Annals of Operations Research 32(1991)215–230

A SURVEY OF SOLUTION TECHNIQUES FOR THE PARTIALLY OBSERVED MARKOV DECISION PROCESS*

Chelsea C. WHITE, III
Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, Michigan 48109-2117, USA

Abstract

We survey several computational procedures for the partially observed Markov decision process (POMDP) that have been developed since the Monahan survey was published in 1982. The POMDP generalizes the standard, completely observed Markov decision process by permitting the possibility that state observations may be noise-corrupted and/or costly. Several computational procedures presented are convergence accelerating variants of, or approximations to, the Smallwood–Sondik algorithm. Finite-memory suboptimal design results are reported, and new research directions involving heuristic search are discussed.

1. Introduction

This paper presents a survey of computational algorithms for the partially observed Markov decision process (POMDP) having finite state, action, and observation sets. Emphasis is placed on algorithmic developments that have occurred since the publication of Monahan's survey of the POMDP [11].

The (completely observed) Markov decision process (MDP) is a well-studied model of sequential decision making under uncertainty that has been applied in a variety of significant real-world settings; see [27] for a recent survey of the MDP, and [28,29] for surveys of its application. By completely observed, we mean that the decision maker has access to the exact value of the current state of the system without charge. Many situations exist, however, where such an assumption is invalid. For example, portions of the state value may be inaccessible (e.g. the state of a machine might be comprised of the state of various internal and unobservable components of the machine), sensors used to measure the state may give noise-corrupted readings, and/or exact state observations may be costly. The POMDP generalizes the MDP to include such situations, making the POMDP a particularly robust modeling tool for the control of a variety of discrete event dynamic systems.

The applicability of the POMDP is limited in two ways. First, the POMDP is very data intensive. White and White [27] survey adaptive control techniques for the MDP and techniques for dealing with MDPs having parameter values described

*This research was supported by NSF Grant ECS-8708183 and ARO Contract DAAG-29-85-K0089.

by set inclusion. Such techniques, extended to the POMDP, are likely to mollify this limitation (see also [6]).

The second limitation is that the solution of the POMDP model of most realistic problems can require substantial computational effort. This limitation, and the potential modeling applicability of the POMDP model, serve as the motivation for the study of numerical solution techniques for the POMDP and hence for this survey.

This survey is organized as follows. Section 2 presents the formulation of the POMDP that will be of interest throughout the paper. Sections 3 and 4 present optimality equations for two different sufficient statistics for the POMDP. Properties of an important operator associated with the second sufficient statistic are listed in section 5. The Smallwood–Sondik (SS) algorithm is described in section 6. We present a computationally attractive variant of the SS algorithm and a procedure to accelerate the Monahan variant of the SS algorithm. In section 7, we present three variants of the SS algorithm that have been shown to significantly accelerate its convergence. These variants are based on convergence-enhancing techniques that have proven useful for the MDP. Approximation schemes are discussed in section 8, primarily a recently developed approximation procedure due to Lovejoy. Section 9 considers a finite-memory suboptimal design technique, the development of which was inspired by results due to Platzman. The use of artificial intelligence concepts found in the heuristic search literature provides directions for research discussed in section 10. We present conclusions in section 11.

2. Problem formulation

We now present a formulation of the POMDP. Let $\{s(t), t = 0, 1, \ldots\}$, $\{z(t), t = 1, 2, \ldots\}$, and $\{a(t), t = 0, 1, \ldots\}$ be the *state, observation,* and *action* processes, respectively. The state space S, the observation space Z, and the action space A are each assumed to be finite. These three processes are assumed to be related by the given, stage independent conditional probabilities

$$p_{ij}(z, a) = P[z(t+1) = z, s(t+1) = j \mid s(t) = i, a(t) = a],$$

where $P(z, a) = [p_{ij}(z, a)]$. Note that

$$P[s(t+1) = j \mid s(t) = i, a(t) = a] = \sum_z p_{ij}(z, a),$$

$$P[z(t+1) = z \mid s(t+1) = j, s(t) = i, a(t) = a] = p_{ij}(z, a) / \sum_z p_{ij}(z, a),$$

which are often referred to as the transition and observation probabilities, respectively. Thus, the above notation, identical to that used in [12,15], represents a slight generalization of the usual notation associated with the POMDP.

