An Iterative Aggregation Procedure for Markov Decision Processes

ROY MENDELSSOHN

National Marine Fisheries Service, NOAA, Honolulu, Hawaii

(Received April 1980; accepted June 1981)

An iterative aggregation procedure is described for solving large scale, finite state, finite action Markov decision processes (MDPs). At each iteration, an aggregate master problem and a sequence of smaller subproblems are solved. The weights used to form the aggregate master problem are based on the estimates from the previous iteration. Each subproblem is a finite state, finite action MDP with a reduced state space and unequal row sums. Global convergence of the algorithm is proven under very weak assumptions. The proof relates this technique to other iterative methods that have been suggested for general linear programs.

MOST REAL applications of Markov decision processes (MDPs) give rise to very large problems; this is particularly true if the state is represented as a vector of dimension greater than two or three. The major limitation to solving large scale MDPs appears to be in-core storage, as computers are now capable of performing iterations of algorithms for MDPs quickly. However, a 7-dimensional state with only 5 grid points per dimension would have 78,125 states and a transition matrix for each policy that could not be stored in present-day computers. In this paper, an iterative aggregation procedure is described for solving large scale MDPs which relieves this storage burden considerably.

The procedure to be described uses the linear programming (LP) formulation of a discounted MDP (d'Epenoux [1963]) and employs ideas for aggregation of LPs developed in Vakhutinskii and Dudkin [1973], Vakhutinskii et al. [1973, 1979], Agafanov and Makarova [1976], Zipkin [1977, 1980a, b], and Dudkin [1979]. In particular, for the special case of MDPs the procedure is an extension of Zipkin's weighted row and column aggregation of LPs (Zipkin [1977, 1980b]) with optimal disaggregation, and of the extensive Russian literature on iterative aggregation procedures for LPs (Vakhutinskii and Dudkin [1973], Vakhutinskii et al. [1973, 1979], Dudkin). At each iteration, the restrictions on how an aggregate problem may be formed are similar to those described in Thomas [1977] and Whitt [1978] for approximating MDPs.

Subject classifications: 116 finite state Markov decision processes, 637 linear programming—algorithms.

0030-364X/82/3001-0062 $01.25

The major result of this paper is that if at each iteration the weights for aggregating the rows and columns of the LP are chosen properly, then the iterative procedure consisting of alternated aggregation and optimal disaggregation converges globally to an optimal primal and dual solution of the MDP. Convergence is proven using Zangwill's Convergence Theorem A ([1969], p. 19) for algorithms.

For MDPs, the results extend those of Zipkin [1980a] by showing how to iteratively choose the weights for weighted row and column aggregation, and by proving convergence of this procedure. Also, it is proven that for MDPs the LP described in Zipkin [1977] for optimal disaggregation can be reduced to a smaller MDP with unequal row sums (i.e., state and action dependent discount factors). This allows the subproblems to be solved by iterative techniques which are more efficient than linear programming.

The Russian literature on iterative aggregation procedures for LPs involves sequences of complicated unconstrained quadratic programming problems. Proofs of convergence (particularly global convergence), when they have been found (Dudkin, Vakhutinskii et al. [1979]), are complicated. The procedure presented in this paper was motivated by the realization that solving these quadratic programming problems is equivalent to one iteration of an exterior penalty function algorithm for solving (aggregate) LPs. When the quadratic programs (i.e., the exterior penalty functions) are solved as relaxed LPs many of the constraints are redundant. This yields a simpler iterative procedure and a stronger proof of convergence.

The major drawback of the procedure is that each revision of the row weights is equivalent to a dual variable update when using multiplier methods (Rockafellar [1973]), and requires the computational equivalent of one iteration of successive approximations on the full MDP. It is believed that the iterative aggregation process should converge more quickly for large problems than successive approximations on the original MDP. However, several alternative procedures are suggested for calculating the new row weights of each iteration requiring less computation but for which convergence is not proven.

1. THE MODEL

A Markov process is to be controlled over an infinite planning horizon. At the start of each period, a state i from a finite set of N states is observed, an action k is chosen from a finite set of K actions (where for convenience it is assumed the same K actions are available at each state), and a transition is made to state j at the start of the next period with probability $p(i, j{:}k)$.

