March 1987 LIDS-P-1652

ADAPTIVE AGGREGATION METHODS FOR INFINITE HORIZON 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 infinite horizon 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

Consider a Markov chain with finite state space $S = \{1,...,n\}$. Let $x(t)$ denote the state of the chain at stage t. Assume that there is a finite decision space U, and that, for each state $x(t)$ and decision $u(t)$ at stage t, the state transition probabilities are given and are independent of t. Let $\alpha \in (0,1)$ be a discount factor and $g(x(t), u(t))$ be a given cost function of state and decision. Let $\mu : S \rightarrow U$ denote a stationary control policy. The infinite-horizon discounted optimal control problem consists of selecting the stationary control policy which minimizes, for all initial states i, the cost

$$J_\mu(i) = E\{\sum_{t=0}^{\infty} \alpha^t g(x(t), \mu(x(t)) \mid x(0) = i , \mu\}. \qquad (1)$$

The optimal cost vector J^* of this problem is characterized as the unique solution of the dynamic programming equation [1]

$$J^* = \min_\mu \{g_\mu + \alpha P_\mu J^* \}. \qquad (2)$$

Here the coordinates of J^* are $J^*(i) = \min_\mu J^*(i)$, g_μ is the vector with coordinates $g(i, \mu(i))$, P_μ is the transition probability matrix corresponding to μ, and the minimization is considered separately for each coordinate.

One of the principal methods for solving the problem is the policy iteration algorithm which iterates between a policy improvement step

$$\mu^n = \arg \min_\mu \{g_\mu + \alpha P_\mu J^{n-1}\} \qquad (3)$$

yielding a new policy μ^n, and a policy evaluation step that finds the cost vector J^n corresponding to policy μ^n by solving the equation

$$J^n = g_{\mu^n} + \alpha P_{\mu^n} J^n. \qquad (4)$$

Eq. 4 is a linear nxn system which can be solved by a direct method such as Gaussian elimination. In the absence of specific structure, the solution requires $O(n^3)$ operations, and is impractical for large n. An alternative, suggested in [11], [12] and widely regarded as the most computationally efficient approach for large problems, is to use an iterative technique for the solution of eq. 4, such as the successive approximation method; this requires only $O(n^2)$ per iteration for dense matrices P (see the survey [2]). It appears that the most effective way to operate this type of method is not to insist on a very accurate iterative solution of eq. 4. Two points relevant to the present paper are that:

(a) The choice of iterative method for solving approximately eq. 4 is open.

(b) For convergence of the overall scheme it is sufficient to terminate the iterative method at a vector J such that a norm of the residual vector

$$J - (g_{\mu^n} + \alpha P_{\mu^n} J)$$

is reduced by a certain factor over the corresponding norm of the starting residual

$$J^{n-1} - (g_{\mu^n} + \alpha P_{\mu^n} J^{n-1})$$

obtained when the policy improvement step of eq. 3 is carried out.

This paper proposes a new iterative aggregation method for solving eq. 4 as per (a) above. Its rate of convergence can be superior to that of other competing methods, particularly for difficult problems where there are multiple ergodic classes corresponding to the transition matrix P_{μ^n}. Its convergence is assured through the use of safeguards that enforce a guaranteed reduction of the residual vector norm as per (b) above.

Several authors have proposed the use of aggregation- disaggregation ideas for accelerating the convergence of iterative methods for the solution of eq. 4 (Miranker [4], Chatelin and Miranker [5], Schweitzer, Puterman and Kindle [6], Verkhovsky [7], and Mendelshohn [8]). In [5], Chatelin and Miranker described the basic aggregation technique and derive a bound for the error reduction. However, they did not provide a specific algorithm for selecting the directions of aggregation or disaggregation. In [7], Verkhovsky proved the convergence of an aggregation method which used 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.

There is an important difference between the aggregation algorithms described in this paper 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. States with similar magnitude of residual are grouped together at each aggregation step, and, because the residual magnitudes change drastically in the course of the algorithm, the group membership of the states can also change accordingly. This is in contrast with the approach of [6] for example, 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_μ 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 faster than $2\alpha/m$ where m is the number of aggregate groups.

The rest of the paper is organized as follows. In section 2, we provide some background material on iterative algorithms for the solution of eq. 4, including bounds on the solution error. In section 3, we derive the equations of aggregation and disaggregation as in [5], and obtain a characterization 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. Throughout the paper we emphasize discounted problems. Our aggregation method extends, however, to average cost Markovian decision problems and in Section 6 we describe the extension. In section 7, we discuss and justify the general iterative algorithm combining adaptive aggregation steps with successive approximation steps. Section 8 presents experimental results.

