Improved successive approximation methods for discounted Markov decision processes
van Nunen, J.A.E.E.

Published: 01/01/1974

Document Version
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:
• A submitted manuscript is the author's version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

Citation for published version (APA):

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 07. Jan. 2019
EINDHOVEN UNIVERSITY OF TECHNOLOGY

Department of Mathematics

STATISTICS AND OPERATION RESEARCH GROUP

Memorandum COSOR 74-06

Improved successive approximation methods for discounted Markov decision processes

by

J.A.E.E. van Nunen

Eindhoven, April 1974
Abstract

Successive Approximation (S.A.) methods, for solving discounted Markov decision problems, have been developed to avoid the extensive computations that are connected with linear programming and policy iteration techniques for solving large scaled problems. Several authors give such an S.A. algorithm.

In this paper we introduce some new algorithms while furthermore it will be shown how the several S.A. algorithms may be combined. For each algorithm converging sequences of upper and lower bounds for the optimal value will be given.

§ 1. Introduction.

We consider a finite state, discrete time Markov system that is controlled by a decisionmaker (see for example [4]). After each transition \( n = 0,1,2, \ldots \) the system may be identified as being in one of \( N \) possible states. Let \( S := \{1,2,\ldots,N\} \) represent the set of states. After observing state \( i \in S \) the decisionmaker selects an action \( k \in K(i) \) from a nonempty finite set \( K(i) \).

Now \( p_{ij}^k \) is the probability of a transition to state \( j \in S \), if the system is actually in state \( i \in S \), and action \( k \in K(i) \) has been selected. An (expected) reward \( q_k^i \) is then earned immediately, while future income is discounted by a constant factor \( 0 \leq \alpha < 1 \).

We suppose, which is permitted without loss of generality, that \( q_k^i \geq 0 \) for all \( i \in S \) and \( k \in K(i) \).

The problem is to choose a policy which maximizes the total expected discounted return.

As known (e.g. [2], [10]), it is permitted to restrict the considerations to nonrandomized stationary strategies. A nonrandomized stationary strategy will be denoted by \( f \in K := K(1) \times K(2) \times \ldots \times K(N) \). The coordinates \( u_f(i) \) of the \( N \times 1 \) vector \( u_f \) give the total expected discounted return if the system's initial state is \( i \) and the stationary strategy \( f \in K \) is used. The (stationary) strategy \( f^* \in K \) is called optimal if \( u_{f^*}(i) \geq u_f(i) \) for all \( f \in K \) and for all \( i \in S \).
Because S.A. algorithms are in some sense modifications of the standard
dynamic programming method this method will be discussed first.

As in Blackwell [1] we define for each $f \in K$ the mapping $L_b(f) : \mathbb{R}^N \to \mathbb{R}^N$ which maps an $N \times 1$ column vector $x$ into

$$L_b(f)x := q_f + \alpha P_f x,$$

where $q_f$ is the $N \times 1$ column vector having as its $i$-th component $f(i)(i)$, and $P_f$ is the $N \times N$ Markov matrix with $(i,j)$ element $p_{ij}^f$. $L_b(f)$ is monotone, i.e. if every coordinate of the $N \times 1$ vector $x$ is at least as large as every coordinate of $y \in \mathbb{R}^N (x \geq y)$, then:

$$L_b(f)x \geq L_b(f)y.$$

Furthermore, we define for some map $L_B(f)$:

$$L_B^0(f)x := x;$$

$$L_B^n(f)x = L_B(f)(L_B^{n-1}(f)x).$$

We define the mapping $U_b : \mathbb{R}^N \to \mathbb{R}^N$ by:

$$U_b x := \max_{f \in K} L_b(f)x.$$

It is easily seen that for every $x \in \mathbb{R}^N$ an $f \in K$ exists such that $L_b(f)x$ is maximal for each coordinate.

It may be proved that $U_b$ is a monotone $\alpha$-contraction mapping with fixed point $u^*$. For an optimal strategy $f^*$ we have $u^* = u_{f^*}$, and a stationary strategy $f$ which is optimal follows from $L_b(f)u^* := U_b u^*$ (see for instance [9]).

