PIECEWISE LINEAR MARKOV DECISION PROCESSES WITH AN APPLICATION TO PARTIALLY OBSERVABLE MARKOV MODELS

by

KATSUSHIGE SAWAKI

Bachelor of Economics, Nanzan University, 1968
Master of Economics, Nanzan University, 1970

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

THE FACULTY OF GRADUATE STUDIES

(Faculty of Commerce and Business Administration)

We accept this thesis as conforming

to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

June, 1977

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the Head of my Department or by his representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of Commerce

The University of British Columbia
2075 Wesbrook Place
Vancouver, Canada
V6T 1W5

Date July 21st 1977

Research Supervisor: Professor Shelby L. Brumelle.

ABSTRACT

This dissertation applies policy improvement and successive approximation or value iteration to a general class of Markov decision processes with discounted costs. In particular, a class of Markov decision processes, called piecewise-linear, is studied. Piecewise-linear processes are characterized by the property that the value function of a process observed for one period and then terminated is piecewise-linear if the terminal reward function is piecewise-linear. Partially observable Markov decision processes have this property.

It is shown that there are ε-optimal piecewise-linear value functions and piecewise-constant policies which are simple. Simple means that there are only finitely many pieces, each of which is defined on a convex polyhedral set. Algorithms based on policy improvement and successive approximation are developed to compute simple approximations to an optimal policy and the optimal value function.

TABLE OF CONTENTS

ACKNOWLEDGEMENT

I wish to express my sincere gratitude to Professor Shelby Brumelle, Chairman of my thesis committee, for his invaluable guidance and constant encouragement. This work would not have been possible without his guidance and I am deeply grateful.

I would also like to thank Professors U. Haussmann and M.L. Puterman for their many helpful comments and suggestions.

I also wish to extend my gratitude to Professor L.G. Mitten for his suggestions and encouragement and to Professor I. Vertinsky for his continuous encouragement and financial support during my doctoral programme. I also wish to thank Theresa Fong for her excellent typing.

Special thanks go to my wife, Emiko, for her understanding and patience.

Chapter I

INTRODUCTION

The combined theories of dynamic programming and Markov decision processes have been applied to many areas including inventory, queuing, and machine maintenance problems.

This thesis develops a theory for a general class of dynamic programming models as well as algorithms which yield policies that are both "simple" and ε-optimal. The approach taken is to consider a dynamic programming problem for an arbitrary state Markov decision processes over an infinite horizon. At present there are no computational algorithms for models in which the state space is a continuum. However, algorithms for some partially observable Markov models and finite (at most countable) state Markov decision processes have been developed. The formulation of our general model is motivated by consideration of the special structure which the partially observable models possess.

The partially observable Markov process, introduced by Dynkin [17], consists of two stochastic processes, the core process $\{Z_n, n = 1,2,\ldots\}$, which cannot directly be observed, and the signal process $\{S_n, n = 1,2,\ldots\}$ which becomes known at each decision epoch $n = 1,2,\ldots$. The core process is a Markov chain and the signal process is probabilistically related to the core process by the conditional probability $\gamma_{i\theta}$ of observing a signal θ given that the core process is in state i. Dynkin shows that the state occupancy probability represents a sufficient statistic for the complete past history. Astrom [3] also considered a similar model with finite states and finite actions over a finite horizon, using the method of successive approximation to find ε-optimal cost vectors, however, it is only applicable to problems in two dimensions. Smallwood and Sondik [60] have independently obtained similar results. Later, Sondik [61] extended this model to the infinite horizon and introduced the class of finitely transient policies. The cost functions of these policies which are used to approximate the cost functions of arbitrary stationary policies are piecewise linear with respect to the sufficient statistic. White [67] has considered a partially observable semi-Markov process with a finite horizon where the controller knows the times of the core process transition. Aoki [1] also studies the partially observable control problem with finite states, finite action sets and finite horizon, but does not include an operational algorithm.

Since in partially observable models with finite state space the states of dynamic programming are probability vectors in R^N (the N-dimensional real space), it follows after some modification that if the state space (complete separable metric space) of our model is replaced by R^N, the model then immediately is reduced to a partially observable Markov decision model. The state space of the system will later be assumed to be a non-empty bounded subset Ω of a separable complete metric space X so as to ensure that this thesis includes partially observable Markov processes as a special case.

In this thesis the concepts of simple partitions, simple policies and piecewise linear functionals on the arbitrary state space are introduced to establish an algorithm for determining an "ε-optimal simple stationary policy". The idea is based on the "linearity" of partially observable Markov processes. In addition to these three concepts, assumptions on the immediate costs and on the contraction operators U_a are introduced. Two algorithms are discussed. The first of these is the method of successive approximation which is used for approximating the optimal cost V^* and for finding policies whose cost functions approximate V^*. The second is based on the method of policy improvement.

In Chapter II a formulation of a dynamic programming problem with an abstract state space and finite action space will be considered. Chapter III is a study of operators used in the algorithms. In Chapter IV the methods of successive approximation and of policy improvement will be studied. Chapter V explicitly develops the algorithms for the two methods in a more concrete setting.

Chapter II

MODEL FORMULATION AND ASSUMPTIONS

In this chapter we shall formulate an optimal control problem with discounted costs and with complete observation over an infinite horizon under the setting of Blackwell [7]. Also, we introduce some definitions and assumptions. A Markov decision process is specified by the following four objects:

(i) the state space Ω is a non-empty Borel subset of a separable Banach space X;

(ii) the action set A is finite and a is an element of A;

(iii) for each pair $(x,a) \in \Omega \times A$, $q(\cdot|x,a)$ is the one step transition probability of the system on the Borel subsets of Ω;

(iv) the immediate cost $c(x,a)$ is a bounded Borel measurable function on $\Omega \times A$.

When the system is in state x and action a is chosen, then we incur a cost $c(x,a)$. We define a policy to be a sequence $\{\delta_n, n = 1,2,...\}$, where δ_n tells us what action to choose at the n-th period as a Borel measurable function of the history $H = (x_1,a_1,...,x_n)$ of the system up to period n. Let Δ be a family of policies . A policy $\delta = (\delta,\delta,...)$ which is independent of time n is called stationary. Our expected discounted total cost $V^{\delta}(x)$ at an initial state x under a policy δ is written as

(II.1) $$V^{\delta}(x) = E\{\sum_{n=1}^{\infty} \beta^{n-1} c(X_n, \delta_n(X_n)) | X_1 = x\}$$

where $\{X_n: n = 1,2,\ldots\}$ is a Markov chain with probability transition function $q(\cdot|x, \delta_n(x))$. The discount factor is denoted by β and $0 \leq \beta < 1$. The function V^{δ} is called the <u>cost</u> of policy δ.

Define the <u>optimal cost</u> function V^* by

(II.2) $$V^*(x) = \inf_{\delta \varepsilon \Delta} V^{\delta}(x) \quad \text{for all} \quad x \,\varepsilon\, \Omega.$$

Then, the following is true (see Blackwell [7]).

<u>Theorem II.1</u>. There exists an optimal stationary policy δ^* with $V^{\delta^*} = V^*$. Also, V^* satisfies

(II.3) $$V^*(x) = \min_{a \varepsilon A}\{c(x,a) + \beta\int_{\Omega} V^*(x')q(dx'|x,a)\} \text{ for all } x \,\varepsilon\, \Omega.$$

An <u>ε-optimal cost function</u> V is one satisfying

(II.4) $$\|V^* - V\| = \sup_{x \varepsilon \Omega}|V^*(x) - V(x)| < \varepsilon.$$

A policy δ such that $V = V^{\delta}$ satisfying (II.4) is an <u>ε-optimal policy</u>. Let $B(\Omega)$ be the set of all bounded Borel measurable functions on Ω with the sup norm $\|\cdot\|$ as above. Then, $B(\Omega)$ is a Banach space (see Lusternik and Sobolev [38, p. 18]).

For finding an ε-optimal policy and its cost function we define simple partitions, simple policies and piecewise (abbreviated, hereafter, by p.w.) linear functions.

Definition II.1. A partition $\{E_i\}_{i=1}^m$ of $\Omega \subset X$ is called simple if each E_i is a convex polyhedral set, where a convex polyhedral set is the solution set of a finite system of linear inequalities, i.e.,

$$E_i = \{x \in \Omega: \ell_{ij}(x) < (\text{or } \leq) d_j,\ j = 1,2,\ldots,n_i\},$$

$$i = 1,2,\ldots,m,$$

where each ℓ_{ij} defined on X is a linear functional and d_j is a real number. Note that we always take linear functional to be bounded.

Examples.

(1) Let $E_1^1 = \Omega$. Take any linear functional ℓ_1 on X and a real number d_1. Then $E^2 = \{E_1^2, E_2^2\}$ is a simple partition where $E_1^2 = \{x \in \Omega: \ell_1(x) < d_1\}$ and $E_2^2 = \{x \in \Omega: \ell_1(x) \geq d_1\}$. Furthermore, take another linear functional $\ell_2 \neq \ell_1$ and a real number $d_2 \neq d_1$. Then, $E^3 = \{E_1^3, E_2^3, E_3^3, E_4^3\}$ is a simple partition where $E_1^3 = \{x \in \Omega: \ell_1(x) < d_1, \ell_2(x) < d_2\}$, $E_2^3 = \{x \in \Omega: \ell_1(x) < d_1, \Omega_\ell(x) \geq d_2\}$, $E_3^3 = \{x \in \Omega: \ell_1(x) \geq d_1, \ell_2(x) < d_2\}$ and $E_4^3 = \{x \in \Omega: \ell_1(x) \geq d_1,$

$\ell_2(x) \geq d_2\}$, and so on.

(2) Let $\Omega = R^N$ (the N-dimensional real space). In definition II.1, let $\ell_{ij}(x) = \ell_{ij}x$ where $\ell_{ij} \varepsilon R^N$ and $\ell_{ij}x$ is the inner product of ℓ_{ij} and x. Then $\{E_i\}$ is a simple partition in R^N.

Lemma II.1. Let $P_1 = \{E_i\}$ and $P_2 = \{F_j\}$ be two simple partitions of Ω. Then, the product partition $P_1 \cdot P_2 = \{E_i \cap F_j\}$ is again simple.

Proof: Here we omit $E_i \cap F_j$ if $E_i \cap F_j = \phi$. The sets $E_i \cap F_j$ are disjoint and are convex polyhedral sets. Hence $P_1 \cdot P_2$ is simple.

Definition II.2. A stationary policy δ is simple with respect to a simple partition $\{E_i;\ i = 1,2,\ldots,m\}$ if $\delta(x) = a_i$ for all $x \varepsilon E_i$, $i = 1,2,\ldots,m$.

Definition II.3. A vector valued function V on $\Omega \subset X$ is called p.w. linear if there exists a simple partition $\{E_i\}$ of Ω such that $V(x) = V_i(x)$ for all $x \varepsilon E_i$, $i = 1,2,\ldots,m$, and each V_i is the restriction to E_i of a linear function on X.

Given a policy δ, define the operator U_δ from $B(\Omega)$ into $B(\Omega)$ by

$$(U_\delta V)(x) = c(x,\delta(x)) + \beta\int_\Omega V(x')q(dx'|x,\delta(x)). \tag{II.5}$$

If $\delta(x) = a$ for each $x \in \Omega$, then we write $U_a = U_\delta$.

Define the operator U_* from $B(\Omega)$ into $B(\Omega)$ by

$$(U_*V)(x) = \min_a (U_aV)(x) \quad \text{for } V \in B(\Omega). \tag{II.6}$$

Although V^* is not necessarily p.w. linear and δ^* is not necessarily simple, we will show for a class of Markov decision processes having the structure described in the following assumption that there are ε-optimal cost functions and simple policies.

<u>Assumption I (A.I.)</u>. $(U_aV)(x)$ is p.w. linear on Ω for each a, provided that V is p.w. linear on Ω.

<u>Examples</u>.

<u>Model 1</u>. Let $X = R^N$ and Ω be a convex polyhedral set in R^N such that $q(\cdot|x,a)$ is a probability measure on Ω for each $(x,a) \in \Omega \times A$. The following two assumptions (A.II) and (A.III) imply (A.I).

<u>Assumption II (A.II.)</u>. For each $a \in A$, the immediate cost function $c(\cdot,a)$ is the restriction to Ω of a linear functional on X. Hence for each a, there is a vector c^a such that

$$c(x,a) = c^a \cdot x \quad \text{for } x \in \Omega.$$

<u>Assumption III (A.III.)</u>. For each convex polyhedral set $B \subset \Omega$ and each action $a \in A$,

$$q^a(B,x) = \int_B x' q(dx'|x,a)$$

is p.w. linear in x with respect to a simple partition $\mathcal{P}^a(B) = \{E_j(a,B),\ j = 1,2,\ldots,m_{a,B}\}$.

We will show in Model 2 that partially observable Markov processes are a special case of Model 1.

We next check that (A.I.) is satisfied. Let $a \in A$ be arbitrary but fixed and suppose that V is p.w. linear with respect to a simple partition $\{E_i,\ i = 1,2,\ldots,m\}$. Let $\mathcal{P}^a = \prod_{i=1}^{m} \mathcal{P}^a(E_i) = \{\tilde{E}_j^a;\ j = 1,2,\ldots,r\}$, the product partition, which is again simple from Lemma II.1.

$$\begin{aligned}(U_a V)(x) &= c^a \cdot x + \beta \int_\Omega V(x') q(dx'|x,a) \\ &= c^a \cdot x + \beta \sum_{i=1}^{m} \int_{E_i} V_i x' q(dx'|x,a) \\ &= c^a \cdot x + \beta \sum_{i=1}^{m} V_i q^a(E_i,x) \\ &= [c^a + \beta \sum_{i=1}^{m} V_i \lambda_{i\ell}^a]\, x \quad \text{for } x \in \tilde{E}_j^a\end{aligned}$$

where $\lambda_{i\ell}^a \cdot x = q^a(E_i,x)$ for $x \in E_\ell(a,E_i)$ and the index ℓ depends on i for each $a \in A$. $U_a V$ is linear on each $\tilde{E}_j^a$. Hence $U_a V$ is p.w. linear with respect to the simple partition $\mathcal{P}^a = \{\tilde{E}_j^a,\ j = 1,2,\ldots,r\}$, which satisfies (A.I.). This model 1 is really the basic model studied in the theory.

Model 2. A partially observable Markov Decision Process (Dynkin [17], Smallwood and Sondik [60]).

Consider a Markov decision process (called the core process) with state space $\{1,2,\ldots,N\}$, with action set A, with probability transition matrices $\{P^a, a \in A\}$, and with immediate cost vectors $\{h^a, a \in A\}$. Let Z_n be the state at the n-th transition. Assume that the process $\{Z_n, n = 0,1,2,\ldots\}$ cannot be observed, but at each transition a signal is transmitted to to the decision maker. The set of possible signals Ⓗ is assumed to be finite. For each n, given that $Z_n = j$ and that action a is to be implemented, the signal θ_n is independent of the history of the signals and actions $\{\theta_0, a_0, {}_1, a_1, \ldots, \theta_{n-1}, a_{n-1}\}$ prior to the n-th transition and has conditional probability denoted by $\gamma^a_{j\theta} = P[\theta_n = \theta | Z_n = j, a]$.

