Journal of the Korean Institute
of Industrial Engineers
Vol. 18, No.1, Jun., 1992

Comparison of Control Policy Algorithms for a Optimal System Operations+

Chang Eun Kim*

Abstract

The control policy algorithm is examined and compared in this study. This research investigates a two state partially observable Markov chain in which only deterioration can occur and for which the only actions possible are to replace or to live alone. The goal of this research is to compare the computational efficiencies of control policy algorithm. One is Sondik's algorithms and the other one is jump algorithm.

1. Introduction

The past few years have witnessed and increasing interest in the development and implementation of optimal system operations for stochastically failing system. The practical need for optimal system operations has stimulated theoretical interest and has led to the development of policies that possess theoretical nevelty and practical importance. Optimal system operations have been actively studied at least as long as operation research has been a viable displine. Initial studies obtained optimum policies based only on lifetime information assuming no knowledge regarding the state of the system itself(Sasieni. [8]). The problem of con-trolling stochastic processes with incomplete state information was initially studied by Dynkin[3].

The earliest use of the Partially Observable Markov Decision Process(POMDP) model appears to be in machine inspection and replacement problem. A Markov model of optimal machine replacement and inspection due to Derman [1] is defined in the following way; the deterioration of a machine can be described by a Markov chain where the state represents the level of deterioration. However, the state of the chain at each step is unknown unless a decision is made to inspect, in which case an inspection cost is incurred. The machine may fail and a failure cost is incurred for each time step that a failure remains undetected. The decision-maker must decide when to inspect and when to replace. Derman showed that the optimal replacement rule is a control limit rule; that is,

+본 논문은 1990년도 교육부 지원에 의한 연구되었음.

*명지대학교 산업공학과.

there is a state $i \in E=\{1, 2, \cdots, L\}$ such that if the observed state k satisfies $k \geq i$, then replace the machine; and if $k<i$, do not replace, where E is the state space with State 1 denoting a new machine and State L denoting an inoperative machine.

Drake[2] developed the first explicit POMDP model. An information source is modeled as a two state sysmmertric Markov chain. The source is observed through a noisy communication medium modeled as a binary symmetric communications channel. The problem is to decode the source; that is, to decide which of two symbols are transmitted by the information source at each point in time. The information source transmitted, of course, may or may not be reflected in the actual symbol. A cost is incured if the symbol chosen as the one transmitted is not the actual symbol that was transmitted, and no cost is incurred if the proper symbol is chosen. Drake proposed a scheme that minimizes the expected cost of making decisions and noted that for certain parameter relationships the transmitted symbol can be determined with minimum error on the basis of only the last output symbol.

Kalymon[5] generalizes Derman's model by considering a stochastic replacement cost determined by a Markov chain. His cost function is a random variable which takes on a finite set of values. When the cost function is increasing according to the level of the state, the Markov chain has an increasing failure rate(IFR) which means that the probability of deterioration increases as the initial state increases if no replacement is made. His machine replacement model has a control limit policy for the finite horizon discounted cost function. He also generalized the infinite horizon ergodic chains for both the long run discounted and the long run averaged cost cases using linear programming.

Satia and Lave[9] studied a finite discrete time discounted Markovian decision process when the states are probabilistically observed. They assumed that the deterioration process is observed through a finite state probabilistic observer. They presented an implicit enumeration algorithm which optimizes the total expected discounted cost given the initial state. Their algorithm will converge to within any predetermined interval in a finite number of iterations. They also discussed applications of their model in such areas as replacement, quality control, and brand switching problems.

Rosenfield[7] considered the POMDP machine relacement problem. His model stipulates that the operator must pay an insection cost to determine the state of system with three choices at every time period; repair, inspection, and do nothing. He defined a process that has a state space consisting of pairs of nonnegative integers denoted by $(i, j)$where i is the condition of the machine and j is the number of periods which have elapsed since the machine was in i state. He proved that an optimal maintenance policy is monotonic in the following sense: the optimal policy is defined by control-limit number $F^*(i)$, $i=1, 2, \cdots, n$, which are nondecreasing in i, where, for each state (i, j), a repair is done only if $j \geq F^*(i)$ and either inspection or leave alone is optimal otherwise.. Rosenfield's goal was to establish the structure of the optimal policy and he was not concerned with developing an efficient algorithm for determining the optimal policy.

