AN ITERATIVE AGGREGATION PROCESS FOR MARKOV DECISION PROCESSES

Roy Mendelssohn
Southwest Fisheries Center
National Marine Fisheries Service
National Oceanic and Atmospheric Administration
Honolulu, Hawaii 96812

ABSTRACT

An iterative aggregation procedure is described for solving large scale, finite state, finite action Markov decision processes (MDP). 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. The subproblems are each a finite state, finite action MDP with a reduced state space and unequal row sums. Global convergence is proven for the iterative aggregation process under very weak assumptions. The proof of convergence relates this iterative aggregation technique to other iterative techniques that have been suggested for solving nonaggregate linear programs.

Most real life applications of Markov decision processes (MDPs) require the ability to solve very large problems; this is particularly true if the state is a vector of dimension greater than two or three. The major limitation appears to be in-core storage. Computers can perform large numbers of calculations in a relatively short time frame. However, a 7-dimension state with only five grid points per dimension would have 78,125 states and a transition matrix for each policy that could not be stored in-core in present day computers. In this paper an iterative aggregation procedure is described for solving large scale MDPs. The results are an extension of the work by Zipkin [10, 11, 12] on fixed-weight row and column aggregation and optimal disaggregation of linear programs. The procedure also improves upon the results of an extensive Russian literature on iterative aggregation processes [1, 4, 6, 7, 8].

The major result in this paper is that if the aggregation weights each iteration are chosen properly, then an iterative procedure of aggregation and optimal disaggregation converge to the optimal primal and dual solutions of the MDP. Moreover, it is proven that the subproblems to calculate optimal disaggregation have a special structure that reduces them to a sequence of reduced state MDPs with unequal row sums (i.e., state dependent discount factors). This allows the subproblems to be solved by more efficient iterative techniques, rather than by linear programming. Finally, alternate methods of performing the updates are presented. The algorithm does not necessarily converge using these procedures. However, it is proven that if the full algorithm is used every k^{th} iteration, then the algorithm does converge to an optimum.

The major drawback of the iterative procedure to be described is that the full algorithm requires the computational equivalent of one iteration of successive approximation to update the dual variables. However, it is believed that the value function should converge more quickly than simply performing some version of successive approximations on the full problem. By using one of the alternative dual updates for the majority of the iterations, the computational burden should be reduced considerably.

2. 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, and a transistion 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}, $0 \leq \beta < 1$, and it is desired to minimize the expected total cost over the infinite planning horizon. It is assumed that $c(i, k)$ is bounded, or equivalently that $0 \leq c(i, k) < \infty$ for all i and k. It is well known [3] that a solution to this MDP can be found by solving the following linear programming problem (LP):

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

$$\text{s.t.} \quad \sum_{j=1}^{N} \{\delta_{ij} - \beta p(i, j:k)\} v(j) \leq c(i, k) \qquad i = 1, \ldots, N \quad k = 1, \ldots, K \tag{2.1}$$

where $\delta_{ij} = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases}$. Dual variables are denoted by $u(i, k)$, $i = 1, \ldots, N$; $k = 1, \ldots, K$, and let $v = \{v(i)\}$, $u = \{u(i, k)\}$. Optimal 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. At each iteration, it is assumed that the same rows and columns are aggregated. Further, it only is possible to aggregate actions within a (possibly aggregate) state or to completely aggregate (possibly aggregate) states. In the former case only rows are aggregated, while in the latter case rows and columns are aggregated.

More formally, let σ be a partition of $\{1, 2, \ldots, N\}$, and let ρ be a partition of $\{1, 2, \ldots, K\}$. Let $\{s_n\}$, $n = 1, \ldots, N' \leq N$ be the set of indices contained in the n^{th} partition of σ. Similarly, let $\{A_\ell\}$, $\ell = 1, \ldots, K' \leq K$ be the set of indices contained in the ℓ^{th} partition of ρ. Thus, the MDP has been reduced to one with N' states and K' actions per state.