Let $X = R^N$ and $\Omega = \{x = (x_1, x_2, \ldots, x_N) : \sum_{i=1}^{N} x_i = 1, x_i \geq 0, \forall i\}$. Define the i-th component of X_n to be

$$P[Z_n = i | \theta_0, a_0, \theta_1, \ldots, \theta_{n-1}, a_{n-1}, \theta_n], \quad i = 1,2,\ldots,N.$$

It can be shown (see Dynkin [17]) that

$$P[Z_{n+1} = j | \theta_0, a_0, \theta_1, \ldots, \theta_n, a_n, \theta_{n+1}] = P[Z_{n+1} = j | \theta_{n+1}, a_n, X_n].$$

Thus X_n represents a sufficient statistic for the complete past history $\{\theta_0, a_0, \ldots, a_{n-1}, \theta_n\}$. It follows that $\{X_n : n = 0,1,2,\ldots\}$ is a Markov decision process (see Sondik [61]), called the observed process. Its immediate cost is $c(x,a) = h^a \cdot x$. Its action set is A. Its probability transition function is determined by the following calculation. For each measurable subset $B \subseteq \Omega$, $x \in \Omega$, and $a \in A$,

$$\begin{aligned}
q(B|x,a) &= P[X_{n+1} \in B | X_n = x,\ a_n = a] \\
&= \sum_\theta P[X_{n+1} \in B | \theta_{n+1} = \theta,\ X_n = x,\ a_n = a] \cdot P[\theta_{n+1} = \theta | X_n \\
&\qquad = x,\ a_n = a] \\
&= \sum_\theta P[X_{n+1} \in B | \theta_{n+1} = \theta,\ X_n = x,\ a_n = a] \cdot \sum_j P[\theta_{n+1} = \\
&\qquad \theta | Z_{n+1} = j,\ X_n = x, a] \cdot P[Z_{n+1} = j | X_n = x,\ a_n = a] \\
&= \sum_\theta P[X_{n+1} \in B | \theta_{n+1} = \theta,\ X_n = x,\ a_n = a] \cdot \sum_j \gamma^a_{j\theta} \sum_i P[Z_{n+1} = \\
&\qquad j | Z_n = i,\ X_n = x,\ a_n = a] P[Z_n = i | X_n = x,\ a_n = a] \\
&= \sum_\theta P[X_{n+1} \in B | \theta_{n+1} = \theta,\ X_n = x,\ a_n = a] \sum_j \gamma^a_{j\theta} \sum_i P^a_{ij} x_i \\
&= \sum_\theta P[X_{n+1} \in B | \theta_{n+1} = \theta,\ X_n = x,\ a_n = a] \underline{1} P^a(\theta) x
\end{aligned}$$

where $\underline{1} = (1,1,\ldots,1)$ and $P^a(\theta) = [P^a_{ij}\ \gamma^a_{j\theta}]^T$. Define the vector $T(x|\theta,a)$ by

$$T(x|\theta,a) = \frac{P^a(\theta)x}{\underline{1}P^a(\theta)x} .$$

Note that $T(X_n|\theta,a) = X_{n+1}$, and that

$$P[X_{n+1} \in B|\theta_{n+1} = \theta,\ X_n = x,\ a] = \begin{cases} 1 & \text{if } T(x|\theta,a) \in B \\ 0 & \text{if } T(x|\theta,a) \notin B \end{cases}$$

(See Sondik [61]). So,

$$q(B|x,a) = \sum_{\theta\varepsilon\Phi^a(B,x)} \underline{1}\ P^a(\theta)x$$

where $\Phi^a(B,x) = \{\theta: T(x|\theta,a) \in B\}$.

Finally, we show that the observed process $\{X_n\}$ is a special case of Model 1; i.e., $q^a(B,x) = \int_B x'q(dx'|x,a)$ is p.w. linear in x for each convex polyhedral set $B \subseteq \Omega$ and action $a \in A$. Using the previously computed $q(B|x,a)$ we have

$$\begin{aligned} q^a(B,x) &= \int_B x'q(dx'|x,a) \\ &= \sum_{\theta\varepsilon\Phi^a(B,x)} T[x|\theta,a]\ \underline{1}\ P^a(\theta)x \\ &= \sum_{\theta\varepsilon\Phi^a(B,x)} \frac{P^a(\theta)x}{\underline{1}P^a(\theta)x}\ \underline{1}\ P^a(\theta)x \\ &= \sum_{\theta\varepsilon\Phi^a(B,x)} P^a(\theta)x. \end{aligned}$$

Thus it is sufficient to verify that the set valued function $\Phi^a(B,\cdot): \Omega \to 2^{\text{(H)}}$ is p.w. constant on Ω where $2^{\text{(H)}}$ is the power set of (H). To do this we need the following lemma.

Lemma II.2. For each signal θ, action a, and set $B \subseteq \Omega$, define

$$E_\theta^{B,a} = \{x \in \Omega: T(x|\theta,a) \in B\}.$$

Then for any subset of signals $\psi \subseteq$ (H), we have

$$\Phi^a(B,x) = \psi \text{ if and only if } x \in \bigcap_{\theta\in\psi} E_\theta^{B,a} \cap \bigcap_{\theta\in\psi^c} (E_\theta^{B,a})^c.$$

Proof. Note that $E_\theta^{B,a} = \{x: \theta \in \Phi^a(B,x)\}$. Thus if $x \in E_\theta^{B,a}$ for $\theta \in \psi$, then $\theta \in \Phi^a(B,x)$. Consequently, $\psi \subseteq \Phi^a(B,x)$. On the other hand, if $x \in (E_\theta^{B,a})^c$ for $\theta \in \psi^c$, then $\theta \notin \Phi^a(B,x)$. Consequently, $\psi^c \subseteq (\Phi^a(B,x))^c$. It follows that $\psi = \Phi^a(B,x)$.

Conversely, suppose that $\Phi^a(B,\hat{x}) = \psi$. Then $\hat{x} \in E_\theta^{B,a}$ for each $\theta \in \psi$ and $\hat{x} \in (E_\theta^{B,a})^c$ for each $\theta \in \psi^c$, which completes the proof.

Let $E_B^a(\psi) = \{x: \Phi^a(B,x) = \psi\}$. The above lemma gives an explicit representation of $E_B^a(\psi)$ and $q^a(B,x)$ is p.w. linear with respect to the partition $\{E_B^a(\psi): \psi \in 2^{\text{(H)}}\}$ where it is assumed that $q^a(B,x) = 0$ if $E_B^a(\psi) = \phi$ (empty) for all ψ. Although this partition is not simple, it can easily be refined to a simple partition as in the next paragraph.

Suppose that $B \subseteq \Omega$ is a convex polyhedral set. Since for $x \in \Omega = \{x: \sum x_i = 1,\ x_i \geq 0\ \forall i\}$ an inequality $\ell x \leq b$ can be rewritten as $\ell x - b = (\ell - b\underline{1})x \leq 0$, we can without loss of generality assume that B has the representation

$$B = \{x \in \Omega : Kx < \underline{0},\ Lx \leq \underline{0}\}$$

for some matrices K and L where $\underline{0} = (0,0,\ldots,0)^T$. With this representation of B,

$$\begin{aligned} E_\theta^{B,a} &= \{x \in \Omega\colon T(x|\theta,a) \in B\} \\ &= \{x \in \Omega\colon K\frac{P^a(\theta)x}{\underline{1}P^a(\theta)x} < \underline{0},\ L\frac{P^a(\theta)x}{\underline{1}P^a(\theta)x} \leq \underline{0}\} \\ &= \{x \in \Omega\colon KP^a(\theta)x < \underline{0},\ LP^a(\theta)x \leq \underline{0}\} \\ &= \{x \in \Omega\colon K^a(\theta)x < \underline{0},\ L^a(\theta)x \leq \underline{0}\} \end{aligned}$$

where $K^a(\theta) = KP^a(\theta)$ and $L^a(\theta) = LP^a(\theta)$. So each $E_\theta^{B,a}$ is a convex polyhedral set. Each $(E_\theta^{B,a})^c$ can be represented as a union of disjoint convex polyhedral sets. It follows that $E_B^a(\psi)$ is a union of disjoint polyhedral sets, say $E_B^a(\psi) = \bigcup_{j=1}^{n_\psi} \{E_j(\psi)\}$. Thus $q^a(B,x)$ is p.w. linear with respect to the simple partition $\{E_j(\psi)\colon\ j = 1,2,\ldots,n_\psi,\ \psi \in 2^{Ⓗ}\}$.

Model 3. Information acquisition in partially observable models.

Consider a partially observable Markov chain in model 2. Define an information structure as a mapping from the set of states (unobservable) of the core process to the set of distinctive signals θ. The decision maker chooses an information structure from the set of available structures and decides upon an action for the system.

Let $a = (a_1, a_2)$ be the pair of actions, a_1 for the system control and a_2 for information acquisition. More precisely, we have

$$P^{a}_{ij}(\theta) = P^{a_1}_{ij}\, P^{a_2}_{j\theta}$$

and

$$c(x,a) = \sum_{i=1}^{N} x_i \sum_{j=1}^{N} P^{a_1}_{ij} \sum_{\theta=1}^{H} \gamma^{a_2}_{j\theta} h(i,j,\theta,a_1,a_2) \tag{II.7}$$

where $h(i,j,\theta,a_1,a_2)$ is the immediate cost of the core process when a state of the core process moves from i to j and a signal θ observed under actions a_1 for the system and a_2 for the information structure, and $x = (x_1,\ldots,x_N)$ is the probability vector with an interpretation that x_i is the probability that the core process is in state i. Note that $c(x,a)$ is linear in x.

Consider a machine maintenance and repair model (e.g., Smallwood and Sondik [60]) as an application of partially observable models. But this model is a modification of Smallwood

Sondik's. The machine consists of two internal components. The states of the core process $Z_n = i$, $i = 1,2,3$, have the following interpretation. If $i = 1$, then both components are broken down, if $i = 2$ either one is broken down and if $i = 3$ both of them are working. Assume that the machine produces M finished products at each period and the machine cannot be inspected. The actions a^1 for the machine control are to repair and not to repair the machine. The actions a^2 for information acquisition are the numbers of a sample to choose out of the M finished products. The signals θ are the number of defective products in the sample, which forms the signal process $\{\theta_n, n = 1,2,\ldots\}$. The core process $\{Z_n, n = 1,2,\ldots\}$ is the unknown states of the components of the machine. Let $x_i = P\{Z_n = i\}$, $i = 1,2,3$ and put $x = (x_1,x_2,x_3)$. Then, the process $\{(Z_n,\theta_n), n = 1,2,\ldots\}$ becomes a partially observable machine maintenance and repair model with actions $a = (a^1,a^2)$ and immediate cost $c(x,a)$ defined by (II.7).

<u>Model 4</u>. A partially observable semi-Markov model (White [67]).

Let $\{Z_t, t \geq 0\}$ be a semi-Markov chain with the finite set of states and let t_n be the time the transition occurred. Let $\tau_n = t_n - t_{n-1}$, $n = 1,2,\ldots$, with $t_0 = 0$. Then $\{(Z_{t_n},\tau_n), n = 1,2,\ldots\}$ is a Markov chain (Ross [54]). Let $Y_n = (Z_{t_n},\tau_n)$ denote the partially observable core process. Let $\{\theta_{t_n}, n = 1,2,\ldots\}$ be the signal process where each signal is observed at the time of the core process transition. The controller knows the times of the core process transition which take only finitely many integer values, i.e., $\tau_n = 1,2,\ldots,M$, for each n. Then the core process $\{Y_n, n = 1,2,\ldots\}$ is a finite state, discrete time Markov chain and the pair of two stochastic processes $\{(Y_n,\theta_n), n = 1,2,\ldots\}$ becomes the same partially observable Markov chain as in model 2, provided that the immediate cost h^a represents the expected cost with respect to the τ_n and the set of actions, $a \in A$, is finite. This model differs from White [67] in that he allows τ_n to be countable.

<u>Model 5</u>. A classical linear economic model (Walras [66])

Let x be a price vector of N commodities (or N securities) in the market and assume that a new price vector x' can be written as

$$x' = P_\theta^a x$$

where P_θ^a is an N x N matrix depending on the present economic situation θ and on an economic alternative a. Let $P[\theta|x,a]$ be the conditional probability of θ forecasted, given x and a. Assume that there exists a simple partition $\{E_i\}$ of the set of price vectors x such that

$$P[\theta|x,a] = P_{\theta i}^a \text{ for } x \in E_i,$$

which is p.w. constant with respect to $\{E_i\}$. Therefore, the model belongs to the class of model 1, provided (A.II.) is satisfied.

Chapter III

PROPERTIES OF U_δ AND U_*

This chapter is a study of the properties of U_δ and U_*. Most of these properties will be used later in the development of algorithms to find ε-optimal approximations to V^* and δ^*.

The following properties are well-known and proofs may be found in Blackwell [7], Denardo [10], and Ross [54].

Lemma III.1.

(i) U_δ and U_* are contraction mappings on $B(\Omega)$ with contraction coefficient $\beta < 1$.

(ii) U_δ, U_* are monotone; i.e., if $f, g \in B(\Omega)$ with $f \leq g$, then $U_\delta f \leq U_\delta g$ and $U_* f \leq U_* g$.

(iii) $V^\delta = V$ is the unique solution of the operator equation $U_\delta V = V$.

(iv) $V^* = V$ is the unique solution of the operator equation $U_* V = V$.

The following theorem shows how the structure in Assumption I implies that U_* and U_δ preserve the p.w. linearity of value functions and the simplicity of policies.

Theorem III.1. Suppose that (A.I.) holds and that V is p.w. linear. Then

(i) $U_\delta V$ is p.w. linear whenever δ is simple;

(ii) $U_* V$ is p.w. linear; and

(iii) there exists a simple policy δ such that $U_\delta V = U_* V$.

Proof.

(i) Suppose that δ is simple with respect to a simple partition $\{E_i\}$. Let E_i be an arbitrary but fixed cell from the partition and suppose that $\delta(x) = a$ for $x \in E_i$. Then

$$(U_\delta V)(x) = (U_a V)(x) \quad \text{for} \quad x \in E_i.$$

From (A.I.), $U_a V$ is p.w. linear for each $a \in A$. Hence $U_\delta V$ is p.w. linear on each cell E_i, and is consequently p.w. linear on Ω.

(ii & iii) The functions $U_a V$ are each p.w. linear by (A.I.). Suppose that $U_a V$ is p.w. linear with respective to the simple partition P^a. Let $P = \prod_{a \in A} P^a$. Then P is finer than each P^a, and so each $U_a V$ is p.w. linear with respect to P. For each $F \in P$ and $a \in A$, there is some linear functional α_F^a such that

$$(U_a V)(x) = \alpha_F^a(x) \quad \text{for} \quad x \in F.$$

For each $F \in P$, define the sets G_F^b, $b \in A = \{1,2,\ldots,p\}$, by

$$G_F^b = \{x\colon \alpha_F^b x < \alpha_F^a x,\ a = 1,2,\ldots,b-1 \text{ and } \alpha_F^b x \le \alpha_F^a x,\ a = b+1,\ldots,p\}.$$

Then $\{G_F^a\colon a \in A\} = \mathcal{P}^F$ is a partition of F and $\hat{\mathcal{P}} = \prod_{F \in \mathcal{P}} \mathcal{P}^F$ is a partition of Ω with the property that

$$(U_* V)(x) = \alpha_F^a(x) \quad \text{if} \quad x \in G_F^a \in \hat{\mathcal{P}}.$$

The policy δ defined by $\delta(x) = a$ for $x \in G_F^a \in \hat{\mathcal{P}}$ satisfies $U_\delta V = U_* V$.

<u>Corollary</u>. Suppose that (A.I.) holds and that $V^o \in B(\Omega)$ is p.w. linear.

(i) Define $V^n(x) = (U_\delta V^{n-1})(x)$, $n = 1,2,\ldots$.

(ii) Define $V^n(x) = (U_* V^{n-1})(x)$, $n = 1,2,\ldots$.