In each period, if state i is observed and action k is selected, a cost

$c(i, k)$ is incurred. The cost in period t is discounted by a factor β^{t-1}, $0 \le \beta < 1$, and it is desired to minimize the expected total cost over the infinite planning horizon. It is assumed that $c(i, k)$ is bounded (equivalently that $0 \le c(i, k) < \infty$ for all i and k). It is well known (d'Epenoux) that a solution to this MDP can be found by solving the following linear programming problem (LP):

$$\text{Maximize } \textstyle\sum_{i=1}^{N} v(i)$$

$$\text{s.t. } \textstyle\sum_{j=1}^{N} (\delta_{ij} - \beta p(i, j{:}k))v(j) \le c(i, k) \qquad (1.1)$$

$$\text{for} \quad i = 1, \cdots, N; \quad k = 1, \cdots, K$$

where

$$\delta_{ij} = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases}.$$

Dual variables are denoted by $u(i, k)$, $i = 1, \cdots, N$; $k = 1, \cdots, K$, and let $v = \{v(i)\}$, $u = \{u(i, k)\}$. Opitmal primal and dual variables are denoted by $\bar{v} = \{\bar{v}(i)\}$ and $\bar{u} = \{\bar{u}(i, k)\}$.

In forming an aggregate problem attention is restricted to a reasonable subset of possible aggregations, similar to those described in Whitt. The idea is to form a reduced MDP with $N' \le N$ aggregate states and $K' \le K$ actions. In the LP formulation, each state has an associated column and each state-action pair has an associated row. Let σ be a partition of $\{1, 2, \cdots, N\}$ and let ρ be a partition of $\{1, 2, \cdots, K\}$. Let S_n, $n = 1, \cdots, N'$ be the nth subset of σ. Similarly, let A_l, $l = 1, \cdots, K'$ be the lth subset of ρ. Hence S_n references both the states and the associated columns that are aggregated into the nth state and column, and A_l references the actions and associated rows that are aggregated into the lth aggregate action and the associated (n, l)th row.

Following Dudkin, and Vakhutinskii et al. [1979], assume at the tth iteration estimates v^t, u^t for $\bar{v}$, $\bar{u}$ are given. Define the following aggregated terms (Zipkin [1980b]):

$$c^{t+1}(n, l) = \left(\textstyle\sum_{k\in A_l} \sum_{i\in S_n} c(i, k)u^t(i, k)\right)\Big/\left(\textstyle\sum_{k\in A_l} \sum_{i\in S_n} u^t(i, k)\right)$$
$$\text{for} \quad n = 1, \cdots, N'; \quad l = 1, \cdots, K'. \qquad (1.2a)$$

$$\hat{p}^{t+1}(i, m{:}k) = \left(\textstyle\sum_{j\in S_m} (\delta_{ij} - \beta p(i, j{:}k))v^t(j)\right)\Big/\left(\textstyle\sum_{j\in S_m} v^t(j)\right)$$
$$\text{for} \quad i = 1, \cdots, N; \quad m = 1, \cdots, N'; \quad k = 1, \cdots, K. \qquad (1.2b)$$

$$p^{t+1}(n, m{:}l) = \left(\textstyle\sum_{k\in A_l} \sum_{i\in S_n} \hat{p}^{t+1}(i, m{:}k)u^t(i, k)\right)\Big/$$
$$\left(\textstyle\sum_{k\in A_l} \sum_{i\in S_n} u^t(i, k)\right) \qquad (1.2c)$$
$$\text{for} \quad n = 1, \cdots, N'; \quad m = 1, \cdots, N'; \quad l = 1, \cdots, K'.$$

The aggregate terms in (1.2) are defined so as to ensure that for any subset of states either the states are totally aggregated (both across columns and across rows by actions) or else the states are not aggregated at all across either rows or columns. For notational convenience, it is assumed that the partitions do not change with each iteration. However, the proofs do not depend on this assumption, and some basis for choosing the partitions at each iteration is supplied by the algorithm.

Let $f_0(x)$, $f_i(x)$, $i = 1, \cdots, k$ be concave functions. For the problem: Max $f_0(x)$, s.t. $f_i(x) \leq 0$ for $i = 1, \cdots, k$, the Lagrangean $L(x, \lambda)$ is given by:

$$\begin{cases} f_0(x) + \sum_i \lambda_i f_i(x) & \text{if } \lambda_i \geq 0 \quad \text{for} \quad i = 1, \cdots, k \\ \infty & \text{otherwise.} \end{cases}$$

Let $x^* = \operatorname{argmax}_x L(x, \lambda)$ and $\lambda^* = \operatorname{argmin}_\lambda L(x, \lambda)$. The well known result that $L(x^*, \lambda) \geq L(x^*, \lambda^*) \geq L(x, \lambda^*)$ is used in Theorem 2.2.

2. THE ALGORITHM AND ITS PROPERTIES

The iterative aggregation procedure is as follows. Choose any v^0, u^0 with $\infty > v^0 \geq 0$ and $\infty > u^0 \geq 0$. Assume that after iteration t, v^t and u^t are given.

Step (*i*). Form the aggregate coefficients defined in (1.2).

