September 1986 LIDS-P-1607

Proceedings of the 25th IEEE Conference on Decision and Control, Athens, Greece, December 1986

ADAPTIVE AGGREGATION METHODS FOR DISCOUNTED DYNAMIC PROGRAMMING

by

Dimitri P. Bertsekas*

David A. Castañon**

* Department of Electrical Engineering and Computer Science
Laboratory for Information and Decision Systems
Massachusetts Institute of Technology
Cambridge, MA 02139

**ALPHATECH, Inc.
111 Middlesex Turnpike
Burlington, MA 01803

This work was sponsored by the Office of Naval Research under contract no. N00014-84-C-0577

ABSTRACT

We propose a class of iterative aggregation algorithms for solving discounted dynamic programming problems. The idea is to interject aggregation iterations in the course of the usual successive approximation method. An important new feature that sets our method apart from earlier proposals is that the aggregate groups of states change adaptively from one aggregation iteration to the next, depending on the progress of the computation. This allows acceleration of convergence in difficult problems involving multiple ergodic classes for which methods using fixed groups of aggregate states are ineffective. No knowledge of special problem structure is utilized by the algorithms.

SECTION 1: Introduction

We consider a Markov chain with finite state space $S = \{1,...,n\}$ and transition probability matrix P. Let $\alpha \in (0,1)$ be a discount factor and $g \in R^n$ be a given cost vector. We want to find the unique solution $J \in R^n$ of the equation

$$J = T(J) \equiv g + \alpha P J. \qquad (1)$$

This is the discounted dynamic programming equation [1] associated with a single policy. We discuss later the extension of our method to the case of multiple policies.

Equation 1 can be solved by a direct method such as Gaussian elimination. However, in the absence of specific structure, the solution requires $O(n^3)$ operations, and is impractical for large n. The alternative is to use iterative techniques such as the successive approximation method having a typical cost of $O(n^2)$ per iteration for dense matrices P (see the survey [2]). Several interesting proposals for accelerating the convergence of iterative methods are based on aggregation- disaggregation ideas (Miranker [4], Chatelin and Miranker [5], Schweitzer, Puterman and Kindle [6], Verkhovsky [7], and others [8]). In [5], Chatelin and Miranker describe the basic aggregation technique and derive a bound for the error reduction. However, they do not provide a specific algorithm for selecting the directions of aggregation or disaggregation. In [7], Verkhovsky proves the convergence of an aggregation method which uses the current estimate of the solution J as a direction of aggregation, and a positive vector as the direction for disaggregation. This idea was extended in [6] by selecting fixed segments of the current estimate J as directions for aggregation, and certain nonnegative vectors as directions for disaggregation. By using the current estimate of the solution to generate directions for aggregation, the iteration becomes inherently nonlinear, and convergence is difficult to establish.

There is an important difference between our algorithms and those developed by the previous authors. In our work, aggregation and disaggregation directions are selected adaptively based on the progress of the algorithm. In particular, the membership of a particular state in an aggregate group changes dynamically throughout the iterations. This is in contrast with Schweitzer, Puterman and Kindle's approach, where the aggregate groups are fixed through all iterations. We show via analysis and experiment that the adaptive aggregate group formation feature of our algorithm is essential in order to achieve convergence acceleration for difficult problems involving multiple ergodic classes. For example, when P is the n x n identity matrix no algorithm with fixed aggregate groups can achieve a geometric convergence rate better than α. By contrast, our algorithm converges at a rate better than $2\alpha/m$ where m is the number of aggregate groups.

The rest of this paper is organized as follows. In section 2, we provide some background material on iterative algorithms for the solution of eq. 1, including bounds on the solution error. In section 3, we derive the equations of aggregation and disaggregation as in [5], and obtain different characterizations of the error reduction produced by an aggregation step. In section 4, we describe and motivate the adaptive procedure used to select the directions of aggregation and disaggregation. Section 5 analyzes in detail the error in the aggregation procedure when two aggregate groups are used. In section 6, we discuss and justify the general iterative algorithm combining adaptive aggregation steps with successive approximation steps. Section 7 contains a discussion of experimental results.

SECTION 2: Successive Approximation and Error Bounds

The successive approximation method for the solution of eq. 1 starts with an arbitrary vector J, and successively computes T(J), $T^2(J)$, ... where the mapping T was defined in eq. 1. Since P is a stochastic matrix and $\alpha \in (0,1)$, it follows that T is a contraction mapping with modulus α. Hence, we have

$$T^k(J) = T(T^{k-1}(J)) \tag{2}$$

$$\lim_{k\to\infty} T^k(J) = J^* \tag{3}$$

where J* is the solution of equation 1. The difficulty with equation 3 is that the matrix P is stochastic, and hence has spectral radius of 1. Hence, eq. 3 will converge at a geometric rate α, which is quite slow when α is close to 1. The rate of convergence can often be substantially improved thanks to the availability of error bounds derived by McQueen [9] and Porteus [3] (see [1] for a concise derivation) which we discuss next.

Let J(i) denote the ith component of the vector J. Let e denote the n-dimensional vector of all ones. The McQueen-Porteus bounds are based on the residual difference of T(J) and J. Let γ and β be defined as

$$\gamma = \min_i [T(J)(i) - J(i)] \tag{4}$$

$$\beta = \max_i [T(J)(i) - J(i)] \tag{5}$$

Then, the solution J* of eq. 1 satisfies

$$T(J) + \frac{\alpha\gamma e}{1-\alpha} \leq J^* \leq T(J) + \frac{\alpha\beta e}{1-\alpha} \tag{6}$$

Furthermore, the bounds of eq. 6 are monotonic and approach each other at a rate equal to the complex norm of the subdominant eigenvalue of αP, as discussed in [2] and shown in Section 4 of this paper. Also, the bounds are invariant under translation of J by a multiple of e. Hence, the iteration in eq. 2 can be stopped when the difference between the lower and upper bounds in eq. 6 is below a specified tolerance. The value of J* in this case is approximated by selecting a value between the two bounds.

Porteus discusses in [2] a number of variations of extrapolation methods based on these bounds. Most of these variations achieve the same rate of convergence as the techniques above. In addition, [2] mentions other iterative schemes such as Gauss -Seidel iteration, successive over-relaxation [10], and Jacobi iteration. For specific problems, the convergence rate of any one of these techniques may be better than the rate of the successive approximation iteration of eq. 2. However, for general stochastic P matrices, these techniques offer no significant advantages over successive approximation.