Then V^n is p.w. linear and the stationary policy, δ_n, defined by $U_{\delta_n} V^{n-1} = U_* V^{n-1}$ is simple.

<u>Remark III.1</u>. Part (i) of the Theorem can be generalized as follows: if δ is simple with respect to a simple partition $\mathcal{P}^\delta$ and $g(\cdot,a)$ is p.w. linear with respect to $\mathcal{P}^a$ for each $a \in A$, then $g(\cdot,\delta(\cdot))$ is p.w. linear with respect to the partition $\mathcal{P} = \mathcal{P}^\delta \cdot \prod_{a \in A} \mathcal{P}^a$.

<u>Remark III.2</u>. Suppose that instead of Assumption I, we assume that Ω is convex and that for each $a \in A$, $U_a V$ is concave whenever V is concave and non-negative. Then $U_\delta V$ and $U_* V$ are non-negative and concave whenever V is. Although this structure

will not be developed further in this thesis, we note that it is somewhat analogous to the p.w. linear structure in (A.I.).

We next consider the effects of iterating montone contraction mappings such as U_* and U_δ, citing some results of Denardo [10].

<u>Lemma III.2</u>. Suppose that U is a contraction mapping on $B(\Omega)$ with contraction coefficient $\beta < 1$. Let $V^o \in B(\Omega)$ be given and define the functions V^n, $n = 1,2,\ldots$ by

$$V^n(x) = (UV^{n-1})(x).$$

Then

(i) $\{V^n\}$ converges in norm to the fixed point $\hat{V}$ of U; i.e., $U\hat{V} = \hat{V}$.

Now assume that U is also montone.

(ii) If $V^1 \leq V^o$, then $\{V^n\}$ is monotonically decreasing to $\hat{V}$.

(iii) If $V^1 \geq V^o$, then $\{V^n\}$ is monotonically increasing to $\hat{V}$.

<u>Remark III.3</u>. The fixed point $\hat{V}$ need not be p.w. linear since the cells in the limiting partition are not necessarily finite in number nor polyhedral.

In the remainder of this chapter, U will be a contraction mapping with contraction coefficient $\beta < 1$ and fixed point $\hat{V}$. The function $V^o \in B(\Omega)$ is assumed to have been given and the functions V^n for $n = 1,2,\ldots$ are defined by $V^n = UV^{n-1}$. By the previous lemma, $\{V^n\}$ converges to $\hat{V}$. The following results concern the rate of convergence of $\{V^n\}$ to $\hat{V}$ and error bounds on $\|V^n - \hat{V}\|$.

The following two lemmas imply that $\{V_n\}$ converges to $\hat{V}$ linearly (due to Denardo [10]).

Lemma III.3.

$$\|V^n - \hat{V}\| \leq \beta\|V^{n-1} - \hat{V}\|.$$

Proof.

$$\begin{aligned}\|V^n - \hat{V}\| &= \|UV^{n-1} - U\hat{V}\| \\ &\leq \beta\|V^{n-1} - \hat{V}\|.\end{aligned}$$

Lemma III.4.

$$\|V^n - \hat{V}\| \leq \frac{1}{1-\beta} \|V^n - UV^n\|$$

Proof.

$$\begin{aligned}\|V^n - \hat{V}\| &\leq \|V^n - UV^n\| + \|UV^n - U\hat{V}\| \\ &\leq \|V^n - UV^n\| + \beta\|V^n - \hat{V}\|.\end{aligned}$$

The result is obtained by a rearranging the last expression.

<u>Lemma III.5</u>. Let $V \in B(\Omega)$. If $\|V - UV\| \leq (1-\beta)\varepsilon$, then

$$\|\hat{V} - V\| \leq \varepsilon.$$

<u>Proof</u>.

$$\|\hat{V} - V\| \leq \|U\hat{V} - UV\| + \|UV - V\|$$

$$\leq \beta\|\hat{V} - V\| + \|UV - V\|$$

Therefore $\quad \|\hat{V} - V\| \leq \|UV - V\|/(1 - \beta) \leq \varepsilon.$

<u>Theorem III.2</u>. If $\beta^n\|V^o - UV^o\| \leq (1 - \beta)\varepsilon$, then

$$\|\hat{V} - V^n\| \leq \varepsilon.$$

<u>Proof</u>. $\quad \|V^n - UV^n\| = \|UV^{n-1} - U^2V^{n-1}\|$

$$\leq \beta\|V^{n-1} - UV^{n-1}\|$$

$$\vdots$$

$$\leq \beta^n\|V^o - UV^o\| \leq (1 - \beta)\varepsilon.$$

Applying Lemma III.5. immediately gives us the result.

Chapter IV

THE ALGORITHMS

Section IV.1. The Method of Successive Approximation

The method of successive approximation is a well known and popular method for solving equations. In the context of a solution technique for solving stationary Markov decision processes it appears in Blackwell [7]. The method is to start with a cost function V^o, and to iterate U_*, constructing a sequence of cost functions $V^n = U_* V^{n-1}$, $n = 1,2,\ldots$. By Lemma III.1, U_* is a contraction mapping with fixed point V^* and by Lemma III.2, $\{V^n\}$ converges to V^*. By Theorem III.2, n can be chosen sufficiently large, so that V^n is an ε-optimal cost function. In fact by taking logarithms of the expression in Theorem III.2,

$$n > \log \left[\frac{(1-\beta)\varepsilon}{\|V^o - V^1\|}\right] / \log \beta$$

is adequate.

The next theorem provides a means of constructing an ε-optimal policy from an ε'-optimal cost function and specifies the relationship between ε and ε'. The algorithm will first construct an ε'-optimal cost function. From this cost function, an ε-optimal policy is constructed.

<u>Theorem IV.1</u>. Let $V^o \in B(\Omega)$ be p.w. linear, define $V^n = U_* V^{n-1}$, $n = 1,2,\ldots$, and let δ_n be a policy satisfying $U_{\delta^n} V^{n-1} = U_* V^{n-1}$. If $\|V^* - V^{n-1}\| < \frac{(1-\beta)\varepsilon}{2\beta}$, then

$$\|V^* - V^{\delta_n}\| < \varepsilon.$$

<u>Proof</u>. By Lemma III.1 U_δ for any stationary policy δ is a contraction mapping and the fixed point is V^δ, i.e., $V^\delta = U_\delta V^\delta$. Consider

$$\begin{aligned}\|V^* - V^{\delta_n}\| &= \|U_{\delta_n} V^{\delta_n} - U_* V^*\| \\ &\leq \|U_{\delta_n} V^{\delta_n} - U_{\delta_n} V^*\| + \|U_{\delta_n} V^* - U_{\delta_n} V^{n-1}\| \\ &\quad + \|U_* V^{n-1} - U_* V^*\| \\ &\leq \beta \|V^{\delta_n} - V^*\| + \beta\|V^* - V^{n-1}\| + \beta\|V^{n-1} - V^*\|\end{aligned}$$

where we used the equality $U_* V^{n-1} = U_{\delta_n} V^{n-1}$. Arranging the above inequality, we obtain

$$(1 - \beta)\|V^{\delta_n} - V^*\| \leq 2\beta\|V^* - V^{n-1}\| < (1 - \beta)\varepsilon,$$

which completes the proof.

If the state space X is uncountable, or even countably infinite, then this procedure is difficult to implement on a computer. However, if the Markov decision process has the structure of (A.I.) and V^o is p.w. linear, then each V^n is p.w. linear and each δ^n constructed as in the previous theorem is simple (by Theorem III.1.). In this case, the cost functions and policies can be specified by a finite number of items - the inequalities describing each cell of a simple partition and the corresponding action or linear function.

<u>Algorithm to find an ε-optimal simple policy.</u>

(i) Start with any p.w. linear function V^o.

(ii) Compute $V^1 = U_* V^o$.

(iii) Choose an integer n such that

$$\beta^n \| V^o - V^1 \| \leq (1 - \beta)\varepsilon',$$

where $\varepsilon' = (1 - \beta)\varepsilon/2\beta$. I.e., choose $\hat{n}$ larger than

$$\log \left[\frac{(1-\beta)^2 \varepsilon}{2\beta \| V^o - V^1 \|}\right] / \log \beta.$$

(iv) Compute $V^n = U_* V^{n-1}$ successively until $n = \hat{n}$.

(v) Consequently, we obtain $V^{\hat{n}}$ such that

$$\| V^* - V^{\hat{n}} \| \leq \varepsilon'.$$

(vi) Construct a policy δ satisfying

$$U_\delta V^{\hat{n}} = U_* V^{\hat{n}}.$$

Then δ is ε-optimal.

<u>Remark IV.1</u>. The algorithm can be started with $V^o \equiv 0$.

<u>Remark IV.2</u>. The termination criterion, $n = \hat{n}$, in the algorithm has the advantage that $\|V^o - V^1\|$ is computed only once. However, it has the disadvantage that $\hat{n}$ will probably be larger than necessary, causing unnecessary iterations.

An alternative would be to compute $\|V^n - V^{n-1}\|$ at each iteration and stop whenever $\|V^n - V^{n-1}\| \leq (1 - \beta)\varepsilon'/\beta$. Theorem III.2 guarantees that V^n is an ε'-optimal cost function. However, the computations of $\|V^n - V^{n-1}\|$ will, in general, be expensive.

The best procedure is undoubtedly to check $\|V^n - V^{n-1}\|$ at some, but not all, iterations. For example, $\hat{n}$ might be computed based on $\|V^o - V^1\|$. Then at some iteration n near $\frac{\hat{n}}{2}$, recompute $\hat{n}$ based on $\|V^n - V^{n-1}\|$. This is the procedure suggested in the next chapter.

Section IV.2. The Method of Policy Improvement

Another commonly proposed method for solving Markov decision problems is policy improvement (Howard [26]). Policy improvement is actually Newton's method applied to the convex operator equation $(I - U_*)V = 0$ to find the solution V^*. Newton's method converges super-linearly in many situations, and this property is maintained when applied to some Markov decision problems (Brumelle & Puterman [8], Puterman & Brumelle [49]). Since the successive approximation method converges only linearly (Lemma III.3.), it is desirable to adapt the policy improvement method to our model. Our version of policy improvement includes the successive approximation method as a special case.

Given a policy δ with cost V^{δ}, an iteration of policy improvement consists of finding a policy δ' such that $U_{\delta'}V^{\delta} = U_*V^{\delta}$, and then solving the linear equation $V = U_{\delta'}V$ for $V^{\delta'}$.

One method of solving the operator equation $V = U_{\delta}V$ for V^{δ} is the method of successive approximation, i.e., by iterating U_{δ}. More explicitly, start with a cost function V^{o} and iterate U_{δ}, constructing a sequence of cost functions $V^{n} = U_{\delta}V^{n-1}$, $n = 1,2,\ldots$. By Lemma III.1, U_{δ} is a contraction mapping with a fixed point V^{δ}, and by Lemma III.2, $\{V^{n}\}$ converges to V^{δ}. By Theorem III.2, for any given $\varepsilon > 0$, n can be chosen sufficiently large so that $\|V^{n} - V^{\delta}\| \leq \varepsilon$. However, we will show that it is not necessary to approximate V^{δ} at all closely in the policy improvement algorithm.

In the remainder of this section, we discuss the algorithm in general terms and then discuss the specific points of starting the algorithm, choosing the parameters $\{k_n\}$ which specify the degree of approximation of V^δ in the n-th iteration, terminating the algorithm, and a proof that the algorithm converges. Since $c(x,a)$ is bounded, there exists a constant M such that $c(x,a) \le M$ $\forall x,a$. Let $\hat{c}(x,a) = c(x,a) - M \le 0$ and define

$$\hat{V}_\delta(x) = E_\delta[\sum_{n=1}^{\infty} \beta^{n-1}\hat{c}(X_n,\delta(X_n))\,|X_1 = x].$$

Then $\hat{V}_\delta(x) = V_\delta(x) - M/(1-\beta)$. Hence a minimization of $\hat{V}_\delta$ is equivalent to a minimization of V_δ over $\delta \in \Delta$. It is, therefore, easy to find a p.w. linear function $\hat{V}$ such that $U_\delta\hat{V} \le \hat{V}$. For instant, put $\hat{V} = 0$ which is p.w. linear and satisfies $U_\delta\hat{V} \le \hat{V}$.

Algorithm for finding an ε-optimal policy under (A.I.).

Start with a simple policy δ^o and a p.w. linear function $y^o \in B(\Omega)$ satisfying $y^o \ge U_{\delta^o} y^o$.

An iteration of the algorithm is described as follows: $n = 0,1,2,\ldots$. At the start of the n-th iteration, we have a simple policy δ^n and a p.w. linear function $y^n \in B(\Omega)$ satisfying $y^n \ge U_{\delta^n} y^n$.

(i) Compute $U_{\delta^n}^{k_n} y^n$ where the integer k_n is the number of iterations of U_{δ^n} which are to be performed.

(ii) Set $y^{n+1} = U_{\delta^n}^{k_n} y^n$ and find a policy δ^{n+1} such that $U_{\delta^{n+1}} y^{n+1} = U_* y^{n+1}$.

(iii) If $\|y^n - y^{n+1}\| \le (1-\beta)\varepsilon$, then stop with y^n ε-optimal and δ_n ε-optimal. Moreover, $V^* \le V^{\delta_n} \le y^{n+1}$.

(iv) If $\|y^n - y^{n+1}\| \leq (1 - \beta)\varepsilon$, then increment n by 1 and perform another iteration.

To start, the algorithm needs a simple policy δ and a p.w. linear function y satisfying $y \geq U_\delta y$. There is no difficulty in finding a simple policy; for example, $\delta(x) = a$ for all $x \in \Omega$ is satisfactory. Finding a satisfactory y is more difficult unless the model is specified further. For example, on page 12 in Model 2, $q^a(r,x) = (p^a)^T x$. So if $\delta(x) = a$ for all $x \in \Omega$, then $V^\delta(x) = c^a[I - \beta(p^a)^T]^{-1}x$ for all $x \in \Omega$. Setting $y = V^\delta$ provides a starting vector.

If y^n is a p.w. linear function and δ^n is simple, it follows from Theorem III.1. and (A.I.) that y^{n+1} is p.w. linear and that δ^{n+1} is simple. Theorem III.1 also implies that each of the intermediate iterates $U^j_{\delta^n} y^n$, $j = 1,2,\ldots,k_n$ are p.w. linear. Consequently, the algorithm can start and the iterations are well defined.

The question of how best to establish the appropriate values of the parameters $\{k_n\}$ in the algorithm has not been resolved. If each $k_n = 0$, then the algorithm reduces to that of successive approximation described in the last section and which is known to converge linearly. However, the effort per iteration is small. If each $k_n = \infty$, then the method is known to converge super-linearly in many situations ([8], [49]). However, in this case the effort per iteration is large. In general, it seems appropriate to take k_n small, perhaps even 0, in the early iterations. However, once the neighborhood of V* is reached, k_n should be large enough so that $U^{k_n}_{\delta^n} y_n$ approximates V^{δ^n} in order to take advantage of the super-linear convergence.

<u>Theorem IV.1</u>. For each iteration, $n = 0,1,2,\ldots$, in the policy improvement algorithm,

$$y^n \geq U_{\delta^n} y^n \geq U^2_{\delta^n} y^n \geq \cdots \geq U^{k_n}_{\delta^n} y^n = y^{n+1}.$$

<u>Proof</u>. First, it is true for $n = 0$. Since $y^0 \geq U_{\delta^0} y^0$ and since by Theorem III.1 U_{δ^0} is monotone, it follows that $y^0 \geq U_{\delta^0} y^0 \geq U^2_{\delta^0} y^0 \geq \cdots \geq U^{k_0}_{\delta^0} y^0 = y^1 \geq U_{\delta^0} y^1$. By definition δ_1 satisfies $U_{\delta_1} y^1 = U_* y^1$. However, $U_* y^1 \leq U_{\delta^0} y^1 \leq y^1$, and so not only is the Theorem established for $n = 0$, but we have also shown that $U_{\delta^1} y^1 \leq y^1$.

