Draft Version: September 1989

Computing Approximate Solutions to Markov Renewal Programs with Continuous State Spaces

Pierre L'Ecuyer[1]

ABSTRACT

Value iteration and policy iteration are two well known computational methods for solving Markov renewal decision processes. Value iteration converges linearly, while policy iteration (typically) converges quadratically and is therefore more attractive in principle. However, when the state space is very large (or continuous), the latter asks for solving at each iteration a large linear system (or integral equation) and becomes unpractical.

We propose an "approximate policy iteration" method, targeted especially to systems with continuous or large state spaces, for which the Bellman (expected cost-to-go) function is relatively smooth (or piecewise smooth). These systems occur quite frequently in practice. The method is based on an approximation of the Bellman function by a linear combination of an *a priori* fixed set of base functions. At each policy iteration, we build a linear system in terms of the coefficients of these base functions, and solve this system approximately. We give special attention to a particular case of *finite element* approximation where the Bellman function is expressed directly as a convex combination of its values at a finite set of grid points.

In the first part of the paper, we survey and extend slightly some basic results concerning convergence, approximation, and bounds. All along the paper, we consider both the discounted and average cost criteria. Our models are infinite horizon and stationary.

[1]Département d'informatique, Université Laval, Ste-Foy, Québec, Canada, G1K 7P4.

1 Introduction

When solving a Markov Renewal Decision Process (MRDP) with large (sometimes continuous) state or action spaces [17], it is usually necessary to use some sort of discretization or approximation procedure [1, 2, 8, 11, 12, 13, 16, 18, 19, 22, 34, 35]. The discretized (or "smaller") problem can then be solved using either linear programming, policy iteration, value iteration, or some hybrid combination of these [2, 13, 18, 19, 26, 27, 28, 32, 34].

A natural approach consists in partitioning the state and action spaces into a finite class of subsets, selecting a representative element in each subset, and defining and approximate but more tractable model, with finite state and action spaces. This form of approximation has been proposed and studied theoretically in [1, 2, 11, 12, 35]. Typically, for the approximation schemes suggested by these authors, the value function in the Dynamic Programming (DP) functional equation is approximated by a piecewise constant function, constant on each subset of the partition of the state space. More sophisticated approachs like polynomial or spline approximation or interpolation, finite element methods, etc., were also suggested in [8, 13, 22, 34].

Schweitzer and Seidmann [34] considered a MRDP with finite state and action sets and suggested polynomial approximation of the value function. They proposed three computational algorithms: linear programming, policy iteration with least squares approximation, and global least squares fit.

Daniel [8] suggested spline approximation for a deterministic, finite horizon, continuous control problem. Haurie and L'Ecuyer [13] considered a discounted MRDP model and suggested spline or finite element methods to approximate the value function at each step of the *value iteration* algorithm. The approximation method and/or grid may vary from iteration to iteration. They provide formulas to compute (or estimate) bounds on the optimal value function and on the value function associated to the optimal policy, taking into account the approximation error at each iteration and the fact that only a finite number of iterations are made. This kind of approach can also be used (heuristically) with Schweitzer's algorithm [32] for the undiscounted case (see [18, 19] for applications and numerical results). Value iteration converges geometrically (at a linear rate) [4, 9, 32], but often with a factor of almost one, which makes it rather slow in terms of the number of iterations. *Policy iteration*, on the other hand, converges typically at a quadratic rate [26, 27], and is thus much more attractive in principle. One difficulty is that when the state space is continuous, or has large cardinality, each policy evaluation step asks for solving either an integral equation or

a huge linear system, which is too costly or impossible to solve in practice. This is the main reason why value iteration is often suggested for such cases (see for instance [13, 32]) despite the fact that it is sometimes very time consuming, especially when each iteration involves a numerical integration at each evaluation point [13, 18, 19].

In this paper, we consider a MRDP model with general (Borel) state and action spaces, as introduced in [13, 17]. We describe a finite element computational approach to deal with continuous or very large state spaces. The general idea is to express the Bellman (expected cost-to-go) function V as a linear combination of a finite number of (simple) base functions $B_1, \ldots, B_J$:

$$V(s) = \sum_{j=1}^{J} d_j B_j(s). \tag{1}$$

We replace V by this linear combination (with unknown coefficients d_j) into the DP functional equation that corresponds to a fixed policy (equation (38) below).

A direct generalization of the approach proposed in [34] is to integrate this equation (numerically or symbolically) for a finite number of states, say $\sigma_1, \ldots, \sigma_I$. If $I = J$, we obtain from the functional equation a linear system in terms of the coefficients $d_1, \ldots, d_J$. If $I > J$ (more evaluation points than unknowns), we can use least squares fitting to determine the d_j's: again, to minimize the quadratic form, a J-dimensional linear system must be solved. Schweitzer and Seidmann [34] proposed this approach for the case of finite state and action spaces, and where $S = \{\sigma_1, \ldots, \sigma_I\}$. The linear system is not easy to solve in general for large values of J.

A second approach is to use a finite element method [14]. We first define a scalar product on the space of real-valued functions of the state s. Then, the basic idea is that after replacing V by (1) in the DP equation (38), we ask the scalar product of this equation by any arbitrary function of the form (1) to be a valid equation. This yields a system of J equations in J unknowns. In a special case of this approach, $V(s)$ turns out to be expressed directly as a *convex* combination of the values of V at J evaluation points $\sigma_1, \ldots, \sigma_J$:

$$V(s) = \sum_{j=1}^{J} V(\sigma_j) B_j(s) \tag{2}$$

where $0 \leq B_j(s) \leq 1$, $B_j(\sigma_i) = \delta_{ij}$ (the Kronecker's delta), and $\sum_j B_j(s) = 1$ for each s. The σ_j's are in fact the *nodes* of the finite elements. When inserting this V into the DP equation (38), one obtains a linear system whose matrix is substochastic, and where the unknowns are $V(\sigma_1), \ldots, V(\sigma_J)$. In this case the system can be partially solved using just a few iterations of an iterative method, starting from the previous value of V. Another useful

property of this scheme is that if we change the set of evaluation points σ_j and functions B_j after a number of iterations, it is easy to compute the *new* values of $V(\sigma_j)$ from the old ones, and continue the iterations with them.

Our suggested algorithm with the finite element approach can be viewed as an extension of the "modified policy iteration" algorithm studied by Puterman and Shin [26] for discounted Markovian decision process (MDP) models. We start with some policy μ, and iterate over the following steps: partition (triangulate) the state space into finite elements, with nodes $\sigma_1, \ldots, \sigma_J$ and base functions $B_1, \ldots, B_J$, construct the linear system associated with this set and the policy μ, solve this system approximately for the $V(\sigma_j)$'s, put this solution back into the DP functional equation, perform a minimization step to obtain a new policy μ, and repeat. To solve the system approximately, one can just perform a few iterations of an iterative method, possibly combined with some aggregation steps [4, 5]. The "extension" lies mostly in the way that we approximate the Bellman function to treat the case of continuous state spaces. We also consider the undiscounted case.

Aggregation-disaggregation techniques have been proposed for accelerating convergence in MRDPs. See [4, 5] and the references cited there. Basically, the state space is partitioned into a finite number of subsets, which form the "states" of the aggregate process. In practice, defining the partition is usually difficult. Bertsekas and Castañon [5] have introduced an interesting aggregation approach in which the partition is modified adaptively based on the progress of the algorithm. These aggregation methods have been proposed for dealing with large, but finite state and action spaces. In contrast, the methods described in this paper are targeted primarily towards continuous state spaces. In fact, they can be combined with aggregation methods.

We now give an outline of the paper. In section 2, we state the two basic MRDP models, one with a discounted cost criterion, the other with an average cost criterion, and we give their associated DP functional equations. We also give formulas to compute (or estimate) bounds on the optimal value function, and on the value function associated with the retained policy. For the discounted case, we give results for a N-stage locally contracting model that generalizes the model considered in [13, 17] (which was one-stage locally contracting). We also provide bounds for the average cost case.

In section 3, we recall the usual value iteration and policy iteration algorithms, with some comments. In sections 4 we describe a finite element computational approach, using an *approximate policy iteration* algorithm. We consider both the discounted and average

cost criteria. We focus on the basic practical ideas. We do not perform a formal complexity analysis. Such an analysis is non-trivial and would require introducing other technical conditions (e.g., for smoothness) on the model. In the conclusion, we comment on further possible variations and we mention some experience with numerical examples.

