ELSEVIER

European Journal of Operational Research 88 (1996) 622–636

EUROPEAN JOURNAL OF OPERATIONAL RESEARCH

Theory and Methodology

A K-step look-ahead analysis of Value Iteration algorithms for Markov decision processes

Meir Herzberg [a,*], Uri Yechiali [b]

[a] *Telecom Australia Research Laboratories, 770 Blackburn Rd., Clayton, Vic. 3168, Australia*

[b] *Department of Statistics and Operations Research, School of Mathematical Sciences, Raymond and Beverly Sackler Faculty of Exact Sciences, Tel Aviv University, Tel Aviv 69978, Israel*

Received April 1993; revised June 1994

Abstract

We introduce and analyze a general look-ahead approach for Value Iteration Algorithms used in solving both discounted and undiscounted Markov decision processes. This approach, based on the value-oriented concept interwoven with multiple adaptive relaxation factors, leads to accelerating procedures which perform better than the separate use of either the concept of value oriented or of relaxation. Evaluation and computational considerations of this method are discussed, practical guidelines for implementation are suggested and the suitability of enhancing the method by incorporating Phase 0, Action Elimination procedures and Parallel Processing is indicated. The method was successfully applied to several real problems. We present some numerical results which support the superiority of the developed approach, particularly for undiscounted cases, over other Value Iteration variants.

Keywords: Markov processes; Value iteration; Modified policy iteration; Adaptive relaxation factor; Look-ahead analysis

1. Introduction

The successive substitution technique for solving Markov Decision Processes (MDPs) appears to be the best computational method for solving large Markov decision models, by avoiding either dealing with huge Linear Programming models or repeatedly solving large sets of linear equations (see Tijms [23]).

The classical way of using the above technique is the standard Value iteration Algorithm (VIA), applied to both discounted and undiscounted MDPs. For discounted cases it relies on a basic recursive equation of the form

$$V_n(i) = \min_{a \in A_i} \left\{ C_i^a + \beta \sum_{j \in I} P_{ij}^a \cdot V_{n-1}(j) \right\}, \quad i \in I \tag{1}$$

* Corresponding author.

SSDI 0377-2217(94)00208-8

(this direction was mentioned as a future study by Puterman and Shin [19]); to present the merged approach in a uniform way for both discounted and undiscounted VIAs; (iii) To suggest practical guidelines for the effective use of this method.

The merged procedure may therefore be classified within the area of Fathoming and Relaxation Criteria, being used in speeding-up Dynamic Programming algorithms (see Morin and Marsten [10]). The new criteria for selecting appropriate ARFs, developed in [3,4] and based on a one-step look-ahead, fits very well with the concept of value oriented. Merging value oriented and relaxation will be done by developing a general procedure that looks ahead K value-oriented steps (called briefly K-step look-ahead) incorporated with multiple ARFs for discounted and undiscounted MDPs. The approach has the potential of performing better than separately using either value oriented or relaxation. This task seems significant, in particular for undiscounted MDPs, for three main reasons: (i) The effectiveness of the new ARF criteria is usually better for higher DFs (see [4]); (ii) the convergence of VIAs is usually slow for discounted MDPs with DFs close to one (see Scherer and White [21]) – the convergence rate of these cases and of undiscounted MDPs is similar; (iii) to the best of our knowledge, no practical considerations have been published for undiscounted MDPs when using the value-oriented concept.

The structure of the paper is as follows: In Section 2 we formulate the K-step look-ahead approach for the main versions of discounted and undiscounted VIAs. Appendix A details a complete derivation of that formulation for the PJ case. In Section 3 we evaluate the merged approach, discuss various computational aspects and suggest some practical improved guidelines for selecting the parameter K and for utilizing effectively the concept of relaxation. Appendix B is added to present the theoretical background which supports the developed approach. We conclude by presenting a few numerical results which demonstrate the effectiveness of the merged approach when compared to other variants of VIAs tested.

2. The K-step look-ahead formulation

The main idea of merging relaxation and value oriented is the following: After the completion of the n-th iteration we try to look ahead K value-oriented steps incorporated with relaxation considerations and estimate the future values of $V_{n+K}(i)$, $i \in I$. We denote these estimators by $\hat{V}_{K,n}(i)$ and will use them in the various VIA schemes for the $(n+1)$st iteration instead of the known values $V_n(i)$, $i \in I$.

We now apply the merged approach to various VIA schemes.

2.1. Discounted VIA schemes

For discounted MDPs we consider the four main VIA schemes, starting with the PJ case. By using Eq. (1) for K value-oriented steps *without relaxation*, one can derive the anticipated values $\hat{V}_{K,n}(i)$, $i \in I$, as follows:

$$\hat{V}_{K,n} = V_n + \sum_{k=1}^{K} \beta^k [P_n(R)]^k (V_n - V_{n-1}) = V_n + \sum_{k=1}^{K} \beta^k [P_n(R)]^k \delta_n, \tag{3}$$

where $[P_n(R)]_{ij} = P_{ij}^{R_i}$, $\forall i,\ j \in I$,

$$R_i \in \arg\min_{a \in A_i} \left\{ C_i^a + \beta \sum_{j \in I} P_{ij}^a \cdot V_{n-1}(j) \right\}, \quad i \in I, \tag{4}$$

and $\hat{V}_{K,n}$, V_n, V_{n-1} and δ_n are $|I|$ column vectors with the components $\hat{V}_{K,n}(i)$, $V_n(i)$, $V_{n-1}(i)$ and $\delta_n(i)$, $i \in I$, respectively.

Based on Eq. (3), one can derive a direct relation between the vector δ_n and the vector $\hat{\delta}_{k,n}$ which represents the contribution of the k-th value-oriented step to the vector $\hat{V}_{K,n}$:

$$\hat{\delta}_{k,n} = \beta^k \left[P_n(R)\right]^k \delta_n, \quad k = 1, 2, \ldots, K. \tag{5}$$

The intermediate estimators $\hat{V}_{k,n}(i)$, $k \geqslant 1$. $i \in I$, obtained after the completion of the k-th value-oriented step, can recursively be calculated in the following way:

$$\hat{V}_{k,n}(i) = \hat{V}_{k-1,n}(i) + \hat{\delta}_{k,n}(i) \equiv \hat{V}_{k-1,n}(i) + \beta \cdot \sum_{j \in I} P_{ij}^{R_i} \cdot \hat{\delta}_{k-1,n}(j)$$

$$= V_n(i) + \sum_{m=1}^{k} \hat{\delta}_{m,n}(i), \quad i \in I, \quad k = 1, 2, \ldots, K, \tag{6}$$

where

$$\hat{V}_{0,n}(i) \equiv V_n(i), \quad i \in I, \quad \text{and} \quad \hat{\delta}_{0,n}(i) \equiv \delta_n(i), \quad i \in I. \tag{7}$$

In order to obtain more effective estimators $\hat{V}_{K,n}(i)$ we apply the concept of Adaptive Relaxation (see, for example [3], [4] and [23]). The common way of applying relaxation is to use a *single* relaxation factor at each iteration. We propose to utilize this concept *several* times within each iteration. Thus, in the process of formulating $\hat{V}_{K,n}(i)$ we use K different Adaptive Relaxation Factors (ARFs) $w_{1,n}$, $w_{2,n}, \ldots, w_{k,n}, \ldots, w_{K,n}$, one for each value-oriented step.