Now suppose $U_{\delta^n} y^n \leq y^n$. The same argument as in the first paragraph establishes the Theorem for n and also that $U_{\delta^{n+1}} y^{n+1} \leq y^{n+1}$. Hence the proof is completed by induction.

<u>Corollary</u>. $y^n \geq V^*$ for $n = 1,2,\ldots$.

<u>Proof</u>. For an arbitrary n, $y^n \geq U_{\delta^n} y^n \geq U_* y^n$. Since U_* is monotone (Lemma III.1), $y^n \geq U^j_* y^n$ for each j. By Lemma III.2, $U^j_* y^n$ decreases monotonically and converges to V^* as $j \to \infty$. Consequently, $y^n \geq V^*$ and the proof is complete.

We next show that if the algorithm terminates then it will provide an ε-optimal cost function and an ε-optimal policy.

<u>Theorem IV.2</u>. If $\|y^n - y^{n+1}\| \leq (1-\beta)\varepsilon$, then $\|y^n - V^*\| \leq \varepsilon$, i.e., y^n is ε-optimal. Moreover, δ_n is also ε-optimal and $V^* \leq V^{\delta_n} \leq y^n$.

<u>Proof</u>. Note that $U_{\delta^n} y^n = U_* y^n$ and that by the previous corollary $y^n \geq V^*$.

$$\|y^n - V^*\| \leqq \|y^n - U_* y^n\| + \|U_* y^n - U_* V^*\|$$

$$\leqq \|y^n - U_{\delta^n} y^n\| + \beta\|y^n - V^*\|$$

$$\leqq \|y^n - U^m_{\delta^n} y^n\| + \beta\|y^n - V^*\| \quad \text{for } m = 1,2,\ldots,$$

because $\quad y^n \geq U_{\delta^n} y^n \geq U^m_{\delta^n} y^n \quad$ for $m = 1,2,\ldots$. (Theorem IV.1)

Thus $\quad (1-\beta)\|y^n - V^*\| \leq \|y^n - U^m_{\delta^n} y^n\| = \|y^n - y^{n+1}\| \leq (1-\beta)\varepsilon,$

and so $\quad \|y^n - V^*\| \leq \varepsilon.$

The last statement in the Theorem follows by Theorem IV.1.

The following theorem has been shown by Doshi [16] for continuous time Markov processes. But our proof is different from and simpler than his.

<u>Theorem IV.3</u>. Let V^δ be the cost of any stationary policy δ. Let δ' be a policy defined by $U_{\delta'} V = U_* V$.

(i) If $U_{\delta'}V^{\delta} = V^{\delta}$, then $V^{\delta'} = V^{\delta}$, and δ' and δ are optimal.

(ii) $V^{\delta'} \leq V^{\delta}$. Furthermore, if for some $x_0 \in \Omega(U_{\delta'}V^{\delta})(x_0) < V^{\delta}(x_0)$, then

$$V^{\delta'}(x_0) < V^{\delta}(x_0).$$

Proof.

(i) From the definition of δ' we have

$$U_*V^{\delta} = U_{\delta'}V^{\delta} = V^{\delta}.$$

Since the optimal cost V^* is the unique solution of U_*, $V^{\delta} = V^*$. By induction on n, $U_{\delta'}^n V^{\delta} = V^{\delta}$ since $U_{\delta'}V^{\delta} = V^{\delta}$. But

$$U_{\delta'}^n V^{\delta} \rightarrow V^{\delta'} \text{ as } n \rightarrow \infty \text{ by Lemma III.2.}$$

Hence $V^{\delta'} = V^{\delta}$ because $V^{\delta'}$ is the unique fixed point of $U_{\delta'}$. (Lemma III.1.)

(ii) By definition of δ' and $V^{\delta} = U_{\delta}V^{\delta}$ (Lemma III.1.)

$$U_{\delta'}V^{\delta} \leq U_{\delta}V^{\delta} = V^{\delta}.$$

By induction on n

$$U_\delta^n, V^\delta \leq V^\delta$$

$$U_\delta^n, V^\delta \to V^{\delta'} \quad \text{as } n \to \infty.$$

$$V^{\delta'} \leq V^\delta.$$

Suppose $\quad (U_\delta, V^\delta)(x_0) < V^\delta(x_0)$ for some $x_0 \in \Omega$.

$$V^{\delta'}(x_0) = (U_\delta, V^{\delta'})(x_0) \quad \text{(From Lemma III.1.)}$$

$$\leq (U_\delta, V^\delta)(x_0) \quad (V^{\delta'} \leq V^\delta)$$

$$< V^\delta(x_0) \quad \text{(the assumption).}$$

Lemma IV.1. Let $\{y^n\}$ be a sequence generated by the policy improvement algorithm. If y^n converges pointwise to y, then

$$U_* y^n \text{ converges to } U_* y.$$

Proof. In this proof all limits are with respect to the pointwise topology.

Let $z_a^n = U_a y^n$ and $z_a = U_a y$ for each a, $n = 1,2,\ldots$.
By the monotone convergence theorem,

$$\begin{aligned}
\lim_n (z_a^n)(x) &= \lim_n (U_a y^n)(x) \\
&= \lim_n \{c(x,a) + \beta \int_\Omega y^n(x')\,q(dx'|x,a)\} \\
&= c(x,a) + \beta \int_\Omega \lim_n y^n(x')\,q(dx'|x,a) \\
&= c(x,a) + \beta \int_\Omega y(x')\,q(dx'|x,a) \\
&= (U_a y)(x) \\
&= (z_a)(x) \quad \text{for each } a \in A,\ x \in \Omega.
\end{aligned}$$

To show $U_* y^n \searrow U_* y$ is equivalent to showing that

$$\min_a (z_a^n)(x) \searrow (\min_a z_a)(x) \quad \text{for all } x \in \Omega.$$

Since A is finite,

$$\left(\min_a z_a^n\right)(x) = \min_a\left(z_a^n\ (x)\right) \text{ and } (\min_a z_a)(x) = \min_a\left(z_a(x)\right).$$

Let $x \in \Omega$ be arbitrary but fixed, and define

$$\alpha_a^n = z_a^n(x) \text{ and } \alpha_a = z_a(x)$$

which are just numbers. Since $(z_a^n)(x) \searrow (z_a)(x)$ pointwise, then

$$\alpha_a^n \searrow \alpha_a \quad \text{for each } a \in A.$$

It remains to show that $\min_{a \in A} \alpha_a^n \searrow \min_a \alpha_a$. It is clear that $\min_a \alpha_a^n$ is monotone decreasing. Since $\alpha_a^n \searrow \alpha_a$ it follows that

$$\min_a \alpha_a \le \min_a \alpha_a^n \quad \text{for } n = 1,2,\ldots\ .$$

Hence

$$\min_{a \in A} \alpha_a \le \lim_n \min_a \alpha_a^n .$$

To show the other way suppose that $\bar{a}$ is the action such that

$$\min_{a \in A} \alpha_a = \alpha_{\bar{a}}\ .$$

Then

$$\min_{a \in A} \alpha_a = \alpha_{\bar{a}} = \lim_n \alpha_{\bar{a}}^n \ge \lim_n \min_a \alpha_a^n$$

Therefore

$$\lim_n \min_a \alpha_a^n = \min_a \alpha_a$$

which completes the proof.

<u>Theorem IV.4</u>. Suppose that $\{y^n\}$ is a sequence of costs generated

by the policy improvement algorithm.

(i) y^n converges pointwise to $y \in B(\Omega)$.

(ii) $y = U_* y$, i.e., y is optimal.

In other words, the policy improvement algorithm converges.

Proof.

(i) First of all we shall show that $\{y^n\}$ is bounded below. By Theorem IV.1 we have $y^n \geq U^m_{\delta^n} y^n$ for each $m = 1,2,\ldots$. By Theorem III.2 $U^m_{\delta^n} y^n \to V^{\delta^n}$ as $m \to \infty$. Therefore $y^n \geq V^{\delta^n}$. Since the cost $c(x,a)$ is bounded below, i.e., $|c(x,a)| \leq M$ for all x, a, $|V^{\delta^n}| \leq \frac{M}{1-\beta}$. Hence $y^n(x) \geq \frac{-M}{1-\beta}$ for all x. From Theorem IV.1 y^n is a decreasing sequence. Hence y^n converges pointwise.

(ii) By a choice of y^0 and Theorem IV.1 we know that

$$y^n \geq U_{\delta^n} y^n \geq U_* y^n. \tag{IV.1}$$

To show the other way we have

$$\begin{aligned} y^n &= U^m_{\delta^{n-1}} y^{n-1} && \text{(By definition of } y^n) \\ &\leq U_{\delta^{n-1}} y^{n-1} && (U^m_\delta y \leq Uy,\ \forall y \in B(\Omega)) \\ &= U_* y^{n-1} && \text{(By definition of } \delta^{n-1}). \end{aligned} \tag{IV.2}$$

Then, from (IV.1.), and (IV.2), we obtain

$$U_* y^n \leq y^n \leq U_* y^{n-1}.$$

From the statement (i) $y^n \searrow y$ and then, from Lemma IV.1. $U_* y^n \to U_* y$. Therefore, we must have

$$U_* y = y$$

which completes the proof.

Chapter V

IMPLEMENTATION OF THE ALGORITHMS FOR MODEL 1

Section 1. Introduction

In this chapter we shall consider in a more concrete setting the methods of successive approximation and of policy improvement.

To show how each method is actually handled, we assume in this chapter that X is the N-dimensional real space (i.e., $X = R^N$) and that Ω is a bounded convex polyhedral set of R^N. Let $c(x,a) = c^a \cdot x$, which is the inner product of two vectors c^a, $x \in R^N$, so that (A.II.) holds. Let $A = \{1,2,\ldots,p\}$. We repeat (A.III.) a bit more explicitly than in Chapter 2.

Assumption III (A.III.) For each convex polyhedral set $B \subset R^N$ and each action $a \in A$, the function $q^a(B,x)$ defined by

$$q^a(B,x) = \int_B x' q(dx' \mid x,a)$$

is p.w. linear in x with respect to a simple partition

$$\mathcal{P}^a(B) = \{E_j(a,B) : j = 1,2,\ldots,m_{a,B}\}.$$

We write $q^a(B,x) = q_j^a(B) \cdot x$ when $x \in E_j(a,B)$.

Remarks V.1. Note that (A.III.) places us in the context of model 1 in Chapter II. Recall from the discussion there and in model 2 that under (A.III.), $U_a V$ is p.w. linear whenever V is, and that partially observable models satisfy (A.III.).

Suppose that f is a p.w. linear function, linear on the cells of the partition $\{E_i,\ i = 1,2,\ldots,n\}$, that $f(x) = f_i \cdot x$ on E_i, and that $E_i = \{x\colon K^i x < b^i;\ L^i x \leq d^i\}$, $i = 1,2,\ldots,n$. Each b^i and d^i is an N-dimensional vector and each K^i and L^i is a matrix with N-dimensional rows. This situation will be denoted by

$$f \sim \{(f_i;\ K^i,\ b^i;\ L^i,\ d^i)\colon i = 1,2,\ldots,n\}$$

and

$$E_i \sim (K^i,b^i;\ L^i,d^i).$$

If δ is a simple policy with respect to the partition $\{E_i,\ i = 1,2,\ldots,n\}$, say $\delta(x) = a_i$ for $x \in E_i$, then we will represent δ by

$$\delta \sim \{(a_i;\ K^i,b^i;\ L^i,d^i)\colon i = 1,2,\ldots,n\}.$$

Define a operator $\circ$ by

$$(K,b;\ L,d) \circ (K',b';\ L',d') = \left[\binom{K}{K'},\binom{b}{b'};\ \binom{L}{L'},\binom{d}{d'}\right].$$

If A and B are matrices each having the same number of columns then $\binom{A}{B}$ is the matrix whose first rows are those of A and whose latter rows are those of B. This operator forms the intersection of the convex polyhedral sets characterized by $(K,b;\ L,d)$ and $(K',b';\ L',d')$. This representation of p.w. linear functions simple policies, and convex polyhedral sets is convenient for machine storage.

We will normally use the same symbol for the p.w. linear function (convex polyhedral set, simple policy, respectively) and the array which represents it. The only aspect of this abuse of notation which is likely to cause any confusion concerns convex polyhedral sets. Let $E \sim (K,b;\ L,d)$ be a convex polyhedral set. The set E is empty if $\{x\colon Kx < b;\ Lx \leq d\} = \phi$. The array E is empty if there are no entries in the array, as when the array is initialized.

The user of either of these methods must specify the values $q^a(B,x)$ for each convex polyhedral set B and each $x \in \Omega$. We assume that this specification is provided by a subroutine, called Q, which has as its arguments an action a, matrices K and L, and vectors b and d. The arrays K, L, b and d specify the convex polyhedral set $B = \{x\colon Kx < b,\ Lx \leq d\}$. The subroutine Q has as its output an array

$$\{(\lambda_j;\ K^j,b^j;\ L^j,d^j)\colon j = 1,2,\ldots,m\}$$

which characterizes the p.w. linear function $q^a(B,\cdot)$. The subroutine Q appropriate for model 2 is described in detail in section 6.

Sections 2 and 3 describe subroutines UDELTA and USTAR which respectively compute $U_{\delta}V$ for a given δ and V, and compute $U_{*}V$ for a given V.

Sections 4 and 5 describe implementations of the methods of successive approximation and of policy improvement.

Section 2. Subroutine UDELTA (δ,V,U_δV)

The inputs to this subroutine are a simple policy δ which takes the value $\delta(x) = a_i$ for $x \in E_i$, $i = 1,2,\ldots,n$, and a p.w. linear function V which takes the values $V(x) = V_j \cdot x$ for $x \in F_j$, $j = 1,2,\ldots,m$. Let $P^\delta = \{E_i: i = 1,2,\ldots,n\}$ and $P_V = \{F_j: j = 1,2,\ldots,m\}$. We let

$$E_i \sim \{(K^{ij},\ b^{ij};\ \ L^{ij},\ d^{ij}),\ j = 1,2,\ldots,n_i\}$$

and

$$F_j \sim \{(\bar{K}^{jk},\ \bar{b}^{jk};\ \ \bar{L}^{jk},\ \bar{d}^{jk}),\ k = 1,2,\ldots,m_j\}.$$

We also assume that the vectors c^a, $a = 1,2,\ldots,p$, and the discount factor β are available in common.

The subroutine outputs the p.w. linear function $U_\delta V$ and is based on the following computation.

$$\begin{aligned}
(U_\delta V)(x) &= c(x,\delta(x)) + \beta\int_\Omega V(x')q(dx'|x,\delta(x)) \\
&= c^{\delta(x)} \cdot x + \beta\sum_{j=1}^{m} V_j \int_{F_j} x'q(dx'|x,\delta(x)) \\
&= c^{\delta(x)} \cdot x + \beta\sum_{j=1}^{m} V_j q^{a_r}(F_j,x) \text{ for } x \in E_r.
\end{aligned}$$

Then, using the notation of (A.III.),

$$(U_\delta V)(x) = [c^{a_r} + \beta \sum_{j=1}^{m} V_j \lambda_{j\ell}^{a_r}] \cdot x$$

for $x \in E_r \cap G_\ell$ where G_ℓ is the ℓ-th cell of the partition $P^{a_r}(F_j)$. Note that the index ℓ depends on j.