Let $T \leq \infty$ represent the number of stages, or decision epochs, in the planning horizon. Let $\bar{d}(t) = \{z(t), \ldots, z(1), a(t-1), \ldots, a(0)\}$ for $t \geq 1$, $\bar{d}(0) = \varnothing$, the null set, and $x(0) = \{x_i(0), i \in S\}$, where $x_i(0) = P[s(0) = i]$. Thus, $\bar{d}(t)$ is the collection of all past and present observations and all past actions at stage t, and $x(0)$ is the a priori probability mass vector. We call $\{d(t), t = 0, 1, \ldots\}$ the *data* process, where $d(t) = \{\bar{d}(t), x(0)\}$; $d(t)$ represents the data available to the decision maker on which to base action selection at stage t. A *policy* at stage t is therefore a function $\delta(t)$: $\{d(t)\} \rightarrow A$. A *strategy* π is an ordered sequence of policies, i.e. $\pi = \{\delta(0), \delta(1), \ldots, \delta(T-1)\}$ for the finite horizon problem and $\pi = \{\delta(0), \delta(1), \ldots\}$ for the infinite horizon problem.

We let $r(i, a)$ be the reward received at stage $t < T$, given $s(t) = i$ and $a(t) = a$. If the problem horizon is finite, then we assume a terminal reward $\bar{r}(i)$ is accrued at the end of the planning horizon, given $s(T) = i$. Both $r(\cdot, \cdot)$ and $\bar{r}(\cdot)$ are real-valued functions.

The criteria that we will consider are

$$E_{x(0)}\left\{\sum_{t=0}^{T-1} \beta^t r[s(t), a(t)] + \beta^T \bar{r}[s(T)]\right\}$$

for the finite horizon case and

$$E_{x(0)}\left\{\sum_{t=0}^{\infty} \beta^t r[s(t), a(t)]\right\}$$

for the infinite horizon case, where E_x is the expectation operator, conditioned on probability mass vector x, and $\beta \geq 0$ is the discount factor. We will assume $\beta < 1$ for the infinite horizon case in order to ensure that the latter criterion is well defined. The problem objective is to determine a strategy that maximizes the criterion of interest, with respect to the set of all strategies. Primary interest will be in determining optimal strategies dependent on $x(0)$ for all values of $x(0)$ in $X = \{x \geq 0 : x\mathbf{1} = 1\}$, where $y\mathbf{1} = \sum_i y_i$, rather than those dependent on a given specific value of $x(0)$.

3. An optimality equation dependent on $d(t)$

We now present a recursive procedure and related results based on dynamic programming for solving the finite horizon POMDP. Justification of these results can be found in [3]. Let $v_t[d(t)]$ be the optimal expected total discounted reward to be accrued from stage t until the end of the problem horizon, given data $d(t)$. The function v_t can be described in terms of v_{t+1} by the following recursive equation:

$$v_t[d(t)] = \max_{a \in A} \left\{ \sum_{i \in S} r(i,a) P[s(t) = i \,|\, d(t)] \right.$$

$$\left. + \beta \sum_{z \in Z} P[z(t+1) = z \,|\, d(t), a(t) = a] v_{t+1}[d(t+1)] \right\}, \tag{1}$$

where $d(t+1) = \{d(t), z, a\}$ and where the boundary condition is

$$v_T[d(T)] = \sum_{i \in S} \bar{r}(i) P[s(T) = i \,|\, d(T)].$$

A policy $\delta(t)$ is optimal if and only if it achieves the maximum in eq. (1), for all values of $d(t)$. A strategy is optimal if and only if it is composed of optimal policies.

Note that $d(t) \in Z^t \times A^t \times X$, the cardinality of which grows geometrically in t. Hence, if T is large, determining $\{v_0, \ldots, v_{T-1}\}$, given v_T, can be very computationally demanding. We therefore seek a more computationally attractive representation of $d(t)$, which in part has motivated the results to be presented next.

4. An optimality equation dependent on $x(t)$

Let $x(t) = \{x_i(t), i \in S\}$, where $x_i(t) = P[s(t) = i \,|\, d(t)]$, and call $\{x(t), t = 0, 1, \ldots\}$ the *information* process. We note that for all t, $x(t) \in X$, and hence the state space of the information process is stage invariant. The information process is a controlled Markov process in that there exists a function λ such that $x(t+1) = \lambda[z(t+1), x(t), a(t)]$, where

$$\lambda(z, x, a) = xP(z, a)/\sigma(z, x, a),$$

$$\sigma(z, x, a) = xP(z, a)\mathbf{1},$$

and

$$[xP]_j = \sum_i x_i p_{ij}$$

for $P = \{p_{ij}\}$. Note that λ is simply a representation of Bayes' rule and that $\sigma[z, x(t), a] = P[z(t+1) = z \,|\, d(t), a(t) = a]$.

