, x_n) \in X$ and $y = (y_1, \ldots, y_n) \in X$,

$$\begin{aligned} T_i(x) - T_i(y) &\leqslant \sum_j p_{ij}^k (x_j - y_j) \\ &= \sum_{j \neq 1} \big(p_{ij}^k \omega_j \big)(x_j - y_j)/\omega_j \\ &\leqslant \gamma \omega_i \max_j \{ (x_j - y_j)/\omega_j \}, \end{aligned}$$

where k is an element of D_i for which $T_i(y) = T_i^k(y)$. Similarly we have

$$T_i(y) - T_i(x) \leqslant \gamma \omega_i \max_j \{ (y_j - x_j)/\omega_j \}.$$

Dividing both inequalities by ω_i then yields

$$\| T(x) - T(y) \|^{\omega} \leqslant \gamma \| x - y \|^{\omega}, \quad \forall x \in X, \quad \forall y \in X, \tag{13}$$

where $\| \cdot \|^{\omega}$ denotes the L_∞-norm scaled by $(1, \omega_2, \ldots, \omega_n)$, i.e. $\| x \|^{\omega} = \| (x_1, x_2/\omega_2, \ldots, x_n/\omega_n) \|$.

Since $\eta = z/\delta$ for some integer z and $r \leqslant n-1$, it can be seen from (10)–(11) that each $\log(\omega_i)$ and $\log(\gamma)$ is a polynomial in L and that $1 - \gamma \geqslant \eta^{2r}$. Also, since $g_1^k = 0$ for all $k \in D_1$, $T(X) \subseteq X$. Then by a contraction argument analogous to that used in the proof of Lemma 1, we obtain that an optimal stationary policy can be identified after a number of successive approximation iterations (i.e. eq. (3a)) that is bounded by a polynomial in η^{-2r} and L. Notice that we do not need to know the S_s's in order to compute this bound; it suffices to know η and an upper bound on r. On the other hand, if a tight upper bound on r is not available, then we can compute the S_s's by using, say, a labeling algorithm similar to Dijkstra's shortest path algorithm, whereby at the s-th iteration all $i \in S_s$ are labeled. The time complexity of this labeling algorithm can be shown to be $O(r\Sigma_i m_i + \overline{m})$ (see Appendix D). In the special case where $m_i = 1$ for all i, the time for computing the S_s's can be further reduced to $O(\overline{m})$ (see Fox [5]).

We remark that, by refining the analysis given in Example 3.3 of [2, Section 4.3], we can obtain a set of weights, different from those given by (10), whose corresponding modulus of contraction is $1 - \eta^{r+1}$, an improvement of that given by (11). However, the analysis of this is rather intricate and, for brevity, is omitted.

Alternatively, it can be seen that T restricted to X is an r-stage contraction with respect to the ordinary L_∞-norm, and that the modulus of contraction estimated by $1 - \min_{i \neq 1, \mu_1, \ldots, \mu_r} [P^{\mu_1} \cdots P^{\mu_r}]_{i1}$. This estimate is difficult to compute in general, but it can be upper bounded by $1 - \eta^r$. To see this, note from the definition of the S_s's that from any state $i \in S_s$, there is a positive probability (which is lower bounded by η) of entering a state $j \in \{1\} \cup S_1 \cup \cdots \cup S_{s-1}$ in one transition, regardless of the control used. By applying the same argument to j and so forth, we find that the probability of reaching state 1 from state i within s transitions is at least η^s, regardless of the control used at each transition. Hence $[P^{\mu_1} \cdots P^{\mu_r}]_{i1}$ is lower bounded by η^r for any i and any sequence of controls $\mu_1, \ldots, \mu_r$.

5. Finite horizon, undiscounted case

In this case $H < +\infty$ and $\alpha = 1$. Under Assumption B (in addition to Assumption A) and by combining the arguments in Section 4 with arguments analogous to those made in Sections 2 and 3, we can find an ε-approximate solution problem in time that is a polynomial in $\log(1/\varepsilon)$, $1/\eta^r$, L and $\log(H)$, when η and r are defined as in Lemma 3. If the ∞-horizon problem has a unique optimal stationary policy, then we can find an exact solution in time that is a polynomial in $1/\eta^r$, L and $\log(H)$. These times are unfortunately very slow even for moderately large values of r and $1/\eta$. In a recent work [5], Fox has suggested *simulation* as an effective approach for solving the special case where $m_i = 1$ for all i. Such an approach perhaps can be extended to solve the general case (or at least to solve problems with small control sets).