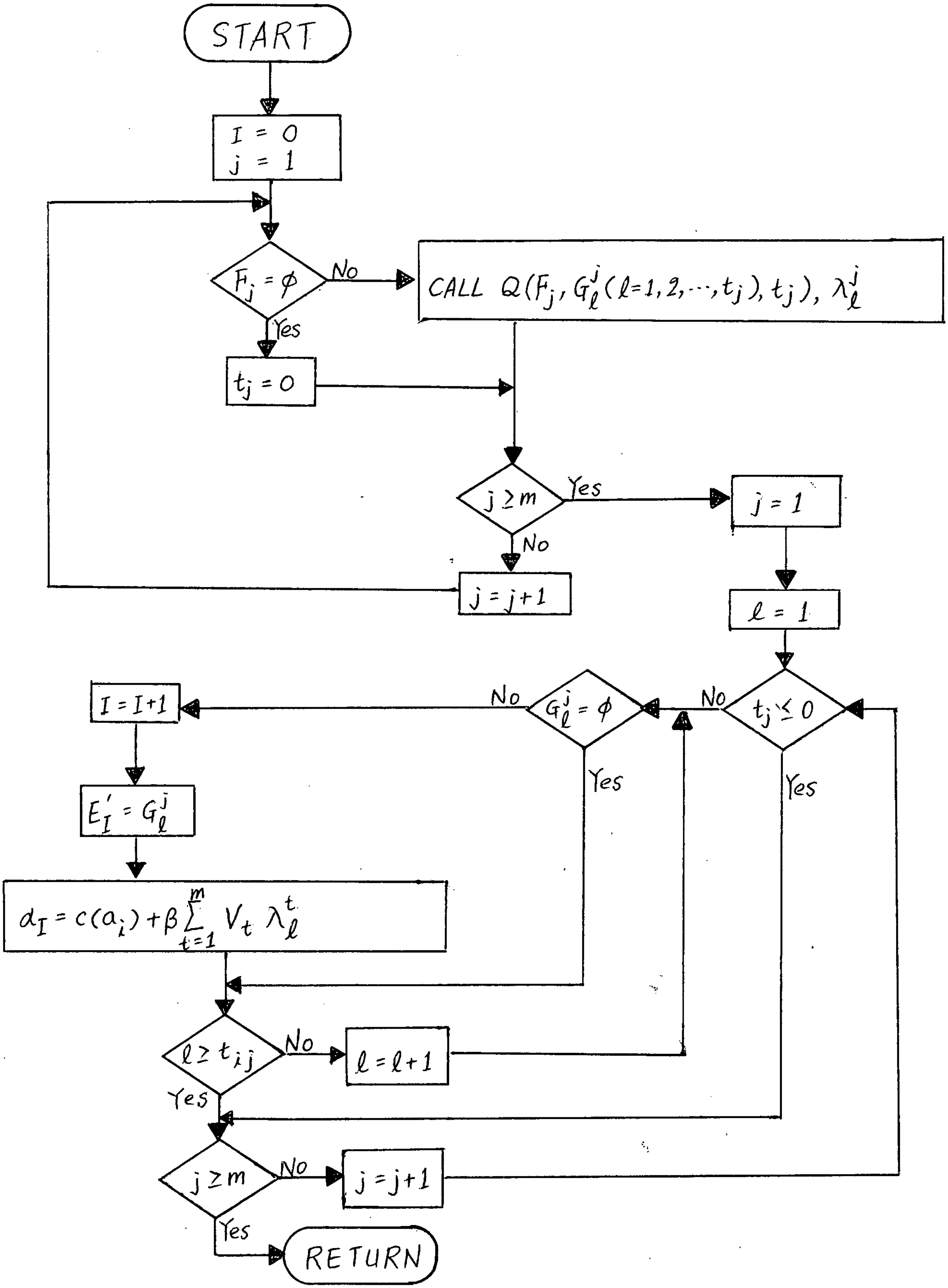
Set I = 0. I will count the number of cells in the partition for $U_\delta V$. For j = 1,2,...,n call $Q(F_j, a_i)$, which will return with an array characterizing the p.w. linear function $q^{a_r}(F_j,\cdot)$, say

$$q^{a_r}(F_j,\cdot) \sim \{(\lambda_\ell^j;\ K^{j\ell};\ b^{j\ell};\ L^{j\ell}, d^{j\ell}),$$

$$\ell = 1,2,\ldots,t\}.$$

Then for j = 1,2,...,m and ℓ = 1,2,...,t do the following. For r = 1,2,...,n form $E_r \cap G$ where $G \sim (K^{j\ell}, b^{j\ell};\ L^{j\ell}, d^{j\ell})$. If $E_r \cap G$ is empty, then do the next r. If $E_r \cap G \neq \phi$ then increment I by 1 and store $E_r \cap G$ as E_I'. Compute $c(a_r) + \beta \sum_{j=1}^{m} V_j \lambda_\ell^j$ and store as α_I.

The subroutine is now completed and $(U_\delta V)(x) = \alpha_i \cdot x$ for $x \in E_i'$, i = 1,2,...,I. It returns with the array $U_\delta V \sim \{I,\ (\alpha_i;\ E_i'),\ i = 1,2,\ldots,I\}$ as output.


START
I = 0
j = 1
$F_j = \phi$
No
Yes
CALL $Q(F_j, G_\ell^j(\ell=1,2,\cdots,t_j), t_j)$, λ_ℓ^j
$t_j = 0$
$j \geq m$
Yes
No
$j = 1$
$j = j+1$
$\ell = 1$
$I = I+1$
No
$G_\ell^j = \phi$
No
$t_j \leq 0$
Yes
Yes
$E_I' = G_\ell^j$
$d_I = c(a_i) + \beta \sum_{t=1}^{m} V_t \lambda_\ell^t$
$\ell \geq t_{ij}$
No
$\ell = \ell+1$
Yes
$j \geq m$
No
$j = j+1$
Yes
RETURN

Section 3. Subroutine USTAR (V, U_*V, δ)

Suppose that V is p.w. linear with respect to a simple partition $\{E_i, i = 1,2,...,n\}$. The subroutine USTAR computes U_*V and finds a simple policy δ such that $U_\delta V = U_*V$.

The argument of USTAR is a p.w. linear function $V \sim \{(V_i, E_i): i = 1,2,...,n\}$.

An array describing the convex polyhedral set Ω, the discount factor β and the vectors c^a, $a = 1,2,...,p$ should be available in common.

The subroutine outputs I and the array $(U_*V, \delta) \sim \{(\alpha_i, E_i', a_i): i = 1,2,...,I\}$. The function U_*V is obtained by $(U_*V)(x) = \alpha_i \cdot x$ for $x \in E_i'$. The policy δ defined by $\delta(x) = a_i$ for $x \in E_i'$, $i = 1,2,...,I$, satisfies $U_\delta V = U_*V$.

The paragraph summarizes the procedure in USTAR. The subroutine first computes U_aV for $a \in A$ using UDELTA. Let $\mathcal{P}^a$ be the simple partition for U_aV. USTAR next forms the product partition $\mathcal{P} = \prod_{a \in A} \mathcal{P}^a$. Then $\mathcal{P}$ is finer than each $\mathcal{P}^a$, and so each U_aV is p.w. linear with respect to $\mathcal{P}$. For each $F \in \mathcal{P}$ and $a \in A$, there is some vector α_F^a such that

$$(U_aV)(x) = \alpha_F^a \cdot x \quad \text{for} \quad x \in F.$$

For each $F \in \mathcal{P}$, define the sets G_F^b, $b \in A$, by

$$G_F^b = \{x: \alpha_F^b x < \alpha_F^a x, \ a = 1,2,...,b-1 \text{ and } \alpha_F^b x \leq \alpha_F^a x, \ a = b+1,...,p\}.$$

Then $\{G_F^a: a \in A\} = \mathcal{P}^F$ is a partition of F and $\hat{\mathcal{P}} = \prod_{F \in \mathcal{P}} \mathcal{P}^F$ is a partition of Ω with the property that

$$(U_*V)(x) = \alpha_F^a \cdot x \quad \text{if} \quad x \in G_F^a \in \hat{P}.$$

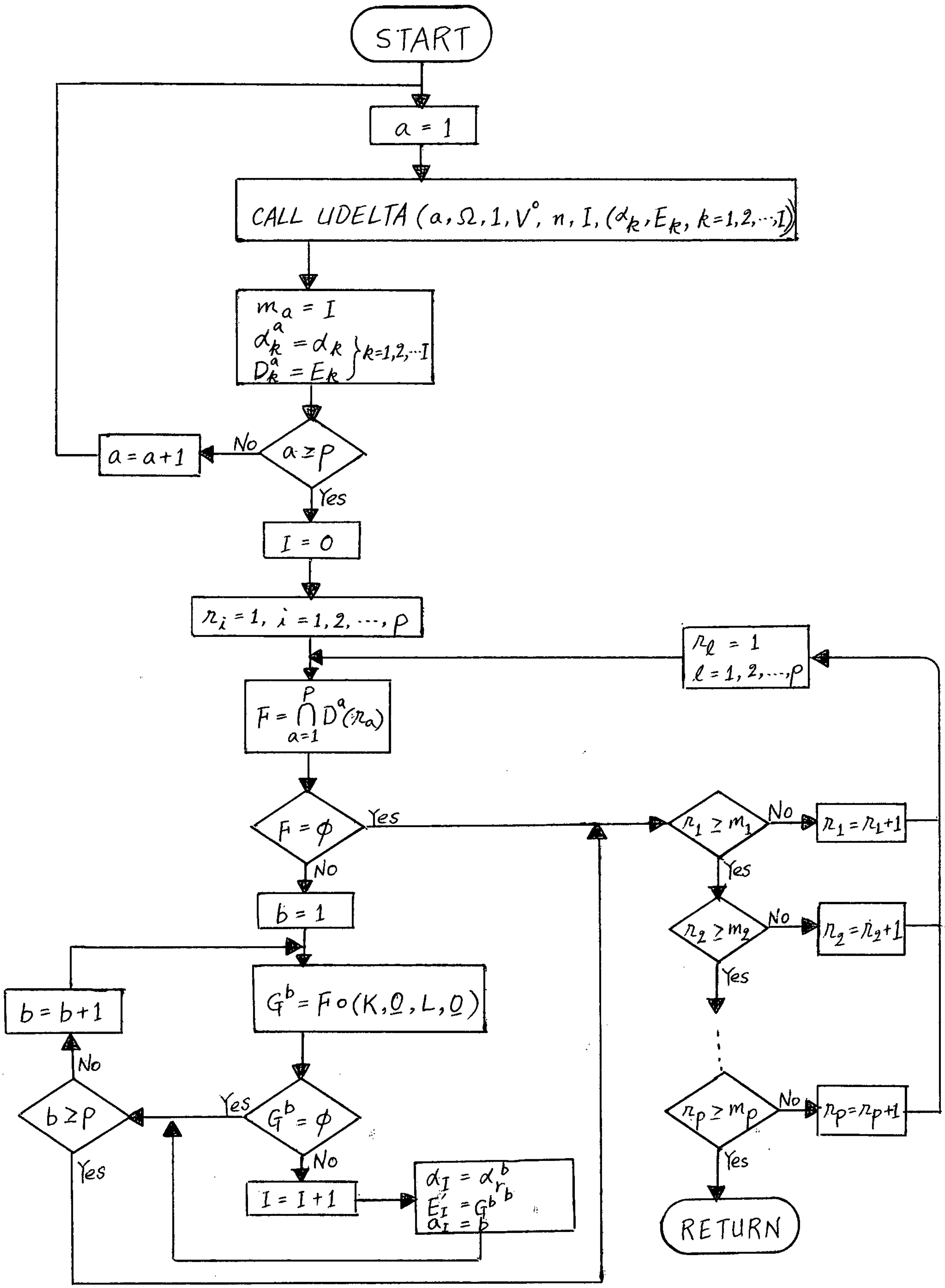
The policy δ defined by $\delta(x) = a$ for $x \in G_F^a$ $(\in \hat{P})$ satisfies $U_\delta V = U_*V$.

We now consider the subroutine in more detail. For each $a \in A$, call UDELTA with the arguments $V \sim \{(V_i, E_i): i = 1,2,\ldots,n\}$ and $\delta \sim \{(a,\Omega)\}$. This generates the arrays $\{(\alpha_j^a, D^a(j))$, $j = 1,2,\ldots,m_a\}$. Recall that each of the convex polyhedral sets E_i, $D^a(j)$, and Ω are themselves arrays of the form $\{(K^i, b^i;\ L^i, d^i): i = 1,2,\ldots,m\}$. The index I will count the cells in the partition for U_*V. Set $I = 0$.

Let R be the set of all p-dimension vectors with the i-th component, r_i, between 1 and m_i for $i = 1,2,\ldots, p$. Systematically construct each $r \in R$ in turn. Compute the set $F \sim \underset{a=1,2,\ldots,p}{\circ} D^a(r_a)$. The set F is a cell of the product partition $P = \prod_{a=1}^{p} P^a$. If F is empty, then construct the next $r \in R$. Otherwise, for each $b \in A$ construct the set

$$G_F^b = F \circ (K, \underline{0};\ L, \underline{0})$$

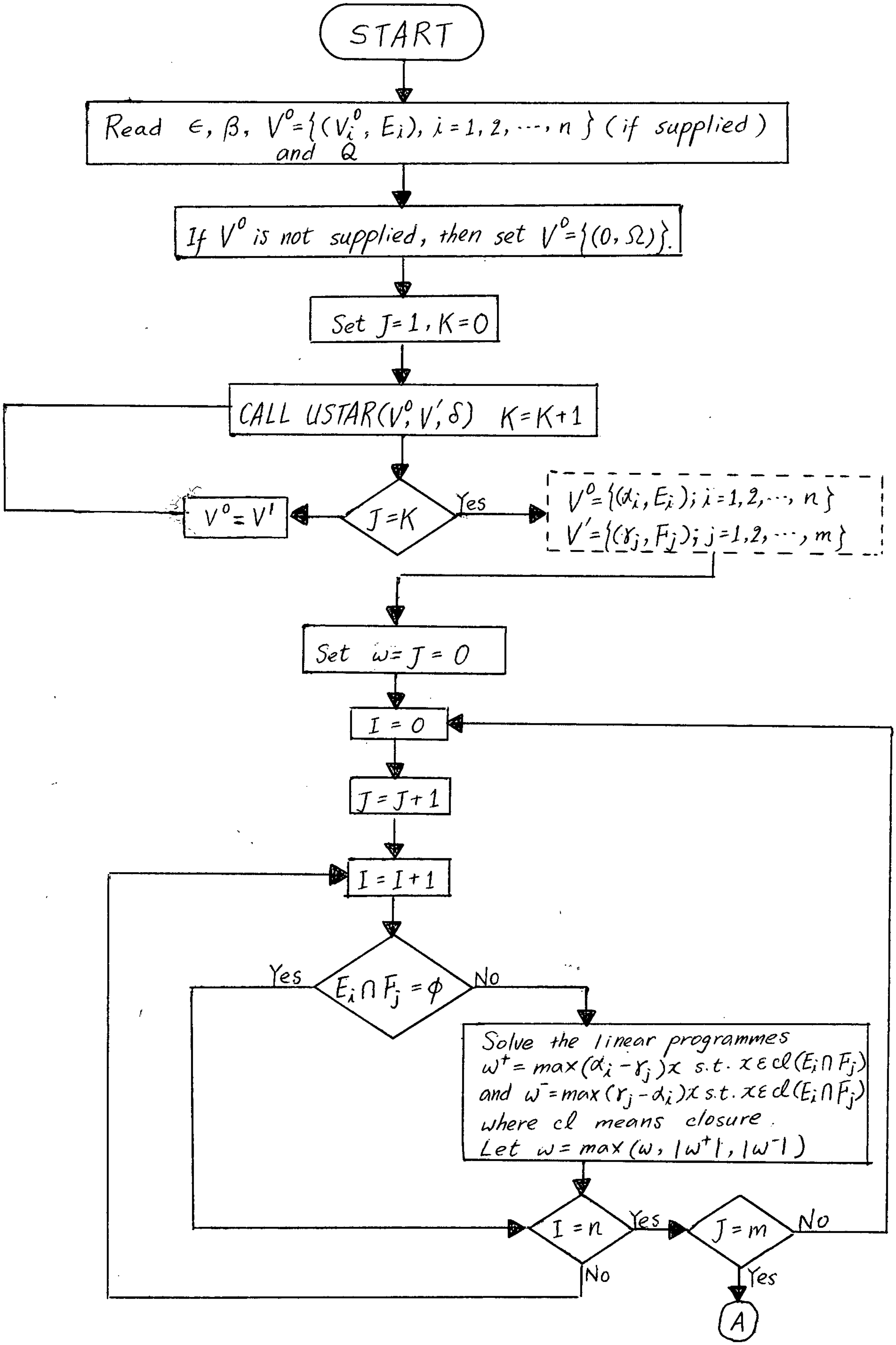
where K is a $(b-1) \times N$ matrix with rows $\alpha_{r_b}^b - \alpha_{r_a}^a$, $a = 1,2,\ldots,b-1$ and L is a $(p-b) \times N$ matrix with rows $\alpha_{r_b}^b - \alpha_{r_a}^a$, $a = b+1,\ldots,p$. If G_F^b is empty, then construct the set G for the next $b \in A$. If $G_F^b \neq \phi$, then increment I by 1 and store $\alpha_I = \alpha_{r_b}^b$, $E_I' = G_F^b$, and $a_I = b$. When each $r \varepsilon R$ has been considered, the subroutine returns.