2 Two MRDP models and their functional equations

Consider the following (stationary) *Markov Renewal Decision Process* (MRDP) model [13]. The state space S and action space A are Borel spaces. Each state s in S has a nonempty set of admissible actions $A(s)$. At each of an infinite sequence of stages (events), the decision maker observes the state s and selects an action a (also called a decision) from $A(s)$. Let $0 = t_0 \leq t_1 \leq \cdots \leq t_n \leq t_{n+1} \leq \cdots$ such that t_n is the time of occurence of stage n, and let s_n and a_n be the state and selected action at that stage. A cost $c(s_n, a_n)$ is incurred at stage n, and the next state s_{n+1} and time of the next stage t_{n+1} are determined as

$$(t_{n+1}, s_{n+1}) = (t_n + \zeta, s)$$

where the pair (ζ, s) is generated randomly according to a probability measure $Q(\cdot \mid s_n, a_n)$ over $[0, \infty) \times S$. A new action a_{n+1} is selected from $A(s_{n+1})$, and so on. As in [3, 13, 17], we assume that $\Gamma = \{(s, a) \mid s \in S, a \in A(s)\}$ is an analytic subset of $S \times A$, that c is a lower semi-analytic function, and that Q is a Borel measurable stochastic kernel on $[0, \infty) \times S$ given $S \times A$. This is less restrictive than the lower semi-continuity assumptions that are often made, and which do not always hold in practice. But we must consider more general policies than Borel-measurable.

A *policy* is a universally measurable function $\mu : S \to A$ such that $\mu(s) \in A(s)$ for each s in S. Let U be the set of all policies. In this paper, we consider only nonrandomized stationary Markov policies. Associated with any initial state $s_0 = s$, and policy μ, there is a uniquely defined probability measure $P_{\mu,s}$ on the set of infinite sequences $(s_0, a_0, s_1, a_1, \ldots)$, where $a_n = \mu(s_n)$ for each n. Let $E_{\mu,s}$ be the corresponding mathematical expectation.

2.1 The discounted model

Here, we assume that the costs are discounted at rate $\rho > 0$. Hence, a cost of 1 incurred in t units of time is equivalent to a cost of $e^{-\rho t}$ incurred now. For each policy μ and initial state $s_0 = s$, we introduce, when they exists, the values

$$V_\mu(s) = \lim_{n \to \infty} E_{\mu,s} \left[\sum_{i=0}^{n-1} e^{-\rho t_i} c(s_i, a_i) \right] \tag{3}$$

and

$$V_*(s) = \inf_{\mu \in U} V_\mu(s). \tag{4}$$

The functions V_μ and V_* represent respectively the total expected discounted cost associated with policy μ, and the optimal total expected discounted cost. A policy $\mu \in U$ is said to be ϵ-optimal, for $\epsilon > 0$, if $V_\mu(s) \leq V_*(s) + \epsilon$ for all $s \in S$.

For any integer $n \geq 1$, the expected n-stage discount factor associated with policy μ and state s is

$$\alpha_n(\mu, s) = E_{\mu,s}\left[e^{-\rho t_n}\right]$$

and satisfies $0 \leq \alpha_n(\mu, s) \leq 1$.

Under condition C or LC below, this model admits of analysis via contraction mappings.

CONDITION C (N-stage contracting model): There exists an integer $N \geq 1$, and real numbers $\alpha_1 < 1$, $c_0 \leq 0$ and $c_1 \geq 0$ such that for all admissible policies μ,

$$\begin{aligned} \alpha_N(\mu, s) &\leq \alpha_1 \\ c_0 \leq c(s, a) &\leq c_1. \blacksquare \end{aligned}$$

A policy μ is called *N-stage distinguished*, for an integer $N \geq 1$, if there exists three constants $\delta_1 < 1$, c_0 and c_1 such that for all s in S :

$$\begin{aligned} \alpha_N(\mu, s) &\leq \delta_1 \\ c_0 \leq c(s, \mu(s)) &\leq c_1. \end{aligned}$$

CONDITION LC (N-stage locally contracting model): There exists a N-stage distinguished policy $\tilde{\mu}$ and two constants K_1 and K_2 such that

$$K_1 + K_2 > 0 \tag{5}$$

and for every policy μ, all $s \in S$, and every integer n such that $N \leq n < 2N$,

$$K_1 + K_2\alpha_n(\mu, s) \leq E_{\mu,s}\left[\sum_{i=0}^{n-1} e^{-\rho t_i} c(s_i, a_i)\right]. \blacksquare \tag{6}$$

Let $\mathbb{B}$ be the set of all extended real-valued functions $V : S \to [-\infty, \infty]$, endowed with the supremum norm $\|V\| = \sup_{s \in S} |V(s)|$, and $\mathbb{B}_0$ be the Banach space of all bounded functions in $\mathbb{B}$. An operator ϕ mapping a closed subset of $\mathbb{B}_0$ into itself is said to be *contracting* with factor α if $\alpha < 1$ and $\|\phi(V_2) - \phi(V_1)\| \leq \alpha\|V_2 - V_1\|$ for all V_1 and V_2 in that subset.

Defined below are three standard dynamic programming operators. For $V \in \mathbb{B}$, $s \in S$ and $a \in A(s)$, let (when the integral exists):

$$H(V)(s,a) = c(s,a) + \int_{[0,\infty)\times S} V(s')e^{-\rho\zeta} Q(\mathrm{d}\zeta \times \mathrm{d}s' \mid s,a) \tag{7}$$

$$T(V)(s) = \inf_{a\in A(s)} H(V)(s,a). \tag{8}$$

For every policy μ, let

$$T_\mu(V)(s) = H(V)(s,\mu(s)). \tag{9}$$

Let $\mathbb{B}_1$ be the subset of universally measurable and lower semi-analytic functions in $\mathbb{B}_0$. The operators H, T and T_μ are well defined on $\mathbb{B}_1$ (the integrals exist), and their image is also in $\mathbb{B}_1$ (see e.g. [3, 17]). Let

$$\mathbb{B}_* = \left\{ V \in \mathbb{B}_1 \mid \frac{Nc_0}{1-\alpha_1} \le V \le \frac{Nc_1}{1-\alpha_1} \right\} \tag{10}$$

for the N-stage contracting model, and

$$\mathbb{B}_* = \left\{ V \in \mathbb{B}_1 \mid K_1 + \min(0,K_2) \le V \le \frac{(2N-1)c_1}{1-\delta_1} \right\} \tag{11}$$

for the N-stage locally contracting model. The following properties are proven in [15, 17] for $N = 1$. The proofs can be generalized to $N > 1$ using the same reasoning as in reference [6].

THEOREM 1. Let $V \in \mathbb{B}_*$. Under assumption C or LC:

(a) Let $\mu \in U$. Under LC, suppose also that $T_\mu^{kN}(K_1 + \min(0,K_2)) \le (2N-1)c_1/(1-\delta_1)$ for any $k \ge 1$. Then,

$$T_\mu(V) = V \tag{12}$$

if and only if $V = V_\mu$, and

$$\lim_{n\to\infty} \|T_\mu^n(V) - V_\mu\| = 0. \tag{13}$$

(b) $\mathbb{B}_*$ is closed under T^n for every $n \ge N$,

$$T(V) = V \tag{14}$$

if and only if $V = V_*$, and

$$\lim_{n\to\infty} \|T^n(V) - V_*\| = 0. \ \blacksquare \tag{15}$$

From an approximate solution to the functional equation (14) and the following theorem, one can obtain bounds on V_* and on the suboptimality of the retained policy (provided it satisfies a contraction condition).