Following [4], assume that at the t^{th} iteration the estimates v^t, u^t for v, u are given. Define the following doubly aggregated terms [11]:

$$c^{t+1}(n, \ell) = \frac{\sum_{k \in A_\ell} \sum_{i \in s_n} c(i, k)\, u^t(i, k)}{\sum_{k \in A_\ell} \sum_{i \in s_n} u^t(i, k)} \qquad n = 1, \ldots, N'; \quad \ell = 1, \ldots, K'. \tag{2.2a}$$

$$\hat{p}^{t+1}(i, m:k) = \frac{\sum_{j \in s_m} \left(\delta_{ij} - \beta p(i, j:k)\right) v^t(j)}{\sum_{j \in s_m} v^t(j)} \qquad i = 1, \ldots, N; \quad m = 1, \ldots, N'; \quad k = 1, \ldots, K. \tag{2.2b}$$

$$p^{t+1}(n, m:\ell) = \frac{\sum_{k\varepsilon A_\ell} \sum_{i\varepsilon s_n} \hat{p}(i, m:k)u^t(i, k)}{\sum_{k\varepsilon A_\ell} \sum_{i\varepsilon s_n} u^t(i, k)} \qquad \begin{array}{l} n = 1, \ldots, N'; \\ m = 1, \ldots, N'; \\ \ell = 1, \ldots, K'. \end{array} \tag{2.2c}$$

It is easy to see that the objective function coefficients on the aggregate variables will still be one using this aggregation scheme.

3. THE ALGORITHM AND ITS PROPERTIES

The iterative aggregation process to be described is similar to the ones in [4, 6, 7, 8] when it is realized that the quadratic objective functions suggested are exactly those terms that would arise from using an exterior penalty method to solve a linear programming problem. Each master problem is a doubly aggregated LP as in [11], and each subproblem is a solution of an optimal disaggregation subproblem [12, Chapter 7]. Assume that after iteration t, v^t and u^t are given.

<u>Step (i)</u> Form the aggregate coefficients defined in (2.2)

<u>Step (ii)</u> Solve the master program:

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

$$\text{s.t.} \quad \sum_{m=1}^{N'} p^{t+1}(n, m:\ell) \leq c^{t+1}(n, \ell) \qquad \begin{array}{l} n = 1, \ldots, N' \\ \ell = 1, \ldots, K' \end{array} \tag{3.1}$$

$$z(n) \geq 0 \qquad n = 1, \ldots, N'$$

(The additional constraints $z(n) \geq 0$ is justified since it is assumed $c(i, k) \geq 0$.) Denote a primal solution to (3.1) by $z^{t+1} = \{z^{t+1}(n)\}$, and the dual solution by $\lambda^{t+1} = \{\lambda^{t+1}(n, \ell)\}$.

Step (iii) Solve N' MDPs

Each MDP has as its state space the states indexed in s_n; its action space is the original action space; the one-period cost of action k from state i is $z^{t+1}(n)\hat{p}^t(i, n:k)$. And the transition probabilities are the same as in the original MDP. Denote the updated values by $v^{t+1} = \{v^{t+1}(n)\}$. Let $\pi_n(i, k)$ be the dual variables for each of the N' MDPs, where the subscript n denotes that i is an index in s_n.

Step (iv) Update the dual variables

$$u^{t+1}(i, k) = \left\{\frac{u^t(i, k)\lambda^{t+1}(n, \ell)}{\sum\limits_{k\varepsilon A_\ell} \sum\limits_{i\varepsilon s_n} u^t(i, k)} - \left(c(i, k) + \beta \sum_{j=1}^{N} p(i, j:k)v^{t+1}(j) - v^{t+1}(i)\right)\right\}^+ \qquad i\varepsilon s_n;\ k\varepsilon A_\ell \tag{3.2}$$

where $\{a\}^+ = \max(a, 0)$.

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^*.