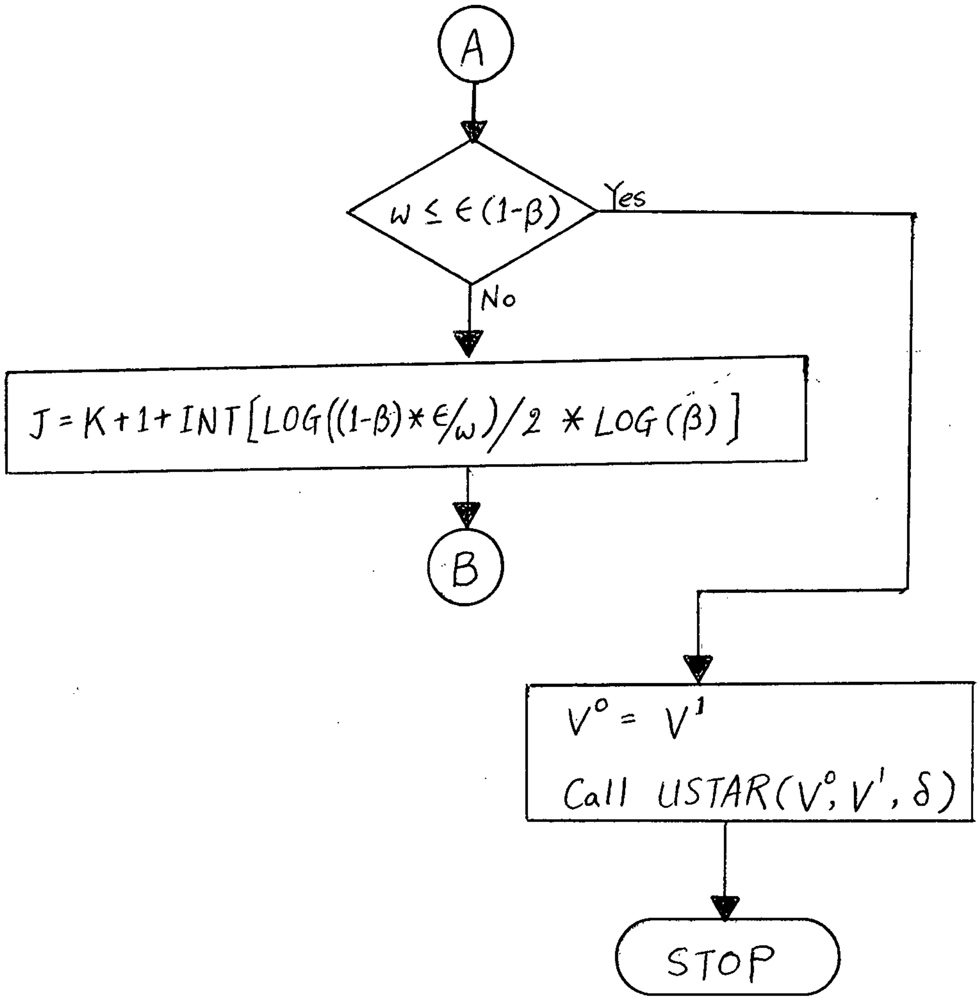

START
a = 1
CALL UDELTA (a, Ω, 1, V°, n, I, (αk, Ek, k=1,2,…,I))
ma = I
αka = αk
Dka = Ek
k=1,2,…I
a ≥ p
No
a = a+1
Yes
I = 0
ri = 1, i = 1,2,…,p
rl = 1
l = 1,2,…,p
F = ⋂ a=1 p Da(ra)
F = φ
Yes
No
b = 1
Gb = F∘(K,0,L,0)
b = b+1
Gb = φ
Yes
No
b ≥ p
No
Yes
I = I+1
αI = α rb b
EI = Gb
aI = b
r1 ≥ m1
No
r1 = r1+1
Yes
r2 ≥ m2
No
r2 = r2+1
Yes
rp ≥ mp
No
rp = rp+1
Yes
RETURN

Section 4. Successive Approximation

The user of the routine must supply a discount factor β, an optimality tolerance $\varepsilon > 0$, a specification of the bounded convex polyhedral set Ω, the cost vectors c^a, and the subroutine Q. If the user does not supply an initial p.w. linear value function V^0, then the routine starts with $V^0 \sim \{(0,\Omega)\}$.

As described in Remark IV.2, if the method of successive approximation iterates U_* until $\|U_*^n V^0 - U_*^{n-1} V^0\| \leq (1 - \beta)\varepsilon'/\beta$ where $\varepsilon' = (1 - \beta)\varepsilon/(2\beta)$ then the policy δ such that $U_\delta V^n = U_* V^n$ is ε-optimal. Let $V^n = U_*^n V^0$. To determine $\|U_* V^n - V^n\|$ requires a fair amount of computation. However, this norm only needs to be computed once by Theorem III.2., since ε'-optimality of the cost function must be achieved with no more than $1 + \mathrm{INT}(\xi)$ iterations, where $\xi = \log(\frac{(1-\beta)\varepsilon'}{V^1 - V^0})/\log \beta$. However, it is likely that ε'-optimality will be achieved in fewer than $1 + \mathrm{INT}(\xi)$ iterations. So we compromi e with the following procedure which checks for ε'-optimality at about half of the maximum number of iterations. Compute $\|V^1 - V^0\|$. Let $J = 1 + \mathrm{INT}(\xi/2)$. Then check for ε'-optimality at iteration $J + 1$. If, at that point, ε'-optimality has not been achieved, recompute J using $\|V^{J+1} - V^J\|$ in place of $\|V^1 - V^0\|$. Check ε'-optimality next after J iterations and continue with this procedure.


START
Read $\epsilon, \beta, V^0=\{(V_i^0, E_i), i=1,2,\cdots,n\}$ (if supplied) and Q
If V^0 is not supplied, then set $V^0=\{(0,\Omega)\}$.
Set $J=1, K=0$
CALL USTAR(V^0, V', δ) $K=K+1$
$J=K$
Yes
$V^0=V'$
$V^0=\{(\alpha_i, E_i); i=1,2,\cdots,n\}$
$V'=\{(\gamma_j, F_j); j=1,2,\cdots,m\}$
Set $\omega=J=0$
$I=0$
$J=J+1$
$I=I+1$
$E_i \cap F_j = \phi$
Yes
No
Solve the linear programmes
$\omega^+ = \max(\alpha_i - \gamma_j)x$ s.t. $x \in cl(E_i \cap F_j)$
and $\omega^- = \max(\gamma_j - \alpha_i)x$ s.t. $x \in cl(E_i \cap F_j)$
where cl means closure.
Let $\omega = \max(\omega, |\omega^+|, |\omega^-|)$
$I=n$
Yes
No
$J=m$
No
Yes
A



Remarks V.2. To check that a convex polyhedral set B is non-empty, minimize a Phase I cost function on *cl* B. Range the right-hand side of those inequalities defining B which are strict.

This check provides a feasible solution to each of the two linear programmes which follow.

Also note that as we increment I for fixed J, the previous solution to the linear programmes (including the Phase I programme) remains feasible for these inequalities corresponding to F_J. Usually, F_J will have more inequalities than E_J.

Section 5. Policy Improvement

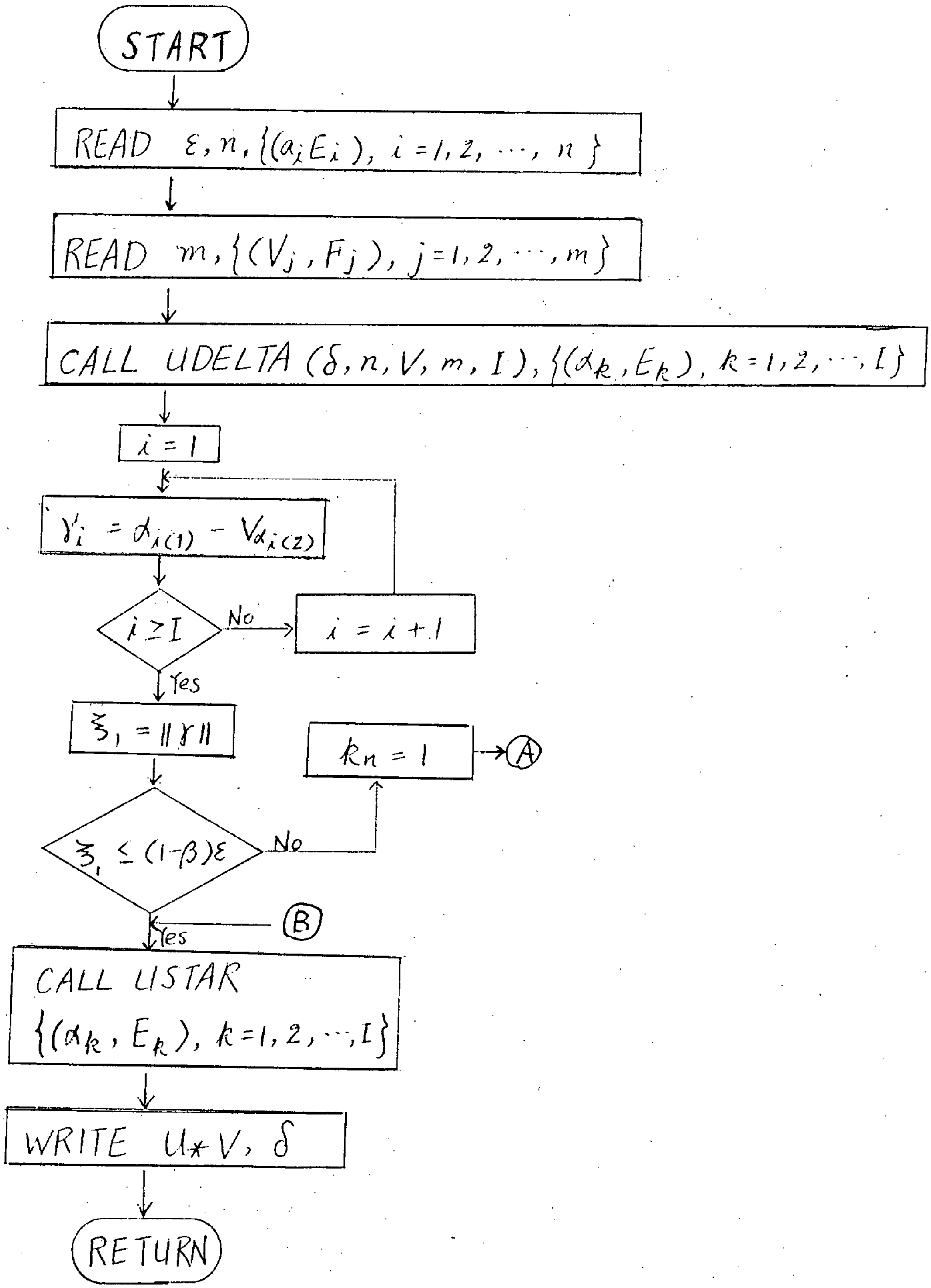
The user of this routine must specify a discount factor β, an optimality tolerance ε, a specification of the bounded convex polyhedral set Ω, the cost vectors c^a, the subroutine Q, a simple policy δ, and a p.w. linear function V such that $V \leq U_\delta V$.

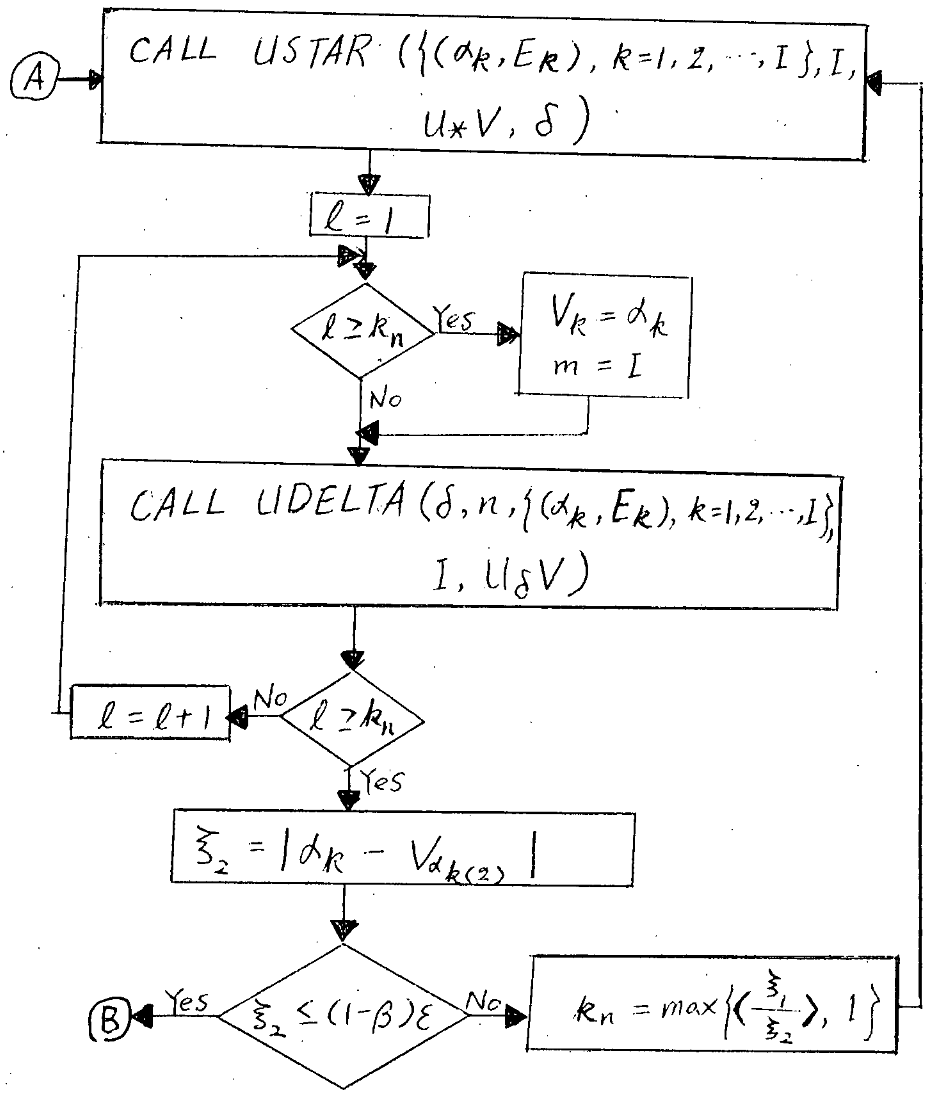
The n-th iteration of this routine starts with a simple policy δ_n and a p.w. linear function y^n, where $\delta_0 = \delta$ and $y_0 = V$. The operator U_{δ_n} is iterated some number of times, say k_n times, using the subroutine UDELTA. This provides $y^{n+1} = U_{\delta_n}^{k_n} y^n$. The policy δ_{n+1} is obtained from $U_{\delta_{n+1}} y^{n+1} = U_* y^{n+1}$ using the subroutine USTAR.