THEOREM 2. We assume condition C or condition LC. Under condition C, let α_1 and N satisfy that condition, and $n_0 = N$. Under condition LC, let $\alpha_1 \in (0,1)$, k_0 be the smallest integer larger than

$$\eta = \frac{(2N-1)c_1/(1-\delta_1) - K_1 - \min(0, K_2)}{(K_1 + K_2)\alpha_1}, \tag{16}$$

and $n_0 = Nk_0$. Let $V, W \in \mathbb{B}_*$, $\delta^- \geq 0$ and $\delta^+ \geq 0$ such that

$$-\delta^- \leq T(V) - W \leq \delta^+. \tag{17}$$

For any real number x, let

$$\begin{aligned} \phi^+(x) &= \sup_{s \in S} \max(0, W(s) - V(s) + x), \\ \phi^-(x) &= \sup_{s \in S} \max(0, V(s) - W(s) + x). \end{aligned}$$

Define

$$\begin{aligned} \epsilon^+ &= n_0\delta^+ + (n_0 - 1)\phi^+(0), \\ \epsilon^- &= n_0\delta^- + (n_0 - 1)\phi^-(0). \end{aligned}$$

Then,

$$-\epsilon^- - \frac{\alpha_1}{1-\alpha_1}\phi^-(\epsilon^-) \leq V_* - W \leq \epsilon^+ + \frac{\alpha_1}{1-\alpha_1}\phi^+(\epsilon^+). \tag{18}$$

Moreover, for any $\delta_0 > \delta^+$, since $T_\mu(V) \leq T(V) + \delta_0 - \delta^+$, there exists a policy μ such that

$$T_\mu(V) \leq W + \delta_0, \tag{19}$$

and if T_μ^M is contracting with modulus $\alpha < 1$ for some integer $M \geq 1$, and ϵ_0 is defined as

$$\epsilon_0 = M\delta_0 + (M-1)\phi^+(0),$$

then

$$0 \leq V_\mu - V_* \leq \epsilon^- + \frac{\alpha_1}{1-\alpha_1}\phi^-(\epsilon^-) + \epsilon_0 + \frac{\alpha}{1-\alpha}\phi^+(\epsilon_0). \tag{20}$$

PROOF. The derivation of (18) is done exactly as in the proof of theorem 3.1 (a) in [13]. The remainder of the proof can also be done along the lines of theorem 3.1 (b) in [13], except that $\mathbb{B}_2$ is replaced by $\mathbb{B}_*$, $g_2 = (2N-1)c_1/(1-\delta_1) + \delta_0 - \delta^+ + \|\underline{V}\|$, $\overline{V} = (Mg_2 + (1+\alpha)\|\underline{V}\|)/(1-\alpha)$, equations (A.12) and (A.13) are replaced by

$$T_\mu(V)(s) \geq c(s, \mu(s)) - \|\underline{V}\| \qquad \forall s \in S$$

and

$$c(s, \mu(s)) \leq (2N-1)c_1/(1-\delta_1) + \delta_0 - \delta^+ + \|\underline{V}\| = g_2$$

respectively, and after equation (A.13), T_μ, g_2 and V_1 are respectively replaced by T_μ^M, Mg_2 and W. To get the last inequality in the proof, we use the fact that $T_\mu^M(V) \leq W + \epsilon_0$, which can be shown by the same argument as in the proof of (A.9) in [13]. ∎

In theorem 2, the choice of α_1 is left open. Note that η, n_0, ϵ^- and ϵ^+ are $O(1/\alpha_1)$. $\phi^+(\epsilon^+)$ and $\phi^-(\epsilon^-)$ are also $O(1/\alpha_1)$ for ϵ^- and ϵ^+ large enough (and do not depend on α_1 otherwise). Hence, the right-hand side of (20) is $O(1/\alpha_1 + 1/(1-\alpha_1))$, and therefore, α_1 should be kept away from 0 or 1. To keep things simple, taking $\alpha_1 = 0.5$ is not a bad idea in general. A more sophisticated approach is to minimize the upper bound in (20) with respect to α_1. Obviously, this involves more computation. The best thing to do depends on how much one is willing to pay to get a (possibly) better bound for the current solution.

One particular case of the above theorem is when $T(V)$ can be used directly for W. In that case, we have $\delta^- = \delta^+ = 0$. Under condition C with $N = 1$, assuming that $W = T(V)$, the bound in (20) becomes

$$V_\mu - V_* \leq \delta_0 + \frac{\alpha_1}{1-\alpha_1}\left(\phi^-(0) + \phi^+(\delta_0)\right). \tag{21}$$

Note that tighter bounds can also be obtained for that particular case as in Porteus [24].

2.2 The average cost case

We define the average expected cost under policy μ, starting from state $s_0 = s$, by (when this expression exists):

$$\psi_\mu(s) = \limsup_{n\to\infty} \frac{E_{\mu,s}\left[\sum_{i=0}^{n-1} c(s_i, a_i)\right]}{E_{\mu,s}\left[\sum_{i=0}^{n-1} t(s_i, a_i)\right]}. \tag{22}$$

The optimal average cost is

$$\psi_*(s) = \inf_{\mu\in U} \psi_\mu(s). \tag{23}$$

For $\epsilon > 0$, a policy μ is said to be ϵ-optimal if $\psi_\mu(s) \leq \psi_*(s) + \epsilon$ for all $s \in S$.

We now give conditions under which the above expressions are well defined. The condition C below is equivalent to the one formulated in the previous subsection for the discounted case, with $N = 1$. Let

$$t(s, a) = \int_{[0,\infty)} \zeta Q(\mathrm{d}\zeta \times S \mid s, a).$$

It represents the expected time to the next transition if the current state is s and action a is chosen. We suppose that t is Borel-measurable in both s and a.

CONDITION C1. There exists constants $\tau > 0$, c_0 and c_1 such that for every $s \in S$ and $a \in A(s)$,

$$\begin{aligned} \tau &\leq t(s,a), \\ c_0 &\leq c(s,a) \leq c_1. \end{aligned}$$

Condition C1 means that the cost per stage is bounded, and that the expected duration of a stage is bounded away from zero, uniformly over the state—action pairs. Under condition C1, the process can be uniformized as suggested in [23, 32], by replacing c, t and Q respectively by:

$$\begin{aligned} \tilde{c}(s,a) &= \frac{\tau c(s,a)}{t(s,a)} \\ \tilde{t}(s,a) &= \tau \\ \tilde{Q}(\bar{S} \mid s,a) &= \frac{\tau}{t(s,a)} Q([0,\infty) \times \bar{S} \mid s,a) + \left(1 - \frac{\tau}{t(s,a)}\right) \varphi(s,\bar{S}) \end{aligned}$$

for any $s \in S$, $a \in A(s)$ and $\bar{S}$ Borel subset of S, where

$$\varphi(s,\bar{S}) = \begin{cases} 1 & \text{if } s \in \bar{S}; \\ 0 & \text{otherwise.} \end{cases}$$

For each policy μ and initial state s, let $\tilde{P}_{\mu,s}$ and $\tilde{E}_{\mu,s}$ be the probability measure and corresponding mathematical expectation associated with policy μ for the uniformized process. The effect of the uniformization is to force a transition every τ units of time, yielding a discrete-time process equivalent to the original one. In fact, $\psi_\mu(s)$ can be rewritten as

$$\psi_\mu(s) = \limsup_{n\to\infty} \frac{1}{n\tau} \tilde{E}_{\mu,s} \left[\sum_{i=0}^{n-1} \tilde{c}(s_i, a_i) \right]. \tag{24}$$

Most of the tools developped for discrete-time average cost dynamic programming models (see [4], chap. 7) can then be adapted to the model studied here. The dynamic programming operators can be defined as in [4] for the uniformized model, and then rewritten in terms of the original model, yielding the operators defined below.

Let $\tilde{s} \in S$ be some (arbitrary) fixed reference state. Let $\mathbb{B}_1$ be the subspace of universally measurable and lower semi-analytic functions V in $\mathbb{B}_0$ such that $V(\tilde{s}) = 0$. We define the

following dynamic programming operators: for $V \in \mathbb{B}_1$, $s \in S$, and $a \in A(s)$, let

$$\begin{aligned} B(V)(s,a) &= \tilde{c}(s,a) + \int_S V(s')\tilde{Q}(\mathrm{d}s' \mid s,a) - V(s) \\ &= \frac{\tau}{t(s,a)}\left(c(s,a) - V(s) + \int_S V(s')Q([0,\infty) \times \mathrm{d}s' \mid s,a)\right) \qquad (25) \\ J(V)(s) &= \inf_{a \in A(s)} B(V)(s,a) \qquad (26) \\ T(V)(s) &= V(s) + J(V)(s) - J(V)(\tilde{s}). \qquad (27) \end{aligned}$$

For every policy μ, let

$$\begin{aligned} J_\mu(V)(s) &= B(V)(s,\mu(s)) \qquad (28) \\ T_\mu(V)(s) &= V(s) + J_\mu(V)(s) - J_\mu(V)(\tilde{s}). \qquad (29) \end{aligned}$$

Note that $T(V)$ and $T_\mu(V)$ are also elements of $\mathbb{B}_1$, while $B(V)$, $J(V)$ and $J_\mu(V)$ are universally measurable and lower semi-analytic elements of $\mathbb{B}_0$. For $V \in \mathbb{B}_1$, $V = 0$ means $V(s) = 0$ for all $s \in S$.