Step (*ii*). Solve the master program:

$$\text{Maximize } \textstyle\sum_{n=1}^{N'} z(n)$$

$$\text{s.t. } \textstyle\sum_{m=1}^{N'} p^{t+1}(n, m\!:\!l) z(m) \leq c^{t+1}(n, l) \tag{2.1a}$$

$$\text{for} \quad n = 1, \cdots, N' \quad \text{and} \quad l = 1, \cdots, K'.$$

Denote a primal solution to (2.1a) by $z^{t+1} = \{z^{t+1}(n)\}$, and the dual solution by $\lambda^{t+1} = \{\lambda^{t+1}(n, l)\}$.

Step (*iii*). Solve an LP for each $n = 1, \cdots, N'$:

$$\text{Maximize } \textstyle\sum_{i \in S_n} v(i)$$

$$\text{s.t. } \textstyle\sum_{j \in S_n} (\delta_{ij} - \beta p(i, j\!:\!k)) v(j) \leq z^{t+1}(n) \hat{p}^{t+1}(i, n\!:\!k) \tag{2.1b}$$

$$\text{for} \quad i \in S_n \quad \text{and} \quad k = 1, \cdots, K.$$

Let v^{t+1} be the vector consisting of the optimal solutions to all of these problems, and denote the optimal dual variables by $\pi_n^{t+1}(i, k)$, where the subscript n denotes that $i \in S_n$.

Step (*iv*). Update the dual variables

$$\begin{aligned} u^{t+1}(i, k) = \{&(u^t(i, k)\lambda^{t+1}(n, l)/\textstyle\sum_{k \in A_l} \sum_{i \in S_n} u^t(i, k)) - (c(i, k) \\ &+ \beta \textstyle\sum_{j=1}^{N} p(i, j\!:\!k) v^{t+1}(j) - v^{t+1}(i))\}^+ \quad \text{for} \quad i \in S_n; \quad k \in A_l \end{aligned} \tag{2.2}$$

where $\{a\}^+ = \max(a, 0)$. Return to Step (i), or else stop if fixed point has been found.

Comment. The Russian literature (Vakhutinskii and Dudkin) also suggests updating the dual variables using the formula (similar but not identical to (2.2))

$$u^{t+1}(i, k) = (\lambda^{t+1}(n, l)/\textstyle\sum_{k\in A_l}\sum_{i\in S_n} u^t(i, k))\{u^t(i, k) \\ - (c(i, k) + \beta \textstyle\sum_{j=1}^{N} p(i, j:k)v^{t+1}(j) - v^{t+1}(i))\}^+.$$

A fixed point of the iterative process, if one exists, is denoted by v^*, u^*. The corresponding values of z, λ are denoted by z^*, λ^*. Note that the N' LPs (2.1b) have dual LPs that are Leontief (Koehler et al. [1975]), and hence can be solved by iterative techniques. Each such LP can also be viewed as an MDP with unequal row sums. The update (2.2) is a modification of the usual update in the method of multipliers.

It is conjectured that the dual LP to (2.1a) is also Leontief. This is true for the few small numerical examples solved to date using the algorithm.

An alternative to Step (iii) requiring less computational effort is fixed-weight disaggregation (Zipkin [1977; 1980a, b]):

$$v^{t+1}(i) = (v^t(i)z^{t+1}(n))/(\textstyle\sum_{j\in S_n} v^t(j)) \quad \text{for} \quad i \in S_n. \tag{2.3}$$

Similarly, Step (iv) can be replaced by a fixed-weight disaggregation:

$$u^{t+1}(i, k) = (u^t(i, k)\lambda^{t+1}(n, l))/(\textstyle\sum_{k\in A_l}\sum_{i\in S_n} u^t(i, k)) \\ \text{for} \quad i \in S_n; \quad k \in A_l \tag{2.4}$$

or, noting that Step (iii) of the algorithm yields a unique value of $\pi_n^{t+1}(i, k)$ for each (i, k):

$$u^{t+1}(i, k) = \pi_n^{t+1}(i, k). \tag{2.5}$$

Yet another alternative is "optimal disaggregation" (Zipkin [1977]), which finds a solution with higher objective value than (2.3):

$$\text{Maximize } \textstyle\sum_{i\in S_n} v(i)$$

$$\text{s.t. } \textstyle\sum_{j\in S_n} (\delta_{ij} - \beta p(i, j:k))v(j) \leq z^{t+1}(n)\hat{p}^t(i, n:k) \\ \text{for} \quad i \in S_n; \quad k = 1, \cdots, K \tag{2.6a}$$

$$\textstyle\sum_{j\in S_n} (-\beta p(i, j:k))v(j) \leq z^{t+1}(n)\hat{p}^t(i, n:k) \\ \text{for} \quad i \notin S_n; \quad k = 1, \cdots, K. \tag{2.6b}$$

Denote as before the optimal dual variables to (2.6a) as $\pi_n^{t+1}(i, k)$, and those of (2.6b) by $w_p^{t+1}(i, k)$, $p \neq n$. Before proving properties of the iterative algorithm procedure, it is necessary to prove that a solution to (2.1b) solves (2.6).