Note that all of the preceding acceleration techniques converge at a rate of α when P has more than one ergodic class, in which case the subdominant eigenvalue of P has a norm of unity.

SECTION 3: Aggregation Error Estimates

The basic principle of aggregation-disaggregation is to approximate the solution of eq. 1 by solving a smaller system of equations obtained by lumping together the states of the original system into a smaller set of aggregate states. We start with an initial guess J to the solution of eq. 1, and we carry out a successive approximation step, obtaining T(J). The idea is to make an additive correction to J of the form Wy, where y is an m-dimensional vector and W is an n x m matrix, so that

$$J + Wy \approx J^* \tag{7}$$

Assumption 1. Q , an m x n matrix, and W an n x m matrix, are chosen so that $Q(I - \alpha P)W$ is nonsingular.

From eq. 1, we get

$$T(J) - J = (I - \alpha P) (J^* - J) \tag{8}$$

Multiplying eq. 8 on the left by Q yields

$$Q (T(J) - J) = Q (I - \alpha P) (J^* - J) \tag{9}$$

Substituting the approximation of eq. 7 in the right hand side of eq. 9 yields

$$Q (T(J) - J) = Q (I - \alpha P) W y \tag{10}$$

Eq. 10 can be solved for y, to obtain an approximate solution

$$J_1 = J + Wy = J + W [Q(I - \alpha P)W]^{-1} Q (T(J) - J) \tag{11}$$

The aggregation step is the conversion of eq. 8 to eq. 10. The disaggregation step is the use of eq. 11 to approximate the solution J*. A successive approximation iteration of J_1 yields

$$T(J_1) = T(J) + \alpha P W y \tag{12}$$

It is important to characterize the error $T(J_1) - J^*$ in terms of the error $J - J^*$. For general W, Q, we get from eq. 12

$$T(J_1) - J^* = (T(J) - J) + (J - J^*) + \alpha PWy \tag{13}$$

which, using eqs. 7 and 10 yields

$$T(J_1) - J^* = \alpha P\{ I - W [Q(I - \alpha P)W]^{-1} Q (I- \alpha P)\} (J -J^*) \tag{14}$$

Note that the matrix $W [Q(I - \alpha P)W]^{-1} Q (I- \alpha P)$ above is an approximation to the identity, and is a projection on the range of W. Indeed, if eq. 8 was exactly satisfied for some y, the right-hand side of eq. 14 would be exactly 0. Hence, the only contributing errors come from the components of J-J* which are outside of the range of W. We will assume that QW=I, which is a typical choice in aggregation methods.

Assumption 2. QW = I, the m-dimensional identity.

Under assumptions 1 and 2, Wy is what is known as the Galerkin solution to eq. 8 using the projection matrix

$$\Pi = WQ. \tag{15}$$

If Π is symmetric (as it will be in this paper) then Π is an othrogonal projection matrix on the range of W.

Eq. 14 is in effect the same error equation obtained by Chatelin and Miranker [5] to characterize the error obtained by additive corrections based on Galerkin approximations. It applies to general linear equations where the matrix P is not necessarily stochastic. In order to better understand this equation, we will derive an expression for the error in the residual obtained after an aggregation-disaggregation step. From eqs. 14 and 8, we get

$$\begin{aligned} T(J_1) - J_1 &= (I - \alpha P)\{ I - W [Q(I - \alpha P)W]^{-1} Q (I- \alpha P) \}(J^* -J) \\ &= \{ I - (I - \alpha P) W [Q(I - \alpha P)W]^{-1} Q\} (I- \alpha P) (J^* -J) \\ &= (I - \Pi) (T(J) -J) + \{W[I - \alpha QPW] - (I - \alpha P) W\}[I - \alpha QPW]^{-1} Q (T(J) -J) \\ &= (I - \Pi) (T(J) -J) + \alpha(I - \Pi) PW [I - \alpha QPW]^{-1} Q (T(J) -J) \end{aligned} \tag{16}$$

Equation 16 is the basic error equation which we will be working with. Our choice of W and Q will be based on trying to minimize a measure of the first error term on the right above. The second term is a measure of how well is the action of the stochastic matrix P represented by the aggregation-disaggregation projections based on W. Note that, if the range of W was invariant under P, the second term would be zero since, from eq. 15 and Assumption 2, $(I - \Pi)W = 0$. Hence, the second term is small when the range of W is closely aligned with an invariant subspace of P. When this is not the case, note that the inverse in this second term can lead to unstable iterations.

SECTION 4: Adaptive Aggregation

We will estimate errors using the following pseudonorm:

$$F(J) = \max_i (J(i)) - \min_i (J(i)) \tag{17}$$

Note that, for a vector J, the scalar F(T(J) - J) is proportional to the spread between the upper and lower bounds in eq. 6. Hence, reducing F(T(J) - J) to 0 is equivalent to having the upper and lower bounds converge to each other, therefore obtaining J^*.

We will select W and Q such that QW = I and the matrix QPW is also a stochastic m-dimensional matrix. As a result, Assumptions 1 and 2 will be satisfied. Partition the state space $S = \{1,2, \ldots, n\}$ into m disjoint sets G_j, $j = 1, \ldots m$ (also called aggregate groups). Define the vectors w_j in R^n as:

$$w_j(i) = 1 \quad \text{if } i \in G_i \tag{18}$$
$$= 0 \quad \text{otherwise.}$$

Let the matrices W and Q be defined as:

$$W = [w_1, \ldots, w_m] \tag{19}$$
$$Q = (W^T W)^{-1} W^T. \tag{20}$$

Note that W^TW is a diagonal matrix, where the i-i entry is the number of elements in group G_i. If one of the groups is empty, then, we can view the inverse above as a pseudoinverse.

Lemma 1. Assume Q and W are defined by eqs. 18, 19, 20. Then,
i. QW = I
ii. $P_a = QPW$ is a stochastic matrix.

Proof: Part i is immediate from the definition of W. Showing part ii is a matter of straightforward verification and is left for the reader. q.e.d.