An alternative to step (iii) that reduces the computational effort is to use the fixed-weight disaggregate value [10, 11]:

$$v^{t+1}(i) = \frac{v^t(i)z^{t+1}(n)}{\sum\limits_{j\varepsilon s_n} v^t(j)} \qquad i\varepsilon s_n \tag{3.3}$$

Similarly, step (iv) can be replaced by a fixed-weight disaggregate:

$$u^{t+1}(i, k) = \frac{u^t(i, k)\lambda^{t+1}(n, \ell)}{\sum_{k\varepsilon A_\ell} \sum_{i\varepsilon s_n} u^t(i, k)} \quad i\varepsilon s_n;\ k\varepsilon A_\ell \tag{3.4}$$

or, by noting that the $\pi_n^{t+1}(i, k)$ in step (iii) are disjoint, let

$$u^{t+1}(i, k) = \pi_n^{t+1}(i, k)\ . \tag{3.5}$$

Before proving properties of the iterative aggregation scheme, it is necessary to prove that step (iii) is equivalent to optimal disaggregation as in [12, Chapter 7].

Lemma 3.1 In the LP:

$$\text{Maximize} \sum_{i\varepsilon s_n} v(i)$$

$$\text{s.t.} \quad \sum_{j\varepsilon s_n} \left(\delta_{ij} - \beta p(i, j:k)\right) v(j) \leq z^{t+1}(n)\hat{p}^t(i, n;k) \quad i\varepsilon s_n,\ k = 1, \ldots, K \tag{3.6a}$$

$$\sum_{j\varepsilon s_n} \left(-\beta p(i, j:k)\right) v(j) \leq z^{t+1}(n)\hat{p}^t(i, n:k) \quad i\not\varepsilon s_n,\ k = 1, \ldots, K \tag{3.6b}$$

$$v(i) \geq 0 \quad i\varepsilon s_n \tag{3.6c}$$

at an optimal solution, none of the constraints (3.6b) are binding, i.e., $\pi_n^{t+1}(i, k) = 0$ for $i\not\varepsilon s_n$.

Proof. Let w be the cardinality of s_n. Then there can be at most w constraints at equality at an optimal solution to (3.6). Suppose one of the rows in (3.6b) is at equality at an optimal solution. This implies there exist some state j' such that no action has been found optimal for it, i.e., $\pi_n^{t+1}(j', k) = 0$ for k = 1, ..., K. This implies in the dual problem to (3.6) that for the dual row associated with $j' \varepsilon s_n$

$$\sum_{\substack{i=1 \\ i \neq j'}}^{N} \sum_{k=1}^{K} - \beta p(i, j':k)\pi_n^{t+1}(i, k) = 1 \tag{3.7}$$

However, $p(i, j':k) \geq 0$ for all j, k, and $\pi_n^{t+1}(i, k) \geq 0$. The only solution satisfy (3.7) is the trivial one $\pi_n^{t+1} \equiv 0$.

□

The importance of lemma 3.1 is that the objective Maximize $\sum_{i \varepsilon s_n} v(i)$ subject to (3.6a) is itself an LP for solving an MDP. Hence more efficient iterative techniques rather than linear programming can be used to find the optimal disaggregate values.

The existence of an optimal solution to (2.1) 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 (2.1).

Theorem 3.1 (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 (2.1).

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 $z^{t+1}(n) = \sum_{i\varepsilon s_n} \bar{v}(i)$, and $\lambda^{t+1}(n, \ell) = \sum_{i\varepsilon s_n} \sum_{k\varepsilon A_\ell} \bar{u}(i, k)$. These values are optimal in the master problem if they are feasible and if they satisfy complementarity conditions. The latter is

$$\sum\left\{\left(\sum_{n=1}^{N}\left(\sum_{i\varepsilon s_n} \bar{v}(i)\right) \sum_{j\varepsilon s_n}\left(\delta_{ij} - \beta p(i, j:k)\right) \frac{\bar{v}(i)}{\sum_{i\varepsilon s_n} \bar{v}(i)} - c(i, k)\right)\left(\frac{\bar{u}(i, k)}{\sum_{i\varepsilon s_n} \sum_{k\varepsilon A_\ell} \bar{u}(i, k)}\right)\right\}$$

$$\left(\sum_{i\varepsilon s_n} \sum_{k\varepsilon A_\ell} \bar{u}(i, k)\right) = 0$$

After cancelling out terms, this reduces to:

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

the complementarity condition for (2.1), 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 $z^{t+1}(n) = \sum_{i\varepsilon s_n} \bar{v}(i)$ and $\lambda^{t+1}(n, \ell) = \sum_{i\varepsilon s_n} \sum_{k\varepsilon A_\ell} \bar{u}(i, k)$ are primal and dual feasible in the master problem.