We observe that $v_T[d(T)]$ depends on $d(T)$ only through $x(T)$; i.e.

$$v_T[d(T)] = v_T[x(T)] = x(T)\bar{r} = \sum_i x_i(T)\bar{r}(i).$$

Assume v_{t+1} depends on $d(t+1)$ only through $x(t+1)$. It is then easily shown that $v_t = Hv_{t+1}$, where

$$[Hv](x) = \max \left\{ xr(a) + \beta \sum_z \sigma(z, x, a) v[\lambda(z, x, a)] : a \in A \right\}.$$

Thus, v_t depends on $d(t)$ only through $x(t)$. Note also that the maximum in Hv_{t+1} is attained as a function of $d(t)$ only through $x(t)$. A simple induction argument verifies that these results hold for all t.

5. Properties of H

The operator H has several interesting and useful properties. Let V^X be the set of all bounded, real-valued functions on X having supremum norm $\| v \| = \sup\{ | v(x) | : x \in X\}$. We remark that $(V^X, \| \cdot \|)$ is a Banach space. Let Δ be the set of all functions $\delta: X \to A$. Define the operator $H_\delta : V^X \to V^X$ as

$$[H_\delta v](x) = xr[\delta(x)] + \beta \sum_z \sigma[z, x, \delta(x)] v[\lambda[z, x, \delta(x)]],$$

for $\delta \in \Delta$, and note that $H : V^X \to V^X$ is such that $Hv = \sup_\delta H_\delta v$. It is shown in [3] and elsewhere that for the infinite horizon ($T = \infty$ and $\beta < 1$) case:

(1) The operators H_δ and H are contraction mappings having modulus β, guaranteeing the existence of unique fixed points v_δ^* and v^*, respectively.

(2) The real number $v_\delta(x)$ represents the expected total discounted reward to be accrued over the infinite horizon by the stationary strategy $\pi = \{\delta, \delta, \dots\}$, given a priori probability mass vector x. Similarly, $v^*(x)$ represents the optimal expected total discounted reward to be accrued over the infinite horizon, given a priori probability mass vector x.

(3) Let the sequences $\{v_{\delta n}\}$ and $\{v_n\}$ be defined as $v_{\delta n+1} = H_\delta v_{\delta n}$ and $v_{n+1} = Hv_n$. Then,

$$\lim_{n \to \infty} \| v_\delta^* - v_{\delta n} \| = 0, \quad \lim_{n \to \infty} \| v^* - v_n \| = 0,$$

given that $v_{\delta 0} \in V^X$ and $v_0 \in V^X$.

(4) There exists a stationary optimal strategy for the infinite horizon problem, and the stationary strategy $\{\delta, \delta, \dots\}$ is optimal if and only if $H_\delta v^* = Hv^*$.

(5) Assume $v \in V^X$ is piecewise-linear and convex (pwl&c), or equivalently, assume there exists a finite set Γ such that $v(x) = \max\{x\gamma : \gamma \in \Gamma\}$ for all $x \in X$. Then Hv is also pwl&c.

6. Smallwood–Sondik algorithm

The Smallwood–Sondik (SS) algorithm [18] is a successive approximations approach for solving the POMDP and as such is based on determining Hv from v. See [19] for a related policy iteration algorithm for the infinite horizon case. The SS algorithm makes extensive use of the fact that H preserves pwl&c. Let Γ_n be such

that $v_n(x) = \max\{x\gamma: \gamma \in \Gamma_n\}$, and note that $\Gamma_0 = \{\bar{r}\}$. The SS algorithm determines Γ_{n+1} from Γ_n. It is therefore appropriate to consider the process of determining Γ', given Γ, where $v(x) = \max\{x\gamma: \gamma \in \Gamma\}$ and $[Hv](x) = \max\{x\gamma' : \gamma' \in \Gamma'\}$. Note that

$$\begin{aligned}
[Hv](x) &= \max\{xr(a) + \beta \sum_z \sigma(z,x,a) v[\lambda(z,x,a)] : a \in A\} \\
&= \max\{xr(a) + \beta \sum_z \sigma(z,x,a) \max\{\lambda(z,x,a)\gamma : \gamma \in \Gamma\} : a \in A\} \\
&= \max\{xr(a) + \beta \sum_z \{xP(z,a)\gamma : \gamma \in \Gamma\} : a \in A\} \\
&= \max\{xr(a) + \beta \sum_z xP(z,a)g(z,x,a) : a \in A\},
\end{aligned}$$

where $g : Z \times X \times A \to \Gamma$ is any function such that

$$xP(z,a)[g(z,x,a) - \gamma] \geq 0$$

for all $\gamma \in \Gamma$. We remark that in this setting

$$\Gamma' = \cup_a \{r(a) + \beta \sum_z P(z,a)g(z,x,a) : x \in X\}.$$

The fact that the function g is a function of $x \in X$ can significantly complicate the determination of Γ'.