This property legitimates the standard dynamic programming algorithm that can be based on:

$$\begin{cases}
    x^b_0 := 0 \\
    x^b_n := U_b x^b_{n-1} =: L_b(f^b)x^b_{n-1}.
\end{cases}$$

It is possible to take $x^b_n$ and $f^b_n$ as estimates for $u^*_{f^*}$ and $f^*$ as follows from:
As starting vector we choose \( x_0 = 0 \). As appears from (1) and (2), this choice guarantees monotone convergence of \( x_n \) to \( u_{f^*} \).

As known, the convergence, depending on \( \alpha \), may be relatively slow. MacQueen [5] constructed upper and more sophisticated lower bounds for \( u_{f^*} \).

The S.A. methods discussed in the following sections are based on contraction mappings (see Denardo [2]).

In section 2, S.A. methods based on mappings \( U_0, U_a, U_s \) of the same type as \( U_b \) will be given; i.e. also these mappings are monotone contraction mappings with fixed point \( u_{f^*} \). Furthermore, combinations of these mappings which lead to mappings \( (U_{hs}, U_{h0}) \) with the same property will be discussed. In section 3 extension of the above algorithms will be given, while in section 4 upper and lower bounds for several methods are discussed. This enables us to incorporate a test for the suboptimality of actions (see also [6]). Finally (section 5) some examples are given to illustrate several methods.

§ 2. "Improved" successive approximation methods.

2.1. Hastings [3] introduced the following (Gauss-Seidel) idea to modify the policy improvement procedure in Howard's policy iteration algorithm. Let \( u_f \) for a given strategy \( f \in K \) be computed. Determine a better strategy \( g \in K \) with components \( g(i) \) as follows:

\[
g(1) \text{ follows from } v_g(1) := \max_{k \in K(1)} \{ q_k(1) + \alpha \sum_{j=1}^{N} p_{ij} u_f(j) \} = \{ q_g(1)(1) + \alpha \sum_{j=1}^{N} p_{ij} u_f(j) \},
\]

where

\[
x_n \leq x_n^b \leq u_n^b \leq u_{f^*}.
\]
This idea can also be used in an S.A. algorithm.

Let \( x \in \mathbb{R}^N \). Define \( L_h(f) \) by:

\[
L_h(f)(x)(i) := q^f(i)(i) + \alpha \sum_{j < i} p_{ij} (L_h(f)(x)(j)) + \alpha \sum_{j \geq i} p_{ij} u_f(j), \quad i \in S.
\]

Define the mapping \( U_h \) by:

\[
U_h x := \max_{f} L_h(f)x.
\]

It is easily verified that \( L_h(f) \) and \( U_h \) are monotone \( \alpha \)-contractions with fixed point \( u_f \) and \( u_f \), respectively, so an S.A. algorithm might be based on

\[
\begin{align*}
   x_0^h &= 0 \\
   x_n^h &= U_h x_{n-1}^h =: L_h(f_n)^h x_{n-1}^h.
\end{align*}
\]

As in standard dynamic programming, the sequence \( \{x_n^h\} \) will have the following properties:

(3) \( x_{n-1}^h \leq x_n^h \leq u_{f_n}^h \leq u_f^* \),

(4) \( \lim_{n \to \infty} x_n^h := u_{f_n}^* \).

Furthermore, a comparison with the \( x_n^b \) of the dynamic programming algorithm yields inductively

(5) \( x_n^h \geq x_n^b \).
2.2. Also "overrelaxation" (see [8], [9]) may be used in successive approximation algorithms. Where the overrelaxation factor appears, for instance, if we try to find better estimates for $u_f$ by computing for certain paths the exact contribution to the total expected discounted reward.

Let $f \in K$ be given, then

$$u_f = L_f u_f = q_f + \alpha P_f u_f.$$  

Another expression for $u_f(i)$ may be found by computing the contribution to the expected reward until the time the system leaves $i$ explicitly ($f(i) = k$):

$$u_f(i) = q^k(i) + \alpha \sum_{j \neq i} \frac{k}{j} p_{ij} u_f(j)$$

$$+ \alpha \sum_{j \neq i} \frac{k}{j} p_{ij} u_f(j)$$

$$= \frac{1}{1 - \alpha p_{ii}} q^k(i) + \alpha \sum_{j \neq i} \frac{k}{j} p_{ij} u_f(j).$$