SECTION 2: Successive Approximation and Error Bounds

For the sake of simplicity, we will drop the argument μ from equation 4, thereby focusing on obtaining an iterative solution to the equation

$$J = T(J) \qquad (5a)$$

where the mapping $T : R^n \rightarrow R^n$ is defined by

$$T(J) \triangleq g + \alpha P J. \tag{5b}$$

A successive approximation iteration on a vector J simply replaces J with T(J). The successive approximation method for the solution of eq. 5 starts with an arbitrary vector J, and sequentially computes T(J), $T^2(J)$, ... Since P is a stochastic matrix (and hence has spectral radius of 1) and $\alpha \in (0,1)$, it follows that T is a contraction mapping with modulus α. Hence, we have

$$\lim_{k\rightarrow\infty} T^k(J) = J^* \tag{6}$$

where J* is the solution of eq. 5 and T^k is the composition of the mapping T with itself k times. The rate of convergence in eq. 6 is geometric at a 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 due to McQueen [9] and Porteus [3] (see [1] for a derivation). These bounds are based on the residual difference of T(J) and J. Let J(i) denote the ith component of a vector J. Let γ and β be defined as

$$\gamma = \min_i [T(J)(i) - J(i)] \tag{7a}$$
$$\beta = \max_i [T(J)(i) - J(i)] \tag{7b}$$

Then, the solution J* of eq. 1 satisfies

$$T(J)(i) + \frac{\alpha\gamma}{1-\alpha} \leq J^*(i) \leq T(J)(i) + \frac{\alpha\beta}{1-\alpha} \tag{8}$$

for all states i. Furthermore, the bounds of eq. 8 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. Hence, the iterations can be stopped when the difference between the lower and upper bounds in eq. 8 is below a specified tolerance for all states i. The value of J* in this case is approximated by selecting a value between the two bounds.

There are also several variations of the successive approximation method such as Gauss - Seidel iteration, successive over-relaxation [10], and Jacobi iteration [2]. Depending on the

problem at hand these schemes may converge faster than the successive approximation method. However their rate of geometric convergence is often close to α when α is large and 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 have a vector J and we want to make an additive correction to J of the form Wy, where y is an m-dimensional vector and W is an nxm matrix, so that

$$J + Wy \approx J^* \tag{9}$$

Our method makes use of two matrices Q and W. We will later assume that $Q = (W^TW)^{-1}W^T$ (superscript T denotes transpose), but it is worthwhile to postpone this assumption for later so as to develop the following error equations in generality. We thus assume:

Assumption 1. Q is an m x n matrix, and W is an n x m matrix, chosen so that $Q(I - \alpha P)W$ is nonsingular, and $QW = I$ where I is the m-dimensional identity.

From eq. 5, we get

$$T(J) - J = (I - \alpha P)(J^* - J) \tag{10}$$

Multiplying this equation on the left by Q yields

$$Q(T(J) - J) = Q(I - \alpha P)(J^* - J) \tag{11}$$

It follows that, if $J^* - J$ is approximately Wy as in eq. 9, then a good choice for y is the unique solution of the following mxm system obtained by replacing $J^* - J$ with Wy in eq. 11

$$Q(T(J) - J) = Q(I - \alpha P)W y, \tag{12}$$

or, using the fact $QW = I$,

$$Q(T(J) - J) = (I - \alpha QPW)\, y$$

Thus we define

$$y = (I - \alpha QPW)^{-1} Q\,(T(J) - J),$$

and view the vector J_1 defined by

$$J_1 = J + Wy = J + W\,(I - \alpha QPW)^{-1} Q\,(T(J) - J) \qquad (13)$$

as an approximation to J^* (cf. eq. 9). The conversion of eq. 10 to the lower dimensional eq. 12 is known as the aggregation step. The disaggregation step is the use of eq. 13 to approximate the solution J^*. Note that there is no claim or guarantee that J_1 approximates well J^* ; this depends on the choice of the subspace W which is the key for the success of aggregation methods. If $J - J^*$ lies on the range space of W then $J_1 = J^*$. Generally J_1 will be close to J^* if $(J - J^*)$ nearly lies on the range space of W.

After obtaining J_1 using eq. 13, the aggregation method performs a successive approximation iteration on it yielding

$$T(J_1) = T(J) + \alpha\, PWy \qquad (14)$$

(this improves the quality of the solution and is also a necessary first step for the subsequent aggregation step as seen from eq. 13). In some cases it is desirable to perform several successive approximation iterations between aggregation steps (see the discussion of section 4). We thus define the iterative aggregation method as a sequence of iterations of the form of eq. 13 with each pair of consecutive iterations possibly separated by one or more successive approximation iterations.

To understand the properties of the iterative aggregation method it is important to characterize the error $T(J_1) - J^*$ in terms of the error $J - J^*$. From eq. 14 we get

$$T(J_1) - J^* = (T(J) - J) + (J - J^*) + \alpha\, PWy \qquad (15)$$

which, using eqs. 10 and 12, yields

$$T(J_1) - J^* = \alpha P\{\, I - W\,(I - \alpha QPW)^{-1} Q\,(I - \alpha P)\}\,(J - J^*) \qquad (16)$$