6. Conclusion and extensions

In this article we have shown that an ϵ-approximate solution of the H-horizon Markov decision problem with $H < +\infty$ is computable in time proportional to $\log(1/\varepsilon)$ and $\log(H)$ and, under an additional stability assumption, an exact solution is computable in time proportional to $\log(H)$. For the discounted case where the discount factor is bounded away from 1, we obtain respectively, a fully polynomial approximation scheme and a polynomial-time algorithm. However, in view of the stability assumptions needed to obtain an exact solution and the absence of negative results, we are still far from a complete complexity theory for this problem. If the stability assumptions are removed, the example in Appendix C shows that we must consider policies that have a certain *periodic* property. Optimal policies having such a periodic property remain poorly understood.

Appendix A

We briefly explain below the complexity terms PSPACE, NP, P, P-hard, P-complete and NC. (see [15] for a more detailed explanation. Also see [3,6,10,14,16] for comprehensive discussions.)

PSPACE is the class of problems that can be solved using polynomial space.

NP is the class of problems that can be solved *nondeterministically* in polynomial time (e.g. independent set, Hamilton circuit problem, integer program).

P is the class of problems that can be solved in polynomial time (e.g. linear program).

A problem is P-hard if any problem in P is reducible to it using logarithmic space.

A problem is P-complete if it is both P-hard and in P.

NC is the class of problems that can be solved in parallel using a polynomial number of processors in time that is a polynomial in the logarithm of the input size.

The following hierarchy (in order of increasing difficulty) for the above problem classes are known to hold: $\mathrm{NC} \subseteq \mathrm{P} \subseteq \mathrm{NP} \subseteq \mathrm{PSPACE}$ and P-complete $\subseteq$ P. Notice that if any P-hard problem is shown to be in NC, then P = NC.

Appendix B

Let A be any $n \times n$ matrix. We show below that, for any integer $k \geqslant 1$, we can compute A^k and $I + A + \cdots + A^k$ in $O(\log(k))$ matrix multiplications. First suppose that k is a power of 2. Then, by using the recursive equations

$$A^k = (A^{k/2})(A^{k/2}),$$

$$I + A + \cdots + A^k = (I + A + \cdots + A^{k/2}) + A^{k/2}(I + A + \cdots + A^{k/2}),$$

we see that if $A^{k/2}$ and $I + A + \cdots + A^{k/2}$ are computable in $3\log(\frac{1}{2}k)$ matrix multiplications, then A^k and $I + A \cdots + A^k$ are computable in $3\log(\frac{1}{2}k) + 3 = 3\log(k)$ matrix multiplications. Hence, by induction, we can compute A^d and $I + A + \cdots + A^d$ for *all* $d = 2^0, 2^1, 2^2, \ldots, k$ in $3\log(k)$ matrix multiplications. Now suppose that k is not a power of 2. Let us first compute and store the matrices A^d and $I + A + \cdots + A^{d-1} + A^d$, for all $d = 2^0, 2^1, 2^2, \ldots, 2^h$, where $h = \lfloor \log(k) \rfloor$. (This takes $3h$ matrix multiplications as we argued above.) We claim that, given the above matrices, A^i and $I+A+\cdots+A^i$ are computable in $3\lceil\log(i)\rceil$ matrix multiplications for any positive integer $i \leqslant k$. This claim clearly holds for $i=1$. Suppose that it holds for all i up to (but not including) some $r \in \{2, 3, \ldots, k\}$. Then by first computing A^{r-d} and $I+A+\cdots+A^{r-d}$, where d is the largest power of 2 less than r, and then using the identities

$$A^r = (A^d)(A^{r-d}),$$
$$I+A+\cdots+A^r = (I+A+\cdots+A^d) + A^d(I+A+\cdots+A^{r-d}),$$

we can compute A^r and $I+A+\cdots+A^r$ in $3\lceil\log(r-d)\rceil + 3$ matrix multiplications. Since $r-d \leqslant d$, this bound is less than $3\log(d)+3 = 3\log(2d)$. Since d is the largest power of 2 less than r, we have $d < r \leqslant 2d$ so that $\lceil\log(r)\rceil = \log(2d)$. This then completes the induction.

Appendix C

Consider the following H-horizon, discounted Markov decision problem:

$$\alpha = 0.5, \quad g^1 = (0,\ 0), \quad g^2 = (0,\ 0), \quad H < +\infty,$$
$$P^1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad P^2 = \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix}.$$

Then

$$T(x) = 0.5\begin{pmatrix} \min\{x_2,\ 0.5(x_1+x_2)\} \\ \min\{x_1,\ 0.5(x_1+x_2)\} \end{pmatrix},$$

and $T_1(x) < T_2(x)$ if $x_1 > x_2$ while $T_1(x) > T_2(x)$ if $x_1 < x_2$. Therefore if $x_1(H) \neq x_2(H)$, then the optimal control at time t would *alternate* between (1, 2) and (2, 1), depending on whether t is odd or even. If $x_1(H) = x_2(H)$, then any sequence of controls is optimal. Note that, for this example, any of the four stationary policies is optimal for the ∞-horizon version of the problem.