Using the LP form of step (iii) given in (3.6), 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 (3.2) 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

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

(v^*, u^*) is optimal in (2.1)

The proof is to show that if $v^t = v^*$, $u^t = u^*$, then $z^{t+1}(n) = \sum_{i \varepsilon s_n} v(i)$ and $\lambda^{t+1}(n, \ell) = \sum_{i \varepsilon s_n} \sum_{k \varepsilon A_\ell} \bar{u}(i, k)$. Suppose not. Since v^*, u^* is a fixed point, then from (3.12), v^* is feasible in (2.1), or else $u^{t+1} \neq u^*$. However, v^* is optimal in step (iii), or equivalently in the N' LPs (3.6). This implies v^* is feasible in (2.1) and optimal, contradicting the assumption that $v^* \neq \bar{v}$. Since $v^* = \bar{v}$, if $u^* \neq \bar{u}$, then it is straightforward to show that the master problem and (3.2) assure that u^* and $\bar{v}$ satisfy the complementarity conditions for (2.1). Hence u^* must be $\bar{u}$.

□

That $(\bar{v}, \bar{u})$ is a fixed point is suggested in [10, 11]. There it is shown that optimal weightings should be proportional to optimal solutions. The main result of this paper is that under very weak conditions the iterative process described in steps (i)-(iv) coverges to an optimum solution of (2.1). The algorithm in steps (i)-(iv) can be seen to be a composite point to set map $A : R_+^N \times R_+^{N\times K} \to R_+^N \times R_+^{N\times K}$.

Theorem 3.2 Assume for $(v^0, u^0) \geq 0$ that v', u' 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. (i) If the algorithm converges, it converges to a fixed point

Denote the assumed limit by (v', u'). Since the sequence $\{(v^t, u^t)\}$ converges on a metric space, there exists a metric d and a t^* such that $d\left((v^t, u^t), (v', u')\right) < \varepsilon$, for every $\varepsilon > 0$. Also, for every $\varepsilon > 0$ there exists a $\bar{t}$ such that for $t' \geq \bar{t}$, $t'' \geq \bar{t}$, $d\left((v^{t'}, u^{t'}), (v^{t''}, u^{t''})\right) < \varepsilon$. Combining inequalities says that $d\left((v^t, u^t), (v^{t+1}, u^{t+1})\right) \to 0$ as t approaches infinity, hence (v', u') must be a fixed point.

(ii). The point to set map A is upper-semicontinuous (closed)

Since the partitions being used are the same each iteration, there exists three constant matrices $T^1\left((N'K') \times 1)\right)$, $T^2(1 \times N')$, 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 forms a convex set, the mapping A^1 is closed. Zangwill [9] proves that the maximization operator is closed, call this A^2. Similarly, the constraints for the subproblems form closed maps, call then A^3, A^4, ..., $A^{N'+2}$, and equation (3.2) is trivially closed, call this map A^0. Therefore the map A can be written as $A = A^0A^2A^{N'+2} \ldots A^2A^3A^2A^1$. 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 [9, p. 96].

(iii) The penalty function

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

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 [5]. Both are developed around the penalty function method for solving LPs.