In Appendix A we recursively derive the values $\hat{V}_{k,n}(i)$ and $\hat{\delta}_{k,n}(i)$, $k, = 1, 2, \ldots, K$, $i \in I$, which lead to the generalization of Eq. (6), namely

$$\hat{V}_{k,n}(i) = \hat{V}_{k-1,n}(i) + w_{k,n} \cdot g_{k,n}(i) = V_n(i) + \sum_{m=1}^{k} w_{m,n} \cdot g_{m,n}(i), \quad i \in I, \quad k = 1, 2, \ldots, K, \tag{8}$$

where

$$g_{k,n}(i) = \beta \cdot \sum_{j \in I} P_{ij}^{R_i} \cdot \hat{\delta}_{k-1,n}(j), \quad i \in I, \quad k = 1, 2, \ldots, K, \tag{9}$$

$$\hat{\delta}_{k,n}(i) = \hat{\delta}_{k-1,n}(i) + w_{k,n} \cdot \left[g_{k,n}(i) - \hat{\delta}_{k-1,n}(i)\right], \quad i \in I, \quad k = 1, 2, \ldots, K, \tag{10}$$

and $\hat{V}_{0,n}(i)$ as well as $\hat{\delta}_{0,n}(i)$ are defined by Eq. (7) (see Eqs. (A.14)-(A.17) in Appendix A).

It is readily seen that Eq. (6), which represents value oriented with no relaxation, is a special case of Eq. (8), for which the values $w_{m,n} \equiv 1$, $m \geqslant 1$, and consequently $\hat{\delta}_{m,n}(i) = g_{m,n}(i)$ (see Eq. (10)). It should be noted that generalizing Eqs. (3) and (5) is not that straightforward. Multiplying each of the summed terms of Eq. (3) by the relaxation factor $w_{k,n}$, for example, does not properly reflect the merged approach.

For the *Jacobi* (J) VIA we use the recursive equation

$$V_n(i) = \min_{a \in A_i} \left\{ \left[C_i^a + \beta \sum_{j \neq i} P_{ij}^a \cdot V_{n-1}(j) \right] \Big/ \left[1 - \beta P_{ii}^a\right] \right\}, \quad i \in I. \tag{11}$$

Performing for the Jacobi scheme a very similar analysis to the one presented in Appendix A and introducing the superscript J, the terms $\hat{V}_{k,n}^{\mathrm{J}}(i)$ and $\hat{\delta}_{k,n}^{\mathrm{J}}(i)$ can be calculated according to Eqs. (8) and (10), respectively. Eq. (7) holds as well. The only change is in the terms $g_{k,n}^{\mathrm{J}}(i)$, for which

$$g_{k,n}^{\mathrm{J}}(i) = \beta \cdot \sum_{j \neq i} P_{ij}^{R_i} \cdot \hat{\delta}_{k-1,n}^{\mathrm{J}}(j) / \left[1 - \beta P_{ii}^{R_i}\right], \quad i \in I, \quad k = 1, 2, \ldots, K. \tag{12}$$

Next consider the *Pre-Gauss-Seidel* (PGS) scheme:

$$V_n(i) = \min_{a \in A_i} \left\{ C_i^a + \beta \sum_{j=1}^{i-1} P_{ij}^a \cdot V_n(j) + \beta \sum_{j=i}^{|I|} P_{ij}^a \cdot V_{n-1}(j) \right\}, \quad i \in I, \tag{13}$$

For this case we use the superscript PGS, so that the terms $\hat{V}_{k,n}^{\mathrm{PGS}}(i)$ and $\hat{\delta}_{k,n}^{\mathrm{PGS}}(i)$ can be calculated according to Eqs. (8) and (10), respectively. Eq. (7) holds as well, while the terms $g_{k,n}^{\mathrm{PGS}}(i)$ are calculated by the recursion

$$g_{k,n}^{\mathrm{PGS}}(i) = \beta \sum_{j=1}^{i-1} P_{ij}^{R_i} \cdot g_{k,n}^{\mathrm{PGS}}(j) + \beta \sum_{j=i}^{|I|} P_{ij}^{R_i} \cdot \hat{\delta}_{k-1,n}^{\mathrm{PGS}}(j), \quad i \in I, \quad k = 1, 2, \ldots, K. \tag{14}$$

Finally, we consider the *Gauss-Seidel* (GS) scheme using the recursive equation

$$V_n(i) = \min_{a \in A_i} \left\{ \left[C_i^a + \beta \sum_{j=1}^{i-1} P_{ij}^a \cdot V_n(j) + \beta \sum_{j=i+1}^{|I|} P_{ij}^a \cdot V_{n-1}(j) \right] \Big/ \left[1 - \beta P_{ij}^a \right] \right\}, \quad i \in I. \tag{15}$$

This time we use the superscript GS where the terms $\hat{V}_{k,n}^{\mathrm{GS}}(i)$ and $\hat{\delta}_{k,n}^{\mathrm{GS}}(i)$ are again calculated via Eqs. (8) and (10), respectively. Once more, Eq. (7) holds, while the terms $g_{k,n}^{\mathrm{GS}}(i)$ are calculated by

$$g_{k,n}^{\mathrm{GS}}(i) = \left[\beta \sum_{j=1}^{i-1} P_{ij}^{R_i} \cdot g_{k,n}^{\mathrm{GS}}(j) + \beta \sum_{j=i+1}^{|I|} P_{ij}^{R_i} \cdot \hat{\delta}_{k-1,n}^{\mathrm{GS}}(j) \right] \Big/ \left[1 - \beta P_{ij}^{R_i} \right], \quad i \in I, \quad k = 1, 2, \ldots, K. \tag{16}$$

To summarize, *for all four schemes* we have

$$\hat{V}_{k,n}^{(\cdot)}(i) = V_n(i) + \sum_{m=1}^{k} w_{m,n} \cdot g_{m,n}^{(\cdot)}(i) = \hat{V}_{k-1,n}^{(\cdot)}(i) + w_{k,n} \cdot g_{k,n}^{(\cdot)}(i), \quad i \in I, \quad k = 1, 2, \ldots, K,$$

$$\hat{\delta}_{k,n}^{(\cdot)}(i) = \hat{\delta}_{k-1,n}^{(\cdot)}(i) + w_{k,n} \cdot \left[g_{k,n}^{(\cdot)}(i) - \hat{\delta}_{k-1,n}^{(\cdot)}(i) \right], \quad i \in I, \quad k = 1, 2, \ldots, K, \tag{17}$$

where $\hat{V}_{0,n}^{(\cdot)} \equiv V_n(i)$ and $\hat{\delta}_{0,n}^{(\cdot)}(\mathrm{i}) \equiv \delta_n(i)$, while $g_{k,n}^{(\cdot)}(i)$ are calculated by Eqs. (9), (12), (14) or (16) for the PJ, I, PGS or GS scheme, respectively. The order of calculating the variables at each look-ahead step k, $k = 1, 2, \ldots, K$, is as follows: $g_{k,n}^{(\cdot)}(i)$, $i \in I$, $w_{k,n}$, $\hat{V}_{k,n}^{(\cdot)}(i)$, $\hat{\delta}_{k,n}^{(\cdot)}(i)$, $i \in I$.