The most significant work in POMDP has been done by Sondik[10, 11] who structured the problem as a traditional Markov decision problem and then developed a special algorithm for obtaining the optimum. He defined a core process and an observation process. The core process was a Markov chain that could not be directly observed. The observation process was the sequence of state that were actually observed, and that were determined by the core process. Specifically, a matrix of probabilities R $=[r_{i\theta}]$ was define such that $r_{i\theta}$ denote the probability of observing state θ, given that the core process was in state i. Although the state of the core process cannot be known with certainty, the decision maker can obtain the probabilities of the state of the core process based on the observation process. It is these probabilities that are used to form Sondik's decision process for a POMDP. He developed a procedure, called the one pass algorithm, that was based on dynamic programming and linear programming. He used the one pass algorithm to compute the optimal policy for finite horizon POMDP's He also developed a Howard-like policy iteration algorithm to compute the optimal policy for infinite horizon POMDP's

White[12, 13] generalized the POMDP to allow for a semi-Markov core process. He extened Sondik's computational procedures to compute policies for finite horizon POMDP model with a semi-Markov core process. He also gave conditions which yield montone optimal policies where there is either perfect observability or no observability.

Kim[6] Jump algorithm is based on an invariant distribution for a continuous state Markov chain. The Markov chain is the process used by Sondik in his decision approach; however, it is shown here that the chain's invariant distribution has some special structure which allows for the development of a new algorithm. Sondik algoithms and jump algorithm will be compared to show the computational efficiency.

2. Sondik's One Pass Algorithm

Using same notation in Kim[6], Sondik formulates a dynamic programming approach to find the optimal control policy for a partially observable Markov process. He defines $V^n(w)$ as the minimum expected cost that the system can incur during the lifetime of the process if the current core probability is w, and there are n control intervals remaining before the process terminates. Then, expanding over all possible next transitions and observations yields the recursive equation:

$$V^n(w)=\min_a[wC^a+\sum_\theta P(\theta \mid w, a]V^{n-1}[T(w \mid \theta, a)]]. \tag{2.1}$$

Equation(2.1) is composed of a finite number of piecewise linear segments which allows $V^n(w)$ to be computed in a very simple fashion:

$$V^n(w)=\min_j [wb^n_j]\ 1\leq j. \tag{2.2}$$

Performing the substitution, he obtained the computational expression from Equation(2.1)

$$V^n(w)=\min_a[wC^a+\sum_\theta \min_i wP^aR^a_\theta b^{n-1}_j]. \tag{2.3}$$

Heshowed that A^n, which is the set of value of b_j^n, contains a finite number of elements. He also defined that R_j^n be the region in W where $b^n(w)$ has the vector value $b_j^n=[b_{j1}^n, b_{j2}^n]^T$; that is,

$$R_j^n=[W : V^n(w)=wb_j^n]. \tag{2.4}$$

$V^n(w)$ is now completely determined by the set A^n, and associated with each element b_j^n in A^n is a region of W. From Equation (2.3) for $w\in R_j^n$, he obtained the important recursive expression such that:

$$\begin{aligned}V^n(w) &= wb_j^n \\ &= \min_a[wC^a+\sum_\theta \min_k wP^a R_\theta^a b_k^{n-1}] \qquad (2.5) \\ &= w[C^a+\sum_\theta P^a R_\theta^{a\prime} b_{d_{j\theta}^{naj}}^{n-1}].\end{aligned}$$

where $d_{j\theta}^{naj}$ is the subscript variable k minimizing Equation(2.5). The optimal control for time n, $\delta^n(w)$, is a piecewise constant with values defined by the minimizing of Equation(2.5); that is

$$\delta_n(w)=a_j\equiv\delta_j^n \text{ for } w\in R_j^n$$

The partition R_j^n describes a refinement of R^n. For a point w, the following sets;

$$\begin{aligned}&wP^a R_\theta^a(b_{d_{j\theta}^{na}}^{n-1}-b_k^{n-1})\le 0,\ \forall\theta, \forall k, \forall a \\ &w(b_j^n-b_j^{na})\le 0,\ \forall_a\ne\delta_j^n. \qquad (2.6) \\ &\sum_i \omega(i)=1 \\ &\omega(i)\ge 0\end{aligned}$$

defines a region with R_j^n The linear programming routine is used to find this region which is the important region for the computation of b_j^n.

