The solution of an undiscounted completely ergodic Markov decision process by successive approximation

EINDHOVEN UNIVERSITY OF TECHNOLOGY

Department of Mathematics

STATISTICS AND OPERATIONS RESEARCH GROUP

Memorandum COSOR 74-05

The solution of an undiscounted completely ergodic Markov decision process by successive approximation

by

J. van der Wal

Eindhoven, March 1974

Abstract

In this paper we consider a completely ergodic Markov decision process with finite state and decision spaces using the average return per unit time criterion. An algorithm is derived which approximates the optimal solution. It will be shown that this algorithm is finite and supplies upper and lower bounds for the maximal average return and a near optimal policy with average return between these bounds.

1. Introduction and notations

We will consider a system which at any time $t = 1,2,\dots$ is in one of the states $1,2,\dots,N$. In each state i there is a finite set K_i of actions which may be chosen. If in state i action $u_i \in K_i$ is selected we receive the expected immediate return $q(u_i)$. For each $j \in S := \{1,2,\dots,N\}$ $[p(u_i)]_j$ is the probability of making a transition to state j if i is the current state and action u_i has been chosen. With $p(u_i)$ we denote the row-vector $([p(u_i)]_1,\dots,[p(u_i)]_N)$. A vector $u \in K := K_1 x \,..\, xK_N$ will be called a policy. A policy prescribes for each state which action will have to be selected. If $u = (u_1,\dots,u_N)$ then $q(u)$ denotes the column-vector $(q(u_1),\dots,q(u_N))^T$ and $P(u)$ is the transition probability matrix with $[P(u)]_{ij} = [p(u_i)]_j$.

We assume that for each $u \in K$ $P(u)$ is completely ergodic (i.e. the Markov chain associated to u has a single aperiodic recurrent class and no transient states).

Moreover $g(u)$ and $v(u)$ will be the gain (average return per unit time) and the vector of relative values (with N-th component zero) belonging to policy u (see R.A. HOWARD [2]). If $p \in \mathbb{R}^N$ we will write:

$[p]_j$ for the j-th component of p ,

$\overline{p}, \underline{p}$ for the largest respectively smallest component of p, and

Δp for the difference $\overline{p} - \underline{p}$.

In section 2 we will derive our algorithm from the "policy iteration" algorithm of R.A. HOWARD. We will prove that our algorithm produces upper and lower bounds for the maximal gain and near optimal policies.

In section 3 we demonstrate that it might be possible to prove the same for the ergodic case (i.e. the Markov chain associated to u has a single aperiodic recurrent class and might have one or more transient states).

2. The algorithm

Since our algorithm has been derived from the "Policy Iteration Algorithm" of R.A. HOWARD [2] we rewrite his algorithm below in our notation:

Policy Iteration Algorithm

STEP 0 Select an initial policy $u = (u_1,\dots,u_N)$;

STEP 1 (Value-Determination Operation)

Solve the system

$$\begin{cases} g.e + v = q(u) + P(u)v & (e = (1,\dots,1)^T) \\ [v]_N = 0 \end{cases}$$

STEP 2 (Policy-Improvement Routine)

Find for all $i \in S$ an action $w_i \in K_i$ which maximizes $q(w_i) + p(w_i)v$.
If for some $i \in S$ $q(w_i) + p(w_i)v \neq q(u_i) + p(u_i)v$ then for all $i \in S$ $u_i := w_i$ and go to STEP 1.

STOP The policy u is optimal and g is the maximal average return per unit time.