Appendix D

Below we describe an $O(r\Sigma_i m_i + \overline{m})$ time labeling algorithm for computing the partition $S_1, \ldots, S_r$ discussed in Section 4. The algorithm proceeds as follows:

Labeling Algorithm

Iteration 0. Set $S_0 \leftarrow \{1\}$ and $\pi_i^k \leftarrow p_{i1}^k$ for all $k \in D_i$ and all $i \neq 1$.

Iteration s. We are given $S_0, \ldots, S_{s-1}$ and $\pi_i^k = \Sigma_{j \in T_{s-1}} p_{ij}^k$ for all $k \in D_i$ and all $i \notin T_{s-1}$, where $T_{s-1} = S_0 \cup \cdots \cup S_{s-1}$. If $T_{s-1} = S$, then we stop. Otherwise, for each $i \notin T_{s-1}$, if $\min_{k \in D_i} \pi_i^k > 0$, then add i to S_s, and for all $j \notin T_{s-1}$ and all $k \in D_j$ such that $p_{ji}^k > 0$, update $\pi_j^k \leftarrow \pi_j^k + p_{ji}^k$.

The above algorithm terminates after $r+1$ iterations and, at the s-th iteration $(1 \leqslant s \leqslant r)$, takes $O(\Sigma_{i \notin T_{s-1}} m_i)$ time to check if $\min_{k \in D_i} \pi_i^k > 0$ for every $i \notin T_{s-1}$ and an additional $O(\Sigma_{i \in S_s} \hat{m}_i)$ time to update the π_j^k's, where $\hat{m}_i =$ number of positive p_{ji}^k's. Hence, the total time for the algorithm is $O(\Sigma_{s=1}^r \Sigma_{i \notin T_{s-1}} m_i + \Sigma_{s=1}^r \Sigma_{i \in S_s} \hat{m}_i)$ which is at most $O(r\Sigma_i m_i + \overline{m})$. (To be precise, one would need to specify the data structure used, but this is not difficult.)

Acknowledgement

We gratefully acknowledge the many helpful comments made by Professors B.L. Fox and J.N. Tsitsiklis.

This research is partially supported by the U.S. Army Research Office, contract DAAL-3-86-K-0171 (Center for Intelligent Control Systems), and by the National Science Foundation under grant NSF-ECS-8519058.

References

[1] D.P. Bertsekas, *Dynamic Programming: Deterministic and Stochastic Models*, Prentice-Hall, Englewood Cliffs, NJ, 1987.

[2] D.P. Bertsekas and J.N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Prentice-Hall, Englewood Cliffs, NJ, 1989.

[3] S.A. Cook, "Towards a complexity theory of synchronous parallel computation", *Enseign. Math.*. **2**/27, 99–124 (1981).

[4] A. Federgruen and P.J. Schweitzer, "Discounted and undiscounted value-iteration in Markov decision problems: A survey", in: M.L. Puterman (ed.), *Dynamic Programming and Its Applications*, Academic Press, New York, 1978, 23–52.

[5] B.L. Fox, "Computing the gradient of expected reward up to absorption: Deterministic versus simulation methods", Technical Report, University of Colorado, Denver, CO, July, 1989.

[6] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, San Francisco, CA, 1979.

[7] A.S. Householder, *The Theory of Matrices in Numerical Analysis*, Dover Publications, New York, 1964.

[8] R.A. Howard, *Dynamic Programming and Markov Processes*, MIT Press, Cambridge, MA, 1960.

[9] N. Karmarkar, "A new polynomial-time algorithm for linear programming", *Combinatorica* **4**, 373–395 (1984).

[10] H.R. Lewis and C.H. Papadimitriou, *Elements of the Theory of Computation*, Prentice-Hall, Englewood Cliffs, NJ, 1981.

[11] N. Megiddo (ed.), *Progress in Mathematical programming: Interior-Point and Related Methods*, Springer-Verlag, New York, 1989.

[12] J. Orlin, "The complexity of dynamic languages and dynamic optimization problems", *Proc. 13th STOC*, 218–227 (1981).

[13] C.H. Papadimitriou, "Games against nature", *J. Comput. System Sci.* **31**, 288–301 (1985).

[14] C.H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall, Englewood Cliffs, NJ, 1982.

[15] C.H. Papadimitriou and J.N. Tsitsiklis, "The complexity of Markov decision processes", *Math. Oper. Res.* **12**, 441–450 (1987).

[16] I. Parberry, *Parallel Complexity Theory*, Pitman, London, 1987.

[17] A.F. Veinott, Jr., "Discrete dynamic programming with sensitive discount optimality criterion", *Ann. Math. Stat.* **40**, 1635–1660 (1969).

[18] D.J. White, "Elimination of nonoptimal actions in Markov decision processes", in: M.L. Puterman (ed.), *Dynamic Programming and Its Applications*, Academic Press, New York, 1978, 131–160.