Eq. 16 is in effect the 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 residual obtained after an aggregation-disaggregation step. Define the matrix

$$\Pi = WQ \tag{17}$$

which is a projection on the range space of W. Generally Π is not an orthogonal projection but with the choice $Q = (W^TW)^{-1}W^T$ that will be used later in this paper Π becomes the orthogonal projection matrix on the range of W. From eqs. 16 and 10 we get

$$\begin{aligned} T(J_1) - J_1 &= (I - \alpha P)\{ I - W [Q(I - \alpha P)W]^{-1}Q (I- \alpha P) \}(J^* - J) \\ &= \{ I - (I - \alpha P) W [Q(I - \alpha P)W]^{-1}Q\} (I- \alpha P) (J^* - J) \\ &= (I - \Pi) (T(J) - J) + \{W[I - \alpha QPW] - (I - \alpha P) W\}[I - \alpha QPW]^{-1}Q (T(J) - J) \\ &= (I - \Pi) (T(J) - J) + \alpha(I - \Pi) PW [I - \alpha QPW]^{-1}Q (T(J) - J) \\ &= (I - \Pi) (T(J) - J) + \alpha(I - \Pi) PWy \end{aligned} \tag{18}$$

Equation 18 is the basic error equation which we will be working with. There are two error terms in the right side of eq. 18 (see Figure 1). Our subsequent choice of W and Q will be based on trying to minimize the first error term on the right above. We generally estimate errors using the pseudonorm

$$F(J) = \mathrm{Max}_i (J(i)) - \mathrm{Min}_i (J(i)) \tag{19}$$

Since the scalar F(T(J) - J) is proportional to the difference between the upper and lower bounds in eq. 8 we see that reducing F(T(J) - J) to 0 is equivalent to having the upper and lower bounds converge to each other, thereby obtaining J*. The second error term in eq. 18 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 P maps the range of W into itself, the second term is zero since, from eq. 17 and the condition QW = I of Assumption 1, we have $(I - \Pi)W = 0$. 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, the inverse in this second term introduces a tendency for instability. Despite this fact it will be seen that the effect of this term can be adequately dealt with.

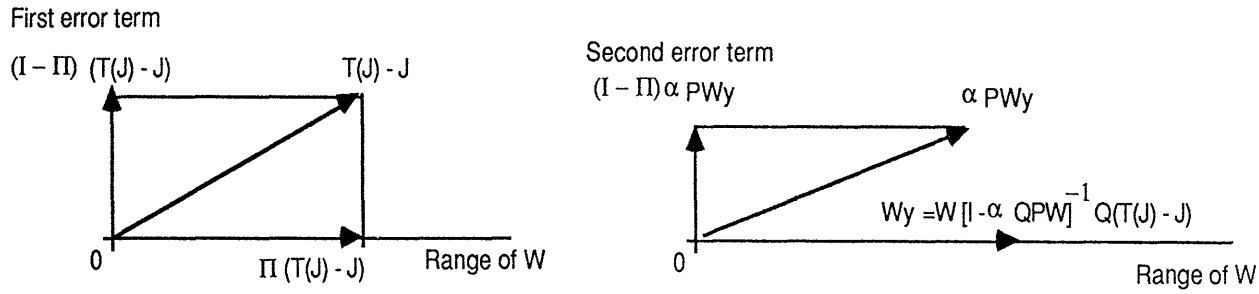

Figure 1: Geometric illustration of the two error terms of eq. 18. The matrix Π projects orthogonally on the range space of W. Note that if the range of W is invariant under P, the second error term is zero.

SECTION 4: Adaptive Choice of the Aggregation Matrices Based on Residual Size

We introduce a specific choice of Q and W. 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 with ith coordinates given by

$$w_j(i) = 1 \quad \text{if } i \in G_j \qquad (20)$$
$$= 0 \quad \text{otherwise.}$$

Let the matrices W and Q be defined by

$$W = [w_1, \ldots, w_m] \qquad (21)$$
$$Q = (W^TW)^{-1}W^T. \qquad (22)$$

Note that W^TW is a diagonal matrix with i-ith entry equal to 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. 20, 21, 22. Then,

(a) $QW = I$

(b) $P_a \triangleq QPW$ is a stochastic matrix

(c) Q and W satisfy Assumption 1.

Proof: (a) Immediate from the definition of eq. 22.
(b) By straightforward calculation we can verify that the (i,j)th element of P_a is

$$[P_a]_{ij} = \frac{1}{|G_i|} \sum_{k \in G_i} \sum_{m \in G_j} p_{km}$$

where $|G_i|$ is the number of states in G_i. It follows that $[P_a]_{ij} \geq 0$ for all i, j, and

$$\sum_{j=1}^{m} [P_a]_{ij} = 1, \qquad \text{for all } i = 1, \ldots, m.$$

Therefore P_a is a stochastic matrix.
(c) The eigenvalues of P_a lie within the unit disk, so, in view of $\alpha < 1$, the matrix $I - \alpha P_a$ cannot have a zero eigenvalue and must therefore be invertible. This combined with part (a) shows that Assumption 1 is satisfied. q.e.d.