The *relative value iteration* (or *successive approximation*) algorithm for the average cost case consists in choosing an initial $V_0 \in \mathbb{B}_1$, and defining recursively $V_n = T(V_{n-1})$. In the following condition C2, we *assume* that this algorithm converges to a fixed point of the operator T.

CONDITION C2. For any $V_0 \in \mathbb{B}_1$, if $V_n = T(V_{n-1})$ for $n = 1, 2, \ldots$, then $\lim_{n\to\infty} \|V_n - V_*\| = 0$ for some $V_* \in \mathbb{B}_1$ solution of the functional equation:

$$T(V) = V, \qquad (30)$$

and $\lim_{n\to\infty} J(V_n)(\tilde{s}) = J(V_*)(\tilde{s})$. ∎

This condition might seem difficult to verify, but as indicated by the following lemma, this should not be a burden for most practical applications. In most cases, indeed, we know intuitively that the *optimal* average cost should be independent of the initial state, and all computations in practice are done on "finite state" computers.

LEMMA 3. Suppose that the state and action spaces are finite. Then,

(a) The following are equivalent: (1) C2 holds; (2) $\psi_*(s)$ is independent of s; (3) $T(V) = V$ for some $V \in \mathbb{B}_1$.

(b) If all states communicate (are "weakly connected" in the sense of [23]), i.e. if for each pair s, s' in S, there is a policy μ and an integer $n > 0$ such that $P_{\mu,s}[s_n = s'] > 0$, then C2 holds.

PROOF. See Schweitzer [33] and Platzman [23]. ∎

In the general case, condition C2 also implies that the *optimal* average cost is independent of the initial state. This is stated in the following theorem. On the other hand, there might be policies having different average costs for different initial states (even for finite state and action spaces).

THEOREM 4. Under condition C1 above, if there exists a bounded function $V_* \in \mathbb{B}_1$ such that $T(V_*) = V_*$, then $J(V_*)$ is constant and the optimal average cost is given by $g_* = J(V_*)(s)/\tau$, independently of s or the initial state. Also, any policy μ_* for which $T_{\mu_*}(V_*) = V_*$ is optimal.

PROOF. From the definition of T, $T(V_*) = V_*$ is equivalent to $J(V_*)(s) = J(V_*)(\tilde{s})$ for all $s \in S$, which is equivalent to

$$\inf_{a \in A(s)} \left(\tilde{c}(s,a) - V_*(s) + \int_S V_*(s') \tilde{Q}(ds' \mid s,a) - \tau g_* \right) = 0, \tag{31}$$

where $g_* = J(V_*)(\tilde{s})/\tau$. Equation (31) corresponds to equation (3) in [30], and the remainder of the proof goes like the proof of theorem 2 in [30]. ∎

THEOREM 5. Under conditions C1 and C2, let $V \in \mathbb{B}_1$ and g, δ^-, δ^+ in $\mathbb{R}$ such that

$$\delta^- \leq J(V)(s) \leq \delta^+ \tag{32}$$

for all $s \in S$. Then,

$$\delta^- \leq g_*/\tau \leq \delta^+. \tag{33}$$

Furthermore, if μ is also a policy such that

$$J_\mu(V)(s) \leq \delta^+ \tag{34}$$

for all $s \in S$, then

$$\delta^- \leq g_*/\tau \leq \psi_\mu(s)/\tau \leq \delta^+ \tag{35}$$

for all $s \in S$, and μ is $(\delta^+ - \delta^-)/\tau$-optimal.

PROOF. Let $V_0 = V$ and for $n = 1, 2, \ldots$, let $V_n = T(V_{n-1}) = T^n(V_0)$, $\gamma_n = \inf_{s \in S} J(V_n)(s)$ and $\bar{\gamma}_n = \sup_{s \in S} J(V_n)(s)$. Let $\epsilon > 0$, $s \in S$ and $a \in A(s)$ such that

$$B(V_n)(s,a) \leq J(V_n)(s) + \epsilon.$$

Then, we have

$$\begin{aligned} J(V_n)(s) &\geq B(V_n)(s,a) - \epsilon \\ &= \tilde{c}(s,a) + \int_S V_n(s')\tilde{Q}(\mathrm{d}s' \mid s,a) - V_n(s) - \epsilon \\ &= \tilde{c}(s,a) + \int_S \left(V_{n-1}(s') + J(V_{n-1})(s')\right)\tilde{Q}(\mathrm{d}s' \mid s,a) \\ &\quad -J(V_{n-1})(\tilde{s}) - V_n(s) - \epsilon. \end{aligned}$$

On the other hand,

$$\begin{aligned} V_n(s) &= T(V_{n-1})(s) \\ &\leq V_{n-1}(s) + B(V_{n-1})(s,a) - J(V_{n-1})(\tilde{s}) \\ &= \tilde{c}(s,a) + \int_S V_{n-1}(s')\tilde{Q}(\mathrm{d}s' \mid s,a) - J(V_{n-1})(\tilde{s}). \end{aligned}$$

Combining these two inequalities, we obtain

$$J(V_n)(s) \geq \int_S J(V_{n-1})(s')\tilde{Q}(\mathrm{d}s' \mid s,a) - \epsilon.$$

Since this holds for any $s \in S$ and $\epsilon > 0$, we obtain

$$\gamma_n = \inf_{s\in S} J(V_n)(s) \geq \inf_{s\in S} J(V_{n-1})(s) = \gamma_{n-1}.$$

By a similar argument, we can also show that $\bar{\gamma}_n \leq \bar{\gamma}_{n-1}$. From condition C2 and theorem 4, we know that for all $s \in S$,

$$\lim_{n\to\infty} J(V_n)(s) = \lim_{n\to\infty} J(V_n)(\tilde{s}) = J(V_*)(\tilde{s}) = g_*/\tau.$$

Therefore,

$$\lim_{n\to\infty} \gamma_n = \lim_{n\to\infty} \bar{\gamma}_n = g_*/\tau.$$

Since $\{\gamma_n\}$ is increasing and $\{\bar{\gamma}_n\}$ is decreasing, we obtain

$$\delta^- \leq \gamma_0 \leq g_*/\tau \leq \bar{\gamma}_0 \leq \delta^+.$$

For the second part of the proof, suppose μ is a policy such that

$$J_\mu(V)(s) \leq \delta^+.$$

Define the operator H_μ by

$$H_\mu(V)(s) = V(s) + J_\mu(V)(s) = \tilde{c}(s,\mu(s)) + \int_S V(s')\tilde{Q}(\mathrm{d}s' \mid s,\mu(s)). \tag{36}$$

Then, for all $s \in S$,

$$\begin{aligned} H_\mu(V)(s) &= V(s) + J_\mu(V)(s) \;\le\; V(s) + \delta^+ \\ H_\mu^2(V)(s) &\le H_\mu(V)(s) + \delta^+ \;\le\; V(s) + 2\delta^+ \\ &\vdots \\ H_\mu^n(V)(s) &\le V(s) + n\delta^+ \\ &\vdots \end{aligned}$$

and therefore

$$\tau\psi_\mu(s) = \lim_{n\to\infty} \frac{H_\mu^n(V)(s)}{n} \le \lim_{n\to\infty} \left(\delta^+ + \frac{V(s)}{n}\right) = \delta^+.$$

The fact that $g_* \le \psi_\mu(s)$ is obvious and this yields (35). From (35), we also have $\psi_\mu(s) - g_* \le (\delta^+ - \delta^-)/\tau$, and this completes the proof. ∎

In the next section, we describe a general approach for solving the functional equations (14) and (30).

3 Value iteration and policy iteration

Value iteration and policy iteration are two general methods for solving dynamic programs like those described in the previous section. They operate as follows.

Value iteration.

Select initial V_0 in $\mathbb{B}_1$;

For $n := 1$ to $\bar{n}$ do

$$V_n := T(V_{n-1}); \tag{37}$$

Retain $\bar{\mu}$ such that $T_{\bar{\mu}}(V_{\bar{n}}) = T(V_{\bar{n}})$;

End.

Policy iteration.