In order to specify W and Q, we must specify the partition G_j, $j = 1, \ldots, m$. We select this partition to minimize the error in the first term on the right hand side of eq. 16 as measured by the pseudonorm F. We write eq. 16 as

$$T(J_1) - J_1 = R_1(J) + R_2(J) \tag{21}$$

where

$$R_1(J) = (I - \Pi)(T(J) - J) \tag{22 a}$$
$$R_2(J) = \alpha(I - \Pi) PW [I - \alpha QPW]^{-1} Q (T(J) - J) \tag{22 b}$$
$$= \alpha(PW - W P_a) [I - \alpha P_a]^{-1} Q (T(J) - J)$$

We want to select the partition G_j such that $F[R_1(J)]$ is minimized. For a given value of F(T(J) - J), and number of aggregate groups m, the following procedure, based on residual size, is minimax optimal against the worst possible choices of P and J. The idea is to select G_j so that the variation of residuals within each group is relatively small.

Consider

$$\gamma = \min_i [T(J)(i) - J(i)] ; \quad \beta = \max_i [T(J)(i) - J(i)]$$

Divide the interval $[\gamma,\beta]$ into m equal length intervals, of length L, where

$$L = \frac{\beta - \gamma}{m} = \frac{F(T(J) - J)}{m} \tag{23}$$

Then, for $j < m$, we select

$$G_j = \{ i \mid \gamma + (j-1)L \leq (T(J) - J)(i) < \gamma + jL \} \tag{24a}$$

$$G_m = \{ i \mid \gamma + (m-1)L \leq (T(J) - J)(i) \leq \beta \} \tag{24b}$$

To understand the idea behind this choice, note that if j(i) is the index of the group containing state i and $n_{j(i)}$ the number of states in $G_{j(i)}$, the ith coordinate of a vector Πx is

$$(\Pi x)(i) = \sum_{k \varepsilon G_{j(i)}} \frac{x(k)}{n_{j(i)}} \tag{25}$$

i.e. the average value of Πx over the group $G_{j(i)}$. Therefore, the ith coordinate of $R_1(J) = (I - \Pi)(T(J) - J)$ is the difference of the ith residual and the average residual of the group containing the ith coordinate. As a result of the choice of eqs. 23 and 24, the coordinates of $R_1(J)$ are also relatively small.

Figure 1 illustrates the choice of G_j for a typical vector T(J) - J using three aggregate groups. In figure 2, we display the vector $R_1(J)$. Note that the spread between the maximum element and the minimum element has been reduced significantly. We have the following estimate.

Lemma 2. Let G_j be defined by eqs. 23 and 24. Then, for $m > 1$,

$$\frac{F[R_1(J)]}{F[T(J)-J]} \leq \frac{2}{m} \tag{26}$$

Proof: From eq. 25, Π (T(J) -J) is the vector of average values of residuals within each group G_j. The operation (I - Π) (T(J) - J), as shown in fig. 2, subtracts the average value of the residuals in each group from the value of the residuals in each group. Since all of the residuals in each group belong to the same interval in $[\gamma,\beta]$, so does the average value, which establishes that:

$$F[(I - \Pi)(T(J) -J)] \leq 2L . \tag{27}$$

The result follows using eq. 23. q.e.d.

We note that the argument in the proof above can be refined to give the improved estimate

$$\frac{F[R_1(J)]}{F[T(J) - J]} \leq \frac{2\,[.5n]}{m\,([.5n] + 1)} \tag{28}$$

where [x] denotes the largest interger less than x. For large n, the improvement is small. Also, the bound above is a worst-case estimate. In practice, one usually gets a reduction factor better than 1/m (as opposed to 2/m). This has been varified computationally and can also be deduced from the proof of Lemma 2.

Lemma 2 establishes the rationale for our choice of W and Q in equations 18-20, 23, 24. With this choice, we get a guaranteed reduction in the error term $R_1(J)$ which is proportional to the number of aggregate groups. Hence,

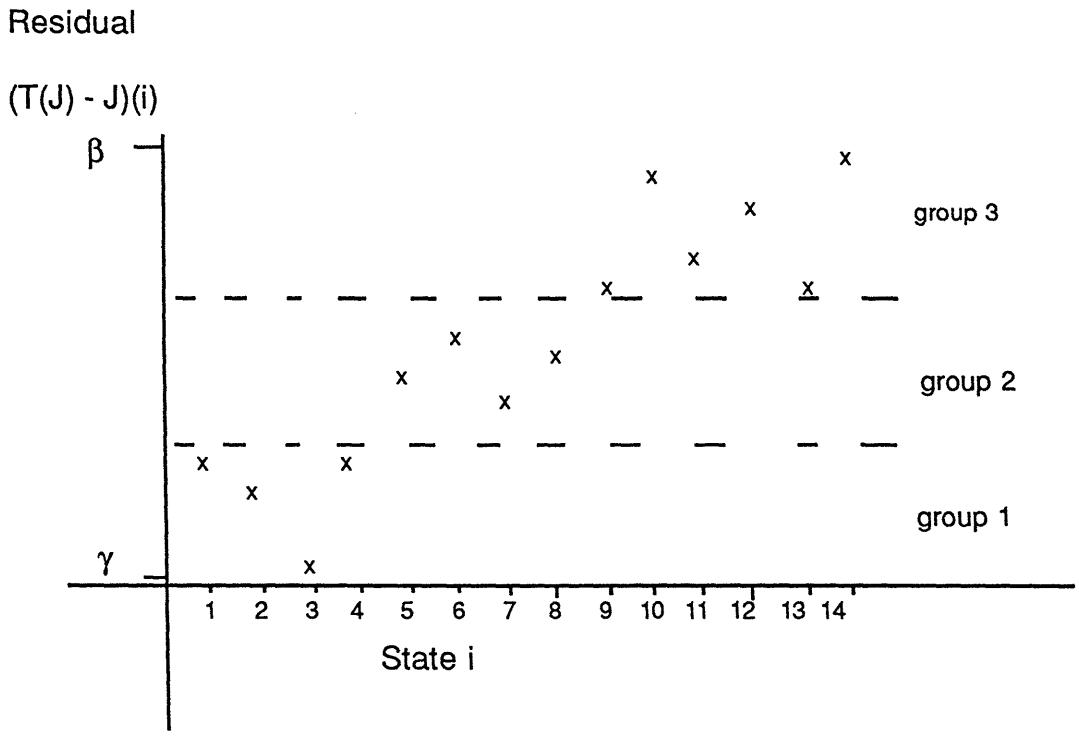

Figure 1: Formation of Aggregate groups is based on magnitude of the residuals. Here the three aggregate groups are obtained by dividing the residual range into three equal portions and grouping together the states with residuals in the same portion.