2.2. Undiscounted VIA schemes

In this subsection we develop a K-step look-ahead analysis for the undiscounted VIA, applied to MDP and to Semi-MDP (SMDP). Consider first the MDP:

$$V_n(i) = \min_{a \in A_i} \left\{ C_i^a + \sum_{j \in I} P_{ij}^a \cdot V_{n-1}(j) \right\}, \quad i \in I. \tag{18}$$

For the purpose of formulation one can apply the derivation of the discounted PJ scheme, using the superscript M for MDP and substituting $\beta = 1$. The terms $\hat{V}_{k,n}^{\mathrm{M}}(i)$ and $\hat{\delta}_{k,n}^{\mathrm{M}}(i)$ are calculated by the general Eq. (17). Once again Eq. (7) holds, while the terms $g_{k,n}^{\mathrm{M}}(i)$ are calculated by Eq. (9) with $\beta = 1$, namely,

$$g_{k,n}^{\mathrm{M}}(i) = \sum_{j \in I} P_{ij}^{R_i} \cdot \hat{\delta}_{k-1,n}^{\mathrm{M}}(j), \quad i \in I, \quad k = 1, 2, \ldots, K. \tag{19}$$

Next consider the Semi-MDP VIA for undiscounted cases:

$$V_n(i) = \min_{a \in A_i} \left\{ C_i^a / \tau_i^a + \sum_{j \in I} \tilde{P}_{ij}^a \cdot V_{n-1}(j) \right\}, \quad i \in I, \tag{20}$$

where the one step transition probabilities P_{ij}^a are transformed to artificial transition probabilities $\tilde{P}_{ij}^a$ aimed at satisfying the uniformization conditions (see Schweitzer [22]), namely,

$$\tilde{P}_{ij}^a = \begin{cases} P_{ij}^a \cdot \tau / \tau_i^a, & i \neq j, \\ P_{ij}^a \cdot \tau / \tau_i^a + 1 - \tau / \tau_i^a, & i = j, \end{cases} \tag{21}$$

with τ_i^a being the expected sojourn time of the process in state i when selecting decision a, and

$$0 < \tau < \min_{i \in I, a \in A_i} \{\tau_i^a\}. \tag{22}$$

Using this time the superscript SM, the terms $\hat{V}_{k,n}^{\mathrm{SM}}(i)$ and $\hat{\delta}_{k,n}^{\mathrm{SM}}(i)$ can be calculated by using Eq. (17). Once more, Eq. (7) holds, while the terms $g_{k,n}^{\mathrm{SM}}(i)$ are obtained by the recursion

$$g_{k,n}^{\mathrm{SM}}(i) = \sum_{j \in I} \tilde{P}_{ij}^{R_i} \cdot \hat{\delta}_{k-1,n}^{\mathrm{SM}}(j), \quad i \in I, \quad k = 1, 2, \ldots, K. \tag{23}$$

3. Computational considerations of the proposed approach

In Section 2 we have presented a general K-step look-ahead procedure incorporated with relaxation which can be summarized as follows: At the end of iteration n, $n \geq 1$, we successively apply K value-oriented steps, each of which is incorporated with an adaptive relaxation value $w_{k,n}$. We thus modify the various VIAs by using for the $(n+1)$st iteration the values $\hat{V}_{K,n}^{(\cdot)}(i)$, $i \in I$, instead of the original values $V_n(i)$, $i \in I$, obtained at the end of iteration n. Modifying the VIAs in the above manner is aimed at reducing the gap between $\mathrm{Max}_i\{\delta_{n+1}(i)\}$ and $\mathrm{Min}_i\{\delta_{n+1}(i)\}$ in order to faster satisfy the stopping criterion and thus reducing the total number of iterations although at the expense of calculating the values $\hat{V}_{K,n}^{(\cdot)}(i)$, $i \in I$. This is justified only if the savings in total computational effort required for convergence, due to the reduction in the number of iterations, is larger than the extra effort assigned for calculating the value $\hat{V}_{K,n}^{(\cdot)}(i)$. It is therefore required to find an effective way of implementing this method in a dynamic fashion that

(i) controls the parameter K and determines its actual value for each iteration n; and
(ii) controls relaxation by assigning values $w_{k,n}$ only to selected value-oriented steps, while using no relaxation (i.e. $w_{k,n} = 1$) for the other steps.

To do so some principles related to MPI and value oriented have to be discussed. Under the practical assumptions that the Markov chain, for every possible decision policy R is finite, aperiodic and irreducible, the elements of the vector $\hat{\delta}_{k,n}^{(\cdot)}$ (or alternatively the vector $g_{k,n}^{(\cdot)}$) tend to converge with the growth of the value k for both discounted and undiscounted MDPs. The convergence is ensured when applying value oriented to the PJ scheme with no relaxation, for which $w_{k,n} = 1$, $k = 1, 2, \ldots, K$, and $\hat{\delta}_{k,n}^{(\cdot)} = g_{k,n}^{(\cdot)}$ (see van der Wal [25]). Consider for this example the vector $\hat{\delta}_{k,n}$, as in Eq. (5), even for the case $\beta = 1$. For large values k each row of the matrix $[P(R)]^k$ tends to converge to the stationary distribution of the policy R which causes $\hat{\delta}_{k,n}(i)$, $i \in I$, to converge to the limiting value $\Sigma_j \Pi_R(j) \cdot \delta_n(j)$, where $\Pi_R(j)$ represents the proportion of time the process is at state j, $j \in I$, under the decision policy R.

From the above, the higher the level of convergence of the vector $\hat{\delta}_{K,n}^{(\cdot)}$ the closer we are to a complete contribution of the policy R, following the use of $\hat{V}_{K,n}^{(\cdot)}$ instead of V_n, and thus the closer to a Policy Iteration. Appendix B clarifies this issue in more detail. The advantage here, for both discounted and

undiscounted cases, is derived from the use of the vector δ_n to reach fast convergence of $\hat{\delta}^{(\cdot)}_{K,n}$ and that effort assigned to each policy can be controlled by the parameter K. Furthermore, the concept of relaxation can be applied to speed-up the convergence of $\hat{\delta}^{(\cdot)}_{K,n}$ following the use of ARFs such as in [3], [4] and [15]. Analysing the level of convergence of the vectors $\hat{\delta}^{(\cdot)}_{k,n}$, $k = 1, 2, \ldots, K$, is therefore a key consideration in controlling the look-ahead process. Our task is in fact to define for each iteration n, $n \geqslant 1$, the required level of convergence of the vector $\hat{\delta}^{(\cdot)}_{K,n}$ (look-ahead depth) and to achieve this level with as few as possible look-ahead steps, denoted K_n, $n \geqslant 1$. On the other hand, a further increase of K_n may lead to a smaller gap between $\mathrm{Max}_i\{\delta_{n+1}(i)\}$ and $\mathrm{Min}_i(\delta_{n+1}(i)\}$. Exploring this tradeoff requires:

1) To define a representative measure which will indicate the level of convergence of the vector $\hat{\delta}^{(\cdot)}_{K,n}$, $k = 1, 2, \ldots, K_n$, $n \geqslant 1$; and
2) To analyse the computational effort associated with Value Iteration and look-ahead steps.