Select initial policy μ_0;

For $n := 1$ to $\bar{n}$ do

Policy evaluation: find V such that

$$T_{\mu_{n-1}}(V) = V; \tag{38}$$

Policy update: find μ_n such that

$$T_{\mu_n}(V) = T(V); \tag{39}$$

Retain $\mu_{\bar{n}}$;

End.

In both cases, the value of $\bar{n}$ may be chosen in advance or depend on some stopping criterion. The operators T and T_μ are defined in (8–9) for the discounted case, and in (27, 29) for the average cost case.

Value iteration converges to V_* for all the models studied in the previous section, under conditions C or LC for the discounted case (taking $V_0 \in \mathbb{B}_*$), or under C1 and C2 for the average cost case. But for policy iteration, there might be intermediate policies μ (not optimal), reached at some iteration, for which V_μ is infinite for some states (under LC), or for which the average cost depends on the initial state (under C1 and C2). In the latter case, for the average cost model, μ is *periodic* or *multichain*, and there is no V for which

$T_\mu(V) = V$. However, there exists adaptations of the policy iteration algorithm that can work for that case (see e.g. [10]).

Obviously, for continuous (or very large) state spaces, these algorithms cannot be applied exactly in general. Some form of approximation must be used. Since solving (38) is usually too difficult when the state space is very large, the use of value iteration has been advocated for that case [13, 32]. For continuous state spaces, Haurie and L'Ecuyer [13] (see also [16, 18]) compute (37) at a finite number of points in the state space (using numerical integration), and use these values to approximate V_n over the whole state space. This approximation is used in the next iteration and the process is repeated. This approach is very time consuming.

It is also well known that value iteration converges linearly (sometimes very slowly), while policy iteration (when it works) is equivalent to applying Newton's method to the equation $T(V) - V = 0$ (see [26, 27]). When V is not too far from V_*, it typically has quadratic convergence. For that reason, trying to adapt policy iteration for the case of large state spaces has been a subject of interest in the recent years. One adaptation is the so-called "modified policy iteration" method [20, 26], where at each iteration, (38) is solved only approximately by applying only a few iterations of the value iteration method with a fixed policy μ_{n-1}, starting from the previous V. In the next section, we examine this idea, combined with a finite element approximation of the value function, in the general setting of section 2.

4 A finite element approach

We now introduce an approximate policy iteration algorithm, with finite element approximation of the value function. Generally speaking, we assume that the value function can be approximated reasonably well by a linear combination of a small set of (already known) functions. Typically, these functions will be *local*, in the sense that their support will be a (small) subregion of S (with some exceptions e.g. for the case of unbounded state spaces). For more details on the finite element method in general and specific presentation of various finite element schemes, see e.g. [14].

4.1 Approximation of $\mathbb{B}_1$ by a finite dimensional space

Let $\Phi = \{B_1, \ldots, B_J\} \subset \mathbb{B}_1$ be a finite set of linearly independent functions, and let $\mathbb{B}_1^\Phi$ be the subspace of $\mathbb{B}_1$ spanned by the B_j's, i.e. $\mathbb{B}_1^\Phi$ is the set of functions V that can be expressed as

$$V(s) = \sum_{j=1}^{J} d_j B_j(s) \tag{40}$$

for some real constants $d_1, \ldots, d_J$. Let ψ be a measure on S, such that $\psi(S) = \int_S \psi(\mathrm{d}s)$ is finite and strictly positive. For each policy μ and each pair (V, W) in $\mathbb{B}_1 \times \mathbb{B}_1$, we define the scalar product

$$\langle V, W \rangle_\mu = \int_S (T_\mu(V) - V)(s) W(s) \psi(\mathrm{d}s). \tag{41}$$

Note that $T_\mu(V) = V$ $\psi(\cdot)$-almost everywhere in S is equivalent to $\langle V, W \rangle_\mu = 0$ for all W in $\mathbb{B}_1$. Instead of solving $T_\mu(V) = V$, we will examine the latter. Solving it is quite difficult in general and what we will do instead is to solve an approximate version of the problem: find V in $\mathbb{B}_1^\Phi$ such that $\langle V, W \rangle_\mu = 0$ for all W in $\mathbb{B}_1^\Phi$, or equivalently, such that $\langle V, B_j \rangle_\mu = 0$ for $j = 1, \ldots, J$. This gives rise to J equations with J unknowns (the unknowns are the coefficients defining V).

4.2 Building a linear system

If we replace V by the expression (40) in the equation $\langle V, B_i \rangle_\mu = 0$, we obtain the equation

$$b_\mu(i) + \sum_{j=1}^{J} m_\mu(i, j) d_j = 0, \tag{42}$$

where for the discounted case, we have:

$$b_\mu(i) = \int_S c(s, \mu(s)) B_i(s) \psi(\mathrm{d}s)$$

and

$$m_\mu(i,j) = \int_S \left(-B_j(s) + \int_{[0,\infty)\times S} B_j(s') e^{-\rho\zeta} Q(d\zeta \times ds' \mid s, \mu(s)) \right) B_i(s) \psi(\mathrm{d}s),$$

while for the average cost case:

$$b_\mu(i) = \int_S \left(\tilde{c}(s, \mu(s)) - \tilde{c}(\tilde{s}, \mu(\tilde{s})) \right) B_i(s) \psi(\mathrm{d}s)$$

and

$$\begin{aligned} m_\mu(i,j) &= \int_S \Big(-B_j(s) + \int_S B_j(s') \tilde{Q}(\mathrm{d}s' \mid s, \mu(s)) \\ &\qquad - \int_S B_j(s') \tilde{Q}(\mathrm{d}s' \mid \tilde{s}, \mu(\tilde{s})) \Big) B_i(s) \psi(\mathrm{d}s) \\ &= \int_S \left(\frac{\tau}{t(s,\mu(s))} \left(-B_j(s) + \int_S B_j(s') Q([0,\infty) \times \mathrm{d}s' \mid s, \mu(s)) \right) \right. \\ &\qquad \left. - \frac{\tau}{t(\tilde{s},\mu(\tilde{s}))} \int_S B_j(s') Q([0,\infty) \times \mathrm{d}s' \mid \tilde{s}, \mu(\tilde{s})) \right) B_i(s) \psi(\mathrm{d}s) \end{aligned}$$

(assuming that these quantities exist and are finite). For a fixed μ, let b and d be the column vectors $b = (b_\mu(1), \ldots, b_\mu(J))'$ and $d = (d_1, \ldots, d_J)'$, and let M be the $J \times J$ matrix whose element (i,j) is $m_\mu(i,j)$. Note that b and M depend on μ. The solution of $T_\mu(V) = V$ will be approximated by V defined in (40), where d is a solution of

$$b + Md = 0. \tag{43}$$

4.3 Approximate policy iteration

In general, policies must also be approximated: it is usually not possible to find μ such that $T_\mu(V) = T(V)$ *exactly*. As we did for the state space, we can define a finite dimensional subspace of the policy space U, and consider only the policies that belong to that subspace. Giving more details on how to do that would require further specification of the structure of the action space A. Note that A is not necessarily a subset of $\mathbb{R}$. In some cases, it can even be a functional space. (The action a can be for instance a continuous time control to be applied until the next transition, which can be viewed as a stopping time.) Since this is

rather problem-dependent, we will content ourselves with the following approach. For any V in $\mathbb{B}_1$ and $\epsilon > 0$, define

$$\Delta_\epsilon(V) = \{\mu \in U \mid T_\mu(V)(s) \leq T(V)(s) + \epsilon \text{ for all } s \in S\}\,. \tag{44}$$

This is the set of policies for which for each state s, the decision $\mu(s)$ brings us no more than ϵ away from the infimum in the definition of $T(V)$. At every "policy update" step of the policy iteration algorithm (equation (39)), we will in fact seek a new policy in $\Delta_\epsilon(V)$, for some value of ϵ. Often, in practice, we will first find a policy μ (by "approximate" optimization) and then *estimate* the smallest ϵ for which $\mu \in \Delta_\epsilon(V)$.