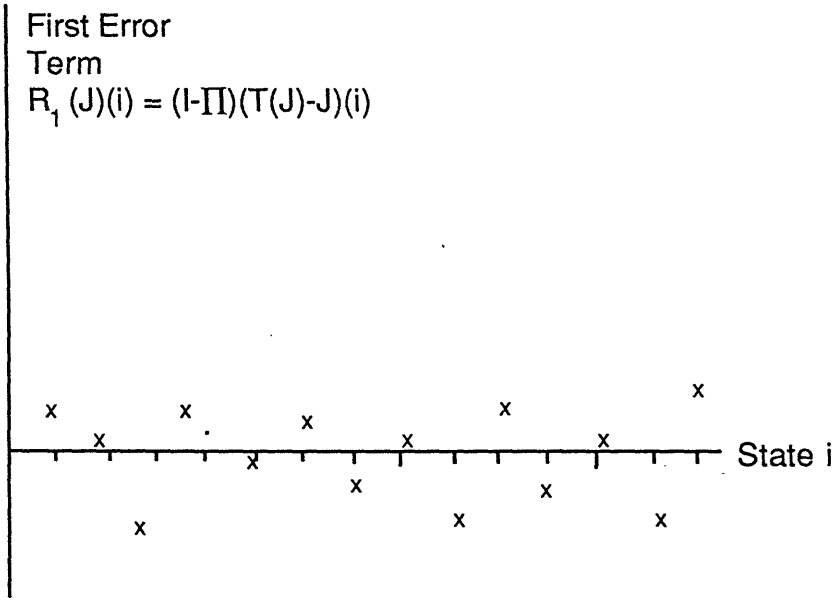

Figure 2: Illustration of the first error term $R_1(J)$ for the case of the residuals of figure 1. $R_1(J)$ is obtained from (T(J) -J) by subtracting the average residual over the group that contains state i.

the acceleration step will work best in problems where the second term $R_2(J)$ is small. To illustrate this, consider the following examples.

Example 1: P = I, the n x n identity

In this case, $R_2(J) = 0$, because PW = W. Hence, the aggregation-disaggregation step reduces the spread between the upper and lower bounds in eqs. 4 and 5 as:

$$F[T(J_1) - J_1] \leq \frac{2\,F[T(J) - J]}{m} \tag{29}$$

In this case, the geometric rate of convergence is accelerated by a minimum factor of 2/m.

Example 2: m = 1, W = e

In this case,we obtain an extrapolation method [2] known as the error sum extrapolation. Starting from J, a successive approximation step is used to compute T(J). Then, an aggregation step is used to compute $T(J_1)$ directly as:

$$T(J_1)(i) = T(J)(i) + \frac{\alpha}{n(1-\alpha)} \sum_{i=1}^{n} (T(J) - J)(i)$$

This aggregation step is followed by a sequence of successive approximation steps and aggregation steps. The rate of convergence of this method can be established using eq. 16. The residual produced by the second successive approximation step is given by

$$\begin{aligned} T(T(J_1)) - T(J_1) &= \alpha P(R_1(J) + R_2(J)) \\ &= \alpha P (I - \Pi) (T(J) - J) \end{aligned}$$

since $R_2(J)$ vanishes (P is a stochastic matrix and Pe=e). After n repetitions of successive approximation and aggregation steps, the residual r_n will be

$$\begin{aligned} r_n &= \alpha^n [P (I - \Pi)]^n (T(J) - J) \\ &= \alpha^n P (I - \Pi) P^{n-1} (T(J) - J) \end{aligned} \tag{30}$$

because from eq. 25, $P\Pi = \Pi$ which implies that $(I - \Pi) P(I - \Pi) = (I - \Pi) P$. Consider a decomposition of P^{n-1} (T(J) -J) into a linear combination of the eigenvectors of P. The eigenvector corresponding to a unity eigenvalue is e and is annihilated by $(I - \Pi)$ (cf. eq. 25). Therefore, r_n will converge to 0 geometrically at a rate determined by the modulus of the largest eigenvalue of αP in a direction other than e (the subdominant eigenvalue).

Example 3: P is block-diagonal and the aggregate groups are aligned with the ergodic classes.

In this case we assume that P has multiple ergodic classes and no transient states. By reordering states if necessary, we can assume that P has the form

$$P = \text{diag}\{ P^1, P^2, \ldots, P^r \} \tag{31}$$

We assume also that each aggregate group G_j, j = 1, . . . , m consists of ergodic classes of states (no two states of the same ergodic class can belong to different groups). The matrix W then has the form

$$W = \begin{bmatrix} 1 \ldots 1\ 0 \ldots 0 \ldots & 0 \\ 0 \ldots 0\ 1 \ldots 1\ 0 \ldots & 0 \\ \vdots & \\ 0 \ldots 0\ 0 \ldots 0\ 0 \ldots 1 \ldots & 1 \end{bmatrix}$$

and it is easily seen that PW = W. Therefore, the second error term $R_2(J)$ vanishes and the favorable rate estimate of eq. 29 again holds. Note that it is not necessary that each aggregate group contains a single ergodic class. This restriction would be needed for fast convergence if the aggregate groups were to remain fixed throughout the computation .

The case of a block diagonal matrix P is important for several reasons. First, block diagonal matrices P present the most difficulties for the successive approximation method, regardless of whether the McQueen-Porteus error bounds are employed. Second, we can expect that algorithmic behavior on block-diagonal matrices will be replicated to a great extent on matrices with weakly coupled or sparsely coupled blocks. This conjecture is substantiated analytically in the next section and experimentally in section 7.