LEMMA 2.1. *In the LP* (2.6), *none of the constraints* (2.6b) *are binding, that is* $w_p^{t+1}(i, k) \equiv 0$.

Proof. The dual LP to (2.6) has the following general form

$$\text{Minimize } \textstyle\sum_{i\in S_n} \sum_{k=1}^{K} \pi_n(i, k)h(i, k) + \sum_{i\notin S_n} \sum_{k=1}^{K} w_p(i, k)h(i, k)$$

$$\text{s.t. } \textstyle\sum_{j\in S_n} \sum_{k=1}^{K} (\delta_{ij} - \beta p(j, i{:}k))\pi_n(i, k)$$

$$+ \textstyle\sum_{J\notin S_n} \sum_{k=1}^{K} (-\beta p(j, i{:}k))w_p(i, k) = 1 \quad \text{for} \quad i \in S_n.$$

The dual LP is obviously Leontief. The lemma follows from Theorem 2.5.1 in Koehler et al.

The existence of an optimal solution to (1.1) has been well-established. The existence of a fixed point (v^*, u^*) for the iterative process will be proven by showing that (v^*, u^*) is a fixed point if and only if it is optimal in (1.1).

THEOREM 2.1. (v^*, u^*) *is a fixed point of the iterative process described in Steps* (i)-(iv) *if and only if it is primal and dual optimal for* (1.1).

Proof. $(\bar{v}, \bar{u})$ *is a fixed point.*

The proof proceeds by showing that if $v^t = \bar{v}$, $u^t = \bar{u}$, then $z^{t+1}(n) = \sum_{i\in S_n} \bar{v}(i)$, and $\lambda^{t+1}(n, l) = \sum_{i\in S_n} \sum_{k\in A_l} \bar{u}(i, k)$. These values are optimal in the master problem if they are feasible and if they satisfy complementarity conditions. The latter is

$$\textstyle\sum_{n=1}^{N'} \sum_{l=1}^{K'} \{(\sum_{n=1}^{N} (\sum_{i\in S_n} \bar{v}(i)) \sum_{j\in S_n} (\delta_{ij} - \beta p(i, j{:}k))(\bar{v}(j)/\sum_{i\in S_n} \bar{v}(i))$$

$$- c(i, k))(\bar{u}(i, k)/\textstyle\sum_{i\in S_n} \sum_{k\in A_l} \bar{u}(i, k))\} (\sum_{i\in S_n} \sum_{k\in A_l} \bar{u}(i, k)) = 0.$$

After cancelling out terms, this reduces to:

$$\textstyle\sum_{i=1}^{N} \sum_{k=1}^{K} (\bar{v}(i) - (c(i, k) + \beta \sum_{j=1}^{N} p(i, j{:}k)\bar{v}(j)))\bar{u}(i, k) = 0$$

the complementarity condition for (1.1), which is true by assumption. Since $\bar{v}$, $\bar{u}$ are primal and dual feasible, positive weighted sums of the rows and columns cannot change this. Hence $z^{t+1}(n) = \sum_{i\in S_n} \bar{v}(i)$ and $\lambda^{t+1}(n, l) = \sum_{i\in S_n} \sum_{k\in A_l} \bar{u}(i, k)$ are primal and dual feasible in the master problem.

Depending on how the partitions are chosen, $\lambda^{t+1}(n, l) = \sum_{i\in S_n} \sum_{k\in A_l} \bar{u}(i, k)$ may be positive for more than N' values of (n, l). However, at $z^{t+1}(n) = \sum_{i\in S_n} \bar{v}(i)$, the (n, l)th constraint in (2.1a) becomes:

$$(1/\textstyle\sum_{k\in A_l} \sum_{j\in S_n} \bar{u}(i, k)) \sum_{k\in A_l} \sum_{j\in S_n} \bar{u}(i, k)$$

$$\cdot[\textstyle\sum_{j=1}^{N} (\delta_{ij} - \beta p(i, j{:}k))\bar{v}(j) - c(i, k)] \leq 0$$

which from the optimality of $(\bar{v}, \bar{u})$ holds with equality for each (n, l). The master LP hence has possibly degenerate optimal solutions. The values $\lambda^{t+1}(n, l) = \sum_{i\in S_n} \sum_{k\in A_l} \bar{u}(i, k)$ in practice can be found by finding all possible optimal solutions to (2.1a).