Under this setting, the (approximate) policy iteration algorithm now becomes:

Algorithm 1. (General form)
Select $\epsilon > 0$, initial policy μ, and initial V in $\mathbb{B}_1$;
If average cost, select $\tilde{s} \in S$;
Loop
Select $\Phi = \{B_1, \ldots, B_J\} \subset \mathbb{B}_1$ and the measure ψ on S;
Compute b and M;
Solve approximately: $b + Md = 0$ for d;
Define $V \in \mathbb{B}_1^\Phi$ by equation (40);
Select ϵ_1 and find new μ in $\Delta_{\epsilon_1}(V)$;
If desired, perform a stopping test:
If discounted case, use e.g. (18) and (20) to compute or estimate
bounds on V_* and V_μ: $V_\mu - V_* \leq \bar{\epsilon}$;
If average cost, use (35) to compute or estimate bounds
on g_* and ψ_μ: $\tau\delta^- \leq g_* \leq \psi_\mu(s) \leq \tau\delta^+$; $\bar{\epsilon} = (\delta^+ - \delta^-)/\tau$;
If $\bar{\epsilon} \leq \epsilon$, or other stopping criteria satisfied, stop;
Endloop
End.

Obviously, as it stands, this algorithm is not completely defined. For instance, the stopping criteria, the way of choosing ϵ, ϵ_1, J, Φ and ψ, the integration method used to compute b and M, the techniques used to solve the linear system (perhaps approximately), to find a new μ (perform the minimization) and to compute (or estimate) $\bar{\epsilon}$, are all left open. These are usually problem dependent. In practice, they may vary from iteration to iteration. Often, the stopping test can be costly and should not be performed at each

iteration. Sometimes, $\bar{\epsilon}$ has to be estimated heuristically, for instance as in [13]: Recompute $T(V)$ and $T_\mu(V)$ at a large number of new points, compute the approximation error at these points, and take the largest and smallest to estimate δ^+, δ^-, $\phi^+(x)$, $\phi^-(x)$ and ϵ_1. The function W in theorem 2 can be taken for instance as $T(V)$, or as the next value of V.

In practice, instead of storing the retained policy, one can store only the vector d. An appropriate decision for a given state s can be recovered as needed using (40) in (8) or (26). The selected decision must be one for which the minimum is attained in (8) or (26) (or, if approximations are used, one for which we get close to the infimum).

4.4 A useful special case

One particular choice for the measure ψ is to select a finite number of points $\sigma_1, \ldots, \sigma_I$ in S, a set of positive weights $\varphi_1, \ldots, \varphi_I$, and for each $\bar{S}$ subset of S, define

$$\psi(\bar{S}) = \sum_{\{i|\sigma_i \in \bar{S}\}} \varphi_i.$$

In this case, the scalar product becomes

$$\langle V, W \rangle_\mu = \sum_{i=1}^{I} (T_\mu(V) - V)(\sigma_i) W(\sigma_i) \varphi_i$$

and the outer integrals in the definitions of $b_\mu(s)$ and $m_\mu(s, j)$ are replaced by sums. Typically, when using a finite element approach, the σ_i's will be the element nodes. We will have $I = J$ and each φ_i equal to 1. Suppose moreover that $B_j : S \to [0, 1]$ for each j, $B_j(\sigma_i) = \delta_{ij}$ (the Kronecker delta), and $\sum_{j=1}^{J} B_j(s) = 1$ for all s in S (most usual finite element schemes have these properties [14]).

As we will see below, one interesting point of this special case is that usually, the eigenvalues of M are such that the system (43) can be solved (approximately) by standard iterative methods, like e.g. Jacobi or Gauss-Siedel iteration [25]. Let $\bar{M} = M + I$, where I denotes the identity matrix, and note that (43) is equivalent to $d = b + \bar{M}d$. One iterative method, called pre-Jacobi, is simply defined by applying iteratively the affectation

$$d := b + \bar{M}d, \tag{45}$$

starting with some initial vector d. Moreover, even if we change σ and Φ between two iterations, it is easy to compute the value of d corresponding to the new σ (and the same V) from the previous one and to continue iterating with it. More specifically, suppose that

$\sigma = (\sigma_1, \ldots, \sigma_I)'$ is changed to $\tilde{\sigma} = (\tilde{\sigma}_1, \ldots, \tilde{\sigma}_K)'$. Then, the new d is $\tilde{d} = (\tilde{d}_1, \ldots, \tilde{d}_K)'$, where $\tilde{d}_i = V(\tilde{\sigma}_i) = \sum_{j=1}^{J} d_j B_j(\tilde{\sigma}_i)$. Note that if Φ is changed, this expression is independent of the new Φ.

Under these assumptions, for the discounted case, we obtain:

$$b_\mu(i) = \sum_{k=1}^{J} c(\sigma_k, \mu(\sigma_k)) B_i(\sigma_k) = c(\sigma_i, \mu(\sigma_i)),$$

$$\begin{aligned} m_\mu(i,j) &= \sum_{k=1}^{J} \left(-B_j(\sigma_k) + \int_{[0,\infty)\times S} B_j(s') e^{-\rho\zeta} Q(d\zeta \times ds' \mid \sigma_k, \mu(\sigma_k)) \right) B_i(\sigma_k) \\ &= -\delta_{ij} + \int_{[0,\infty)\times S} B_j(s') e^{-\rho\zeta} Q(d\zeta \times ds' \mid \sigma_i, \mu(\sigma_i)) \\ &\geq -\delta_{ij} \end{aligned}$$

and

$$\begin{aligned} 1 + \sum_{j=1}^{J} m_\mu(i,j) &= \int_{[0,\infty)\times S} \left(\sum_{j=1}^{J} B_j(s') \right) e^{-\rho\zeta} Q(d\zeta \times ds' \mid \sigma_i, \mu(\sigma_i)) \\ &= \int_{[0,\infty)\times S} e^{-\rho\zeta} Q(d\zeta \times ds' \mid \sigma_i, \mu(\sigma_i)) \\ &\leq 1. \end{aligned}$$

Hence, $\bar{M}$ is a substochastic matrix. In practice, in most cases, it also has a spectral radius $\rho(\bar{M}) < 1$, in which case (45) converges geometrically. But there might be policies μ for which $\bar{M}$ has a spectral radius of one. These (rare) cases correspond to policies μ for which the total expected discounted cost for the discretized model becomes infinite. Such policies are never optimal (at least for the discretized model) and if they are encountered, it is most likely to be at the early iterations.

For the average cost case, we obtain in a similar fashion:

$$b_\mu(i) = \tilde{c}(\sigma_i, \mu(\sigma_i)) - \tilde{c}(\tilde{s}, \mu(\tilde{s})),$$

$$m_\mu(i,j) = -\delta_{ij} + \int_S B_j(s') \tilde{Q}(\mathrm{d}s' \mid \sigma_i, \mu(\sigma_i)) - \int_S B_j(s') \tilde{Q}(\mathrm{d}s' \mid \tilde{s}, \mu(\tilde{s})).$$

We will now suppose that the state $\tilde{s}$ is one of the σ_i's. Without loss of generality, say $\tilde{s} = \sigma_1$. Define

$$\tilde{m}_\mu(i,j) = \int_S B_j(s') \tilde{Q}(\mathrm{d}s' \mid \sigma_i, \mu(\sigma_i)).$$

Let $\tilde{M}$ be the $J \times J$ stochastic matrix formed by these elements, and let $\tilde{m}'$ be its first row. Then the iteration (45) can be rewritten as

$$\tilde{g} := \tilde{m}'d; \tag{46}$$

$$d := b + \tilde{M}d - \mathbf{1}'\tilde{g} \tag{47}$$

where $\mathbf{1}$ is a vector of ones. The scheme (46, 47) is in fact a relative value iteration scheme applied to the finite state discrete time Markov chain defined by the transition matrix $\tilde{M}$ and cost vector b. Its convergence is geometric, with a rate determined by the subdominant (second largest in norm) eigenvalue of $\tilde{M}$ (see [21]), so long as this subdominant eigenvalue is inside the unit circle. Since $\tilde{M}$ is stochastic, its largest eigenvalue is always 1. When the norm of the subdominant one is also 1, this indicates that the Markov chain corresponding to $\tilde{M}$ is multichain or periodic. Periodicity can be eliminated by taking a slightly smaller value of τ. Multichain matrices can still be encountered (rarely) during the iterations, and we will mention below some heuristic ways to cope with that.

For any V in $\mathbb{B}_1$, σ in S^I and $\epsilon > 0$, define

$$\Delta_\epsilon(V, \sigma) = \{\mu \in U \mid T_\mu(V)(\sigma_i) \leq T(V)(\sigma_i) + \epsilon \text{ for } i = 1, \ldots, I\}. \tag{48}$$

This is the set of policies for which for each "selected" state σ_i, the decision $\mu(\sigma_i)$ brings us no more than ϵ away from the infimum in the definition of $T(V)$.