The one pass algorithm is a systematic way of determining A^n from A^{n-1}. It is based on the fact that the complete knowledge of A^{n-1} allows the computation of $V^n(w)$ at some point w. With this background, the algorithm is presented as follows:

Algorithm(2.7). The optimal control for each period is found using a single computational pass over the state space W for each time period the process is to operate. The complete one pass algorithm is as following:

(1) Select an initial point $w\in W$.

(2) Find $V_*^n(w)=wb^{na}$ using Euation(2.5). Insert b^{na} as the first entry in A^n.

(3) Check for table empty. If yes, STOP. Otherwise, go to next step.

(4) Select b_j^{na} from the search table. Using Equations(2.6), find all vectors b_k^n with regions R_k^n that border R_j^n.

(5) Add those vectorsb_k^n that have not been previously selected to the search table:delete b_j^n from the table. Go to step 3.

3. Sondik's Policy Iteration Algorithm

When Howard[4] studied the completely observable problem, he used the expected cost per unit time as the minimiation criterion for the infinite horizon problem. Sondik[10, 11] similarly used Howard's approach for the optimal control of partially observable Markov processes over the infinite horizon. In the completely observable problem, there are only a finite number of alternatives, and thus the policy iteration method of Markov decision theory must converge in a finite number of iterations. This is not the case with the partially observable problem because of its continuous state space which admits an uncontable number of controls. Thus, it is necessary to have a measure of closeness to the optimal control at each iteration. Therefore, Sondik developed additionally a measure of closeness to the optimal control to the policy iteration method of Markov decision theory. Some definitions from his paper are needed to illustrate his algorithm.

Definition(3.1). A real valued function $f(\cdot)$ over W is termed "piecewise linear" if it can be written $f(w)=wb_i$ all $w\in u_j\in W$, where $u_1, \cdots$, is a finite connected partition of W, and b_j is a constant vector $w\in u_{jv}$.

Definition(3.2). A partition $u=\{u_1, u_2, \cdots\}$ of W that possesses the following properties with respect to a stationary policy is said to be a Markov partition.

Property(1) All points in the sets u_j are assigned the same control alternative by δ.[i.e. if $w^1, w^2\in u_j$, then $\delta(w^1)=\delta(w^2)$].

Property(2) We define $D\delta=[w : \delta(w)$ is discontinuous at w]. D^n is defined as follows : $D^0=D\delta \cdot D^n=[w : T(w \mid \theta, \delta)\in D^{n-1}]$.

Property(3) Under the mapping $T(\cdot \mid \theta, \delta)$ all points in u_j map into the same set. The relationship between the sets u_j, and the mappings $T(\cdot \mid \theta, \delta)$ is given by the Markov mapping $\nu(j, \theta)$ such that if $w\in u_j$, then $T(w \mid \theta, \delta)\in u\nu_{(j, \theta)}$.

In order to construct u^k, the sequence of sets D^0, D^1, $\cdots$, D^k must be determined and then arranged to form the boundaries of the sets in u^k. The set D^1 is defined by Property(2) :

$$D^1=[w : T(w \mid \theta, \delta)\in D^0]. \qquad (3.3)$$

D^1 is found by reflecting a point in D^0 back through the curves $T(w \mid \theta, \delta)$. This is equivalent to writing D^1 as

$$D^1=[w=T^{-1}(w^0 \mid \theta, \delta)\in W : w^0\in D^0] \quad (3.4)$$

where T^{-1} is the inverse mapping of T and is easily calculated from

$$T^{-1}(w^0 \mid \theta, \delta)=\frac{w[P^{\delta(w)}R_\theta^{\delta(w)}]^{-1}}{w[P^{\delta(w)}R_\theta^{\delta(w)}]^{-1}\ 1}. \qquad (3.5)$$

Thus D^{n+1} is determined from D^n using the same step as above for D^1. With set D^k completed, ν is constructed by combining the sets D^n into $U_{n=0}^k D^n$ and then ordering these points, thus forming a set of intervals.