Using the LP form of Step (iii) given in (2.6), it is evident that $v^{t+1} = \bar{v}$. Since $\bar{v}$ is feasible and optimal, the second term on the right-hand side of (2.2) is always nonpositive. The first term reduces to $\bar{u}(i, k)$. If $\bar{u}(i, k)$ is zero, then the brackets imply $\bar{u}^{t+1}(i, k) = 0 = \bar{u}(i, k)$. If $\bar{u}(i, k) > 0$, then

$$c(i, k) + \beta \textstyle\sum_{j=1}^{N} p(i, j{:}k)\bar{v}(j) - \bar{v}(i) = 0, \quad \text{so again} \quad u^{t+1}(i, k) = \bar{u}(i, k).$$

(v^*, u^*) is optimal in (1.1).

(v^*, u^*) is optimal in (1.1) if v^* is primal feasible, u^* is dual feasible and (v^*, u^*) satisfies the complementarity conditions. As (v^*, u^*) is a fixed point by assumption, then $z^*(n) = \sum_{i\in S_n} v^*(i)$, $\lambda^*(n, l) = \sum_{i\in S_n} \sum_{k\in A_l} u^*(i, k)$. From (2.2), v^* is feasible in (1.1) or else u^* would not be a fixed point.

As above, writing out the complementarity conditions for the master problem (2.1a) at (z^*, λ^*) and substituting in their values yields:

$$\textstyle\sum_{i=1}^{N} \sum_{k=1}^{K} (v^*(i) - (c(i, k) + \beta \sum_{j=1}^{N} p(i, j{:}k)v^*(j)))u^*(i, k) = 0$$

so (v^*, u^*) indeed satisfies the complementarity condition.

At $\lambda^*(n, l) = \sum_{i\in S_n} \sum_{k\in A_l} u^*(i, k)$, (2.2) becomes

$$u^*(i, k) = (u^*(i, k) - (c(i, k) + \beta \textstyle\sum_{j=1}^{N} p(i, j{:}k)v^*(j) - v^*(i)))^+$$

which is the usual dual update for the method of multipliers. As $u^*(i, k)$ is invariant at v^*, it follows from Rockafellar that u^* is dual optimal.

It is shown in Zipkin [1977] that "optimal" weightings should be proportional to optimal values. A theorem similar to Theorem 2.1 is proven in Vakhutinskii et al. [1979]. The possible degeneracy of the master LP (2.1a) suggests that at each iteration, partitions should be chosen such that $\sum_{k\in A_l} \sum_{j\in S_n} u^t(i, k) > 0$ for only one l for each $n = 1, \cdots, N'$. In this case, the algorithm reduces to an aggregation procedure for policy evaluation, and a multiplier type step for policy improvement. Otherwise, partitions should be chosen such that the dual constraint matrix is Leontief.

Theorem 2.2 states the main results of this paper, that under very weak conditions the iterative aggregation procedure converges to a solution of (1.1). Note that the iterative aggregation procedure can be viewed as a point to set map A: $R_+^N \times R_+^{N\times K} \rightarrow R_+^N \times R_+^{N\times K}$. The theorem is proven by showing that the algorithm satisfies the conditions of Zangwill's Theorem A (Zangwill [1969]).

THEOREM 2.2. *Assume for $(v^0, u^0) > 0$ that v^1, u^1 are nonnegative and bounded when one iteration of the procedure is performed. Let $\{(v^t, u^t)\}_{t=1}^{\infty}$ be the sequence of vectors generated by the algorithm. Then*

either $\{(v^t, u^t)\}$ converges to $(\bar{v}, \bar{u})$ or else there is a convergent subsequence with $(\bar{v}, \bar{u})$ as the limit of the subsequence.

Proof. It is necessary to show that A is an upper semicontinuous point to set map defined on a compact set, and that there exists some function that changes monotonically with each iteration of the algorithm.

(i) *The penalty function.*

$$x(v^t) = \textstyle\sum_{i=1}^{N} v^t(i) - 1/e \sum_{i=1}^{N} \sum_{k=1}^{N} ([\sum_{j=1}^{N} (\delta_{ij} - \beta p(i, j{:}k))v^t(j) - c(i, k)]^+)^2$$

monotonically decreases with t for some fixed value e > 0.

Remark. This result relates iterative aggregation for MDPs to the iterative algorithm for nonaggregated linear programs proposed in Mangasarian [1979].