Figure 2 illustrates the "aggregated Markov chain" corresponding to the stochastic matrix P_a and identifies its states with aggregate groups. This chain provides an insightful interpretation of the aggregated system of eq. 12. By writing this system as

$$Q(T(J) - J) = (I - \alpha P_a)y$$

and by comparing it with the system of eq. 10 we see that y is the cost vector corresponding to the aggregated Markov chain, and to a cost per stage equal to Q(T(J) - J) the ith component of which is the average residual

$$\frac{1}{|G_i|} \sum_{k \in G_i} [T(J)(k) - J(k)]$$

over the ith aggregate group of states. Thus the aggregation iteration solves in effect a (lower dimensional) dynamic programming equation corresponding to the aggregated Markov chain.

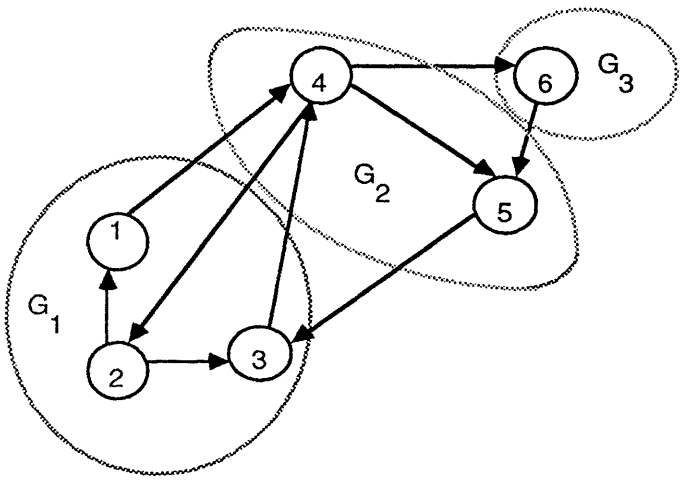

Figure 2: Illustration of the aggregated Markov chain associated with the transition matrix $P_a = QPW$. The aggregate groups are $G_1 = \{1, 2, 3\}$, $G_2 = \{4, 5\}$, $G_3 = \{6\}$ and they correspond to states of the aggregated Markov chain. The transition probability from state G_i to state G_j equals the sum of all transition probabilities from states in G_i to states in G_j An aggregation step can be interpreted as a policy evaluation step involving the aggregated Markov chain.

We now describe the method for selecting the aggregate groups. We write eq. 18 as

$$T(J_1) - J_1 = R_1(J) + R_2(J) \tag{23}$$

where

$$R_1(J) = (I - \Pi)(T(J) - J) \tag{24a}$$

$$R_2(J) = \alpha(I - \Pi) PW (I - \alpha QPW)^{-1} Q (T(J) - J) \tag{24b}$$

We want to select the partition G_j, $j = 1, ..., m$ so 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

$$L = (\beta - \gamma)/m = (F(T(J) - J))/m \tag{25}$$

Then, for $j < m$, we select

$$G_j = \{i \mid \gamma + (j-1)L \le (T(J) - J)(i) < \gamma + jL\}, \qquad j < m \tag{26a}$$

and we select

$$G_m = \{i \mid \gamma + (m-1)L \le (T(J) - J)(i) \le \beta\} \tag{26b}$$

To understand the idea behind this choice, note that if j(i) is the index of the group containing state i and $|G_{j(i)}|$ is the number of states in $G_{j(i)}$, the ith coordinate of a vector $\Pi x = W(W^TW)^{-1}W^Tx$ (cf. eqs 15 and 22) can be calculated to be

$$(\Pi x)(i) = \sum_{k \in G_{j(i)}} \frac{x(k)}{|G_{j(i)}|} \tag{27}$$

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 residual of state i and the average residual of the group containing state i. As a result of the choice of eqs. 25 and 26, the coordinates of $R_1(J)$ are also relatively small.

Figure 3 illustrates the choice of G_j for a typical vector T(J) - J using three aggregate groups. In Figure 4, 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. 25 and 26. Then, for $m > 1$,

$$\frac{F[R_1(J)]}{F[T(J)-J]} \le \frac{2}{m} \tag{28}$$

Proof: From eq. 27, 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. 4, 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 each coordinate of (I - Π) (T(J) - J) lies between -L and L. Therefore, using eq. 25, we have

$$F[(I - \Pi)(T(J) - J)] \leq 2L = 2\,F(T(J) - J) / m \qquad (29)$$

which proves the result. q.e.d.

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

$$\frac{F[R_1(J)]}{F[T(J) - J]} \leq \frac{2\,\lfloor .5n \rfloor}{m\,(\lfloor .5n \rfloor + 1)} \qquad (30)$$

where $\lfloor x \rfloor$ denotes the largest integer 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 verified computationally and can also be deduced from the proof of Lemma 2.