We will change this algorithm in the following way.
Instead of solving the system in STEP 1 we will approximate the values of v and g. A.R. ODONI [4] computes after any execution of the "Policy-Improvement Routine" a new value v but he does not try to improve this approximation. However, before running through the Policy Improvement Routine once more, we improve the approximation of v until we know the gain fairly accurate.
A similar procedure has been suggested by SPREMANN and GESSNER [5]. Their algorithm however, does not produce upper and lower bounds. These authors suggest an other modification which we use as well. During the first iterations we will not look for a better action if in a state the limit probability is small, does not exceed δ.
As suggested in [5] we take for δ_j the sequence $\frac{1}{2}$, $\frac{1}{N}$, 0, 0, ... Applying these modifications we produce the following algorithm:

STEP 0 Select an initial policy u, select $\alpha > 0$ and a monotone non-increasing sequence $\varepsilon_0, \varepsilon_1, \ldots$ with $\varepsilon_j > 0$ for all j and $\lim_{j\to\infty} \varepsilon_j = 0$. For $i \in S$ $[\pi]_i := \frac{1}{N}$, $[v]_i := 0$; $\delta := \frac{1}{2}$; $j := 0$; eps $:= \varepsilon_0$.

STEP 1 $\pi := P^T(u)\pi$; $\pi := P^T(u)\pi$.

STEP 2 While $\Delta(q(u) + P(u)v-v) >$ eps do
$v := q(u) + P(u)v - [q(u) + P(u)v]_N\, e$

STEP 3 Find for all $i \in S$ for which $[\pi]_i \geq \delta$ an action $w_i \in K_i$ which maximizes $q(w_i) + p(w_i)v$.
If $\delta = 0$ and for all $i \in S$
$q(w_i) + p(w_i)v < q(u_i) + p(u_i)v + \alpha$ go to STOP else
if $[\pi]_i \geq \delta$ then $u_i := w_i$; $j := j+1$; eps $:= \varepsilon_j$

STEP 4 If $\delta = \frac{1}{2}$ and $N > 2$ $\delta := \frac{1}{N}$; go to STEP 1
else $\delta := 0$ go to STEP 2

STOP u is near optimal. Let u^* be optimal then we have:

(i) $g(u^*) \leq g(u) + \alpha +$ eps

(ii) $\underline{q(u) + P(u)v-v} \leq g(u) \leq \underline{q(u) + P(u)v-v} +$ eps

(iii) $\underline{q(u) + P(u)v-v} \leq g(u^*) \leq \underline{q(u) + P(u)v-v} + 2$ eps $+ \alpha$.

Remark. The introduction of a $\alpha > 0$ is necessary to prevent cycling if there exists more than one optimal policy.

To prove the finiteness of our algorithm and the correctness of the estimations at STOP, we will show first that the number of successive iterations within STEP 2 is finite and that the value of v in STEP 2 converges to the vector of relative values for the actual policy.

Suppose we arrive at STEP 2 with a policy u and and initial approximation $v_0(u)$ of $v(u)$.
Now define:

(1) $$v_i(u) = q(u) + P(u)v_{i-1}(u) - [q(u) + P(u)v_{i-1}(u)]_N \cdot e \,, \qquad (i = 1,2,\dots)$$

(2) $$g_i(u) = q(u) + P(u)v_{i-1}(u) - v_{i-1}(u) \qquad (i = 1,2,\dots) \,.$$

Obviously $v_i(u)$ is the approximation of $v(u)$ that would be found after improving the approximation $v_0(u)$ i times within STEP 2.
The test for transition from STEP 2 to STEP 3 is the examination whether or not

(3) $\Delta g_i(u) \le eps$ holds.

Substitution of (1) in (2) yields $g_{i+1}(u) = P(u)g_i(u)$, $i = 1,2,\dots$.
Hence

(4) $$g_{\ell+1}(u) = P^{\ell}(u)g_1(u), \qquad \ell = 0,1,\dots \,.$$

Since $P^{\infty}(u) := \lim_{r\to\infty} P^r(u)$ exists and has identical rows there exists a number $g^*(u)$ so that

(5) $$\lim_{i\to\infty} g_i(u) = g^*(u)\cdot e \,.$$

For any policy $u \in K$ there exist b and ρ $(0 \le \rho < 1)$ such that (see [1])

(6) $$\forall_{j,k\in S} \; |[P^r(u) - P^{\infty}(u)]_{jk}| \le b\rho^r \,.$$

Hence we have for all $j,k \in S$ and $x \in R^N$

(7) $$|[P^r(u)x]_j - [P^r(u)x]_k| \le \Delta(P^r(u)x) = \Delta(P^r(u)(x-\underline{x}\cdot e)) \le 2bN\rho^r\Delta x \,.$$

Now we can formulate:

Lemma 1. For any $u \in K$ and for any initial approximation $v_0(u)$ of $v(u)$ STEP 2 is finite.