An approach to determining Γ' from Γ (which has been attributed to Monahan) that avoids dealing directly with the issue of determining the function g is as follows. First, determine the set

$$G = \cup_a \{r(a) + \beta \sum_z P(z,a)\gamma^z : \gamma^z \in \Gamma\}.$$

Note that $[Hv](x) = \max\{x\gamma: \gamma \in G\}$. However, G is a large set compared to Γ ($^{\#}G = {}^{\#}A(^{\#}\Gamma ** {}^{\#}Z)$, where $^{\#}G$ is the cardinality of the set G and $A ** B = A^B$) and may contain many unnecessary elements. Second, eliminate as many elements in G as possible to obtain Γ'.

A direct linear programming approach (see [18] for further discussion) can be used to reduce the number of elements in G. Consider the following sufficient condition for constructing Γ' from G: Choose $\xi \in G$. If $u^* = 0$, then add ξ' to Γ', where

$$\begin{aligned}
& u^* = \text{maximum } u \\
\text{subject to: } & u \leq x(\xi' - \xi) \qquad \forall\, \xi \in G, \\
& x \in X.
\end{aligned}$$

We note that this linear program has ${}^{\#}S + 1$ variables and ${}^{\#}A({}^{\#}\Gamma ** {}^{\#}Z)$ constraints and that ${}^{\#}A({}^{\#}\Gamma ** {}^{\#}Z)$ linear programs must be run in order to determine Γ' in this manner.

Lark [8] has determined another procedure for determining Γ' from elements in G. Let $e_s \in X$ have 1 as its sth entry.

Step 0. Initialization.

(i) Set $\Gamma' = \emptyset$.

(ii) For each $s = 1, \ldots, {}^{\#}S$, find $\gamma_s^* \in G$ such that

$$e_s(\gamma_s^* - \gamma') \geq 0 \quad \forall\, \gamma' \in G.$$

Remove γ_s^* from G and place in Γ'.

Step 1. If $G = \emptyset$, then stop; Γ' has been determined. If $G \neq \emptyset$, then go to step 2.

Step 2. Select $\gamma^* \in G$. Determine u^*, where

$$u^* = \text{maximum } u$$

$$\text{subject to: } u \leq x(\gamma^* - \gamma') \quad \forall\, \gamma' \in \Gamma',$$

$$x \in X.$$

Let $x^* \in X$ be a vector that causes the above maximum to be attained. If $u^* < 0$, then remove γ^* from G and discard. If $u^* \geq 0$, then search G for the element γ'' such that

$$x^*(\gamma'' - \gamma') \geq 0 \quad \forall\, \gamma' \in G.$$

Remove γ'' from G and place in Γ'. Go to step 1.

With respect to the above two linear programs, we note that iterations in each can be halted if the criterion value goes negative. Importantly, we also note that the latter linear program has ${}^{\#}S + 1$ variables and ${}^{\#}\Gamma'$ constraints, generally ${}^{\#}\Gamma' \ll {}^{\#}A({}^{\#}\Gamma ** {}^{\#}Z)$, and hence we conjecture that the second procedure for determining Γ' from G will be significantly less computationally intensive than the first procedure. A preliminary computational analysis supports this conjecture.

We now present a procedure that deals directly with the issue of determining the function g, thereby possibly avoiding having to construct the entire set G. This procedure is a minor variant of the SS algorithm presented in [18]. Other such variants of the SS algorithm can be found in [4].

We say that a subset $X' \subseteq X$ having a nonempty interior is *g-invariant* if $g(z, x, a) = g(z, x', a)$ for all $x, x' \in X'$ and for all $(z, a) \in Z \times A$. Note that the set of all g-invariant subsets in X is a partition of X (on all but a set of Lebesgue measure zero). Let X' be g-invariant. Then, for all $x \in X'$,

$$[Hv](x) = \max\{x\alpha(a) : a \in A\},$$

where $\alpha(a) = r(a) + \beta \sum_z P(z, a)g(z, x, a)$ and where $g(z, x, a)$ is constant for all $x \in X'$. We now present a two-level approach for determining Γ'.