Lemma 2 establishes that with our choice of W and Q we get a substantial reduction in the error term $R_1(J)$. Hence,

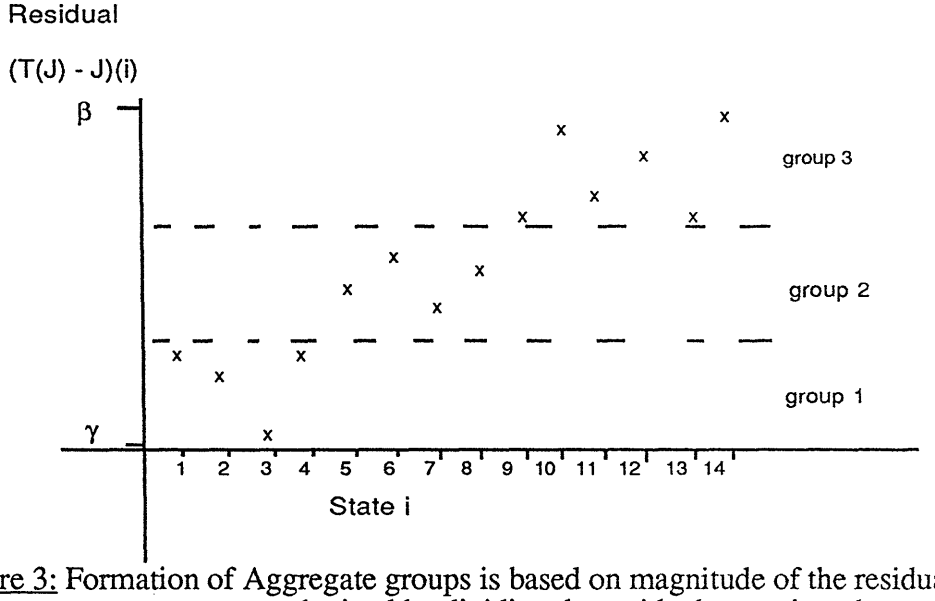

Figure 3: Formation of Aggregate groups is based on magnitude of the residuals. Here the three aggregate groups are obtained by dividing the residual range into three equal portions and grouping together the states with residuals in the same portion.

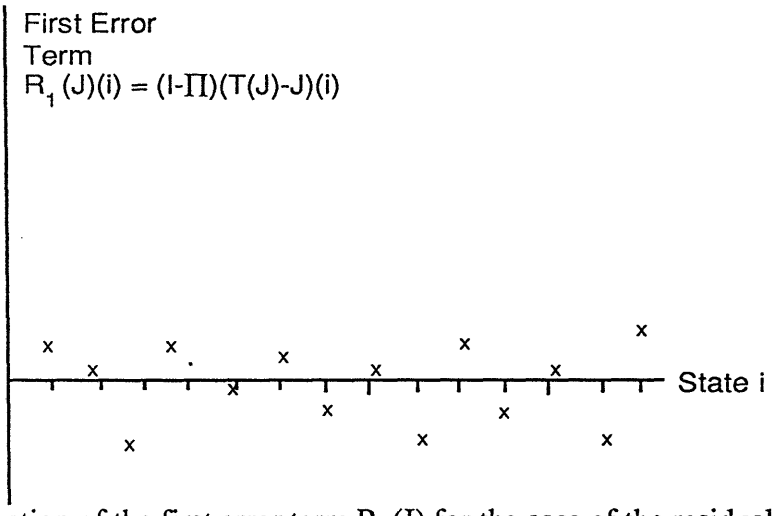

Figure 4: Illustration of the first error term $R_1(J)$ for the case of the residuals of Figure 3. $R_1(J)$ is obtained from (T(J) -J) by subtracting the average residual over the group that contains state i.

the aggregation 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. 7 and 8 as:

$$F[T(J_1) - J_1] \leq \frac{2\,F[T(J) - J]}{m} \tag{31}$$

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

Example 2: m = 1, W = e where e is the unit vector $e^T = [1, 1, ..., 1]$.
In this case,we obtain a scheme known as the error sum extrapolation [2]. 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. 18. The residual produced by the second successive approximation step is given by

$$T(T(J_1) - J_1) = \alpha P(R_1(J) + R_2(J))$$
$$= \alpha P (I - \Pi) (T(J) - J)$$

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

$$r_n = \alpha^n [P (I - \Pi)]^n (T(J) - J)$$
$$= \alpha^n P (I - \Pi) P^{n-1} (T(J) - J) \qquad (32)$$

because from eq. 27, $P\Pi = \Pi$ which implies that $(I - \Pi)P(I - \Pi) = (I - \Pi)P$. Consider a decomposition of $P^{n-1}(T(J) - J)$ along the invariant subspaces of P. There is a subspace corresponding to a unity eigenvalue that is spanned by e, and the component of $P^{n-1}(T(J) - J)$ along that subspace is annihilated by $(I - \Pi)$ (cf. eq. 27). Therefore, r_n will converge to 0 geometrically at a rate determined by the largest complex norm of eigenvalues of αP in a direction other than e (the subdominant eigenvalue norm).

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

$$P = \text{diag} \{ P^1, P^2, \ldots, P^r \} \qquad (33)$$

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

$$W = \begin{bmatrix} 1 \ldots 1\ 0 \ldots 0 \ldots & 0 \\ 0 \ldots 0\ 1 \ldots 1\ 0 \ldots & 0 \\ \vdots & \vdots \\ 0 \ldots 0\ 0 \ldots 0\ 0 \ldots 1 \ldots & 1 \end{bmatrix}^T \qquad (34)$$

and it is easily seen that PW = W. Therefore, the second error term $R_2(J)$ vanishes and the favorable rate estimate of eq. 31 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. 25 and 26, 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 the algorithm to be given in Section 7.