In [14] a convergence indicator for discounted cases was defined by $M_{k,n}$ which is the largest element among $\hat{\delta}^{(\cdot)}_{k,n}(i)$, $i \in I$. We found that this indicator is less suitable for undiscounted cases and suggest to use one of the following two indicators (derived, in fact, from the stopping criteria mentioned in Section 2) for both discounted and undiscounted cases:

(i) $M_{k,n} - m_{k,n}$ or alternatively,
(ii) $M_{k,n}/m_{k,n}$ (when $m_{k,n} > 0$),

where $M_{k,n}$ ($m_{k,n}$) is the largest (smallest) term of $\hat{\delta}^{(\cdot)}_{k,n}(i)$, $k = 1, 2, \ldots, K_n$, $i \in I$. As a third measure, one can use the Standard Deviation of the elements of $\hat{\delta}^{(\cdot)}_{k,n}$, (see [3]).

The Order of Computational Effort (OCE) per iteration of the basic VIAs (neither applying the approach of value-oriented nor using the concept of relaxation) is $\bar{A} \cdot \bar{Z} \cdot |I|$ (derived directly from the recursive equations of VIAs), where:

$\bar{A}$ = Average number of decisions (actions) per state.
$\bar{Z}$ = Average number of non-zero transition probabilities per action per state.
$|I|$ = Total number of states.

The OCE per each look-ahead step is $\bar{Z} \cdot |I|$, required to calculate a value-oriented step, augmented with the effort needed for calculating an ARF $w_{k,n}$, which ranges from $4 \cdot |I|$ to $12 \cdot |I|$, depending on the criterion used for selecting the ARF (see [3], [4]). This OCE analysis calls to limit the look-ahead computations per iteration, thus, limiting the value K_n. The larger the value $\bar{A}$ the larger the limit on K_n, denoted Max K, as computational savings of value iterations are increased with the growth of $\bar{A}$.

Fig. 1 illustrates typical cases of over, under and effective look-ahead when using the measure $M_{k,n} - m_{k,n}$. Each dot on the figure represents the convergence level as a function of n and k, following the completion of either a value iteration (for $k = 0$ recall that $M_{0,n}$ and $m_{0,n}$ are $\mathrm{Max}_i\{\delta_n(i)\}$ and $\mathrm{Min}_i\{\delta_n(i)\}$, respectively) or a look-ahead step ($k \geqslant 1$). It is demonstrated (Case a) that the use of large values K_n might lead to a waste in effort devoted to the look-ahead method. This is reflected by the relation $M_{K,n} - m_{K,n} \ll M_{0,n+1} - m_{0,n+1}$. On the other hand, under look-ahead (case b) does not fully utilize the value-oriented concept, resulting in unsatisfactory reduction of the number of value iterations. Under look-ahead can be recognized by the relation $M_{K,n} - m_{k,n} > M_{0,n+1} - m_{0,n+1}$; the extreme case of under look-ahead is in fact the basic VIA for which $K = 0$ and $M_{0,n} - m_{0,n} > M_{0,n+1} - m_{0,n+1}$. Effective look-ahead (Case c) uses adaptively K_n values and targets to the relation $M_{K,n} - m_{K,n} \leqslant M_{0,n+1} - m_{0,n+1}$. Finally, (Case d), the effective incorporation of relaxation is demonstrated, resulting in reduction of look-ahead steps and consequently savings in computational effort.

General guidelines and observations which were found useful after applying the proposed method to many problems tested are as follows:

- The value Max K. It was observed that affective results are obtained when setting Max K in the range of $1.5\bar{A}$–$2.5\bar{A}$. This limit is significant mainly at first iterations, when the variability of δ_n is high, for problems with low $\bar{A}$ values (about 10 or less). This rule suggests that extra computational effort will be limited to the level of computations of around 2 value iterations.

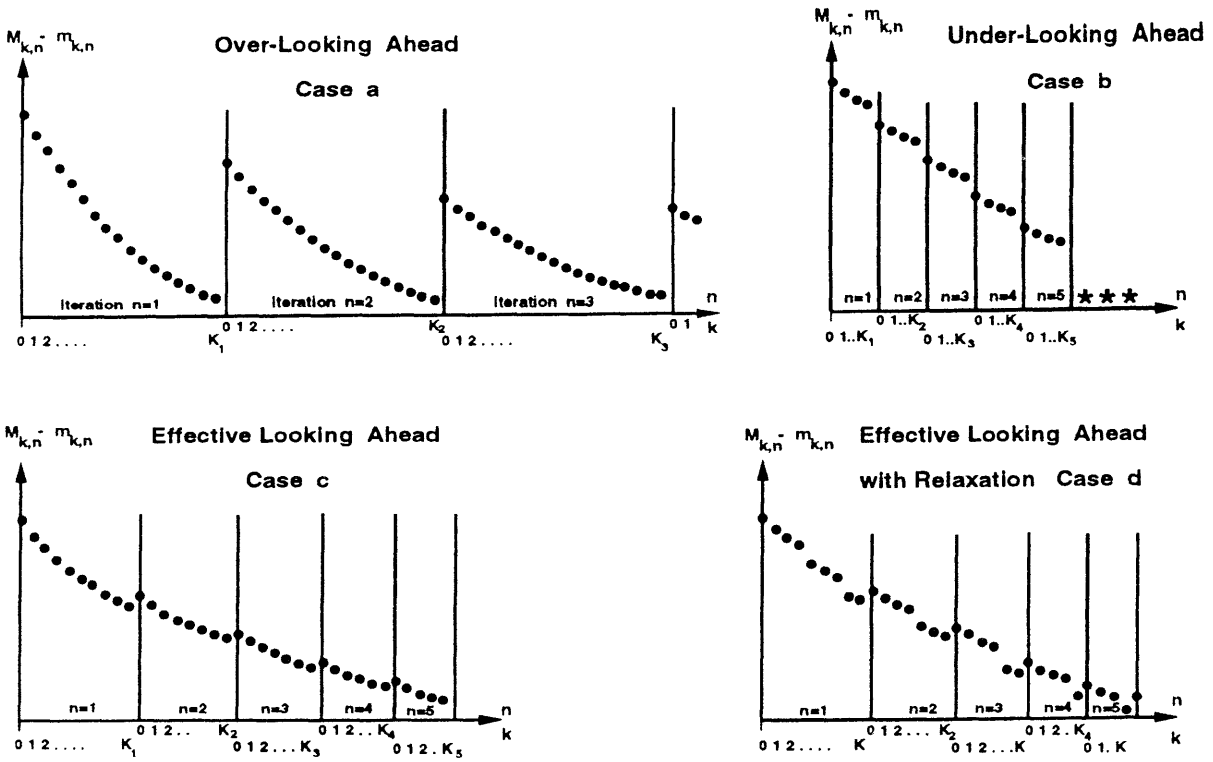

Fig. 1. Over, under and effective K-step look ahead.