Proof. From (7) we have

$$\Delta g_{r+1}(u) \le 2bN\rho^r \Delta g_1(u) = 2bN\rho^r \Delta(q(u) + P(u)v_0(u) - v_0(u)) \,.$$

Repeated application of (1) yields

$$(8) \qquad v_i(u) = \{I + P(u) + \dots + P^{i-1}(u)\}q(u) + P^i(u)v_0(u) +$$

$$- [\{I + P(u) + \dots + P^{i-1}(u)\}q(u) + P^i(u)v_0(u)]_N \cdot e \ .$$

By arranging the terms in (8) in pairs, $q(u)$ and $[q(u)]_N \cdot e$ and so on, and using (7) $\ell+1$ times we get (since $[v_{i+\ell}(u) - v_i(u)]_N = 0$) :

$$(9) \qquad |[v_{i+\ell}(u) - v_i(u)]_j| \le 2bN \ \{\Delta q(u)(\rho^i + \rho^{i+1} + \dots + \rho^{i+\ell-1}) +$$

$$+ \Delta v_0(u)(\rho^i + \rho^{i+\ell})\}.$$

Now (9) implies that the sequence $v_0(u), v_1(u), \dots$ converges.
Let $v^*(u) := \lim_{\ell\to\infty} v_\ell(u^k)$ then we can formulate

Lemma 2. The limits $v^*(u)$ and $g^*(u)$ are just the vector of relative values $v(u)$ and the gain $g(u)$ belonging to u.

Proof. $v^*(u)$, $g^*(u)$ and $v(u)$, $g(u)$ both solve the system

$$\begin{cases} g \cdot e + v = q(u) + P(u)v \\ \quad [v]_N = 0 \end{cases}$$

which possesses a unique solution.

Let now $u^0, u^1, \dots$ be the succession of policies determined by our algorithm, u^0 the selected initial policy, and the approximation in STEP 2 of $v(u^k)$, with initial value $v_0(u^k)$, require n_k iterations.
Now define

$$(10) \qquad \begin{cases} v_0(u^0) = 0 \\ v_0(u^k) = v_{n_{k-1}}(u^{k-1}), \ k = 1,2,\dots \quad \text{and} \\ v_i(u^k), \ g_i(u^k), \ i = 1,2,\dots; \ k = 0,1,\dots \ \text{according to (1) and (2).} \end{cases}$$

If the algorithm did not terminate after completing STEP 3, while we have already $\delta = 0$, then a policy u^k has just been improved to a policy u^{k+1}. We have

$$q(u^{k+1}) + P(u^{k+1})v_0(u^{k+1}) = q(u^k) + P(u^k)v_0(u^{k+1}) + d_{k+1} ,$$

where

$$d_{k+1} \geq 0, \ \exists_{j \in S} \ [d_{k+1}]_j \geq \alpha .$$

Hence

$$g_1(u^{k+1}) = g_{n_k+1}(u^k) + d_{k+1} ,$$

which implies

$$g(u^{k+1}) \geq \underline{g_{n_k+1}}(u^k) + [P^\infty(u^{k+1})d_{k+1}]_1 .$$

Let β be the smallest element of all $P^\infty(u)$, $u \in K$. Since all $P(u)$ are completely ergodic we have $\beta > 0$. Defining $\gamma := \alpha\beta$ we have

$$g(u^{k+1}) \geq \underline{g_{n_k+1}}(u^k) + \gamma \geq g(u^k) + \gamma - \varepsilon_k .$$

Now we have

<u>Lemma 3</u>. If k sufficiently large (so that $\varepsilon_k < \gamma$) then a once improved policy u^k cannot be found again.

<u>Proof</u>. For all $p \geq k$ $g(u^{p+1}) > g(u^p)$ so $g(u^p) > g(u^k)$ for all $p > k$.