SECTION 5: Adaptive Aggregation with Two Groups

The preceding section showed that the contribution of the second error term $R_2(J)$ of eq. 18 is crucial for the success of our aggregation method. The analysis of this contribution seems

very difficult in general, but the case where m = 2 is tractable and is given in this section. Experiment and some analysis show that the qualitative conclusions drawn from this case carry over to the more general case where m>2. Assume that W, Q have been selected according to eqs. 20 - 22. 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

$$P_a = \begin{bmatrix} 1-b & b \\ c & 1-c \end{bmatrix} \tag{35}$$

where

$$b = \frac{1}{k} \sum_{i=1}^{k} b_i \tag{36a}$$

$$c = \frac{1}{n-k} \sum_{i=k+1}^{n} c_i \tag{36b}$$

$$b_i = \sum_{j=k+1}^{n} P_{ij}, \qquad i = 1, \ldots, k \tag{37a}$$

$$c_i = \sum_{j=1}^{k} P_{ij}, \qquad i = k+1, \ldots, n. \tag{37b}$$

The right eigenvectors and eigenvalues of P_a are

$$v_1 = [\,1 \;\; 1\,]^T \;; \quad v_2 = [\,1 \;\; -c/b\,]^T \tag{38}$$

$$\lambda_1 = 1 \;; \quad \lambda_2 = 1 - b - c. \tag{39}$$

assuming $b \neq 0$. If $b = 0$ then v_2 can be chosen as

$$v_2 = [\,0 \;\; 1\,]^T \tag{40}$$

and $\lambda_1 = 1, \lambda_2 = 1 - c$. From eq. 22 and the form of W we obtain

$$Q = \begin{bmatrix} 1 & 0 \\ k & 1 \\ 0 & (n-k) \end{bmatrix} W^T \qquad (41)$$

We can decompose the term Q(T(J) - J) of eq. 18 into its components along the eigenvectors v_1, v_2, as

$$Q(T(J) - J) = a_1 v_1 + a_2 v_2 \qquad (42)$$

We have $(I - \alpha P_a)v_1 = (1-\alpha)v_1$ from which we obtain

$$W(I - \alpha P_a)^{-1} v_1 = (1-\alpha)^{-1} v_1 \qquad (43)$$

Hence

$$\alpha(I - \Pi)\, PW\, (I - \alpha P_a)^{-1} v_1 = \alpha\,(1-\alpha)^{-1}\,(I - \Pi) P v_1 = 0,$$

and it follows that the only contribution to $R_2(J)$ comes from the term $a_2 v_2$ in eq. 42. Using eqs. 35, 38, and 39 we obtain

$$(I - \alpha P_a)^{-1} v_2 = [\, 1-\alpha + \alpha(b + c)\,]^{-1}\, v_2\,. \qquad (44)$$

Thus, using eq. 24b, we obtain

$$R_2(J) = \alpha(I - \Pi)\, PW\, (I - \alpha P_a)^{-1} a_2 v_2 = \alpha a_2 Q(PW - WP_a)[\, 1-\alpha + \alpha(b + c)\,]^{-1} v_2 \qquad (45)$$

From eqs. 34 - 37, we can calculate the (i,1) element of the matrix $PW - WP_a$ to be

$$\begin{aligned} (PW - WP_a)\,(i,1) &= b - b_i \qquad \text{if } i \le k \\ &= -c + c_i \qquad \text{if } i > k \end{aligned} \qquad (46)$$

Similarly,

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

Thus, from eq. 45

$$R_2(J) = \alpha\, a_2 F(v_2) h \tag{47}$$

where h is the vector with coordinates

$$h(i) = \frac{b - b_i}{1 - \alpha + \alpha(b+c)} \qquad \text{if } i \leq k \tag{48}$$

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

and $F(v_2) = 1 + c/b$ (cf. eqs. 19 and 38). From eqs. 36, 37, and 48 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. 47 and 48 we see that the ith coordinate of q is

$$q(i) = \frac{\alpha^2 a_2 F(v_2)}{1 - \alpha + \alpha(b+c)} \left[\sum_{j=1}^{k} p_{ij} (b - b_j) + \sum_{j=k+1}^{n} p_{ij} (c_j - c) \right] \tag{49}$$

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

$$P = \begin{pmatrix} P^1 & P^2 \\ P^3 & P^4 \end{pmatrix} \tag{50}$$

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. 36, 37, 47, and 48 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. 37 are all nearly equal with their averages b and c respectively. Even if $R_2(J)$ is not near zero, from eq. 49 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

$$\begin{aligned} b_i &= \sum_{j=k+1}^{m_2} P_{ij} && \text{if} \quad i \leq m_1 \\ &= \sum_{j=m_2+1}^{n} P_{ij} && \text{if} \quad k \geq i > m_1 \end{aligned} \tag{51}$$

$$\begin{aligned} c_i &= \sum_{j=1}^{m_1} P_{ij} && \text{if} \quad m_2 \geq i > k \\ &= \sum_{j=m_1+1}^{k} P_{ij} && \text{if} \quad m_2 < i \leq n \end{aligned} \tag{52}$$