The favorable rate of convergence described above is predicated on the alignment of the ergodic classes and the aggregate groups. The issue of effecting this alignment is therefore important. We first remark that even if this alignment is not achieved perfectly, we have observed experimentally that much of the favorable convergence rate can still be salvaged, particularly if an aggregation step is followed by several successive approximation steps. We provide some related substantiation in the next section, but hasten to add that we do not fully understand the mechanism of this phenomenon. We next observe that for a block-diagonal P, the eigenvectors corresponding to the dominant unity eigenvalues are of the form

$$e_j = [0 \ldots 0\ 1 \ldots 1\ 0 \ldots 0]^T \qquad j = 1, \ldots, r$$

where the unit entries correspond to the states in the j-th ergodic class. Suppose that we start with some vector J and apply k successive approximation steps. The residual thus obtained will be

$$T^k(J) - T^{k-1}(J) = (\alpha P)^{k-1}(T(J) - J)$$

and for large k, it will be nearly a linear combination of the dominant eigenvectors. This means that $T^k(J) - T^{k-1}(J)$ is nearly constant over each ergodic class. As a result, if aggregate groups are formed on the basis of the residual $T^k(J) - T^{k-1}(J)$ and eqs. 23 and 24, they will very likely be aligned with the ergodic classes of P. This fact suggests that several successive approximation steps should be used between aggregation steps, and provides the motivation for our algorithm to be given in Section 6.

SECTION 5: Adaptive aggregation with two groups

In this section we study in detail the effects of the adaptive aggregation- disaggregation procedure for the case when two aggregate groups are used (m=2). Experiment and in some cases analysis show that the qualitative conclusions drawn from consideration of this case carry over to the more general case where m>2. We will focus on analyzing the contribution of the second error term $R_2(J)$ of eq. 16 or 22 b. Assume that W, Q have been selected according to eqs. 18 - 20. By appropriate renumbering of the states, assume that W is of the form

$$W = \begin{bmatrix} 1 \ldots 1\ 0 \ldots 0 \\ 0 \ldots 0\ 1 \ldots 1 \end{bmatrix}^T$$

Let k be the number of elements in the first group. Then a straightforward calculation shows that

$$P_a = \begin{bmatrix} 1-b & b \\ c & 1-c \end{bmatrix} \qquad (33)$$

where

$$b = \frac{1}{k} \sum_{i=1}^{k} b_i \qquad (34a)$$

$$c = \frac{1}{n-k} \sum_{i=k+1}^{n} c_i \qquad (34b)$$

$$b_i = \sum_{j=k+1}^{n} P_{ij}, \qquad i = 1, \ldots, k \tag{35a}$$

$$c_i = \sum_{j=1}^{k} P_{ij}, \qquad i = k+1, \ldots, n. \tag{35b}$$

The right eigenvectors and eigenvalues of P_a are

$$v_1 = [\,1 \;\; 1\,]^T \;; \quad v_2 = [\,1 \;\; -c/b\,]^T \tag{36}$$
$$\lambda_1 = 1 \;; \quad \lambda_2 = 1 - b - c. \tag{37}$$

Note if $b = 0$ that v_2 can be chosen as

$$v_2 = [\,0 \;\; 1\,]^T \tag{38}$$

Note also that

$$Q = \begin{pmatrix} \frac{1}{k} & 0 \\ 0 & \frac{1}{(n-k)} \end{pmatrix} W^T \tag{39}$$

We can decompose the term $Q(T(J) - J)$ of eq. 16 into its components along the eigenvectors v_1, v_2, as

$$Q(T(J) - J) = a_1 v_1 + a_2 v_2 \tag{40}$$

Note that

$$W\,[I - \alpha P_a]^{-1} v_1 = (1-\alpha)^{-1}\, e \tag{41}$$

Hence, using the fact that e lies in the range of W,

$$\alpha(I - \Pi)\, PW\, [I - \alpha P_a]^{-1} v_1 = \alpha\,(1-\alpha)^{-1}\,(I - \Pi)Pe = 0 \tag{42}$$

Thus, the only contribution to $R_2(J)$ comes from the second component of equation 40 , in the direction of v_2. Using eqs. 33, 36, and 37 we obtain

$$[I - \alpha P_a]^{-1} v_2 = [\,1-\alpha + \alpha(b + c)\,]^{-1}\, v_2\,. \tag{43}$$

From eqs. 32 - 35, we can calculate the (i,1) element of the matrix $PW - WP_a$ to be

$$(PW - WP_a)(i,1) = b - b_i \qquad \text{if } i \le k \tag{44}$$
$$= -c + c_i \qquad \text{if } i > k$$

Similarly,

$$(PW - WP_a)(i,2) = -(PW - WP_a)(i,1).$$

Thus, from eq. 22 b

$$R_2(J) = \alpha\,(PW - WP_a)\,[\,1-\alpha + \alpha(b + c)\,]^{-1}\, a_2 v_2$$
$$= \alpha\, a_2 F(v_2) h \tag{45}$$

where h is the vector with coordinates

$$h(i) = \frac{b - b_i}{1 - \alpha + \alpha(b+c)} \qquad \text{if } i \le k \tag{46}$$

$$= \frac{c_i - c}{1 - \alpha + \alpha(b+c)} \qquad \text{if } i > k$$

and $F(v_2) = 1 + c/b$ (cf. eq. 36). From eqs. 34, 35, and 46 we see that in order for the coordinates of h to be small, the probabilities b_i and c_i should be uniformly close to their averages b and c. If this is not so then at least some coordinates of $R_2(J)$ will be substantial, and it is interesting to see what happens after a successive approximation step is applied to $R_2(J)$. The corresponding residual term is the vector

$$q = \alpha P R_2(J).$$

From eqs. 45 and 46 we see that the ith coordinate of q is

$$q(i) = \frac{\alpha^2 a_2 F(v_2)}{1 - \alpha + \alpha(b+c)} \left[\sum_{j=1}^{k} p_{ij} (b - b_j) + \sum_{j=k+1}^{n} p_{ij} (c_j - c) \right] \tag{47}$$

Since b and c are the averages of b_j and c_j respectively, we see that the coordinates of q can be small even if the coordinates of h are large. For example if P has a totally random structure (e.g. all elements are drawn independently from a uniform distribution), then for large n the coordinates of q will be very small by the central limit theorem. There are several other cases where either h or q (or both) are small depending on the structure of P. Several such examples will now be discussed. All of these examples involve P matrices with subdominant eigenvalues close to unity for which standard iterative methods will converge very slowly.

Case 1: P has uniformly weakly coupled classes of states which are aligned with the aggregate groups

The matrix P in this case has the form

$$P = \begin{pmatrix} P^1 & P^2 \\ P^3 & P^4 \end{pmatrix} \tag{48}$$

where P^1 is k x k and the elements of P^2 and P^3 are small relative to the elements of P^1 and P^4. From eqs. 34, 35, 45, and 46 we see that if b and c are considerably smaller than $(1 - \alpha)$, then $R_2(J) \approx 0$. This will also happen if the terms b_i and c_i of eq. 35 are all nearly equal with their averages b and c respectively. Even if $R_2(J)$ is not near zero, from eq. 47 we see that $q \approx 0$ if the size of the elements within each row of P^1, P^2, P^3 and P^4 is nearly uniform.