Using u^k, Sondik constructs a mapping ν that is used to approximate $V^n(w)$.

Since ν will be constructed from u^k, the interger k will be called the degree of the approximation. The mapping ν is defined as follows:

$$\text{if } T(w^j\in u_j^k \mid \theta, \delta)\in u_l^k, \text{ then } \nu(j, \theta)=l \quad (3.6)$$

Since u^k does not satisfy Porperty (2) there is some set u_j^k outputθ, and $w\in u_j^k$ such that $T(w^0 \mid \theta, \delta)9\notin u^k_{\nu(j, \theta)}$ The mapping ν is used to construct piecewise linear approximation to $V(w \mid \delta)$. The approximation to $V(w \mid \delta)$ denoted $V(w \mid \delta)$, is defined by

$$V(w \mid \delta)=wb_j, \ w\in u_j^k. \qquad (3.7)$$

where the vectors b_j are chosen to satisfy the set of linear equations.

$$b_j=C^{\delta_j}+\sum_\theta P^{\delta_j} R_\theta^{\delta_j} b_{\nu(j, \theta)} \text{ where } \delta_j=\delta(w)$$

$$\text{for } w\in u^k_j \qquad (3.8)$$

With this background, the algorithm is presented as follows:

Algorithm (3.9). The algorithm is to iterate through a succession of approximations to stationary policies, using the expected cost of each approximation policy as a basis for policy improvement.

(1) Pick an arbitrary policy, say $\delta(w)=a$, $\forall w$.

(2) For k chosen to satisfy error requirements, find the partition u^k.

(3) Construct the mapping ν from u^k from Definition (3.2).

(4) Calculate $b\delta$ and $\Psi\delta$ from $\Psi\delta 1+b\delta=P\delta b\delta+C\delta$..

(5) Find the policy $\delta_1(w)$ which minimizes

$wC^a+\sum P\{\theta \mid w, a\}V[T(w \mid \theta, \delta) \mid \delta]$

over a, and where $V(w \mid \delta)=wb_{\nu(w)}$

(6) Evauate $\Psi\delta-\Psi\delta$ from S(w) where.

$S(w)=[\Psi\delta+wb_{\nu(w)}]-[wC^{\delta1(w)}+\sum_{\theta}P\{\theta \mid w, \delta_1\}V[T(w \mid \theta, \delta) \mid \delta]$

and $\min S(w) \leq \Psi\delta-\Psi\delta^* \leq \max S(w)$

(7) If $|\Psi\delta-\Psi\delta^*| < \varepsilon$then Stop; the optimal policy is δ, otherwise, return to Step 2 with δ replaced by δ_1.

4. Comparison between Algorithms

The optimal policy on the basis of the average long-run cost can be found at least conceptually by using the Sondik's one pass algorithm. But this approach definitely requires a lot of computation time. Therefore, we make a comparison between the new algorithm and Sondik's policy iteration algorithm.

Sondik's policy iteration algorithm needs a matrix inversion code which requires that a significant amount of computation time depend ona size of $P\delta$. The new algorithm is a very simple structure and need a search method.

In order to compare the computational efficiency of the two algorithms, sample problems on the following data sets will be used. Sample data sets include two sets of cost data, three transition matrices, and two observation matrices. These sample data sets provide 12 sample problems. All programs with FORTRAN code were run on Amdahl 5850. Execution time of the two algorithms will be compared as a measure of computational efficiency.

Sample data sets

Cost data set

set 1 : $C^1=\begin{bmatrix} 3 \\ 9 \end{bmatrix}$ $C^2=\begin{bmatrix} 5 \\ 11 \end{bmatrix}$

set 1 : $C^2=\begin{bmatrix} 1 \\ 3 \end{bmatrix}$ $C^2=\begin{bmatrix} 4 \\ 6 \end{bmatrix}$