Let $\omega^k_i := \frac{1}{1 - \alpha p_{ii}}$, then with $k = f(i)$, (6) can be given as

$$u_f(i) = \omega^k_i q^k(i) + \alpha \sum_{j \neq i} \frac{k}{j} p_{ij} u_f(j).$$

(7) can also be deduced from:

$$u_f(i) = q^k(i) + \alpha \sum_{j} ^{k} p_{ij} u_f(j),$$

which yields:

$$(1 - \alpha p_{ii}) u_f(i) = q^k(i) + \alpha \sum_{j \neq i} \frac{k}{j} p_{ij} u_f(j),$$

where (7) follows by dividing by $(1 - \alpha p_{ii})$.

On the idea used in (7) we base for any $f \in K$ the mapping $L_0(f)$ defined by:
Furthermore we define the mapping \( U_0 \) by:

\[
U_0 x = \max_{f \in K} \{ L_0(f)x \} .
\]

Let

\[
\omega^-(f) := \min_{i \in S} \{ \omega^i(f) \},
\]

\[
\omega^+(f) := \max_{i \in S} \{ \omega^i(f) \},
\]

\[
\gamma(\omega) := 1 - \omega(1 - \alpha) .
\]

Then \( L_0(f) \) is a monotone \( \gamma(\omega^-(f)) \)-contraction with fixed point \( u_f \). It is easily verified that \( \gamma(\omega^-(f)) \leq \alpha \).

Let \( \omega^* := \min_{i,k} \{ \omega^i_k \} \), then \( U_0 \) is a monotone \( \gamma(\omega^*) \)-contraction with fixed point \( u_{f^*} \) (see [8]).

We have the relation:

\[
\gamma(\omega^-(f)) \leq \gamma(\omega^*) \leq \alpha .
\]

Hence a successive approximation method might be based on

\[
\text{III} \begin{cases} 
  x^0_0 := 0 \\
  x^0_n := U_0x^0_{n-1} =: L_0(f^0_n)x^0_{n-1} ,
\end{cases}
\]

where the following inequalities are easily proved:

\[
x^0_{n-1} \leq x^0_n \leq x^0_{f^*_n} \leq u_{f^*} ,
\]

\[
x^b_n \leq x^0_n .
\]

2.4. It is also possible to simplify algorithm III by using the fixed overrelaxation factor \( \omega^* \), which means that the contribution to the expected reward until the system leaves state \( i \) is only estimated.
Then we define $L_s(f)$ by:

$$L_s(f)(x(i)) := \omega^* q^k(i) + \alpha \omega^* \sum_{j \in S} p_{ij}^k (x(j)) + (1-\omega^*) x(i).$$

$U_s$ is defined by:

$$U_s x := \max_{f \in K} L_s(f)x.$$

$L_s(f)$ and $U_s$ are monotone $\gamma(\omega^*)$-contractions with fixed point $u_f$ and $u_{f^*}$, respectively. So it is possible to construct an S.A. algorithm based on:

$$\begin{cases}
x_0^s := 0 \\
x_n^s := U_s x_{n-1}^s =: L_s(f_n^s)x_{n-1}^s.
\end{cases}$$

Again we have:

$$\begin{align*}
x_{n-1}^s & \leq x_n^s \leq u_{f_n^s} \\
x_n^b & \leq x_n^s.
\end{align*}$$

§ 3. Combinations of S.A. algorithms.

In this section it will be shown that combinations of the mappings $U_h$, $U_0$, $U_s$ lead to mappings $U_{h0}$, $U_{hs}$, with the same properties as the original mappings; i.e. $U_{h0}$, $U_{hs}$ are monotone contractions with fixed point $u_{f^*}$.

First we want to combine the transformations $U_0$ and $U_h$ as is done in a modified form by Reetz [8].

We define the transformation $L_{h0}(f)$ inductively by

$$L_{h0}(f)(x(i)) := \omega_i^k q^k(i) + \alpha \omega_i^k \sum_{j < i} p_{ij}^k L_{h0}(f)(x(j)) + \alpha \omega_i^k \sum_{j > i} p_{ij}^k x(j),$$

with $k = f(i)$,

and $U_{h0}$ by:

$$U_{h0} x := \max_f L_{h0}(f)x.$$
Then, $L_{h_0}(f)$ and $U_{h_0}$ are monotone and $\gamma(\omega^*)$-contractions with fixed point $u_f$ and $u_f^*$, respectively.