What happens when the groups identified by the adaptive aggregation process are not perfectly aligned with the block structure of P? We examine this case next.

Case 2: P block diagonal with the upper k x k submatrix not corresponding to the block structure of P.

Without loss of generality, assume that $i = 1, \ldots, m_1 \leq k$ are all elements of one group of ergodic classes of P, while $i = m_2+1, \ldots, n$, $m_2 \geq k$, are elements of the complementary group of ergodic classes. Note that the states $m_1 \leq i \leq m_2$ are not aligned with their ergodic classes in the adaptive aggregation process.

In this case, we have

$$b_i = \sum_{j=k+1}^{m_2} P_{ij} \qquad \text{if} \quad i \leq m_1$$

$$= \sum_{j=m_2+1}^{n} P_{ij} \qquad \text{if} \quad k \geq i > m_1 \tag{49}$$

$$c_i = \sum_{j=1}^{m_1} P_{ij} \qquad \text{if} \quad m_2 \geq i > k$$

$$= \sum_{j=m_1+1}^{k} P_{ij} \qquad \text{if} \quad m_2 < i \leq n \tag{50}$$

Suppose

$$k - m_1 \approx m_2 - k; \quad k \approx n/2 \; ; \quad k - m_1 << k \tag{51}$$

so that the aggregate groups are nearly aligned with the block structure of P. The ergodic classes corresponding to group 1 consist of the set of states i = 1,. . ., m_1 and i=k+1, . . ., m_2, while the remaining states correspond to the ergodic classes in group 2. From eq. 49 we see that b_i will tend to be small for i=1, . . ,m_1 and large for i=m_1+1,...,k. Similarly c_i will tend to be small for i=m_2+1, ...,n and large for i=k+1, ... ,m_2. It follows from eq. 46 that

$$\begin{aligned} &h(i) > 0 \qquad \text{if } i = 1,. . ., m_1 \text{ or } i = k+1, . . ., m_2 \\ &h(i) < 0 \qquad \text{otherwise.} \end{aligned} \tag{52}$$

Hence, $R_2(J)$ is contributing terms of opposite sign to the ergodic classes in groups 1 and 2. By following the aggregation step with repeated successive approximation iterations, this contribution will be smoothed throughout the ergodic classes. Thus, the next aggregation step will be able to identify groups which are aligned with the block structure of P, thereby reducing the error as in case 1. The following example illustrates this point.

Example 4: Let P be the 20 x 20 matrix

$$P = \begin{bmatrix} P^1 & 0 & .1e & 0 \\ 0 & .1 & 0 & .1e^T \\ .1e^T & 0 & .1 & 0 \\ 0 & .1e & 0 & P^2 \end{bmatrix} \tag{53}$$

where P^1, P^2 are 9x9 blocks with uniform entries .1, and e is a 9 dimensional vector of all 1s. Note that one ergodic class has states i = 1, . . ., 9 and i = 11, while the rest of the states are in the second ergodic class. Assume that J is such that

$$\begin{aligned} (T(J) - J)(i) &= 1 \quad \text{if } i \leq 10 \\ &= -1 \quad \text{if } i \geq 11. \end{aligned}$$

In this case, the aggregation matrix W is defined by

$$\begin{aligned} w_1(i) = 1 - w_2(i) &= 1 \qquad \text{if } i \leq 10, \\ &= 0 \qquad \text{if } i \geq 11. \end{aligned}$$

Note that the groups are almost aligned with the ergodic classes of P. Using eqs. 44, 49 and 50, we get

$$\begin{aligned} &b = c = .18 \\ &h(i) = .08 \, [\, 1-\alpha + \alpha(.36) \,]^{-1} \qquad \text{if } i \leq 9 \\ &h(10) = -.72 \, [\, 1-\alpha + \alpha(.36) \,]^{-1} \\ &h(11) = .72 \, [\, 1-\alpha + \alpha(.36) \,]^{-1} \\ &h(i) = -.08 \, [\, 1-\alpha + \alpha(.36) \,]^{-1} \qquad \text{if } i > 11. \end{aligned}$$

From eqs. 36 and 40, we evaluate $F(v_2) = 2$ and $a_2 = .8$. Hence,

$$F(R_2(J)) = \alpha\, 1.44\, [\, 1-\alpha + \alpha(.36)\,]^{-1}(.8)2 \leq 6.4 \qquad (54)$$

Note also that $R_1(J) = 0$ for the choice of T(J) - J of this example.We can now see the effect of the aggregation step. We started out with F(T(J) - J) = 2 and ended up with $F(T(J_1) - J_1) \approx 6.4$ (assuming $\alpha \approx 1$). Therefore the residual error as measured by F has increased substantially as a result of the aggregation step.

Consider now the effect of a successive approximation step subsequent to the aggregation step. Since

$$\begin{aligned} (Ph)(i) &= .144\, [\, 1-\alpha + \alpha(.36)\,]^{-1} && \text{if } i \leq 9 \text{ or } i = 11 \\ &= -.144\, [\, 1-\alpha + \alpha(.36)\,]^{-1} && \text{otherwise.} \end{aligned}$$

we see that the corresponding residuals $(T^2(J_1) - T(J_1))(i)$ will be constants of opposite sign over the two ergodic classes. (The smoothing of the error after a single successive approximation step in this example is a coincidence. In general, several successive approximation steps will be required to diffuse the effect of the initial aggregation step throughout the ergodic classes.) The end effect is to align the aggregate groups with the ergodic classes at the next aggregation step.

Note also that using eq. 47 we have

$$\begin{aligned} F(\, T^2(J_1) - T(J_1)) &= F(\alpha P R_2(J)) = F(q) \\ &= \alpha^2 .288\, [\, 1-\alpha + \alpha(.36)\,]^{-1}(.8)\, 2 \leq 1.28. \end{aligned}$$

Therefore, after a single successive approximation step, the error will be reduced substantially below the starting error F(T(J) - J) = 2. Thus, we see that the aggregation step itself causes an increase in the error as measured by F. Yet, it produces a vector that is oriented sufficiently away from the dominant eigenvectors of P so that the subsequent successive approximation step is highly effective. This phenomenon was consistently observed during our experimentation and has also been observed by Chatelin and Miranker [5].