The method of choosing k_n has not been satisfactorily resolved. Recall that the larger k_n is, the larger is the step size $y^n - y^{n+1}$. The maximum step size is $y^n - U_{\delta_n}^{\infty} y^n = y^n - V^{\delta_n}$. Thus one trades off larger step size vs. fewer calls of UDELTA. In general, it seems desirable to have k_n small initially and larger as y^n converges. The following procedure has this property. Set $k_n = \text{Max}\left(\text{INT}\dfrac{\|y^1 - y^0\|}{\|y^{n+1} - y^n\|}, 1\right)$.

We compute $\|y^{n+1} - y^n\|$ each iteration and use Theorem IV.2. to check ε-optimality; i.e., δ_n is an ε-optimal policy whenever $\|y^{n+1} - y^n\| \leq (1 - \beta)\varepsilon$ and

$$V^* \leq V^{\delta_n} \leq y^{n+1}.$$


START
READ $\varepsilon, n, \{(a_i E_i), i = 1, 2, \cdots, n\}$
READ $m, \{(V_j, F_j), j = 1, 2, \cdots, m\}$
CALL UDELTA $(\delta, n, V, m, I), \{(\alpha_k, E_k), k = 1, 2, \cdots, I\}$
$i = 1$
$\gamma_i' = \alpha_{i(1)} - V_{\alpha_{i(2)}}$
$i \geq I$
No
$i = i + 1$
Yes
$\xi_1 = \|\gamma\|$
$k_n = 1$
A
$\xi_1 \leq (1-\beta)\varepsilon$
No
B
Yes
CALL USTAR
$\{(\alpha_k, E_k), k = 1, 2, \cdots, I\}$
WRITE $U_* V, \delta$
RETURN


A
CALL USTAR ({(α_k, E_k), k=1, 2, ..., I}, I, U*V, δ)
ℓ = 1
ℓ ≥ k_n
Yes
V_k = α_k
m = I
No
CALL UDELTA (δ, n, {(α_k, E_k), k=1,2,...,I}, I, U_δV)
ℓ = ℓ+1
No
ℓ ≥ k_n
Yes
ξ_2 = | α_k − V_{α_k(2)} |
B
Yes
ξ_2 ≤ (1−β)ε
No
k_n = max{⟨ξ_1/ξ_2⟩, 1}

Section 6. Subroutine Q(B, a, V) for Model 2.

The inputs to this subroutine are an action $a \in A$ and a convex polyhedral set $B \subseteq \Omega$ represented by the array $B = \{K, b; \ L, d\}$, where (K,b) has m rows and (L,d) has r rows. The subroutine has available as its data, the arrays $\{\gamma^a_{j\theta}: j = 1,2,\ldots,N; \ \theta = 1,2,\ldots,q; \ a = 1,2,\ldots,p\}$ and $\{P^a_{ij}; \ i = 1,2,\ldots,N; \ j = 1,2,\ldots,N, \text{ and } a = 1,2,\ldots,p\}$. The array $V = \{I, \ (\lambda^j; \ L^j,b^j; \ K^j,d^j), \ j = 1,2,\ldots,I\}$ is the subroutine output. The array V characterizes the p.w. linear vector-valued function $\hat{q}^a(B,\cdot)$ by $\hat{q}^a(B,x) = \lambda^j \cdot x$ for x satisfying $L^j x < b^j$ and $K^j x \leq d^j$. Note that λ^j is a matrix.

The subroutine is based on Lemma II.2., and the computation preceding the Lemma showing that

$$q^a(B,x) = \sum_{\theta \in \Phi^a(B,x)} P^a(\theta) \cdot x .$$

In this subroutine, the equation convention for describing convex polyhedral sets will be modified slightly. Each convex polyhedral set E considered will always be a subset of Ω, and hence $x \in E$ will always satisfy $\sum^N x_i = 1$. This equality will always be implicit in any description of a convex polyhedral set, even if it is not explicitly included in the list of inequalities. With this convention the set B is represented by the array $\{\hat{K},\underline{0}; \ \hat{L},\underline{0}\}$ where $\hat{K}_{ij} = K_{ij} - b_i$ and $\hat{L}_{ij} = L_{ij} - d_i$ for each i and j.

The first time the subroutine is called the matrices $P^a(\theta)$, $a = 1,2,\ldots,p$, $\theta = 1,2,\ldots,q$, must be computed. Recall from Section 3 that $P^a_{ij}(\theta) = P^a_{ji}\gamma^a_{j\theta}$. Although the matrices $P^a(\theta)$

could be input directly, the quantities P^a_{ij} and $\gamma^a_{j\theta}$ are more natural from the user's point of view.

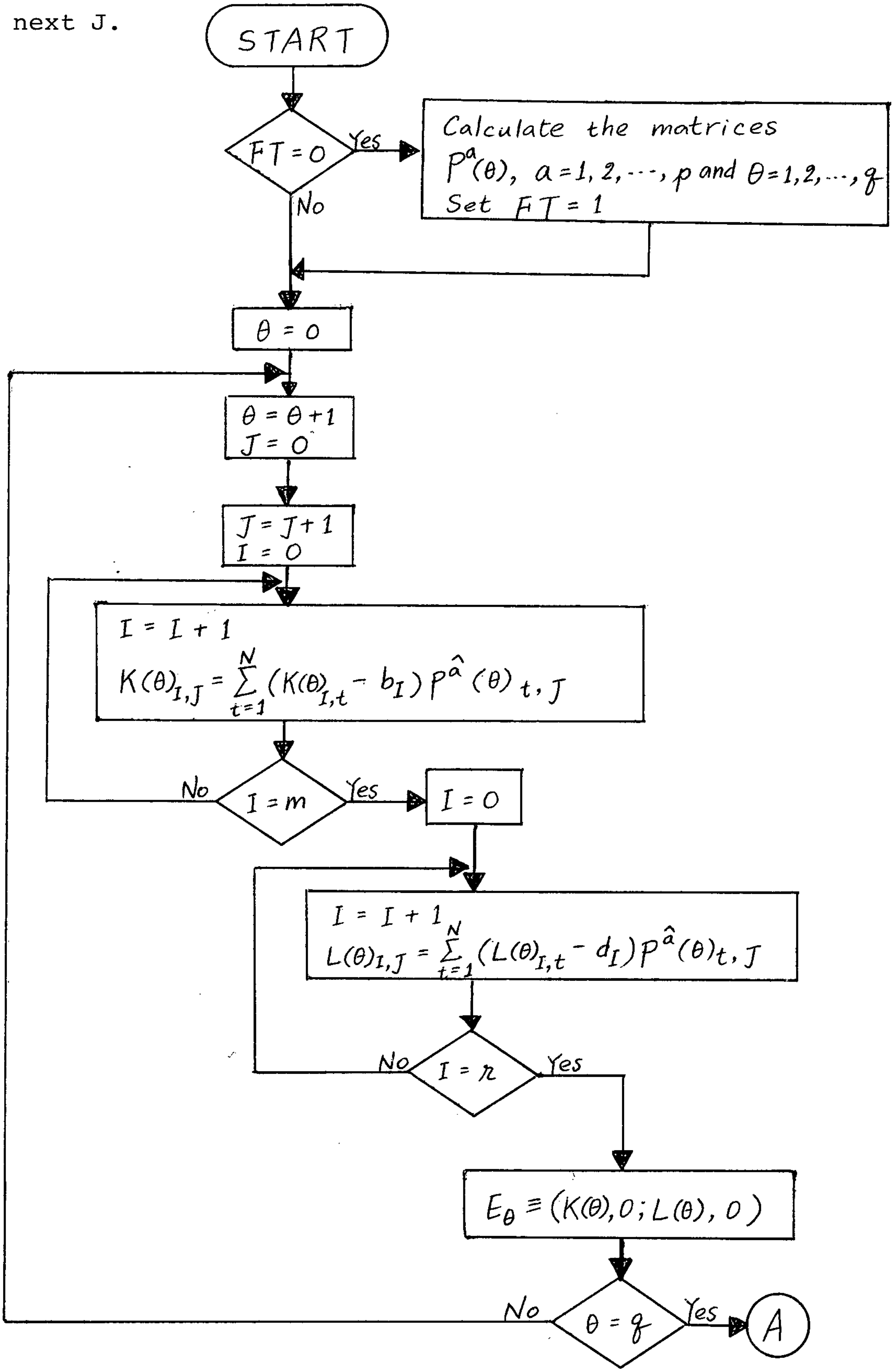
Next compute $K(\theta) = \hat{K}P^{\hat{a}}(\theta)$ and $L(\theta) = \hat{L}P^{\hat{a}}(\theta)$ for each $\theta \in Ⓗ$ and set $E_\theta = \{K(\theta),0;\ L(\theta),0\}$. The array E_θ characterizes the set E^{Ba}_θ in Lemma II.2. The index I will count the cells in the partition of V. Set I = 0.

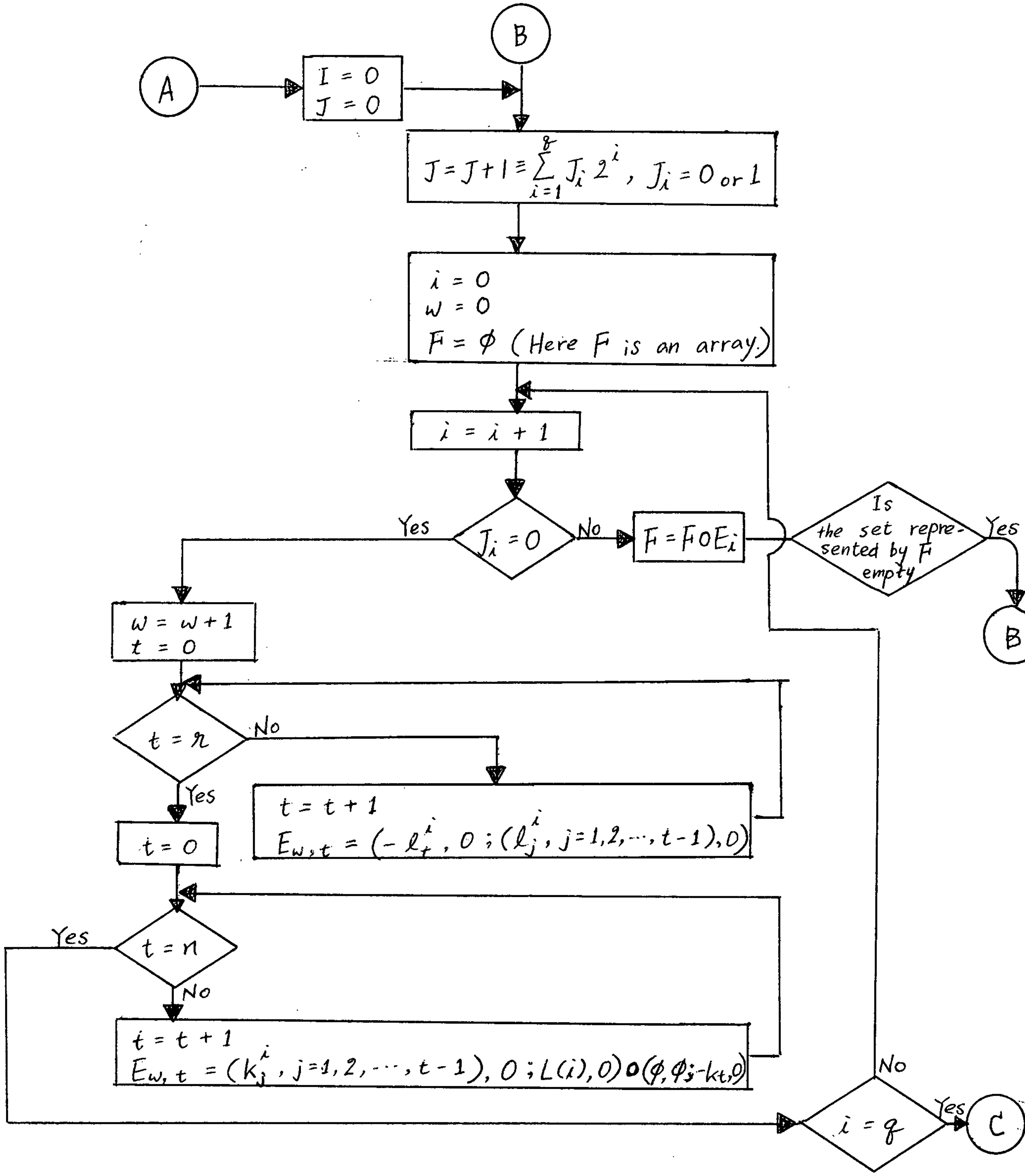
Let J step from 1 through 2^q. Let J_i be the i-th digit of J in its binary representation, i.e., $J = \sum_{i=1}^{q} J_i 2^i$, $J_i \in \{0,1\}$. Each J represents a subset ψ of Ⓗ by $\theta \in \psi$ if and only if $J_\theta = 1$. Form the array $F = \underset{\{i:J_i=1\}}{\circ} E_i$. If F is empty, then look at the next J. Otherwise calculate the matrix $R = \sum_{i=1}^{q} P^{\hat{a}}(i) \cdot J_i$. The array F corresponds to the set $\underset{\theta\varepsilon\psi}{\cap} E^{B,a}_\theta$ in Lemma II.2. The set $\underset{\theta\varepsilon\psi^c}{\cap} (E^{Ba}_\theta)^c$ is a union of convex polyhedral sets, which we now find. Let the vectors k^θ_t, $t = 1,2,\ldots,m$ and ℓ^θ_t, $t = 1,2,\ldots,r$ be the rows of $\hat{K}(\theta)$ and $\hat{L}(\theta)$, respectively. Define

$$E^{B,a}_{\theta,t} = \begin{cases} \{x:\ \ell^\theta_t x>0,\ \ell_j x\le 0,\ j=1,2,\ldots,t-1\} & 1\le t\le r \\ \\ \{x:\ L(\theta)x\le 0,\ k_t x\ge 0, \text{ and } k^\theta_j x<0,\ j=1,2,\ldots,t-r-1\} & r<t\le r+n. \end{cases}$$

Then $(E^{Ba}_\theta)^c = \bigcup_{t=1}^{t+n} E^{Ba}_{\theta t}$ and $\{E^{Ba}_{\theta t}: t = 1,2,\ldots,r+n\} = \mathcal{P}_\theta$ is a partition of E^{Ba}_θ. Let $\mathcal{P} = \prod_{\{i:J_i=0\}} \mathcal{P}_i$.