Level 1. Perform a sweep of the g-invariant subsets in X.

Level 2. Within each g-invariant subset $X' \subseteq X$, if $a' \in A$ is such that there is an $x' \in X'$ such that

$$x'[\alpha(a') - \alpha(a)] \geq 0 \qquad \forall\, a \in A,$$

then add $\alpha(a')$ to Γ'.

We remark that if $u^* \geq 0$, then the above condition holds, where

$$u^* = \text{maximum } u$$

$$\text{subject to: } u \leq x[\alpha(a') - \alpha(a)] \qquad \forall\, a \in A,$$

$$x \in X'.$$

This linear program has ${}^{\#}S + 1$ variables and ${}^{\#}A({}^{\#}Z{}^{\#}\Gamma + 1)$ constraints, where ${}^{\#}Z{}^{\#}A{}^{\#}\Gamma$ is the number of constraints describing X'. Within each g-invariant subset, this linear program must be run ${}^{\#}A$ times.

Level 1 proceeds as follows. Arbitrarily select $x' \in X$ and determine $g(\cdot, x', \cdot)$. Use the level 2 linear programs to identify the hard constraints on the g-invariant subset described by $g(\cdot, x', \cdot)$. Each hard constraint corresponds to another g-invariant subset. For example, assume that $xP(z', a')[g(z', x', a') - \gamma']$ is a hard constraint on the g-invariant subset X'. Then there is a g-invariant subset in X which contains a point x'' such that $g(z, x', a) = g(z, x'', a)$ for all $(z, a) \in Z \times A$ except that $g(z', x'', a') = \gamma'$. (We remark that it is not necessary to know the point x'' in order to describe this new g-invariant subset.) Proceed to consider all g-invariant subsets in X (which is similar to the one-pass procedure described in [18]).

7. Accelerating the SS algorithm, $T = \infty$ case

As has been stated earlier, v_n converges to v^*, given $v_0 \in V^X$, indicating that the SS algorithm can be used to solve, at least approximately, the infinite horizon case. However, this convergence can be slow and often can be accelerated significantly. White and Scherer [24] present three approaches that have been shown to reduce CPU time until convergence, relative to the procedure of determining $\{v_n\}$. We now outline these three approaches.

Approach 1. Approach 1 is based on reward revision [26]. The intent of this approach is to find an operator G^K that has the same fixed point as H and that requires

fewer operations to achieve convergence of the sequence $\{w_n\}$, where $w_{n+1} = G^K w_n$, than is required by the sequence $\{v_n\}$. The integer K is a design parameter, the selection of which is discussed in [24] for the POMDP. The guiding motivation in the construction of the operator G^K, which is given below, is that the solution of a completely observed MDP is almost invariably easier to determine than the solution of its partially observed counterpart. Let $G^0 u = u$, $G^k u = \bar{H}(u, G^{k-1}u)$,

$$\bar{H}(u, v) = \sup_{\delta} \bar{H}_\delta(u, v),$$

$$\bar{H}_\delta(u, v)(x) = \bar{r}[x, \delta(x), u] + \bar{\beta}\, x\, \bar{P}[\delta(x)]\tilde{v},$$

$$\bar{r}(x, a, u) = xr(a) + \beta \sum_z \sigma(z, x, a) u[\lambda(z, x, a)] - \bar{\beta}\, x\, \bar{P}(a)\tilde{u},$$

and where $\tilde{u} = \{u(e_s), s \in S\}$. We think of the nonnegative scalar $\bar{\beta}$ and the stochastic matrix $\bar{P}(\cdot)$ as design parameters.

Let $u^k = G^k u$, $k = 1, \ldots, K$. We observe that $G^1 = H$. Thus, the initial step in determining $G^K u$, given u, is identical to determining Hu. Note that determination of u^K only requires knowledge of $\tilde{u}^{K-1}$. Therefore, determining $G^K u$, given u, requires the following steps:

1. Determine $\bar{r}(\cdot, \cdot, u)$, which requires determination of Hu.
2. Determine $\tilde{u}^{k+1}$, given $\tilde{u}^k$, for $k = 1, \ldots, K-1$.
3. Determine u^K, given $\tilde{u}^K$.