- Last iterations. At the last one or two iterations, when $M_{0,n} - m_{0,n}$ is close to ε, it is recommended to create an over look-ahead case by reducing $M_{K,n} - m_{K,n}$ down to the range of $0.02 \cdot \varepsilon$ to $0.05 \cdot \varepsilon$, where ε is the pre-determined tolerance error required for the stopping criterion selected. The objective is to increase the probability of satisfying the stopping criterion by the end of the $(n+1)$st value iteration.
- Applying relaxation. When applying successive look-ahead steps, it is required in addition to effective selection of ARFs, to determine how often to apply actual relaxation. For the problems tested, it was found that selecting an ARF for each value-oriented step tends to be less efficient than selecting an ARF every X steps, where $3 \leqslant X \leqslant 7$. Selecting such X values eliminates, to a great extent, the cases of 'jamming' which often occur when applying the one-step look-ahead approach (see [3]) and reaches the required level effectively.
- The trend of K_n. As mentioned earlier, the vector δ_n tends to converge with the growth of the iteration number n. This implies that the minimum value K, which achieves any given convergence level of the vector $\hat{\delta}^{(\cdot)}_{K,n}$, tends to decrease with the growth of n. On the other hand, an improved policy R requires a higher level of convergence of the vector $\hat{\delta}^{(\cdot)}_{K,n}$ so as to gain further contribution of the improved policy. Considering these contradicting effects on K_n, we have found that, for problems having large values of $\bar{A}$, the adaptive values K_n usually tend to *decrease* with the growth of n (the effect of δ_n is more dominant). This observation, which is based on the experience gained from solving many real problems of up to 100 000 states, is new.

It is worth noting that the proposed approach can be enhanced in several ways. It can be regarded as a general modified VIA for which one can apply 'Phase 0' and Action Elimination (see Herzberg [5]).

Phase 0 is a preliminary procedure aimed at supplying effective initial values $V_0(i)$, $i \in I$. It was found suitable for problems with high $\bar{A}$ values and relies on an approximate solution of artificial related problems with low values of $\bar{A}$. While the look-ahead approach tries to save computational effort of the policy iteration part, Action Elimination (AE) aimed at saving effort during the value iteration part by excluding from the calculation actions which are not part of the optimal policy (see MacQueen [9], Porteus [18], Hastings and Van Nunen [2] and Puterman and Shin [20]). Permanent AE can be applied to discounted cases immediately after each value iteration while temporary AE, for both discounted and undiscounted cases, may be applied only after the look-ahead stage just before the next value iteration.

Another enhancement relates to the effectivenes of data processing. Parallel Processing capabilities may readily be used for the merged approach. Indeed, most of the computational effort is assigned to calculate the terms $g_{k,n}^{(\cdot)}(i)$, $k = 1, 2, \ldots, K$, $i \in I$. For the schemes mentioned (excluding PGS and GS) calculation of a term $g_{k,n}^{(\cdot)}(i)$, for a certain state i, can be done *independently* of the calculations for other states (given that all calculations for value-oriented step $k-1$ and its associated relaxation $w_{k-1,n}$ have been completed). For PGS and GS the concept of parallel processing is more limited, however, it can still be applied to the second part of the terms $g_{k,n}^{(\cdot)}(i)$, $i \in I$ (see Eqs. (14) and (16)).

Applying the proposed procedure requires the use of memory, allocated for the terms $\hat{V}_{k,n}^{(\cdot)}(i)$, $g_{k,n}^{(\cdot)}(i)$ and $\hat{\delta}_{k,n}^{(\cdot)}(i)$, $k = 1, 2, \ldots, K$, $i \in I$. As these terms are recursively calculated, one can practically reuse the memory assigned to $V_t(i)$, and $\delta_n(i)$, $i \in I$, for the terms $\hat{V}_{k,n}^{(\cdot)}(i)$, and $\hat{\delta}_{k,n}^{(\cdot)}(i)$, $i \in I$, respectively. Similarly, the memory required for the terms $g_{k,n}^{(\cdot)}(i)$, $i \in I$ can be reused for the various k values, so additional memory of *only one* vector with $|I|$ elements is required for the K-step look-ahead analysis.

Table 1
Numerical results of various discounted value iteration schemes

VI schemes ($\varepsilon = 10^{-5}$)	Attribute	$\bar{A} = 8$, $\|I\| = 65$		$\bar{A} = 10$, $\|I\| = 1000$		$\bar{A} = 80$, $\|I\| = 370$		$\bar{A} = 2000$, $\|I\| = 650$	
		$\beta = 0.8$	$\beta = 0.9$	$\beta = 0.8$	$\beta = 0.9$	$\beta = 0.8$	$\beta = 0.9$	$\beta = 0.8$	$\beta = 0.9$
PJ:									
Basic VIA	NOI	52	101	48	98	56	102	52	125
	RSC	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
$K = 1$, $X = 1$	NOI	12	21	16	27	27	35	20	43
	RSC	2.8	3.0	2.4	2.6	2.0	2.8	2.6	2.9
MPI	NOI	4	6	4	5	5	7	4	6
	MxK	20	24	24	41	75	35	126	
	TotK	60	100	54	90	124	313	87	418
	RSC	4.5	5.5	5.1	7.0	8.6	9.4	12.9	20.1
MARVO, $X = 5$	NOI	3	4	3	4	5	6	4	5
	MxK	17	17	18	18	38	54	22	104
	TotK	34	50	35	54	103	189	69	266
	RSC	6.1	7.3	7.0	10.1	8.6	11.3	12.9	24.3
GS:									
Basic VIA	NOI	39	74	41	75	47	78	44	84
	RSC	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
$K = 1$, $X = 1$	NOI	13	22	12	19	21	29	14	25
	RSC	2.3	2.5	2.4	2.8	2.2	2.6	3.1	3.4
MPI	NOI	3	5	3	4	5	7	4	5
	MxK	20	20	24	24	33	64	30	109
	TotK	40	80	39	72	94	240	63	375
	RSC	4.9	4.9	6.0	6.7	7.6	7.8	11.0	16.2
MARVO, $X = 5$	NOI	3	4	3	4	4	5	4	5
	MxK	17	17	18	18	25	47	19	58
	TotK	30	48	31	42	68	148	48	203
	RSC	4.8	6.5	6.3	8.7	9.4	10.8	11.0	16.5