So we have

$$V \begin{cases} 
  x_0^h = 0 \\
  x_n^h := U_{h_0} x_{n-1} = L_{h_0}(f_n^h) x_{n-1} 
\end{cases}$$

with

$$(12) \quad x_{n-1}^h \leq x_n^h \leq u_{f_n^h} \leq u_f^* ,$$

$$\lim_{n \to \infty} x_n^h = u_f^* .$$

Furthermore,

$$(13) \quad x_0^h \leq x_n^h ,$$

$$(14) \quad x_n^h \leq x_n .$$

The original Reetz [8] algorithm can be found as a combination of the transformations $U_s$ and $U_h$.

Let $L_{hs}(f)$ be given by

$$L_{hs}(f)x(i) := \omega^* q^k(i) + \alpha \omega^* \sum_{j \neq i}^{k} p_{ij} L_{hs}(f)x(j) + \alpha \omega^* \sum_{j \neq i}^{k} p_{ij} x(j) + (1 - \omega^*) x(i)$$

and

$$U_{hs}(f)x = \max_{f} L_{hs}(f)x .$$

$L_{hs}(f)$ and $U_{hs}$ are monotone and $\gamma(\omega^*)$-contractions with fixed point $u_f$ and $u_f^*$, respectively.

We have:

$$VI \begin{cases} 
  x_0^{hs} = 0 \\
  x_n^{hs} := U_{hs} x_{n-1} = L_{hs}(f_n^{hs}) x_{n-1} 
\end{cases}$$
with
\[ x_{n-1}^h \leq x_n^s \leq x_n^h \leq u_f^* , \]
\[ \lim_{n \to \infty} x_n^h = u_f^* , \]
\[ x_n^s \leq x_n^h , \]
\[ x_n^h \leq x_n^s . \]


A method to improve the estimations for \( u_f^* \) can also be found by inserting a number of value determination iteration steps in the S.A. algorithm based on \( U_\beta \) where \( \beta \in \mathbb{T} := \{b,h,0,s,hs,h0\} \), see [7].

This idea can also be introduced as the skipping of a number of policy improvement iteration steps in the S.A. algorithms.

We define for each \( x \in \mathbb{R}^N \) and for finite \( \lambda \in \mathbb{N} \) and for \( \beta \in \mathbb{T} \) the mapping \( U(\lambda) \) by:

\[ U(\lambda)x := L(\lambda)(x^\beta)U_\beta x \]

where \( f^{\beta \lambda} \) indicates the strategy that is found by applying \( U_\beta \) on \( x \).

For \( \lambda \in \mathbb{N}, \lambda > 1, U(\lambda) \) is neither necessarily a contraction mapping nor a monotone mapping.

However, we may base an algorithm on such a mapping:

\[
\begin{dcases}
  x_0^{\beta \lambda} = 0 , & \beta \in \mathbb{T} \\
  x_n^{\beta \lambda} = U(\lambda)x_n^{\beta \lambda} =: L(\lambda)(f^{\beta \lambda})x_n^{\beta \lambda} , & \beta \in \mathbb{T} .
\end{dcases}
\]

The monotone convergence of \( x_n^{\beta \lambda} \) to \( u_f^* \) is preserved (see [7]) as follows from the monotonicity of \( U_\beta \) and \( L(\beta)(f^{\beta}) \), i.e.
\[
\lim_{n \to \infty} x_n^{\beta \lambda} = u_{f^*_n},
\]

\[
x_{n-1}^{\beta \lambda} \leq x_n^{\beta \lambda} \leq u_{f_n^{\beta \lambda}} \leq u_{f^*_n}.
\]

A comparison of \(x_n^{\beta \lambda}\) with \(x_n^{\beta}\) yields:

\[
x_n^{\beta} \leq x_n^{\beta \lambda}, \quad n \in \mathbb{N}, \beta \in T.
\]

§ 5. Upper and lower bounds for \(u_{f^*_n}\). 