Case 3: P has sparsely-coupled classes of states

In this case, P has the general form

$$P = \begin{bmatrix} P^1 & P^2 \\ P^3 & P^4 \end{bmatrix}$$

where elements of P^1, P^4, P^2, P^3 are of the same order, and P^1, P^4 are dense while P^2, P^3 are very sparse. Assume that the groups are aligned with the block structure of P. Then we have

$$b_i = \sum_{j=1}^{n-k} P^2{}_{ij} \qquad \text{if } i \leq k \qquad (55a)$$

$$c_i = \sum_{j=1}^{k} P^3{}_{ij} \qquad \text{if } i > k. \qquad (55b)$$

As in case 1, if b_i and c_i are small (of the order of (1-a)), or vary little from the corresponding averages b and c, then $R_2(J) \approx 0$. If the size of the elements within P^1 and P^4 is nearly uniform, then from eq. 47 we see that $q \approx 0$. Furthermore, the behavior observed in case 2 will be replicated in this case where, when the aggregate groups are not aligned with the block structure of the P matrix, the term $R_2(J)$ forces the next aggregation step to be better aligned with the block structure of P.

In conclusion, the cases studied in this section indicate that, for classes of problems where there are multiple eigenvalues with norm near unity, a combination of several successive approximation steps, followed by an aggregation step, will minimize the contribution of $R_2(J)$ to the error, and thereby accelerate the convergence of the iterative process as in lemma 2. In the next section, we formalize these ideas in terms of an overall iterative algorithm.

SECTION 6. Iterative Aggregation Algorithms

The method for imbedding our aggregation ideas into an algorithm is straightforward. Each iteration consists of one or more successive approximation steps, followed by an aggregation step. The number of successive approximation steps in each iteration may depend on the progress of the computation.

One reason why we want to control the number of successive approximation steps per iteration is to guarantee convergence. In contrast with a successive approximation step, the aggregation step need not improve any measure of convergence. We may wish therefore to ensure that sufficient progress has been made via successive approximation between aggregation steps to counteract any divergence tendencies that may be introduced by aggregation. Indeed, we have observed experimentally that the error F(T(J) -J) often tends to deteriorate immediately folowing an aggregation step due to the contribution of $R_2(J)$, while unusually large improvements are made in the next few successive approximation steps. This is consistent with some of the analytical conclusions of the previous section. An apparently effective scheme is to continue with successive approximation steps as long as F(T(J) - J) keeps decreasing by a "substantial" factor.

One implementation of the algorithm will now be formally described:

Step 0: (Initialization) Choose initially a vector J, and scalars $\varepsilon > 0$, β_1, β_2 in (0,1), $\omega_1 \approx \infty$ and $\omega_2 \approx \infty$.

Step 1: (Successive approximation step) Compute T(J).

Step 2: (Termination Test) If $F(T(J) - J) < \varepsilon$, stop and accept

$$T(J) + .5\,\alpha(1-\alpha)^{-1}[\max_i (T(J)-J)(i) - \min_i (T(J)-J)(i)]$$

as the solution (cf. the bounds in eq. 6). Else go to step 3.

Step 3: (Test for an aggregation step) If

$$F(T(J) - J) \leq \omega_1 \tag{56}$$

and

$$F(T(J) - J) \geq \omega_2 \tag{57}$$

set $\omega_1 := \beta_1 F(T(J) - J)$ and go to step 4. Else, set $\omega_2 := \beta_2 F(T(J) - J)$, J:=T(J) and go to step 1.

Step 4: (Aggregation Step)Form the aggregate groups of states G_j, j = 1,. . ., m based on T(J) - J as in eq. 25. Compute $T(J_1)$ using eqs. 10 and 12. Set $J := T(J_1)$, $\omega_2 \approx \infty$, and go to step 1.

Note that ω_1 is reduced by a factor of at least β_1 for each time we carry out an aggregation step. As a result, the test of eq. 56 enforces convergence, since it guarantees that, before step 4 is entered, F(T(J) - J) is reduced to a level below the target ω_1, and ω_1 converges to zero when an infinite number of aggregation steps are performed. If only a finite number of aggregation steps are performed, the algorithm reduces eventually to the convergent successive approximation method. The purpose of the test of eq. 57 is to allow the aggregation step only when the progress made by the successive approximation step is relatively small (a factor no greater than β_2).

An alternative implementation is to eliminate the test of eq. 57 and perform an aggregation step if eq. 56 is satisfied and the number of consecutive iterations during which an aggregation step was not performed exceeds a certain threshold.

There is a similar algorithm that can be used for dynamic programming problems where we want to solve the system of equations

$$J(i) = T(J)(i) \equiv \min_{u \in U} \{ \alpha \sum_{j=1}^{n} P_u(i,j)J(j) + g_u(i) \}, \; i = 1,. . ., n \tag{58}$$

where U is a finite set, and P_u, g_u are transition matrices and cost vectors parametrized by u. The key idea is to employ an approximate policy iteration algorithm in the spirit of Puterman and Shin [11], [12], whereby the policy evaluation step is carried out approximately using the earlier algorithm, i.e., one or more cycles each consisting of multiple successive approximation steps followed by an aggregation step.

SECTION 7: Computational Results

A large number of randomly generated problems with 100 states or less were solved using the adaptive aggregation methods of this paper. The conclusion in summary is that problems that are easy for the succcessive approximation method (single ergodic class, dense matrix P) are also easy for the aggregation method; but problems that are hard for succcessive approximation (several weakly coupled blocks, sparse structure) are generally easier for aggregation and often dramatically so.

Tables 1 and 2 summarize representative results relating to problems with 75 states grouped in three blocks of 25 each. The elements of P are either zero or randomly drawn from a uniform distribution. The probability of an element being zero was controlled thereby allowing the generation of matrices with approximately prescribed degree of density. Table 1 compares various methods on block diagonal problems with and without additional transient states, which are full (100%) dense, and 25% dense within each block. Table 2 considers the case where the blocks are weakly coupled with 2% coupling (size of elements outside the blocks is on the average 0.02 times the average size of the elements inside the blocks), and the case where the blocks are 100% coupled (all nonzero elements of P have nearly the same size). Each entry in the tables is the number of steps for the corresponding method to reach a prescribed difference (10^{-6}) between the upper and lower bounds of Section 2. We have estimated that an aggregation step requires roughly twice as much computation as a succcessive approximation step. The entries for the aggregation methods represent the sum of the number of succcessive approximation and twice the number of aggregation steps. In all cases the starting vector was zero, and the components of the cost vector g were randomly chosen on the basis of a uniform distribution over [0, 1].