4. Numerical results

Table 1 presents numerical results for selected discounted MDP problems. Each problem has been solved for various VIAs, where for each algorithm we use 4 different solution procedures: basic VIA, one-step look-ahead, MPI (value-oriented only) and the proposed scheme called Multiple Adaptive Relaxation with Value Oriented (MARVO). The problems differ from each other mainly by dimension. The attributes considered are: NOI – number of iterations, MxK – Maximal value-oriented steps that actually taken per iteration (MxK $\leqslant$ Max K), TotK – total number of value-oriented steps, and RSC – relative speed until completion in terms of CPU time ratio when compared to the basic VIA. RSC can be used for comparison analysis under the practical assumption that CPU times per iteration are the same for all basic VIAs. For all problems the value $X = 5$ is selected (ARFs are calculated every 5 value-oriented steps). The tolerance error $\epsilon = 10^{-5}$ was chosen and the discount factors used are $\beta = 0.8$ and ($\beta = 0.9$. The same convergence level rules for $\hat{\delta}^{(\cdot)}_{K,n}$ have been applied to the schemas MPI and MARVO. We found that using alternately the ARF criteria Min Difference and Min Variance (see [4]) achieves for discounted cases effective results. No major differences among the various schemes are discovered with respect to MPI and MARVO, however some differences are worth noting. As the results obtained for the J and the PJ (PGS and the GS) schemes were very similar, we show in Table 1 the numerical results which only relate to the schemes PJ and GS. The convergence of PJ under MARVO usually requires less iterations than with MPI, resulting in total computations saving of up to 55% when compared to the MPI. This is less frequent when using the scheme GS but reduction of TotK under MARVO is achieved in any case. As was expected, MARVO is particularly effective for cases of high values of β, resulting in higher RSC values for $\beta = 0.9$ than for $\beta = 0.8$.

Table 2 presents numerical results for selected undiscounted MDP problems. Each problem has been solved several times by using the value X as a parameter. For undiscounted MDPs we use alternately the

Table 2
Numerical results of various undiscounted value iteration schemes

$\epsilon = 10^{-3}$		$\bar{A} = 8, \lvert I\rvert = 65$		$\bar{A} = 9, \lvert I\rvert = 950$		$\bar{A} = 240, \lvert I\rvert = 90$		$\bar{A} = 250, \lvert I\rvert = 650$	
Scheme	Attribute	MDP	SMDP	MDP	SMDP	MDP	SMDP	MDP	SMDP
Basic VIA	NOI	337	44	677	698	109	139	537	528
	RSC	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
$K = 1$	NOI	43	17	94	123	24	27	86	87
	MxK	1	1	1	1	1	1	1	1
	TotK	42	16	93	122	23	26	85	86
	RSC	5.7	2.1	6.1	4.8	4.6	5.3	6.3	6.1
MPI	NOI	13	4	21	14	8	5	7	6
	MxK	19	19	23	23	57	109	174	148
	TotK	241	53	455	307	211	284	874	656
	RSC	7.8	4.1	9.6	15.6	12.3	22.6	51.2	61.2
MARVO, $X = 3$	NOI	6	3	12	10	4	5	5	5
	MxK	17	17	18	18	82	33	122	109
	TotK	85	34	206	172	168	110	396	311
	RSC	14.8	4.4	15.7	19.7	21.9	24.7	73.3	77.8
MARVO, $X = 5$	NOI	6	3	12	12	4	5	5	5
	MxK	17	17	18	18	76	50	105	110
	TotK	72	34	196	205	169	121	376	364
	RSC	18.1	4.9	17.7	17.8	22.4	24.8	77.6	77.0
MARVO, $X = 7$	NOI	7	3	14	12	5	5	5	5
	MxK	17	17	18	18	78	63	136	119
	TotK	100	34	247	211	214	148	428	352
	RSC	14.9	5.3	15.0	18.0	18.1	24.3	76.6	78.8

Min Ratio and the Min Variance ARF criteria. The MPI scheme represents the value-oriented concept with no relaxation. Once again, the same convergence level rules have been used for the MPI and for MARVO. The tolerance error selected is $\varepsilon = 10^{-3}$. The Relative Speed until Completion (RSC) achieved with the combined use of value oriented and relaxation is significantly higher than that in the discounted cases and the superiority of the method over other schemes is more apparent. The larger the dimensions of the problem the higher the RSC obtained.

It is worth noting that although Action Elimination and Parallel Processing have the potential to improve the performance of all variants of discounted and undiscounted VIAs, it is expected that basic VIAs will benefit from these enhancements more than MPI and MARVO. This stems from the fact that the full action space of basic VIAs is always considered while the value-oriented phase, as applied to MPI and MARVO, uses a single action per state.

5. Conclusions

We have analysed an algorithm which can be regarded as a hybrid of Value and Policy Iteration. The Value Determination phase of the Policy Iteration algorithm which requires the solution of a set of $|I|$ simultaneous equations is replaced, under the look-ahead approach, by an easier task aimed at letting the vector $\hat{\delta}^{(\cdot)}_{K,n}$ to converge in a controllable fashion. To carry out this task effectively we have introduced a novel approach of multiple relaxation which is incorporated with value-oriented steps in a uniform mode for both discounted and undiscounted MDPs. Applying practical guidelines to this method in addition to the effective selection of ARF criteria and their use, lead to improved performance when it is compared to other variants of VIAs for solving MDPs. The computational savings are significant, especially for cases of high discount factors and for undiscounted MDPs, as it is demonstrated in Tables 1 and 2. The method seems attractive, particularly for solving large-scale MDPs. It was successfully tested on many state-dependent decision and control processes in the area of private [6] and mobile [7] telecommunication networks by considering problems in the range of up to 100 000 states, having about 100 decisions per state.

Appendix A

In this appendix we present a derivation which formulates the K-step look-ahead approach for the PJ scheme using recursive Eq. (1) of Section 1. To do so we perform, first, a one-step look-ahead analysis. The relaxation part of this step is achieved by calculating the modified values $\bar{V}_n(i)$ as follows:

$$\bar{V}_n(i) = w_{1,n} \cdot V_n(i) + (1 - w_{1,n}) \cdot V_{n-1}(i) = V_{n-1}(i) + w_{1,n} \cdot \delta_n(i), \quad i \in I. \tag{A.1}$$

Following the completion of the first value-oriented step we derive the estimators $\hat{V}_{1,n}(i)$:

$$\hat{V}_{1,n}(i) = C_i^{R_i} + \beta \sum_{j \in I} P_{ij}^{R_i} \cdot \bar{V}_n(j), \quad i \in I. \tag{A.2}$$

$\hat{V}_{1,n}(i)$ would represent the value $V_{n+1}(i)$ if the values $\hat{V}_n(j)$, in the $(n+1)$st iteration, are used instead of the values $V_n(i)$, $i \in I$, respectively, and the action R_i still satisfies Eq. (4).

From Eqs. (A.2) and (A.1),

$$\begin{aligned} \bar{V}_{1,n}(i) &= C_i^{R_i} + \beta \sum_{j \in I} P_{ij}^{R_i} \cdot \left[V_{n-1}(j) + w_{1,n} \cdot \delta_n(j) \right] \\ &= V_n(i) + w_{1,n} \cdot g_{1,n}(i) \quad i \in I, \end{aligned} \tag{A.3}$$

where

$$g_{1,n}(i) = \beta \cdot \sum_{j \in I} P_{ij}^{R_i} \cdot \delta_n(j), \quad i \in I. \tag{A.4}$$

In a similar manner we define estimators $\hat{\delta}_{1,n}(i)$ for $\delta_{n+1}(i)$, $i \in I$, where

$$\hat{\delta}_{1,n}(i) = \hat{V}_{1,n}(i) - \bar{V}_n(i), \quad i \in I. \tag{A.5}$$

Using Eqs. (A.1) and (A.3) one gets

$$\hat{\delta}_{1,n}(i) = \delta_n(i) + w_{1,n} \cdot \left[g_{1,n}(i) - \delta_n(i) \right], \quad i \in I. \tag{A.6}$$

In order to find 'good' values of $w_{1,n}$, we select an ARF by using either the Minimum Difference criterion or the Minimum Variance criterion (see [4]). In this way we apply relaxation after calculating the values $g_{1,n}(i)$, in order to obtain effective values $\hat{\delta}_{1,n}(i)$ and $\hat{V}_{1,n}(i)$, $i \in I$.