Successive approximation algorithms based on the ideas of the previous sections will converge. However, it will be necessary to construct upper and lower bounds for the current and the optimal strategy. Upper and lower bounds enable us to qualify the estimates for \(u_{f_n}, u_{f^*_n}\) and \(f^*_n\), respectively, see for instance MacQueen [5]. Also upper and lower bounds enable us to incorporate a test for the suboptimality of decisions in an algorithm (see MacQueen [6]).

Let the upper bound \(\bar{x}\) and the lower bound \(\underline{x}\) for \(u^*_n\) be given, then we can state the following lemma:

Lemma 1. Strategy \(f\) is suboptimal if

\[
L_\beta(f)\bar{x} < U_\beta \underline{x}, \quad \beta \in T.
\]

Proof.

\[
u^*_n = U_\beta u^*_n \geq U_\beta \underline{x} > L_\beta(f)\bar{x} \geq L_\beta(f)u^*_n,
\]

where the monotonicity property of \(U_\beta\) and \(L_\beta(f)\) is used.

This lemma enables us to determine for each \(i \in S\) decisions which are suboptimal (see for instance [6]).

* Note that in the algorithms where \(U_s\) is used, \(\omega^*\) can be redefined if the decision that causes \(\omega^*\) is suboptimal.
If we want to compare two algorithms it will be necessary to compare the corresponding sequences of upper and lower bounds. However, where the estimates for $u_f^*$ found in the $n$-th iteration step of a specific algorithm may be better than those of another algorithm (as shown in the previous sections), this doesn't mean unfortunately that it is possible to construct bounds that are "better" too.

We will illustrate this phenomenon with some examples (see section 6), however, we want to give without proof some general statements about upper and lower bounds first.

**Lemma 2.** For $U_\beta$, $\beta \in T$, the sequence

$$
-\beta x_n := x_n^{\beta} + \frac{1}{1 - c(\beta)} \|x_n^{\beta} - x_{n-1}^{\beta}\|_\infty , \quad n \in \mathbb{N},
$$

yields monotone nonincreasing upper bounds for $u_f^*$. Where $c(\beta)$ is the contraction factor corresponding with $U_\beta$ and where

$$
\|x - y\|_\infty := \max_i |x(i) - y(i)| , \quad x, y \in \mathbb{R}^N.
$$

Furthermore,

$$
\lim_{n \to \infty} -\beta x_n = u_f^* .
$$

**Lemma 3.** For $U^{(\lambda)}_\beta$, $\beta \in T$, $\lambda \in \mathbb{N}$, the sequence

$$
-\beta \lambda x_n := \min \{ -\beta \lambda x_{n-1}^{\beta}, x_n^{\beta} + \frac{1}{1 - c(\beta)} \|U_\beta x_n^{\beta} - x_{n-1}^{\beta}\|_\infty \}
$$

yields monotone nonincreasing upper bounds for $u_f^*$. Furthermore

$$
\lim_{n \to \infty} -\beta \lambda x_n = u_f^* .
$$

It is also possible to construct a monotone nondecreasing sequence of lower bounds for $u_f$ and $u_f^{\beta \lambda}$, and so for $u_f^*$. Such a sequence can be formed

$$
\lim_{n \to \infty} x_n^{\beta}, x_n^{\beta \lambda}, \beta \in T, \text{ respectively.}
$$
We will now give sequences of lower bounds that might be used for the several methods described in the previous sections.

Lemma 4. For $U_\beta$, $\beta \in T$, the sequence

$$x_n^\beta := x_{n-1}^\beta + \frac{1}{1 - \delta(\beta)} \| x_n^\beta - x_{n-1}^\beta \|_\infty$$

yields monotone nondecreasing lower bounds for $u_{f_n}^\beta$ and so for $u_{f_\ast}^\beta$.

$$\lim_{n \to \infty} x_n^\beta = u_{f_\ast}^\beta,$$

where $\delta(b) = \alpha; \delta(h) = \alpha^N, \delta(0) = \gamma(\omega^+(f_0^0)), \delta(s) = \gamma(\omega^*), \delta(h^0) = \gamma(\omega^+(f_0^0))^N, \delta(hs) = (\gamma(\omega^*))^N$ and

$$\| x_n^\beta - x_{n-1}^\beta \|_\infty := \min_{i \in S} (x_n^\beta(i) - x_{n-1}^\beta(i)).$$