Bertsekas [2] 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 (3.2), the Lagrangean function for the master problem (3.1) at iteration t+1 can be written as

$$\underset{z \geq 0}{\text{Maximize}} \sum_{n=1}^{N'} z(n)$$

$$- \left[\sum_{n=1}^{N'} \sum_{\ell=1}^{K'} w(n, \ell) \right] \left\{ \sum_{k \varepsilon A_\ell} \sum_{i \varepsilon s_n} \left(u^t(i, k) + \sum_{j=1}^{N} \left(\delta_{ij} - \beta p(i, j:k) \right) v^t(j) - c(i, k) \right)^+ \right.$$

$$\left. \left(\sum_{n=1}^{N'} z(n) \sum_{j \varepsilon s_n} \left(\delta_{ij} - \beta p(i, j:k) \right) \frac{v^t(j)}{\sum_{j \varepsilon s_n} v^t(j)} - c(i, k) \right) \right\} \tag{3.8}$$

$\left(u^t(i, k) + \sum_{j=1}^{N} \left(\delta_{ij} - \beta p(i, j:k) \right) v^t(j) - c(i, k) \right)^+$ is used in (3.8) because the total normalizing constant can be cancelled out on both sides of each constraint of the aggregate problem (3.1).

At $w(n, \ell) \equiv \frac{1}{e}$, and at the trial value $z(n) = \sum_{i \varepsilon s_n} v^t(i)$, (3.8) reduces to:

$$\sum_{i=1}^{N} v^t(i) - \frac{1}{e} \sum_{n=1}^{N'} \sum_{\ell=1}^{K'} \left\{ \sum_{k \varepsilon A_\ell} \sum_{i \varepsilon s_n} \left(u^t(i, k) + \sum_{j=1}^{N} \left(\delta_{ij} - \beta p(i, j:k) \right) v^t(j) - c(i, k) \right)^+ \right.$$

$$\left. \left(\sum_{j=1}^{N} \left(\delta_{ij} - \beta p(i, j:k) \right) v^t(j) - c(i, k) \right) \right\} \tag{3.9}$$

Let I^- be the set of (i, k) such that $\left(u^t(i, k) + \sum_{j=1}^{N}\left(\delta_{ij} - \beta p(i, j:k)\right)v^t(j) - c(i, k)\right) \leq 0$. Then (3.9) can be written as:

$$\sum_{i=1}^{N} v^t(i) - \frac{1}{e} \sum_{n=1}^{N'} \sum_{\ell=1}^{K'} \left\{ \sum_{k \varepsilon A_\ell} \sum_{i \varepsilon s_n} u^t(i, k)\left(\sum_{j=1}^{N}\left(\delta_{ij} - \beta p(i, j:k)\right)v^t(j) - c(i, k)\right)\right\}$$

$$- \frac{1}{e} \sum_{(i, k)\varepsilon I^+} \left(\sum_{j=1}^{N}\left(\delta_{ij} - \beta p(i, j:k)\right)v^t(j) - c(i, k)\right)^2$$

$$+ \frac{1}{e} \sum_{(i, k)\varepsilon I^-} u^t(i, k)\left(\sum_{j=1}^{N}\left(\delta_{ij} - \beta p(i, j:k)\right)v^t(j) - c(i, k)\right) \qquad (3.10)$$

Since $u^t(i, k) = \lambda^t(n, \ell) \dfrac{u^{t-1}(i, k)}{\sum_{\ell \varepsilon A_\ell} \sum_{i \varepsilon s_n} u^{t-1}(i, k)}$; $v^t(i) = z^t(n) \dfrac{v^{t-1}(i)}{\sum_{i \varepsilon s_n} v^{t-1}(i)}$

and z^t, λ^t are optimal in (3.1) at iteration t, the second term in (3.10) is identically zero. Moreover, the last term in (3.10) is nonpositive since $(i, k)\varepsilon I^-$ if v^t is feasible in that row. Hence, (3.10) is less than or equal to:

$$\sum_{i=1}^{N} v^t(i) - \frac{1}{e} \sum_{i=1}^{N} \sum_{k=1}^{K} \left[\left(\sum_{j=1}^{N}\left(\delta_{ij} - \beta p(i, j:k)\right)v^t(j) - c(i, k)\right)^+\right]^2 . \qquad (3.11)$$

Moreover, z^t, λ^t maximizes the penalty function (3.11) given the constraints in (3.1) at iteration t. Writing the Lagrangean as L(z, w), this implies that for $w \equiv \frac{1}{e}$, $\max_z L(z, w)$ occurs at $z(n) = \sum_{i \varepsilon s_n} v^t(i)$. Let $L^*(w) = \max_z L(z, w)$.