Suppose

$$k - m_1 \approx m_2 - k; \quad k \approx n/2 ; \quad k - m_1 << k \tag{53}$$

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. 51 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. 48 that

$$\begin{aligned} h(i) &> 0 \qquad \text{if} \quad i = 1,. . ., m_1 \quad \text{or} \quad i = k+1, . . ., m_2 \\ h(i) &< 0 \qquad \text{otherwise.} \end{aligned} \tag{54}$$

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

$$P = \begin{bmatrix} P^1 & 0 & .1e & 0 \\ 0 & .1 & 0 & .1e^T \\ .1e^T & 0 & .1 & 0 \\ 0 & .1e & 0 & P^2 \end{bmatrix} \tag{55}$$

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

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

In this case, the aggregation matrix W is defined by

$$w_1(i) = 1 - w_2(i) = 1 \qquad \text{if} \quad i \leq 10,$$

$= 0 \qquad$ if $i \geq 11$.

Note that the groups are almost aligned with the ergodic classes of P. Using eqs. 46, 51 and 52, we get

$b = c = .18$

$h(i) = .08\,[\,1-\alpha + \alpha(.36)\,]^{-1}$ if $i \leq 9$

$h(10) = -.72\,[\,1-\alpha + \alpha(.36)\,]^{-1}$

$h(11) = .72\,[\,1-\alpha + \alpha(.36)\,]^{-1}$

$h(i) = -.08\,[\,1-\alpha + \alpha(.36)\,]^{-1}$ if $i > 11$.

From eqs. 38 and 42, we obtain $F(v_2) = 2$ and $a_2 = .8$. Hence,

$$F(R_2(J)) = \alpha\, 1.44\,[\,1-\alpha + \alpha(.36)\,]^{-1}(.8)2 \leq 6.4 \qquad (56)$$

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

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

$= -.144\,[\,1-\alpha + \alpha(.36)\,]^{-1}$ otherwise.

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. 49 we have

$$F(\,T^2(J_1) - T(J_1)) = F(\alpha P R_2(J)) = F(q)$$
$$= \alpha^2.288\,[\,1-\alpha + \alpha(.36)\,]^{-1}(.8)\,2 \leq 1.28.$$

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

$$b_i = \sum_{j=1}^{n-k} P^2{}_{ij} \qquad \text{if } i \leq k \tag{57a}$$

$$c_i = \sum_{j=1}^{k} P^3{}_{ij} \qquad \text{if } i > k. \tag{57b}$$

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. 49 we see that $q \approx 0$. Furthermore, the behavior observed in case 2 is replicated in this case and, 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 Section 7, we formalize these ideas in terms of an overall iterative algorithm.

SECTION 6. Extension to the Average Cost Problem

The aggregation procedure described in section 3 can also be used in the policy evaluation step of the policy iteration algorithm in the average cost case. Here the cost vector for a stationary policy μ is given by

$$J_\mu = \lim_{t\to\infty} (1/T)\, E \{ \sum_{t=0}^{T} g(x(t), \mu(x(t))) \mid \mu \} \qquad (58)$$

As in the discounted cost case, the average cost incurred by policy μ can be characterized by the linear equation (see [1] for a detailed derivation)

$$J_\mu + h_\mu = g_\mu + P_\mu h_\mu. \qquad (59)$$

The vector h_μ is the differential cost incurred by policy μ. In what follows we drop the subscript μ.

The solution of eq. 59 can be computed under certain conditions using the successive approximation method [1]. Fix a state which for simplicity is taken to be state 1. Starting with an initial guess h^0 for the differential cost, the successive approximation method computes h^{n+1} as

$$h^{n+1} = T(h^n) - e\, e_1^T\, T(h^n) \qquad (60)$$

where T(h) is defined by

$$T(h) = g + Ph,$$

and $e_1 = [\, 1, 0, \ldots, 0\,]^T$ is the coordinate vector corresponding to the fixed state 1. Eq. 60 can be written as

$$h^{n+1} = g_A + P_A h^n. \qquad (61)$$

where

$$g_A = (I - e\, e_1^T)\, g$$

$$P_A = (I - e\, e_1^T)P.$$

We assume that all eigenvalues of P except for a single unity eigenvalue lie strictly within the unit circle (see [1] for a method that works under the weaker assumption that P has a single ergodic class). A straightforward calculation shows that $P_A^2 = P_A P$ from which we obtain $P_A^k = P_A P^{k-1}$ for all $k > 0$. Since P_A annihilates the eigenvector e corresponding to the unit eigenvalue of P, it follows that the eigenvalues of P_A all lie strictly inside the unit circle, guaranteeing the convergence of the iteration of eq. 61. Furthermore the rate of convergence is specified by the subdominant eigenvalue of P.