Lemma 5. For $U_\lambda^\lambda$, $\beta \in T, \lambda \in N$, the sequence

$$x_n^\beta \lambda := x_{n-1}^\beta \lambda + \frac{1}{1 - \delta(\beta)} \| U_\beta x_n^\beta \lambda - x_{n-1}^\beta \lambda \|_\infty$$

yields monotone nondecreasing lower bounds for $u_{f_n}^\beta \lambda$ and so for $u_{f_\ast}^\beta \lambda$; furthermore

$$\lim_{n \to \infty} x_n^\beta \lambda = u_{f_\ast}^\beta \lambda.$$

For all the bounds we have a monotone convergence to $u_{f_\ast}^\beta \lambda$. So each number of the indicated set of algorithms can be used to estimate the optimal policy $f_\ast$ and the corresponding value vector $u_{f_\ast}$.

The examples in section 6 show that a choice for a specific algorithm may depend on the problem under consideration.
§ 6. Examples.

In this section we will give two simple examples to illustrate that the decision which algorithm has to be chosen, might depend on the problem under consideration.

Example 1. In this example we compare the distance between the upper and lower bounds in the n-th iteration step of algorithm I and this distance in the n-th iteration step of algorithm IV.

Consider a two state problem with in each state only one possible decision. Let the matrix of transition probabilities be given by:

\[
P := \begin{pmatrix} p & 1-p \\ 1-p & p \end{pmatrix}
\]

and the reward vector \( r \) by: \( r := \begin{pmatrix} r_1 \\ r_2 \end{pmatrix} \), with discount factor \( \alpha \).

Then algorithm I yields

\[
x_n^b = \sum_{k=0}^{n-1} \alpha^k p^k r,
\]

so

\[
x_n^b - x_{n-1}^b = \alpha^{n-1} p^{n-1} r = \alpha^{n-1} \left( \begin{array}{c} \frac{1}{2} \\ \frac{1}{2} \end{array} \right) r + (-\alpha(1-2p))^{n-1} \left( \begin{array}{c} \frac{1}{2} \\ -\frac{1}{2} \end{array} \right) r.
\]

This yields:

\[
D^b(n) := \|x_n^b - x_{n-1}^b\|_\infty - \|x_n^b - x_{n-1}^b\|_\infty = (\alpha |1-2p|)^{n-1}(r_1 - r_2).
\]

Using algorithm IV yields in a similar way

\[
D^s(n) := \|x_n^s - x_{n-1}^s\|_\infty - \|x_n^s - x_{n-1}^s\|_\infty = \frac{\alpha(1-p)}{1-\alpha p} \frac{1}{n-1} (r_1 - r_2)
\]

Let \( A_n \) be defined by:

\[
A_n := \frac{D^b(n)}{D^s(n)} = c \frac{|1-2p|^{n-1}(1-\alpha p)^{n-1}}{(1-p)^{n-1}}
\]

where \( c \) is a constant which is independent of \( n \).
Then

\[
\lim_{n \to \infty} Q_n = \infty \quad \text{if} \quad p > p_1 = \left(\frac{3 + \alpha}{4\alpha}\right) - \sqrt{\left(\frac{3 + \alpha}{4\alpha}\right)^2 - 1}
\]

\[
\lim_{n \to \infty} Q_n = 0 \quad \text{if} \quad p < p_1
\]

For this problem this leads to the conclusion that algorithm I is preferable if \(p < p_1\).

Example 2. Consider a two state problem with \(K(1) = \{J\}, K(2) = \{1, 2\}\), \(a = 0, 9\), and

\[
\begin{align*}
p_1^{11} &= 1, & p_1^{12} &= 0, & r^1(1) &= 2, \\
p_1^{21} &= 0, & p_1^{22} &= 1, & r^1(2) &= 2, \\
p_2^{21} &= 1, & p_2^{22} &= 0, & r^2(2) &= 1, 9.
\end{align*}
\]

The Hastings algorithm II will start in state 2 with the suboptimal decision 2, while (MacQueen) algorithm I starts with selecting the optimal decision 1. Furthermore, the upper and lower bounds corresponding to algorithm I are equal, which means that the optimal values \(u_{f^*}\) are known in one step.
References.