Bertsekas [1975] has shown that there exists for linear programs an exact penalty function method such that for all e in the interval $(0, \bar{e})$, the penalty function algorithm converges. Let e be one such value. Using (2.2), the Lagrangean function for the master problem (2.1a) at iteration $t + 1$ can be written as

$$\begin{aligned}\text{Maximize } &\textstyle\sum_{n=1}^{N'} z(n) - \sum_{n=1}^{N'} \sum_{k=1}^{K'} w(n, l)\\ &\cdot \{\textstyle\sum_{k\in A_l} \sum_{i\in S_n} ((\lambda^t(n, l)u^{t-1}(i, k)/\sum_{k\in A_l} \sum_{i\in S_n} u^{t-1}(i, k))\\ &+ \textstyle\sum_{j=1}^{N} (\delta_{ij} - \beta p(i, j{:}k))v^t(j) - c(i, k))^+ \\ &\cdot (\textstyle\sum_{n=1}^{N'} z(n) \sum_{j\in S_n} (\delta_{ij} - \beta p(i, j{:}k))(v^t(j)/\sum_{j\in S_n} v^t(j)) - c(i, k))\}.\end{aligned} \tag{2.8}$$

At $w(n, l) \equiv 1/e$ and at the trial value $z(n) = \sum_{i\in S_n} v^t(i)$, (2.8) reduces to:

$$\begin{aligned}&\textstyle\sum_{i=1}^{N} v^t(i) - (1/e) \sum_{n=1}^{N'} \sum_{l=1}^{K'} \{\sum_{k\in A_l} \sum_{i\in S_n} ((\lambda^t(n, l)u^{t-1}(i, k)/\\ &\quad \textstyle\sum_{k\in A_l} \sum_{i\in S_n} u^{t-1}(i, k)) + \sum_{j=1}^{N} (\delta_{ij} - \beta p(i, j{:}k))v^t(j) - c(i, k))^+\\ &\qquad \cdot (\textstyle\sum_{j=1}^{N} (\delta_{ij} - \beta p(i, j{:}k))v^t(j) - c(i, k))\}.\end{aligned} \tag{2.9}$$

At iteration t, let I^- denote the set of (i, k) such that (2.2) is nonpositive, and I^+ the set of (i, k) such that (2.2) is positive. Then (2.9) can be rewritten as:

$$\begin{aligned}&\textstyle\sum_{j=1}^{N} v^t(i) - (1/e) \sum_{n=1}^{N'} \sum_{l=1}^{K'} \{\sum_{k\in A_l} \sum_{i\in S_n} (\lambda^t(n, l)u^{t-1}(i, k)/\\ &\quad \textstyle\sum_{k\in A_l} \sum_{i\in S_n} u^{t-1}(i, k))(\sum_{j=1}^{N} (\delta_{ij} - \beta p(i, j{:}k))v^t(j) - c(i, k))\}\\ &\quad - (1/e) \textstyle\sum_{(i,k)\in I^+} (\sum_{j=1}^{N} (\delta_{ij} - \beta p(i, j{:}k))v^t(j) - c(i, k))^2\\ &\quad + (1/e) \textstyle\sum_{(i,k)\in I^-} \lambda^t(n, l)u^{t-1}(i, k)/\sum_{k\in A_l} \sum_{i\in S_n} u^{t-1}(i, k)\\ &\qquad \cdot (\textstyle\sum_{j=1}^{N} (\delta_{ij} - \beta p(i, j{:}k))v^t(j) - c(i, k)).\end{aligned} \tag{2.10}$$

If in the second term of (2.10),

$$(\textstyle\sum_{j=1}^{N} (\delta_{ij} - \beta p(i, j{:}k))v^t(j) - c(i, k))$$

were replaced by

$$\textstyle\sum_{n=1}^{N'} (\sum_{j\in S_n} (\delta_{ij} - \beta p(i, u{:}k))z^t(n)(v^{t-1}(j)/\sum_{j\in S_n} v^{t-1}(j))) - c(i, k)$$

then from the optimality of $z^t(n)$, $\lambda^t(n, l)$ in (2.1a), the entire term would be zero. However, (2.1b) ensures that

$$\begin{aligned}&\textstyle\sum_{j=1}^{N} (\delta_{ij} - \beta p(i, j{:}k))v^t(j)\\ &\qquad\le \textstyle\sum_{n=1}^{N'} (\sum_{j\in S_n} (\delta_{ij} - \beta p(i, j{:}k))(z^t(n)v^{t-1}(j)/\sum_{j\in S_n} v^{t-1}(j)))\end{aligned}$$

and that their respective weighted sums (over $k \in A_l$, $i \in S_n$), weighted by

$$(\lambda^t(n, l)u^{t-1}(i, k))/(\textstyle\sum_{k\in A_l} \sum_{i\in S_n} u^{t-1}(i, k)),$$

are identical. Hence the second term in (2.10) is still identically zero. For $(i, k) \in I^-$, by definition

$$\textstyle\sum_{j=1}^{N} (\delta_{ij} - \beta p(i, j{:}k))v^t(j) - c(i, k) \le 0,$$

hence

$$\begin{aligned}&(\lambda^t(n, l)u^{t-1}(n, l))/(\textstyle\sum_{k\in A_l} \sum_{i\in S_n} u^{t-1}(n, l))\\ &\qquad\qquad \cdot (\textstyle\sum_{j=1}^{N} (\delta_{ij} - \beta p(i, j{:}k))v^t(j) - c(i, k)) \le 0.\end{aligned}$$

These two results imply that (2.10) is less than or equal to $x(v^t)$.