<u>Lemma 4</u>. If for each $u \in K$ the Markov chain with matrix $P(u)$ is completely ergodic then the algorithm is finite.

<u>Proof</u>. From Lemma 1, Lemma 3 and the existence of only a finite number of policies.

If u^* is the optimal policy and the algorithm terminates with a policy u^k then we have

(11) $$g_1(u^*) < g_{n_k+1}(u^k) + \alpha \cdot e,$$

and

(12) $$\Delta g_{n_k+1}(u^k) \leq \varepsilon_k \ .$$

So we have

<u>Lemma 5</u>. If the algorithm terminates with a policy u^k, while u^* is an optimal policy, then

(i) $$g(u^*) - g(u^k) < \alpha + \varepsilon_k$$

(ii) $$\underline{g_{n_k+1}}(u^k) \leq g(u^k) \leq \underline{g_{n_k+1}}(u^k) + \varepsilon_k$$

<u>Proof</u>. From (11) and (12) with

$$\underline{g_{n_k}}(u^k) \leq \underline{g_{n_k+1}}(u^k) \leq g(u^k) \leq \overline{g_{n_k+1}}(u^k) \leq \overline{g_{n_k}}(u^k) \ .$$

<u>Theorem 1</u>. If for all $u \in K$ the Markov chain with matrix $P(u)$ is completely ergodic then the algorithm is finite. For the approximation u^k for u^* the following estimates hold:

(i) $g(u^*) - g(u^k) < \alpha + \varepsilon_k$

(ii) $\underline{g_{n_k+1}}(u^k) \leq g(u^k) \leq \underline{g_{n_k+1}}(u^k) + \varepsilon_k$

(iii) $\underline{g_{n_k+1}}(u^k) \leq g(u^*) \leq \underline{g_{n_k+1}}(u^k) + 2\varepsilon_k + \alpha.$

<u>Proof</u>. The finiteness follows by Lemma 4; (i), (ii) by Lemma 5, (i) and (ii) imply (iii).

<u>Remark 2</u>. It is possible to prevent termination of the algorithm while ε_k is still large, e.g. $\varepsilon_k > \alpha$.

3. The ergodic case

In the foregoing we proved our algorithm to be finite for completely ergodic decision processes. We believe that the algorithm is also finite if for each policy in the transition probability matrix P(u) is ergodic (which means that for each policy u the set of states is divided into a set of transient states and one aperiodic recurrent class). It might however be necessary to modify STEP 3 of the algorithm, i.e. to put

"if $[\pi]_i \geq \delta$ then if $q(w_i) + p(w_i)v \geq q(u_i) + p(u_i)v + \alpha$, $u_i := w_i$"

instead of

"if $[\pi]_i \geq \delta$ then $u_i := w_i$" .

This modification enabled us to prove finiteness in the case that for each policy the recurrent class consists of the same N-1 states.

Lemma 6. If P(u) is ergodic then the system

$$\begin{cases} g.e + v = P(u)v + q(u) \\ [v]_N = 0 \end{cases}$$

possesses a unique solution (in g and v).

Proof. The rank of I - P(u) is N-1 (see [3]) and if v,g solve (I-P(u))v = = q(u) - g.e then v + α.e, g as well. So the rank of the system in N.

Let for all $u \in K$ P(u) be ergodic and the recurrent class consist of the same N-1 states then we have the following lemma's.

Lemma 7. If k is sufficiently large and the policy u^k is improved in one of the recurrent states then the algorithm will not generate u^k once more.

Proof. It is obvious that the modification of STEP 3 does not influence any of the proofs in the preceding section. Now let j be the transient state and $S^* := S\backslash\{j\}$. Analogously to section 2 we have

$$g(u^{k+1}) \geq \min_{i \in S^*} [g_{n_k+1}(u^k)]_i + \gamma' \geq g(u^k) + \gamma' - \varepsilon_k$$

with $\gamma' = \alpha\beta'$ where β' is the smallest element of all $P^\infty(u)$ not belonging to the j-th row or column. Again we have $\gamma' > 0$ so if k sufficiently large we have $g(u^p) > g(u^k)$ for all $p > k$.