Under this setting, algorithm 1 for the special case can be rewritten:

Algorithm 1. (Special case)
Select $\epsilon > 0$, initial policy μ, and initial V in $\mathbb{B}_1$;
If average cost, select $\tilde{s} \in S$;
Loop (outer loop):
Select J, $\sigma = (\sigma_1, \ldots, \sigma_J)' \in S^J$ and $\Phi = \{B_1, \ldots, B_J\} \subset \mathbb{B}_1$ such that
$\sigma_1 = \tilde{s}$, $B_j : S \to [0, 1]$, $B_j(\sigma_i) = \delta_{ij}$ and $\sum_{j=1}^J B_j(s) = 1$ for all $s \in S$;
Compute b, $\bar{M}$ and $d := (V(\sigma_1), \ldots, V(\sigma_J))'$;
Inner loop: select k and repeat k times: $d := \bar{M}d + b$;
Define $V \in B_1^\Phi$ by equation (40);
Select ϵ_1 and find new μ in $\Delta_{\epsilon_1}(V, \sigma)$;
If desired, perform a stopping test: (same as for the general case);
Endloop
End.

For $k = 1$, this algorithm becomes the value iteration (or successive approximation) algorithm, as described in [13]. For $k = \infty$, we get policy iteration. Like for the general version, many things are left open. A good choice of k is problem dependent. It could be chosen adaptively, based on the previous iterations. Intuitively, the more costly it is to compute b and $\bar{M}$, the larger the value of k should be. But the inner loop should also stop when progress gets too slow, i.e. when d is not changing significantly enough anymore, or if d does not appear to converge geometrically.

One may wonder why we are coming back to a linearly convergent iterative algorithm to (partially) solve the linear system, in the policy evaluation step, while our primary motivation to adopt the policy iteration algorithm was its quadratic convergence rate! The problem is that for large state spaces, solving the linear system is difficult. However, the gain with respect to straightforward value iteration can still be impressive due to the fact that performing one iteration of the inner loop (45) is usually much less costly than performing the iteration (37). In algorithm 1 above, the numerical integrations to compute $\bar{M}$ are performed only once every outer loop. In value iteration, since the "minimizing" policy usually changes every iteration (at least for continuous action spaces), together with V, the integrals must be recomputed. This often accounts for most of the computational costs.

The choice of σ determines a grid over the state space S and the σ_i's are the *nodes* of the finite elements [14]. Intuitively, a coarser grid should be chosen at the early stages of the algorithm and the grid should be refined only when progress is stalling. Multigrid techniques [7] can also be used: it is often worthwhile to get back to a coarser grid to make "corrections" when progress becomes too slow with a fine grid. Note that the inner loop can be supplemented with aggregation steps, using e.g. the adaptive aggregation method proposed in [4, 5]. Various other techniques for the iterative solution of linear systems can also be used, e.g. overrelaxation, reordering, etc. [25].

Many good choices for σ and Φ are usually available and details on this are covered by a voluminous finite element literature. Aboundant software also exists, some of which aimed at automatic grid construction. However, this software has been designed primarily to solve partial differential equations. Usually, in that context, the scalar product is not defined as in section 4.2, but is typically bilinear, symmetric, and possesses some other nice properties [14]. The matrix of the resulting linear system is usually symmetric and sparse. Here, M is not symmetric in general. Its sparsity depends not only on the "locality" of the base functions B_j, but also on the stochastic kernel Q. Row i of M will be sparse if the set of states reachable in one transition from state σ_i intersects the support of just a few of the B_j's.

Normally, the bounds on V_* should get closer at each iteration of the outer loop, but this is not guaranteed. For the discounted case, from theorems 1 and 2, it follows that for $k = 1$, if the error of approximation of $T(V)$ by W and the difference between W and the next V go to 0 with the iteration number, then $\|V - V_*\|$ converges to 0. For version C of the model, if the sequence of values of ϵ_1 also goes to 0, then for any $\epsilon > 0$, an ϵ-optimal policy is obtained after a finite number of iterations (see also [13, theorem 3.3]). For the average cost case, for $k = 1$, the algorithm becomes relative value iteration, whose convergence follows from condition C2 (assuming that the discretization error goes to 0). In general, one way to "insure" convergence is to adopt the following rule: whenever the distance between the bounds on V_* (in terms of norm) or on g_* is not diminishing at a given iteration, with respect to the best value obtained before, take $k = 1$ for the next iteration. Of course, this is only one way of doing it. In practice, it is also sometimes possible to detect these infinite cost or multichain policies for which (45) might not converge. An alternative heuristic in this case is to modify them slightly so that $\rho(\bar{M}) < 1$ (discounted case) or to make $\tilde{M}$ unichain (undiscounted case). Since any optimal policy should have the latter property, this could usually be done without impairing the algorithm.

For the average cost case, the (two step) scheme (46, 47) is usually preferable to the direct application of (45), because $\tilde{M}$ is usually sparser than $\bar{M}$. In the algorithm, we have assumed that $\tilde{s} = \sigma_1$, but of course, this is not necessary. Things are easier, though, if $\tilde{s}$ is one of the σ_i's, since $\tilde{g}$ can then be computed using the corresponding row of $\tilde{M}$.

4.5 An alternative regression approach

For a finite state and action space model, Schweitzer and Seidmann [34] have proposed a regression approach for polynomial approximation of the value function. A direct generalization of their approach leads to the following. We select a vector of states $\sigma = (\sigma_1, \ldots, \sigma_I)$, for $I \geq J$, and write the expressions $T_\mu(V)(\sigma_i) - V(\sigma_i)$, for $i = 1, \ldots, I$. If we replace V by the expression (40) in these expressions and integrate, we obtain I affine forms in terms of $d_1, \cdots, \mathrm{d}_J$. In matrix form, this can be written as say $b + Md$ (where b and M are not the same as in the previous subsections).

If $I = J$, (38) could be replaced by the linear system $b + Md = 0$. If $I > J$, the latter system would have more equations than unknowns. Rather, we can minimize a (possibly weighted) sum of squares:

$$(Md + b)'W(Md + b) \tag{49}$$

where $W = \text{diag}(w_1, \ldots, w_I)$ is a diagonal matrix with positive diagonal elements. A trivial choice is $W = I$ (the identity matrix), which gives equal weights to all points. The sum of squares (49) is minimized when

$$M'Wb + M'WMd = 0. \tag{50}$$

Note that the matrix $M'WM$ is symmetric. In the previous subsections, the matrices M and $\bar{M}$ were not symmetric in general. If $I = J$ and M^{-1} exists, (50) reduces to $b + Md = 0$.

In [34], S is assumed to be finite and $S = \{\sigma_1, \ldots, \sigma_I\}$. In that case, a sufficient condition for $M'WM$ to be invertible is that $B_1, \ldots, B_J$ are linearly independent [34, lemma 1]. In our case, this result does not hold in general, but a sufficient condition is that M has full rank. Indeed, $M'WM$ not invertible means that there exists a vector $y \neq 0$ such that $M'WMy = 0$. Hence, $y'M'WMy = 0$ and since W is diagonal with only positive elements on the diagonal, $My = 0$, which means that M is not of full rank. Even when $M'WM$ is not invertible, a solution of (50) (that minimizes (49)) can be computed, but the solution is not unique.

Note that this regression approach should be viewed as an *heuristic*. There is no guarantee of convergence to the optimal solution, even when $S = \{\sigma_1, \ldots, \sigma_I\}$. Sometimes, the solution may even deteriorate from one iteration to the next. Deciding what to do when $\bar{\epsilon}$ is not getting small enough is essentially a matter of art. A problem can also occur in the average cost case if a multichain policy is encountered at some iteration. One possible (heuristic) remedy in that case is to modify the policy slightly to make it unichain (it is not always trivial, though, to recognize that a policy is multichain).

Other variants of this approach are also proposed by Schweitzer and Seidmann [34] for finite state and decision spaces: linear programming (LP) and global least-squares fit (GLSF). LP might work well for relatively small state and decision spaces. GLSF is similar to the approach that we have described in this section, and it also offers a guaranteed improvement at each iteration. However, it requires much more work per iteration. The basic idea is that after solving (50) for a, one performs a linear search along the direction that links this d and the previous d, to minimize (49) where M and b depend on μ, μ is the policy for which the minimum is attained in the definition of $T(V)$, and V is defined by (40). Note that V, μ, M and b vary during the linear search.