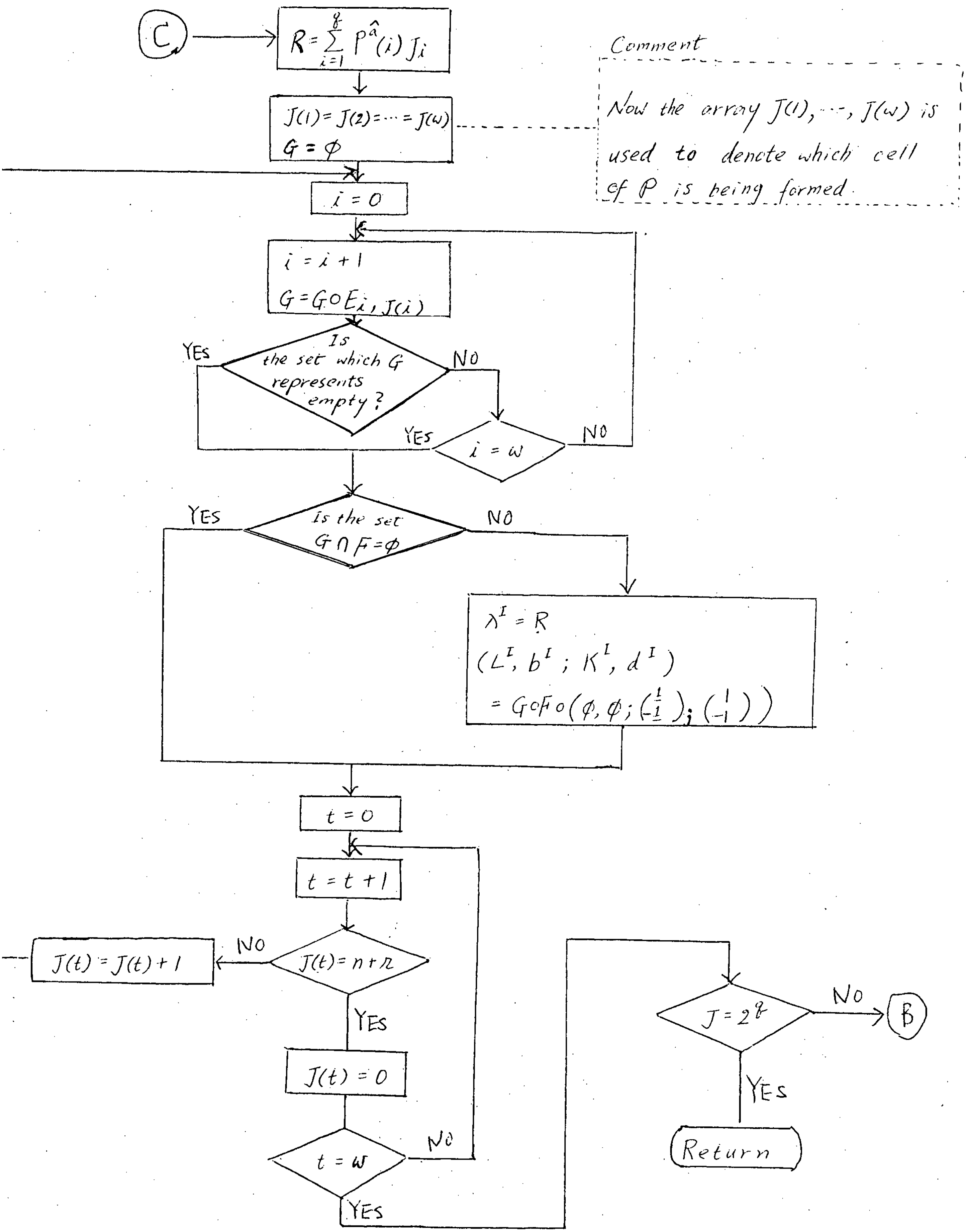
For each $G \in \mathcal{P}$ such that $F \cap G \neq \phi$, increment I by 1. Let $E'_i = \{L^I,0,K^I,d\}$ be the array representing $F \circ G$. The matrix (K^I,d) should also explicitly include the rows $(\underline{1},1)$ and $(\underline{-1},-1)$, unless the equality $\underline{1}x = 1$ is redundant. Set $\lambda^I = R$. Continue until each $G \in P$ has been considered and then proceed to the next J.





Comment

Whether or not the set $F = \phi$ is determined by solving a Phase I linear programme. Since when i is incremented the onlychange in the linear programmes is to add constraints, the L.P. should be started from the previous optimal tableau and the dual simplex algorithm used. Similar arguments apply to the following loop where $G = \phi$ is tested.


C
$R = \sum_{i=1}^{q} P^{\hat{a}}(i) J_i$
$J(1) = J(2) = \cdots = J(w)$
$G = \phi$
Comment
Now the array $J(1), \cdots, J(w)$ is used to denote which cell of P is being formed.
$i = 0$
$i = i + 1$
$G = G \circ E_{i, J(i)}$
Is the set which G represents empty?
YES
NO
$i = w$
YES
NO
Is the set $G \cap F = \phi$
YES
NO
$\lambda^I = R$
$(L^I, b^I ; K^I, d^I)$
$= G \circ F \circ (\phi, \phi ; (\frac{1}{-1}) ; (\frac{1}{-1}))$
$t = 0$
$t = t + 1$
$J(t) = n + r$
NO
$J(t) = J(t) + 1$
YES
$J(t) = 0$
$t = w$
NO
YES
$J = 2^q$
NO
B
YES
Return

BIBLIOGRAPHY

1. Aoki, M., Optimization of Stochastic Systems, Academic Press, New York, 1967.

2. Astrom, K. J., Introduction to Stochastic Control Theory, Academic Press, New York, 1970.

3. Astrom, K. J., Optimal Control of Markov Processes with Incomplete State Information, Journal of Mathematical Analysis and Applications, 10(1965), pp. 174-205.

4. Bather, ., Optimal Decision Procedures for Finite Markov Chains, Part I: Examples, Advances of Applied Probability, 5(1973), pp. 328-339.

5. Bellman, R., Introduction to the Mathematical Theory of Control Processes, Academic Press, New York, 1971.

6. Blackwell, D., Discrete Dynamic Programming, Annals of Mathematical Statistics, 33(1962), pp. 719-726.

7. Blackwell, D., Discounted Dynamic Programming, Annals of Mathematical Statistics, 36(1965), pp. 226-235.

8. Brumelle, S.L. and Puterman, M.L., On the Convergence of Newton's Method for Operators with Supports, ORC76-12, University of California, Berkeley, May 1976.

9. Chitgopekar, S., Continuous Time Markovian Sequential Control Processes, SIAM Journal on Control, 7(1969), pp. 367-389.

10. Denardo, E., Contraction Mapping in the Theory Underlying Dynamic Programming, SIAM Review, 9(1967), pp. 165-177.

11. Denardo, E., Markov Renewal Programs with Small Interest Rates, Annals Mathematical Statistics, 42(1971), pp. 477-496.

12. Denardo, E., and Mitten, L., Elements of Sequential Decision Processes, Journal of Industrial Engineering, SVIII(1967), pp. 106-112.

13. Davis, M.H.A., and Varaiya, P., Dynamic Programming Conditions for Partially Observable Stochastic Systems, SIAMS Journal on Control, 11(1973), pp. 226-261.

14. Derman, C., On Sequential Decisions and Markov Chains, Management Science, 9(1962), pp. 16-24.

15. Derman, C., Denumerable State Markovian Decision Process - Average Cost Criterion, Annals of Mathematical Statistics, 37(1966), pp. 1545-1553.

16. Doshi, B., Continuous Time Control of Markov Procession on Arbitrary State Space: Discounted Rewards, The Annals of Mathematical Statistics, 4(1976), pp. 1219-1235.

17. Dynkin, E.B., Controlled Random Sequences, Theory of Probability and its Applications, X(1965), pp. 1-14.

18. Feller, W., An Introduction of Probability Theory and Its Applications, Volume 2, Wiley, New York. 1966.

19. Fox, B.L., Markov Renewal Programming by Linear Fractional Programming, SIAM Journal of Applied Mathematics, 14(1966), pp. 1418-1432.

20. Fox, B.L., Existence of Stationary Optimal Policies for Some Markov Renewal Programs, SIAM Review, 9(1967), pp. 665-670.

21. Fox, B.L., Finite-State Approximation to Denumerable State Dynamic Programs, Journal of Mathematical Analysis and Applications, 34(1971), pp. 665-670.

22. Furukawa, N., Markovian Decision Processes with Compact Action Spaces, Annals of Mathematical Statistics, 43(1972), pp. 1612-1622.

23. Harrison, J.M., Discrete Dynamic Programming with Unbounded Rewards, Annals of Mathematical Statistics, 43(1972), pp. 636-644.

24. Haussmann, U.G., On the Optimal Long-Run Control of Markov Renewal Processes, Journal of Mathematical Analysis and Applications, 36(1971), pp. 123-140.

25. Hockstra, D., Partially Observable Markov Decision Processes with Applications, Technical Report No. 156, Department of Operations Research, Stanford University, Stanford, September, 1973.

26. Howard, R.A., Dynamic Programming and Markov Processes, Wiley, New York, 1960.

27. Jewell, W., Markov Renewal Programming: II Infinite Return Models, Operations Research, 11(1963), pp. 938-971.

28. Kakumanu, P., Nondiscounted Continuous Time Markovian Decision Process with Countable State Space, SIAM Journal on Control, 10(1972), pp. 210-220.

29. Kakumanu, P., Continuously Discounted Markov Decision Model with Countable State and Action Space, Annals of Mathematical Statistics, 42(1971), pp. 919-926.

30. Kashyap, R.L., Optimization of Stochastic Finite State Systems, IEEE, Transactions on Automatic Control, AC-11(1960), pp. 685-692.

31. Krylov, N.V., Construction of an Optimal Strategy for a Finite Controlled Chain, Theory of Probability and Applications, 10(1965), pp. 45-54.

32. Kushner, H.J., Introduction to Stochastic Control, Holt, Rinehart and Winston, New York, 1971.

33. Lembersky, M.R., On Maximal Rewards and ε-Optimal Policies in Continuous Time Markov Decision Chains, Annals of Statistics, 2(1974), pp. 159-169.

34. Lembersky, M.R., Preferred Rules in Continuous Time Markov Decision Processes, Management Science, 21(1974), pp. 348-357.

35. Li, Yu-ku, Information Structure and Optimal Policy, Computer and Information Science Research Centre, The Ohio State University, Ohio, September, 1970.

36. Lippman, S.A., Semi-Markov Decision Processes with Unbounded Rewards, Management Science, 19(1973), pp. 717-731.

37. Lippman, S.A., Maximal Average - Reward Policies for Semi-Markov Decision Processes with Arbitrary State and Action Space, Annals of Mathematical Statistics, 42(1971), pp. 1717-1726.

38. Lusternik, L.A. and Sobole , V.J., Element of Functional Analysis, Hindustan Publishing Company, Delhi, 1961.

39. Maitra, A., Discounted DynamicProgramming on Compact Metric Spaces, Sankhya, 30A(1968), pp. 211-216.

40. Martin-Lof, A., Optimal Control of a Continuous-Time Markov Chain with Periodic Transition Probabilities, Operations Research, 15(1967), pp. 872-881.

41. Miller, B.L., Finite State Continuous Time Markov Decision Processes with a Finite Planning Horizon, SIAM Journal on Control, 6(1968), pp. 266-280.

42. Miller, B.L., Finite State Continuous Time Markov Decision Processes with an Infinite Planning Horizon, Journal of Mathematical Analysis, 22(1968), pp. 552-569.

43. Miller, B.L., and Veinott, A. Jr., Discrete Dynamic Programming with a Small Interest Rate, Annals of Mathematical Statistics, 40(1969), pp. 366-370.

44. Mine, H. and Tabata, Y., On a Set of Optimal Policies in Continuous Time Markovian Decision Processes, Journal of Mathematical Analysis and Applications, 34(1971), pp. 53-66.

45. Mine, H., and Tabata, Y., A New Optimality Criterion for Discrete Dynamic Programming, Journal of Mathematical Analysis and Applications, 37(1972), pp. 118-129.

46. Mitten, L.G., "Composition Principles for Synthesis of Optimal Multi-stage Processes," Journal of Operations Research, 12(1964), pp. 610-619.

47. Morton, R., Optimal Control of Stationary Markov Processes, Advances of Applied Probability, 5(1973), pp. 18-19.

48. Osaki, S. and Mine, H., Linear Programming Algorithms for Semi-Markovian Decision Processes, Journal of Mathematical Analysis Applications, 22(1968), pp. 356-381.

49. Puterman, M.L. and Brumelle, S.L., On the Convergence of Policy Iteration in Stationary Dynamic Programming, Working Paper No. 392, Faculty of Commerce, U.B.C., June 1976.

50. Raviv, Decision Making in Incompletely known Stochastic Systems, International Journal of Engineering Science, 3(1965), pp. 119-140.

51. Rishel, R., Necessary and Sufficient Dynamic Programming Conditions for Continuous Time Stochastic Optimal Control, SIAM Journal on Control, 8(1970), pp. 559-571.

52. Rolph, J.E. and Strauch, R., A Countable Policy Set for Sequential Decision Problems, Annals of Mathematical Statistics, 43(1972), pp. 2078-2082.

53. Ross, S., Arbitrary State Markovian Decision Processes, Annals of Mathematical Statistics, 39(1968), pp. 2118-2122.

54. Ross, S., Applied Probability Models with Optimization Applications, Holden-Day, San Francisco, 1970.

55. Ross, S., Average Cost Semi-Markov Decision Processes, Journal of Applied Probability, Volume 7 (1970), pp. 649-656.

56. Rudemo, M., Doubly Stochastic Poisson Processes and Process Control, Advances of Applied Probability, 4(1972), pp. 318-338.

57. Sage and Melsa, System Identification, Academic Process, New York, 1971.

58. Sawaki, K., and Ichikawa, A., An Algorithm for Partially Observable Markov Decision Processes Over Infinite Horizon. Bulletin of Operations Research Society of Japan. 1976. September.

59. Schweitzer, P.L., Iterative Solution of the Functional Equations of Undiscounted Markov Renewal Programming, Journal of Mathematical Analysis and Applications, 39(1971), pp. 495-501.

60. Smallwood, R.D., and Sondik, E.J., The Optimal Control of Partially Observable Markov Processes Over a Finite Horizon, Operations Research, 21(1973), pp. 1071-1088.

61. Sondik, E., The Optimal Control of Partially Observable Markov Processes Over the Infinite Horizon: Discounted Costs, Department of Engineering Economic Systems, Stanford University. Stanford, June 1971. (Forthcoming in Operations Research).

62. Stone, L., Necessary and Sufficient Conditions for Optimal Control of Semi-Markov Jump Processes, SIAM Journal on Control, 11(1973), pp. 187-201.

63. Strauch, R.E., Negative Dynamic Programming, Annals of Mathematical Statistics, 37(1966), pp. 871-890.

64. Veinott, A., Discrete Dynamic Programming with Sensitive Discount Optimality Criteria, Annals of Mathematical Statistics, 40(1969), pp. 1635-1660.

65. Wald, A., Statistical Decision Functions, John Wiley & Sons, New York, 1950.

66. Walras, L., Elements d'economie politique pure, L. Corbaz, Lausanne, 1874.

67. White, C.C., Procedures for the Solution of a Finite Horizon, Partially Observable Semi-Markov Optimization Problem, Operations Research, 24(1976), pp. 348-358.