Note that the iteration in eq. 61 is identical to the discounted cost iteration

$$h^{n+1} = g + \alpha P h^n,$$

except that g_A replaces g and P_A replaces αP. Thus, the aggregation and error equations of section 3 can be extended to the average cost problem using the above substitutions. The following lemma establishes that the choice of the matrices Q and W used in section 4 result in a well-posed aggregate problem provided the fixed state 1 forms an aggregate group by itself:

Lemma 3. Assume Q and W are defined by eqs. 20 - 22 with the set G_1 consisting of just state 1, and that all eigenvalues of P except for a single unity eigenvalue lie strictly within the unit circle. Then the aggregate matrix QP_AW has spectral radius less than unity.

Proof: It is straightforward to verify that

$$QP_AW = (I - e_m e_{1,m}^T)P_a, \tag{62}$$

where $P_a = QPW$ is the aggregate stochastic matrix defined in Lemma 1b, e_m is the m-dimensional vector of all 1's, and $e_{1,m}$ is the m-dimensional vector with first coordinate 1, and all other coordinates 0. Therefore, as earlier, we obtain $(QP_AW)^2 = (QP_AW)P_a$ from which

$$(QP_AW)^k = (QP_AW)P_a^{k-1} = (I - e_m e_{1,m}^T)P_a^k, \text{ for all } k > 0. \tag{63}$$

We have $P_a^k = (QPW)^k = QP^kW$ for all $k > 0$, and from this we obtain that P_a has all its eigenvalues strictly within the unit circle except for a single unity eigenvalue. Using this

fact, eq. 63, and the fact that $(I - e_m e_{1,m}^T)$ annihilates the eigenvector e_m corresponding to the single unity eigenvalue of P_a, we see that QP_AW must have all its eigenvalues strictly within the unit circle. q.e.d.

Equation 62 illustrates that the solution to the aggregate linear equation is the solution of an aggregate average-cost problem with transition probabilities P_a. The equations for the aggregation step are:

$$h_1 = h + W(I - QP_AW)^{-1}Q\,(g_A + P_Ah - h)$$

Using this equation we obtain error equations similar to eqs. 23 and 24, indicating that the same choice of Q and W will result in similar acceleration as in the discounted case. This has been verified by the experiments of section 8.

SECTION 7. 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 following 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) < ε, stop and accept

$$T(J) + (1/2)\,\alpha\,(1 - \alpha)^{-1}[\max_i (T(J)-J)(i) - \min_i (T(J)-J)(i)]$$

as the solution (cf. the bounds in eq. 8). Else go to step 3.

Step 3: (Test for an aggregation step) If

$$F(T(J) -J) \leq \omega_1 \tag{64}$$

and

$$F(T(J) -J) \geq \omega_2 \tag{65}$$

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. 26. Compute $T(J_1)$ using eqs. 13 and 14. Set $J := T(J_1)$, $\omega_2 \approx \infty$, and go to step 1.

The purpose of the test of eq. 65 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). The test of eq. 64 guarantees convergence of the overall scheme. To see this note that the test of eq. 64 ensures 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.

An alternative implementation is to eliminate the test of eq. 65 and perform an aggregation step if eq. 64 is satisfied and the number of consecutive iterations during which an aggregation step was not performed exceeds a certain threshold.

SECTION 8: 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

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. Our accounting assumes that an aggregation step requires roughly twice as much computation as a succcessive approximation step which is quite realistic for most problems. Thus 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. 8), 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 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

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

typically contain a mixture of ergodic classes and transient states. This has an adverse effect on both error terms of eq. 18. 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 be 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 7.

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. 24 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

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

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. 12, 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 is then equivalent to the policy iteration algorithm.

Table 3 shows the performance of adaptive aggregation algorithms for the infinite horizon average cost case. In these algorithms, the number of successive approximation steps between aggregation steps was determined adaptively as in the algorithm of section 7, by performing aggregation steps whenever the rate of error reduction of successive approximation steps was slower than .9. Table 3 shows that, while the rate of convergence of successive approximation methods is very sensitive to the strength of the coupling between blocks of P, the rate of convergence of the adaptive aggregation methods remains largely unaffected. In particular, the results for the adaptive algorithms using only two aggregate groups illustrate that major reductions in computation time can be achieved even if the number of aggregate groups is smaller than the number of strongly-connected components of the stochastic matrix P.

TABLE 3: Average Cost Infinite Horizon Problems,
Coupled P, 3 Blocks, 25 states each,
Stopping Tolerance 1.0 E-6

	Successive Approximation	Adaptive Aggregation, 2 Aggregate groups	Adaptive Aggregation 3 aggregate groups
100 % density, 2% coupling	184	62	13
25 % density, 2% coupling	164	26	26
100 % density, 1% coupling	338	64	13
25 % density, 2% coupling	307	43	27
100 % density, .1% coupling	LARGE	71	10
25 % density, .1% coupling	LARGE	50	26

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", Operations Research J., Vol. 33, 1985, pp. 589-606.

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