Transition matrix data set

set 1 : $P^1=\begin{bmatrix} .9 & .1 \\ 0 & 0 \end{bmatrix}$

set 2 : $P^1=\begin{bmatrix} .8 & .2 \\ 0 & 1 \end{bmatrix}$

set 3 : $P^1=\begin{bmatrix} .7 & .3 \\ 0 & 1 \end{bmatrix}$

Observation matrix data set

set 1 : $R^1=\begin{bmatrix} .9 & .1 \\ .2 & .8 \end{bmatrix}$

set 2 : $R^1=\begin{bmatrix} .8 & .2 \\ .3 & .7 \end{bmatrix}$

The following Table 1 shows how 12 sample problems are combined from the sample sata sets:

〈Table 1〉. Combinations of Sample Data Sets

Run No.	Cost	Observation	Transition
1	1	1	1
2	1	1	2
3	1	1	3
4	1	2	1
5	1	2	2
6	1	2	3
7	2	1	1
8	2	1	2
9	2	1	3
10	2	2	1
11	2	2	2
12	2	2	3

The run results of 12 sample problems in Table 1 are summarized as follows:

〈Table 2〉. Summary of Sample Runs

Run	δ^1	$\Psi\delta^1$	1*	δ^2	$\Psi\delta^2$	2*
1	.03, .47	3.95	.09	.14	3.93	.03
2	.07, .67	4.55	.07	.08	4.55	.02
3	.00, .09	4.90	.04	.02	4.90	.02
4	.07, .28	4.07	.13	.12	4.07	.42
5	.00, .09	4.60	.07	.08	4.60	.09
6	.00, .14	4.90	.07	.02	4.90	.02
7	.54, .90	1.64	.14	.57	1.65	.85
8	.07, .67	2.03	.07	.62	2.01	.02
9	.12, .77	2.29	.08	.66	2.29	.02
10	.40, .65	1.74	.17	.50	1.74	.42
11	.56, .58	2.11	.12	.58	2.11	.07
12	.60, .70	2.38	.12	.64	2.38	.03

Note : 1* represents the computation time of the jump algorithm as defined in Algorithm[6] and 2* represents the computation time of the Sondik's policy iteration algorithm as defined in Algorithm(3.9). δ^1 is an optimal control interval for the jump Algorithm and δ^2 is an optimal control limit for Sondik's policy iteration algorithm(3.9)

From the results of these 12 sample runs in Table 1, we have made the following observations:

(1) The jump algorithm is faster than Sondik's policy iteration algorithm in sample runs 4,7, and 10.

(2) The variation of computation time for the jump algorithm is much smaller than Sondik's policy iteration algorithm.

REFERENCES

[1] Derman, C. 1970. ***Finite State Markovian Decision Process.*** Academic Press, New York.

[2] Drake, A. W. 1962 Observation of a Markov process Through a Noisy Channel. Ph. D. Dissertation, M.I.T., Cambridge, Mass.

[3] Dynkin, e. 1965. Controlled Random Sequences. Theory Porb. Appl. 10, 1-14.

[4] Howard, R. 1971. Dynamic Programming and Markov process, Wiley, New York.

[5] Kalymon, B.A. 1972, Machine Replacement with Stochastic Costs. Mgmt. Sci. 18, 288-298.

[6] Kim, C.E. 1990. System Replacement Policy for a Partially Observable Markov Decision Process Model. J. Korean Institute of I. E. 16, 1-9.

[7] Rosenfield, D. 1976, Markovian Deterioration with Uncertain Information. Opns. Res. 24, 141-155.

[8] Sasieni, M. W., A. Yaspen and L. Friedman. 1959. Operations Research Methods

and Problems, John-Wiley, New York.

[9] Satia, J., and R. Lave. 1973, Markovian Decision processes with Uncertain Transition Probalbilities. Opns. Res. 21, 728-740.

[10] Sondik. E. 1971. The Optimal Control of Partially Observable Markov Processes. Ph.D. Dissertation, Stanford University.

[11] Sondik. E. 1978. The Optimal Control of Partially Observable Markov Processes Over the Infinite Horizon : Discounted Costs. Opns. Res. 26, 282-304.

[12] White, C. 1976. Procedures for the Solution of a Finite Horizon, Partially Observed, Semi-Markov Optimization Problem. Opns. Res. 24. 348-358.

[13] White, C. 1977. A Markov Quality Control Process Subject to Partial Observation. Mgmt. Sci. 23, 843-852.