The methods are succcessive approximation (with the error bounds of eq. 6), and six aggregation methods corresponding to all combinations of 3 and 6 aggregate groups, and 3, 5, and 10 succcessive approximation steps between aggregation steps. Naturally these methods do not utilize any knowledge about the block structure of the problem.

TABLE 1. Discount factor .99, Block Diagonal P,
3 Blocks, 25 states each
Tolerance for Stopping: 1.0 E-6

	Successive (SA) Approximation	3 SA Steps per aggregation, 3 aggregate groups	3 SA Steps 6 aggregate groups	5 SA Steps 3 aggregate groups	5 SA Steps 6 aggregate groups	10 SA Steps 3 aggregate groups	10 SA Steps 6 aggregate groups
100 % density, 0 transient states	1195	11	11	15	15	25	25
100% density, 20 transient states	1225	31	16	58	17	170	27
25% density, 0 transient states	1212	23	26	29	23	27	27
25% density, 20 transient states	1197	186	105	177	72	194	50

Table 1 shows the dramatic improvement offered by adaptive aggregation as predicted by Example 3 in Section 4. The improvement is substantial (although less pronounced) even when there are transient states. Generally speaking the presence of transient states has a detrimental effect on the performance of the aggregation method when there are multiple ergodic classes. Repeated successive approximation steps have the effect of making the residuals nearly equal across ergodic classes; however the residuals of transient states tend to drift at levels which are intermediate between the corresponding levels for the ergodic classes. As a result, even if the alignment of aggregate groups and ergodic classes is perfectly achieved, the aggregate groups typically contain a mixture of ergodic classes and transient states. This has an adverse effect on both error terms of eq. 16. As the results of Table 1 show, it appears advisable to increase the number of aggregate groups m when there are transient states. It can be seen also from Table 1 that the number of succcessive approximation steps performed between aggregation steps influences the rate of convergence. Generally speaking there seems to a problem-dependent optimal value for this number which increases as the problem structure deviates from the ideal block diagonal structure. For this reason it is probably better to use an adaptive scheme to control this number in a general purpose code as discussed in Section 6.

TABLE 2. Discount factor .99, coupled P,
3 Blocks, 25 states each,
Tolerance for Stopping: 1.0 E-6

	Successive (SA) Approximation	3 SA Steps per aggregation, 3 aggregate groups	3 SA Steps 6 aggregate groups	5 SA Steps 3 aggregate groups	5 SA Steps 6 aggregate groups	10 SA Steps 3 aggregate groups	10 SA Steps 6 aggregate groups
100 % density, 2% coupling	170	17	17	22	22	37	37
25% density, 2% coupling	167	38	33	36	32	40	40
100% density, 100% coupling	6	7	7	8	7	7	7
3% density, 100% coupling	66	56	66	60	64	64	66

Table 2 shows that as the coupling between blocks increases (and consequently the modulus of the subdominant eigenvalue of P decreases), the performance of both successive approximation and adaptive aggregation improves. When there is full coupling between the blocks the methods become competitive, but when the coupling is weak the aggregation methods hold a substantial edge as predicted by our analysis.

An interesting issue is the choice of the number of aggregate groups m. According to lemma 2, the first error term $R_1(J)$ of eq. 22 is reduced by a factor proportional to m at each aggregation step. This argues for a large value of m, and indeed we have often found that increasing m from two to something like three or four leads to a substantial improvement. On the other hand the benefit from reduction of $R_1(J)$ is usually exhausted when m rises above four, since then the effect of the second error term $R_2(J)$ becomes dominant. Also the aggregation step involves the solution of the m-dimensional linear system of eq. 10, so when m is large the attendant overhead can become substantial. In the extreme case where m=n and each state forms by itself an aggregate group, the solution is found in a single aggregation step. The corresponding dynamic programming method described at the end of the previous section is then equivalent to the policy iteration algorithm.

REFERENCES

1. Bertsekas, D.P., Dynamic Programming: Deterministic and Stochastic Models, Prentice-Hall, Englewood Cliffs, NJ, 1987.
2. Porteus, E. L., "Overview of Iterative Methods for Discounted Finite Markov and Semi-Markov Decision Chains," in Rec. Developments in Markov Decision Processes, R. Hartley, L.C. Thomas and D. J. White (eds.), Academic Press, London 1980.
3. Porteus, E. L, "Some Bounds for Discounted Sequential Decision Processes," Management Science, Vol. 18, 1971.
4. Miranker, W.L., "Hierarchical Relaxation," Computing, Vol. 23, 1979.
5. Chatelin, F. and W. L. Miranker, "Acceleration by Aggregation of Successive Approximation Methods," Linear Alg. Appl., Vol. 43, 1982.
6. Schweitzer, P.J., M. Puterman and K. W. Kindle, "Iterative Aggregation-disaggregation Procedures for Solving Discounted Semi-Markovian Reward Processes," Working paper QM 8331, Univ. Rochester, 1983.
7. Verkhovsky, B. S., "Smoothing System Optimal Design," RC 6085, IBM Research division, Yorktown Heights, NY 1976.
8. Mendelshohn, R., "An Iterative Aggregation Procedure for Markov Decision Processes," Operations Research, Vol. 30, 1982.
9. McQueen, J., "A Modified Dynamic Programming Method for Markovian Decision Problems," Journal Math. Anal. and Appl., vol. 14, 1966.
10. Kushner, H.J. and A. J. Kleinman, "Accelerated Procedures for the Solution of Discounted Markov Control Problems," IEEE Trans. Auto. Control, Vol. AC-16, 1971.
11. Puterman, M.L. and M. C. Shin, "Modified Policy Iteration Algorithms for Discounted Markov Decision Problems," Management Science, Vol. 24, 1979.
12. Puterman, M.L. and M. C. Shin, "Action Elimination Procedures for Modified Policy Iteration Algorithms," Operations Research, vol. 30, 1982