Before performing a look-ahead analysis for the second time (step), we apply once again the concept of relaxation, using this time an APF $w_{2,n}$ to get the modified values $\bar{V}_{1,n}(i)$, $i \in I$, instead of the values $\bar{V}_{1,n}(i)$, $i \in I$, respectively, where

$$\bar{V}_{1,n}(i) = \bar{V}_n(i) + w_{2,n} \cdot \left[\bar{V}_{1,n}(i) - \bar{V}_n(i) \right], \quad i \in I. \tag{A.7}$$

Using again Eqs. (A.1) and (A.3) we get

$$\begin{aligned} \bar{V}_{1,n}(i) &= V_{n-1}(i) + w_{1,n} \cdot \delta_n(i) + w_{2,n} \cdot \left\{ V_n(i) + w_{1,n} \cdot g_{1,n}(i) - \left[V_{n-1}(i) + w_{1,n} \delta_n(i) \right] \right\} \\ &= V_{n-1}(i) + w_{1,n} \cdot \delta_n(i) + w_{2,n} \cdot \left\{ \delta_n(i) + w_{1,n} \cdot \left[g_{1,n}(i) - \delta_n(i) \right] \right\}. \end{aligned}$$

Applying Eq. (A.6), the above can be written as

$$\bar{V}_{1,n}(i) = V_{n-1}(i) + w_{1,n} \cdot \delta_n(i) + w_{2,n} \cdot \hat{\delta}_{1,n}(i), \quad i \in I. \tag{A.8}$$

We can now complete the second value-oriented step by applying Eq. (A.8) to calculate estimators $\bar{V}_{2,n}(i)$ for $V_{n+2}(i)$, $i \in I$, respectively, using the same R_i decisions again:

$$\begin{aligned} \bar{V}_{2,n}(i) &= C_i^{R_i} + \beta \sum_{j \in I} P_{ij}^{R_i} \cdot \bar{V}_{1,n}(j) \\ &= C_i^{R_i} + \beta \sum_{j \in I} P_{ij}^{R_i} \cdot \left[V_{n-1}(j) + w_{1,n} \cdot \delta_n(j) + w_{2,n} \cdot \hat{\delta}_{1,n}(j) \right] \\ &= V_n(i) + w_{1,n} \cdot g_{1,n}(i) + w_{2,n} \cdot g_{2,n}(i), \end{aligned} \tag{A.9}$$

where

$$g_{2,n}(i) = \beta \cdot \sum_{j \in I} P_{ij}^{R_i} \cdot \hat{\delta}_{1,n}(j), \quad i \in I. \tag{A.10}$$

Alternatively, by Eq. (A.3),

$$\bar{V}_{2,n}(i) = \bar{V}_{1,n}(i) + w_{2,n} \cdot g_{2,n}(i). \tag{A.11}$$

The estimators $\hat{\delta}_{2,n}(i)$ for $\delta_{n+2}(i)$, $i \in I$, are defined as

$$\hat{\delta}_{2,n}(i) = \hat{V}_{2,n}(i) - \bar{V}_{1,n}(i), \quad i \in I. \tag{A.12}$$

Using Eqs. (A.8) and (A.9) one gets

$$\hat{\delta}_{2,n}(i) = V_n(i) - V_{n-1}(i) + w_{1,n} \cdot \left[g_{1,n}(i) - \delta_n(i) \right] + w_{2,n} \cdot \left[g_{2,n}(i) - \hat{\delta}_{1,n}(i) \right].$$

Based on Eq. (A.6), the above can be written as

$$\hat{\delta}_{2,n}(i) = \hat{\delta}_{1,n}(i) + w_{2,n} \cdot \left[g_{2,n}(i) - \hat{\delta}_{1,n}(i) \right], \quad i \in I. \tag{A.13}$$

By induction it is straightforward to derive a general formulation of k value-oriented steps incorporated with relaxation (for which Eqs. (A.3) and (A.9) are special cases):

$$\hat{V}_{k,n}(i) = V_n(i) + \sum_{m=1}^{k} w_{m,n} \cdot g_{m,n}(i) = \hat{V}_{k-1,n}(i) + w_{k,n} \cdot g_{k,n}(i), \quad i \in I, \quad k = 1, 2, \ldots, K \tag{A.14}$$

where $\hat{V}_{0,n}(i)$ is defined by Eq. (7).

Alternatively, generalizing Eqs. (A.9), (A.12) and substituting $\hat{V}_{k-1,n}(j) = \hat{V}_{k,n}(j) - \hat{\delta}_{k,n}(j)$,

$$\begin{aligned} \hat{V}_{k,n}(i) &= C_i^{R_i} + \beta \sum_{j \in I} P_{ij}^{R_i} \cdot \bar{V}_{k-1,n}(j) \\ &= C_i^{R_i} + \beta \sum_{j \in I} P_{ij}^{R_i} \cdot \left[\hat{V}_{k,n}(j) - \hat{\delta}_{k,n}(j) \right], \quad i \in I, \quad k = 1, 2, \ldots, K. \end{aligned} \tag{A.15}$$

Eqs. (A.4) and (A.10) can also be generalized to the form

$$g_{k,n}(i) = \beta \cdot \sum_{j \in I} P_{ij}^{R_i} \cdot \hat{\delta}_{k-1,n}(j), \quad i \in I, \quad k = 1, 2, \ldots, K, \tag{A.16}$$

where $\hat{\delta}_{0,n}(i)$ is defined by Eq. (7).

Eqs. (A.6) and (A. 13) are generalized accordingly:

$$\hat{\delta}_{k,n}(i) = \hat{\delta}_{k-1,n}(i) + w_{k,n} \cdot \left[g_{k,n}(i) - \hat{\delta}_{k-1,n}(i) \right], \quad i \in I, \quad k = 1, 2, \ldots, K. \tag{A.17}$$

Appendix B

For completeness, we derive in this appendix further insight of the relation between the look-ahead approach and Policy Iteration for both discounted and undiscounted MDPs.