Then at an optimum to (3.1) at iteration t+1, $L(z^{t+1}, \lambda^{t+1}) \leq L^*(w)$. From (3.11), this implies the penalty function is decreasing with t.

(iv) Convergence

Parts (i)-(iii) of the proof, combined with theorem 3.1, are the conditions required in Convergence Theorem A[9, p. 91].

□

When steps (i)-(iv) of the iterative aggregation process are used only every k^{th} iteration, and the alternatives (3.3), (3.4), or (3.5) are used at all other iterations, proofs of convergence follow closely the proof of theorem 3.2 to derive the conditions necessary for Zangwill's other convergence theorems. Intuitively, one full step of successive approximations still is computed infinitely often if the algorithm does not find a fixed point. 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 converges when applied to MDPs.

Finally, it is conjectured that convergence can be proven in an analogous mammer to theorem 3.2 if only some subset of the dual variables are adjusted by equation (3.2) at each iteration. (This is in the spirit of recent advances in LP algorithms that converge in polynomial time.) For example, one partition at a time could be updated each iteration using equation (3.2).

4. 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 then 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 at each iteration, the dual variables, a method which converges globally. Improvements in this area should lead to truly efficient means for solving large scale MDPs.

ACKNOWLEDGMENT

This paper was completed while I was a Visiting Research Associate at the Center for Coastal Marine Studies, University of California at Santa Cruz. I am grateful to the CCMS for its support during this period.

REFERENCES

1. G. V. Agafanov and A. S. Makarova, "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), p. 132-148 (1976). (Translated from Russian by Wilvan G. Van Campen for the Southwest Fisheries Center Honolulu Laboratory, National Marine Fisheries Service, NOAA, Honolulu, HI 96812, 1980, 18 p. Translation No. 43.) [Limited distribution.]

2. D. P. Bertsekas, "Necessary and Sufficient Conditions for a Penalty Method to be Exact," Mathematical Programming 9, 87-99 (1975).

3. G. d'Epenoux, "A Probabilistic Production and Inventory Problem," Management Science 10, 98-108 (1963).

4. L. M. Dudkin (editor), Iterative Aggregation and Its Application in Planning, Moscow, 1979, 328 p.

5. O. L. Mangasarian, "Iterative Solution of Linear Programs," Computer Sciences Technical Report No. 327, University of Wisconsin, Madison, 1979, 20 p.

6. I.Ya. Vakhutinskii and L. M. Dudkin, "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. Novosibirsk. Seriia Obshchestvendykh Nauk) 3(11), 67-71 (1973). (Translated from Russian by Wilvan G. Van Campen for the Southwest Fisheries Center Honolulu Laboratory, National Marine Fisheries Service, NOAA, Honolulu, HI 96812, 1980, 8 p. Translation No. 41.) [Limited distribution.]

7. I. Ya. Vakhutinskii, L. M. Dudkin, and A. Makarov, "An Iterative Aggregation Algorithm for Connecting a System of Branch Planning Models (with Energetics Example)," Automation and Remote Control 10, 145-159 (1973).

8. I. Ya. Vakhutinskii, L. M. Dudkin, and A. A. Ryvkin, "Iterative Aggregation - A New Approach to the Solution of Large-Scale Problems," Econometrica 47, 821-841 (1979).

9. W. Zangwill, Nonlinear Programming - A Unified Approach, Prentice-Hall, N.Y., 1969, 395 p.

10. P. Zipkin, "Bounds on the Effect of Aggregating Variables in Linear Programs," To appear, Operations Research.

11. P. Zipkin, "Bounds for Row-Aggregation in Linear Programming," To appear, Operations Research.

12. P. Zipkin, Aggregation in Linear Programming, PhD. Dissertation, Yale University, 1977, 189 p.