Lemma 8. A policy $u \in K$ can be improved but a finite number of times in succession in the transient state only.

Proof. Let state j ($j \neq N$) be the only transient state. From (1) and (2) we have

$$g_{i+1}(u^k) = v_{i+1}(u^k) - v_i(u^k) + [q(u^k) + P(u^k)v_i(u^k)]_N . e$$

and therefore

$$g_{i+1}(u^k) - [g_{i+1}(u^k)]_N . e = v_{i+1}(u^k) - v_i(u^k) \;. \qquad (*)$$

If now $\alpha > \varepsilon_k$ and u^k is improved in the transient state only we have

$$[g_1(u^{k+1})]_j - [g_1(u^{k+1})]_N > 0 \;.$$

Hence according to (*)

$$[v_1(u^{k+1})]_j - [v_0(u^{k+1})]_j > 0 \;.$$

While $\Delta g_i(u^{k+1}) > \varepsilon_k$ we have

$$[g_i(u^{k+1})]_j - [g_i(u^{k+1})]_N > 0 \;.$$

Hence

$$[v_i(u^{k+1})]_j - [v_{i-1}(u^{k+1})]_j > 0 \;.$$

Let ℓ be the last integer for which $\Delta g_\ell(u^{k+1}) \geq \varepsilon_k$ then we have

$$[v_\ell(u^{k+1})]_j - [v_0(u^{k+1})]_j \geq \alpha - \varepsilon_k \;.$$

The approximation of $g(u^{k+1})$ and $v(u^{k+1})$ proceeds until $\Delta g(u^{k+1}) \leq \varepsilon_{k+1}$. From lemma 1 we have $\Delta g_{\ell+i}(u) \leq 2bN\rho^{i}\Delta g_{\ell}(u)$.
So these iterations result in a decrease of $[v_i(u^{k+1})]_j$ of at most

$$\Delta g_{\ell+1}(u^{k+1}) + \Delta g_{\ell+2}(u^{k+1}) + \ldots \leq \Delta g_{\ell+1}(u^{k+1}) + \frac{2bN}{1-\rho} \Delta g_{\ell+1}(u^{k+1}) \leq (1 + \frac{2bN}{1-\rho})\varepsilon_k.$$

So we have

$$[v_{n_{k+1}}(u^{k+1})]_j - [v_0(u^{k+1})]_j = [v_0(u^{k+2})]_j - [v_0(u^{k+1})]_j \geq \alpha - (2 + \frac{2bN}{1-\rho})\varepsilon_k \geq \lambda > 0$$

if k sufficiently large, say $k \geq k_0$.
Since $v(u)$ is uniformly bounded for $u \in K$ a policy u^k, $k \geq k_0$, can be improved but a finite number of times in the transient state only.
If N is the transient state and $k \geq k_0$ then each improvement in state N only results in a decrease of at least λ for the components $[v(u^k)]_j$, $j \in S\backslash\{N\}$.

From Lemma's 6, 7 and 8 we now conclude:

Theorem 2. If for each policy u the Markov chain with matrix $P(u)$ is ergodic and the ergodic class consists of the same N-1 states then the modified algorithm is finite.

References

[1] J.L. Doob, Stochastic Processes. John Wiley & Sons, New York 1953, p. 173.

[2] R.A. Howard, Dynamic Programming and Markov Processes, Cambridge M.I.T. press, 5th printing 1969, p. 32-43.

[3] H. Mine en S. Osaki, Markovian decision processes, American Elsevier, New York 1970, pp. 25-26.

[4] A.R. Odoni, On finding the gain for Markov decision processes, O.R. 17 (1969), pp. 857-860.

[5] K. Spremann und P. Gessner, Bewerteter Markovprozesse im stationären Zustand. Ein neuer Algorithmus mit Beispiel. Discussion paper nr. 18 des Institutus für Wirtschaftstheorie und Operations Research der Universität Karlsruhe, Juli 1973.