5 Conclusion

We have described a finite element approach to solve MRDP models with continuous or very large state spaces. It can deal with most reasonably smooth value functions, even if we don't have an a priori idea of their shapes, provided that the state space is bounded and has few (continuous) dimensions. One can also deal with certain non continuous or non differentiable value functions by a proper placement of the element boundaries. For high dimensional state spaces, that kind of approach can also be used if the value function can be approximated reasonably well by a linear combination of a small set of (a priori known) base functions, or if only a rough approximation is sufficient. Note that the dimension of the state space can sometimes be reduced using special techniques. For instance, Saad and Turgeon [31] used principal component analysis to define a linear mapping from S to a smaller dimensional state space $\tilde{S}$, and defined the value function V only on $\tilde{S}$.

Defining the scalar product as we have done in equation (41) is only one way of doing it. There are certainly other interesting ways, leading to different linear systems, that should be explored.

In some numerical experiments, speedups by factors of between 10 and 20 were achieved by using the modified policy iteration algorithm as suggested here, compared to the more naïve value iteration approach described in [13] with an optimization step at each iteration (using similar grids and finite elements in both cases). See [18, 19] for more details. The example in [19] concerns the scheduling of a robot servicing a set of machines on a line. It arises in the context of a textile mill, where the machines are identical winding heads. The state space in that case is comprised of a finite number of closed intervals on the real line. On each interval, the (optimal) value function is continuous, but not differentiable. In [19], it was approximated by a piecewise linear function. The example in [18] deals with the dynamic optimization of checkpointing times for database systems. In one numerical illustration, the state space is the union of a finite segment of the real line, with a rectangle in the plane. That rectangle was partitioned into subrectangles, and a bilinear approximating function was used on each subrectangle.

Acknowledgements

Part of this work was done while the author was enjoying the hospitality of the group "Méta-2" at INRIA, Rocquencourt, France. Another part was done while visiting the Operations Research Department at Stanford University. It has been supported by NSERC-Canada grant # A5463 and FCAR-Québec grant # EQ2831. The author wish to thank J. P. Quadrat and M. Goursat for valuable suggestions, and M. Mayrand for his contribution to the ideas of section 4.4.

References

[1] Bellman, R., Kalaba, R. and Kotkin, B. "Polynomial Approximation — A New Computational Technique in Dynamic Programming", *Math. of Computation*, **17**, 8 (1963), 155–161.

[2] Bertsekas, D. P. "Convergence of Discretization Procedures in Dynamic Programming", *IEEE Trans. on Automatic Control*, **AC-20** (1975), 415–419.

[3] Bertsekas, D. P. and Shreve, S. E. *Stochastic Optimal Control: The Discrete Time Case*. Academic Press, New-York, 1978.

[4] Bertsekas, D. P. *Dynamic Programming: Deterministic and Stochastic Models*. Prentice Hall, 1987.

[5] Bertsekas, D. P. and Castañon, D. A. "Adaptive Aggregation for Infinite Horizon Dynamic Programming", *IEEE Transactions on Automatic Control*, **34**, 6 (1989), 589–598.

[6] Breton, M. and L'Ecuyer, P. "Noncooperative Stochastic Games Under a N-stage Local Contraction Assumption", *Stochastics*, **26** (1989), 227–245.

[7] Briggs, W. L. *A Multigrid Tutorial*, SIAM, Philadelphia, 1987.

[8] Daniel, J. W. "Splines and Efficiency in Dynamic Programming", *J. Math. Anal. Appl.*, **54** (1976), 402–407.

[9] Denardo, E. V. "Contraction Mappings in the Theory Underlying Dynamic Programming", *SIAM Review*, **9** (1967), 165–177.

[10] Federgruen, A. and Spreen, D. "A New Specification of the Multichain Policy Iteration Algorithm in Undiscounted Markov Renewal Programs", *Management Science*, **26**, 12 (1980), 1211–1217.

[11] Fox, B. L. "Discretizing Dynamic Programs", *J. Optim. Theory and Appl.*, **II** (1973), 228–234.

[12] Haurie, A. and L'Ecuyer, P. "A Stochastic Control Approach to Group Preventive Replacement in a Multicomponent System", *IEEE Trans. on Automatic Control*, **AC-27**, 2 (1982), 387–393.

[13] Haurie, A. and L'Ecuyer, P. "Approximation and Bounds in Discrete Event Dynamic Programming", *IEEE Transactions on Automatic Control*, **AC-31**, 3 (1986), 227–235.

[14] Hugues, T. J. R., *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*, Prentice-Hall, Englewood, New Jersey, 1987.

[15] L'Ecuyer, P. and Haurie, A. "Discrete Event Dynamic Programming in Borel Spaces with State Dependent Discounting", Report no. DIUL-RR-8309, Département d'Informatique, Univ. Laval, 1983.

[16] L'Ecuyer, P. and Haurie, A. "The Repair vs Replacement Problem: A Stochastic Control Approach", *Optimal Control Applications and Methods*, **8** (1987), 219–230.

[17] L'Ecuyer, P. and Haurie, A. "Discrete Event Dynamic Programming with Simultaneous Events", *Math. of Oper. Research*, **13**, 1 (1988), 152–163.

[18] L'Ecuyer, P. and Malenfant, J. "Computing Optimal Checkpointing Strategies for Rollback and Recovery Systems", *IEEE Trans. on Computers*, **C-37**, 4 (1988), 491–496.

[19] L'Ecuyer, P., Mayrand, M. and Dror, M. "Dynamic Scheduling of a Robot Servicing Machines on a One-Dimensional Line", submitted for publication, 1988.

[20] Morton, T. E., "Undiscounted Markov Renewal Programming Via Modified Successive Approximation", *Oper. Res.*, **19** (1971), 1081–1089.

[21] Morton, T. E. and Wecker, W. E., "Discounting, Ergodicity and Convergence for Markov Decision Processes", *Managament Science*, **23**, 8 (1977), 890–900.

[22] Morin, T. L. "Computational Advances in Dynamic Programming", in *Dynamic Programming and its Applications*, M. L. Puterman Ed., Academic Press, New-York (1978), 53–90.

[23] Platzman, L. "Improved Conditions for Convergence in Undiscounted Markov Renewal Programming", *Operations research*, **25** (1977), 529–533.

[24] Porteus, E. L. "Bounds and Transformations for Finite Markov Decision Chains", *Operations Research*, **23** (1975), 761–784.

[25] Porteus, E. L. "Computing the Discounted Return in Markov and Semi-Markov Chains", *Nav. Res. Log. Quart.*, **28**, 4 (1981), 567–578.

[26] Puterman, M. L. and Shin, M. C. "Modified Policy Iteration Algorithms for Discounted Markov Decision Problems", *Management Science*, **24**, 11 (1978), 1127–1137.

[27] Puterman, M. L. and Brumelle, S. L. "On the Convergence of Policy Iteration in Stationary Dynamic Programming", *Math. of Oper. Research*, **4** (1979), 60–69.

[28] Rishel, R. "Group Preventive Maintenance: An Example of Controlled Jump Processes", *Proc. 20th IEEE Conf. on Decision and Control*, San Diego, CA, Dec. (1981), 786–791.

[29] Ross, S. M. *Applied Probability Models with Optimization Applications*, Holden Day, 1970.

[30] Ross, S. M. "Average Cost Semi-Markov Decision Processes", *J. Applied Probability*, **7** (1970), 649–656.

[31] Saad, M. and Turgeon, A., "Application of Principal Component Analysis to Long-Term Reservoir Management", *Water Resources Research*, **24**, 7 (1988), 907–912.

[32] Schweitzer, P. J. "Iterative Solution of the Functional Equations of Undiscounted Markov Renewal Programming", *J. Math. Anal. Appl.*, **34** (1971), 495–501.

[33] Schweitzer, P. J. "On the Existence of Relative Values for Undiscounted Markovian Decision Processes with a Scalar Gain Rate", *J. Math. Anal. Appl.*, **104** (1984), 67–78.

[34] Schweitzer, P. J. and Seidmann, A. "Generalized Polynomial Approximations in Markovian Decision Processes", *J. Math. Anal. Appl.*, **110** (1985), 568–582.

[35] Whitt, W. "Approximations of Dynamic Programs I and II", *Math. of Oper. Research*, **3** (1978), 231–243, and **4** (1979), 179–185.