Eq. (3) of Section 2 can be rewritten as follows:

$$\hat{V}_{K,n} = V_n + \sum_{k=1}^{K} \beta^k \left[P_n(R) \right]^k \delta_n = V_{n-1} + \sum_{k=0}^{K} \beta^k \left[P_n(R) \right]^k \delta_n. \tag{B.1}$$

Consider Eq. (B.1) for the limiting case $K \to \infty$:

$$\lim_{K \to \infty} \hat{V}_{K,n} = V_{n-1} + \left[[\mathrm{Id}] - \beta \left[P_n(R) \right] \right]^{-1} \delta_n. \tag{B.2}$$

Eq. (B.2), for the special case where $V_{n-1}(i) = 0$ and $\delta_n(i) = C_i^{R_i}$, $i \in I$ (in fact, using the values $V_{n-1}(i) = 0$, $i \in I$, under a policy R yields $V_n(i) = C_i^{Ri}$ and thus $\delta_n(i) = V_n(i) - V_{n-1}(i) = C_i^{R_i}$, $i \in I$) can be evaluated by the expression

$$\lim_{K \to \infty} \hat{V}_{K,n} = \left[[\mathrm{Id}] - \beta \left[P_n(R) \right] \right]^{-1} C^R, \tag{B.3}$$

where [Id] denotes the unit matrix and C^R is the cost vector whose elements are $C_i^{R_i}$, $i \in \mathrm{I}$.

For undiscounted MDPs the limiting case relates to the vector $\hat{\delta}_{K,n}$ whose elements converge, by substituting as before $\delta_n(i) = C_i^{R_i}$, $i \in I$, to the value

$$\lim_{K \to \infty} \hat{\delta}_{K,n} = \sum_{j \in I} \Pi_R(j) \cdot \delta_n(j) = \sum_{j \in I} \Pi_R(j) \cdot C_j^{R_j}. \tag{B.4}$$

Eq. (B.4) represents the process cost per unit time under the policy R, obtained by Howard's Value Determination phase for undiscounted cases. Assuming that each element $\hat{\delta}_{K,n}(i)$, $i \in I$, converges to the value Δ_n, Eq. (A.15), for undiscounted cases $\beta = 1$ and for $k = K$, can be modified to the form

$$\hat{V}_{K,n} + [\Delta_n] = C^R + [P_n(R)]\hat{V}_{K,n}, \tag{B.5}$$

where $[\Delta_n]$ is an $|I|$ column vector with identical elements equal to Δ_n.

Replacing V_n by $\hat{V}_{K,n}$, according to Eq. (B.3) (Eq. (B.5)) just before the $(n+1)$st value iteration, is exactly performing Howard's Value Determination phase with the policy R for discounted (undiscounted) MDPs before applying the Policy Improvement phase.

Howard's Value Determination phase in the well known Policy Iteration algorithm, for both discounted and undiscounted MDPs, is therefore a special case of the look-ahead approach which under a policy R *always uses* $\delta_n(i) = C_i^{R_i}$, $i \in I$, $n \geqslant 1$, and looks ahead *infinite* number of steps *without relaxation*. For discounted cases it relates to the PJ variant only.

Acknowledgements

The authors are thankful to the referees for valuable remarks and suggestions which significantly improved the presentation of the paper. The authors also wish to thank Dr. Moshe Sneidovich of the University of Melbourne, and Mr. William M. Jolly of Telstra Research Labs for useful discussions. The permission of the Director of Telstra Research Laboratories, to publish this paper is hereby acknowledged.

References

[1] Dembo, R.S., and Haviv, M., "Truncated policy iteration algorithm", *Operations Research Letters* 3/5 (1984) 243–246.
[2] Hastings, N.A.J., and Nunen, J.A.E. van, "The action elimination algorithm for Markov decision processes", in: H.C. Tijms and J. Wessels (eds.), *Markov Decision Processes*, Mathematical Center Tract 93, Amsterdam, 1977, 161–170.
[3] Herzberg, M., and Yechiali, U., "Criteria for selecting the relaxation factor of the value iteration algorithm for undiscounted Markov and semi-Markov decision processes", *Operations Research Letters* 10/4 (1991) 193–202.
[4] Herzberg, M., and Yechiali, U., "Accelerating procedures of the value iteration algorithm for discounted Markov decision processes, based on a one-step look-ahead analysis", *Operations Research* 42/5 (1994) 940–946.
[5] Herzberg, M., "Phase 0 and action elimination for generally modified value iteration algorithms", *Asia-Pacific Journal of Operational Research* 10 (1993) 159–169.
[6] Herzberg, M., "An optimal decision process for routing circuit-switched calls originated by users of a private distribution network", *International Teletraffic Congress (ITC)* 13, Copenhagen, Denmark, 1991, 453–458.
[7] Herzberg, M., and McMillan, D., "State-dependent control of call arrivals in layered cellular mobile networks", *Telecommunication Systems* 1/4 (1993) 365–378.
[8] Kushner, H., and Kleinman, A.J., "Accelerated procedures for the solution of discrete Markov control problems", *IEEE Transactions on Automatic Control* 16 (1971) 147–152.
[9] MacQueen, J, "A test of sub-optimal actions in Markovian decision problems", *Operations Research* (1967) 559–561.
[10] Morin, T.L., and Marsten, R.E., "Branch and bound for dynamic programming", *Operations Research* 24/4 (1976) 611–627.
[11] Morton, T., "Undiscounted Markov renewal programming via modified successive approximations", *Operations Research* 19 (1971) 1081–1089.
[12] Nunen, J.A.E. van, "Contracting Markov decision processes", Tract 71, Mathematical Centre, Amsterdam, 1976.
[13] Nunen, J.A.E. van, "A set of successive approximation methods for discounted Markovian decision problems", *Zeitschrift für Operations Research* 20 (1976) 203–208.
[14] Ohno, K., and Ichiki, K., "Computing optimal policies for controlled tandem queueing systems", *Operations Research* 35/1 (1987) 121–126.
[15] Popyack, J.L., Brown, R.L., and White, III, C.C., "Discrete versions of an algorithm due to Varaiya", *IEEE Transactions on Automatic Control* 24 (1979) 503–504.

[16] Porteus E.L, "Some bounds for discounted sequential decision processes", *Management Science* 18 (1971) 7–11.
[17] Porteus, E.L., and Totten, J., "Accelerated computation of the expected discount return in a Markov chain', *Operations Research* 26 (1978) 350–358.
[18] Porteus, E.L., "Bounds and transformation of finite Markov decision chains", *Operations Research* 23 (1975) 761–784.
[19] Puterman, M.L., and Shin, M.C, "Modified policy iteration algorithms for discounted Markov decision problems", *Management Science* 24/11 (1978) 1127–1137.
[20] Puterman, M.L., and Shin, M.C., "Action elimination procedures for modified policy iteration algorithms", *Operations Research* 30/2 (1982) 301–307.
[21] Scherer, W.T., and White, D.J., "The convergence of value iterations in discounted Markov decision processes", ORSA/TIMS Conference, Washington, DC, 1988.
[22] Schweitzer, P.J., "Iterative solution of the functional equations of undiscounted Markov renewal programming", *Journal of Mathematical Analysis and Applications* 34 (1971) 495–501.
[23] Tijms, H.C., *Stochastic modelling and analysis: A computational approach*, Wiley, Chichester, 1986.
[24] Thomas, L.C., Hartley, R., and Lavercombe, A.C., "Computational comparison of value iteration algorithms for discounted Markov decision processes", *Operations Research Letters* 2 (1983) 72–76.
[25] Wal, J. van der, "Stochastic dynamic programming", Tract 139, Mathematical Centre, Amsterdam, 1984.