Moreover, this implies for $w(n, l) \equiv 1/e$, $\max_z L(z, w)$ occurs at $z(n) = \sum_{i\in S_n} v^t(i)$. Therefore $L(z^{t+1}, \lambda^{t+1}) \le L(z^{t+1}, 1/e) \le L(z^t, 1/e) \le x(v^t)$. The optimality of z^{t+1}, λ^{t+1} in the master LP (2.1a) and the fact that v^{t+1} solves the N' problems (2.1b) guarantee that the penalty function evaluated at v^{t+1} is less than or equal to $L(z^{t+1}, \lambda^{t+1})$.

(ii) *The sequence* $\{v^t, u^t\}$ *is bounded.*

By assumption, $x(v^1)$ is finite, and from part (i) of the proof, $x(v^t)$ is nonincreasing with t, so the sequence $\{v^t, u^t\}$ is bounded above. Straightforward algebra proves that the penalty function for (2.1) evaluated at z^t provides a lower bound for $x(v^t)$. Since z^t is optimal in (2.1) at iteration t, $x(v^t) \ge 0$. By definition, $u^t \ge 0$. Together, these facts imply $\{v^t, u^t\}$ is bounded from below.

(iii) *The point to set map A is upper semicontinuous (closed).*

Since the partitions are the same each iteration, there exists three constant matrices $T^1((N'K') \times 1))$, $T^2(1 \times N')$, and $T^3(NK \times N)$ such that the constraints for the master problem in Step (i) can be written

$$(T^1u')T^3(vT^2) \leq (T^1u')c.$$

As the mapping is linear and continuous, and the constraint defines a convex set, the mapping A^1 is closed. Zangwill [1969] proves that the maximization operator is closed; call this A^2. Similarly, the constraints for the subproblems form closed maps, call them A^3, A^4, $\cdots$, $A^{N'+2}$, and Equation 2.2 is trivially closed, call this map A^0. Therefore the map A can be written as $A = A^0A^2A^{N'+2} \cdots A^2A^3A^2A^1$. The assumption that (v^1, u^1) is finite guarantees that the algorithm produces a sequence of bounded vectors, that is, A is defined on a compact set. Closedness of A then follows from Corollary 4.2.1 in Zangwill ([1969], p. 96).

(iv) *The algorithm converges to a fixed point.*

That the algorithm converges follows from Theorem A in Zangwill [1969]. To show that it converges to a fixed point, the algorithm maps the sequence $\{v^t, u^t\}_{t=0}^{\infty}$ into the sequence $\{v^t, u^t\}_{t=1}^{\infty}$. Two subsequences of a convergent sequence on a compact set have the same limit point. Denote it by (v^*, u^*). However,

$$(v^*, u^*) = \lim_{t\to\infty} A\colon (v^t, u^t) = A(v^*, u^*)$$

the final equality following from A being a closed map.

When Steps (i)-(iv) of the iterative aggregation process are used only every kth iteration, and the alternatives (2.3), (2.4), or (2.5) are used at all other iterations, proofs of convergence follow closely the proof of Theorem 2.2 to derive the conditions necessary for Zangwill's other convergence theorems. Intuitively, either a fixed point is found, or else every kth iteration of the algorithm, the equivalent of one iteration of successive approximations is calculated. As long as the intermediate steps do not force the value function v and the dual variables u in undesirable directions, the algorithm converges since successive approximations converge when applied to MDPs.

Finally, it is conjectured that convergence can be proven in an analogous manner to Theorem 2.2 if only some subset of the dual variables are adjusted by Equation 2.2 at each iteration. (This is in the spirit of several relaxation algorithms (Agmon [1954], Motzkin and Schoenberg [1954]).) For example, one partition at a time could be updated each iteration using Equation 2.2.

Theorem 2.2 can also be proven using the augmented Lagrangean in part (iii) instead of the penalty function. As the dual update (2.2) is a method of multipliers update, the algorithm approximates the maximum v for each u^{t+1} in the augmented Lagrangean. Sufficient conditions for such approximate procedures to converge are given in Rockafellar.

Clearly the algorithm need not be run until it converges if approximate solutions are satisfactory. Since at the end of each iteration π_n^t describes a nonrandomized policy that can be followed, any of the bounds for aggregate LPs or MDPs given in Whitt, Zipkin [1980b], and Mendelssohn [1980] can be used as stopping rules.

3. CONCLUSION

An iterative aggregation procedure for MDPs has been presented which converges globally to an optimal value function and to optimal dual variables. The process requires less in-core storage at any point than does solving the full MDP. However, each iteration requires at least the computational equivalent of one iteration of successive approximations. Convergence should be more rapid using the iterative aggregation process.

To reduce the computational burden, several alternative procedures are presented at key steps. Convergence has not been proven when these procedures are used; however, if the full iterative aggregation process is used every *k*th iteration, then again the algorithm converges globally.

There should exist a more efficient computational method for updating the dual variables at each iteration, a method which converges globally. Improvements in this area should lead to truly efficient means for solving large-scale MDPs.

ACKNOWLEDGMENT

I am grateful to Paul Zipkin, to an anonymous referee, and to the area editor for insightful comments and to the Center for Coastal Marine Studies, University of California at Santa Cruz, for its support.

REFERENCES

AGAFANOV, G. V., AND A. S. MAKAROVA. 1976. An Algorithm for Iterative Aggregation of Economic Hierarchy Systems as an Instrument for Consistency of Solutions of Multilevel Systems. *Optimization Methods and Operations Research* (*Metody Optimizatsii i Issledovanie Operatsii*, Akademiia Nauk SSSR, Sibirskoe Otdelenie, Energeticheskii Institut, pp. 132-148). (Translated from Russian by Wilvan G. Van Campen for the Southwest Fisheries Center Honolulu Laboratory, Natl. Mar. Fish. Serv., NOAA, Honolulu, HI 96812, 1980, 18 pp. Translation No. 43, limited distribution.)

AGMON, S. 1954. The Relaxation Method for Linear Inequalities. *Can. J. Math.* **6,** 382-392.

BERTSEKAS, D. P. 1975. Necessary and Sufficient Conditions for a Penalty Method to Be Exact. *Math. Program.* **9,** 87-99.

D'EPENOUX, G. 1963. A Probabilistic Production and Inventory Problem. *Mgmt. Sci.* **10,** 98-108.

DUDKIN, L. M. 1979. *Iterative Aggregation and Its Application in Planning.* Moscow.

KOEHLER, G. J., A. B. WHINSTON AND G. P. WRIGHT. 1975. *Optimization over Leontief Substitution Systems.* North Holland/American Elsevier, New York.

MAÑGASARIAN, O. L. 1979. Iterative Solution of Linear Programs, Computer Sciences Technical Report No. 327, University of Wisconsin, Madison.

MENDELSSOHN, R. 1980. Improved Bounds for Aggregated Linear Programs. *Opns. Res.* **28,** 1450–1453.

MOTZKIN, TH., AND I. J. SCHOENBERG. 1954. The Relaxation Method for Linear Inequalities. *Can. J. Math.* **6,** 393–404.

ROCKAFELLAR, R. T. 1973. The Multiplier Method of Hestenes and Powell Applied to Convex Programming. *J. Optim. Theory Its Appl.* **12,** 555–562.

THOMAS, A. 1977. *Models for Optimal Capacity Expansion*, Ph.D. thesis, Yale University, New Haven, Conn.

VAKHUTINSKII, I. YA., AND L. M. DUDKIN. 1973. Algorithm of Iterative Aggregation for the Solution of the Problem of Linear Programming of a General Nature. *Izvestiia of the Siberian Section of the Academy of Sciences of the USSR Social Science Series* (Akademiia Nauk SSSR, Sibirskoe Otdelenie, Izvestiia, Novosibirski, Seriia Obshchestvendykh Nauk s(11), 67–71). (Translated from Russian by Wilvan G. Van Campen for the Southwest Fisheries Center Honolulu Laboratory, Natl. Mar. Fish. Serv., NOAA, Honolulu, HI 96812, 8 p. Translation No. 41, limited distribution.)

VAKHUTINSKII, I. YA., L. M. DUDKIN AND A. MAKAROV. 1973. An Iterative Aggregation Algorithm for Connecting a System of Branch Planning Models (with Energetics Example). *Automation and Remote Control* **10,** 145–159.

VAKHUTINSKII, I. YA., L. M. DUDKIN AND A. A. RYVKIN. 1979. Iterative Aggregation—A New Approach to the Solution of Large-Scale Problems. *Econometrica* **47,** 821–841.

WHITT, W. 1978. Approximations of Dynamic Programs. *Math. Opns. Res.* **3,** 231–243.

ZANGWILL, W. 1969. *Nonlinear Programming—A Unified Approach.* Prentice-Hall, Englwood Cliffs, N.J.

ZIPKIN, P. 1977. *Aggregation in Linear Programming*, Ph.D. dissertation, Yale University, New Haven, Conn., 189 pp.

ZIPKIN, P. 1980a. Bounds on the Effect of Aggregating Variables in Linear Programs. *Opns. Res.* **28,** 403–418.

ZIPKIN, P. 1980b. Bounds for Row-Aggregation in Linear Programming. *Opns. Res.* **28,** 903–916.