Details of these three steps and conditions on $\bar{\beta}$ and $\bar{P}(\cdot)$ that guarantee that the sequence $\{w_n\}$ converges to v^* are given in [24].

We observe that step 2 represents $K-1$ successive approximation iterations associated with a completely observed MDP. The resulting $\tilde{u}^K$ is then used in step 3 to "adjust" the vectors in Γ', where Γ' is such that $[Hu](x) = \max\{x\gamma: \gamma \in \Gamma'\}$. This vector adjustment represents the key to accelerating the convergence of $\{w_n\}$.

Approach 2. Approach 2 is a variation of approach 1. The essential difference is that instead of performing $K-1$ successive approximation iterations on $\tilde{u}^1$ using the operator

$$\max\{\bar{r}(e_i, a, u) + \bar{\beta} \sum_j \bar{p}_{ij}(a) v(e_j) : a \in A\},$$

we use the operator

$$\bar{r}(e_i, a', u) + \bar{\beta} \sum_j \bar{p}_{ij}(a') v(e_j),$$

where $a' = \delta(u)(e_i)$ and where $\delta(u) \in \Delta$ is such that $H_{\delta(u)} u = Hu$. Observe that the first operator requires $\#A$ times as many operations per iteration as does the second operator. Thus, approach 2, which combines reward revision and a modified policy

iteration [16,17] appears to be more numerically attractive per iteration than does approach 1. This conjecture has been supported by numerical testing. Theoretical results associated with approach 2 are not nearly as well developed as those for approach 1; see [24] for details.

Approach 3. Approach 3 is a generalization of the Bertsekas extrapolation [3] to the POMDP. Let $H^{k+1}v = H(H^k v)$, where $H^1 = H$. Then, by proposition 3 [24],

$$(H^k v)(x) + c_k \leq (H^{k+1}v)(x) + c_{k+1}$$

$$\leq v^*(x) \leq (H^{k+1}v)(x) + \bar{c}_{k+1} \leq (H^k v)(x) + \bar{c}_k,$$

where for all $k = 0, 1, \ldots,$

$$c_k = \beta \inf\{(H^k v)(x) - (H^{k-1}v)(x) : x \in X\}/(1-\beta),$$

$$\bar{c}_k = \beta \sup\{(H^k v)(x) - (H^{k-1}v)(x) : x \in X\}/(1-\beta).$$

A numerically simple procedure for approximating c_k and $\bar{c}_k$ is given in [24].

An in-depth discussion of a numerical evaluation of the above three approaches and the approach involving the determination of $\{v_n\}$, which we will call successive approximations (SA), is presented in [24]. This evaluation indicates that all three of the approaches presented above are superior (in terms of CPU time and iterations to convergence) to SA. Also, approach 2 tends to be superior to approach 1, which tends to be superior to approach 3. On average, approaches 1 and 2 required roughly 15% the CPU time and 20% the number of iterations as SA, while approach 3 required roughly 22% the CPU time and 30% the number of iterations as SA.

8. Approximating Hv

We have noted earlier that a real-valued function v, having the uncountably infinite space X as its domain, has a finite representation if v is pwl&c; i.e. there exists a finite set of vectors G such that $v(x) = \max\{x\gamma: \gamma \in \Gamma\}$. This fact was a key element in the development of the SS algorithm and its variants. The impressive numerical results reported in the previous section enhance the usefulness of the SS algorithm. However, two facts constrain the ultimate usefulness of any SS-type algorithm. First, the finite representation of the optimal value function requires computationally expensive reconstruction at each iteration. (We note that this representation is equivalent to a finite partition of X.) Second, the cardinality of the finite representation can grow geometrically as a function of stage; recall that ${}^{\#}G = {}^{\#}A({}^{\#}\Gamma ** {}^{\#}Z)$.

Consider the following alternative approach, which we will call the *fixed grid* approach. Assume that X' is a finite subset of X, that an expected value function is determined for all $x' \in X'$, and that these values are used to determine the value

of this expected value function for all $x \in X$. The expected value function is invariably an approximation of the optimal value function. However, both of the aforementioned constraints of the SS algorithm are avoided.

Kakalik [7] and Eckles [5] both used a linear interpolation method between fixed points in X in order to determine an approximation of the optimal value function. Bounds on the quality of the Eckles' approximation are presented in Sondik [20]. Sondik and Mendelssohn [21] used a grid of points that represents only those points in X that can be visited by a specified policy. Lovejoy [10] presented a procedure for approximating X by a finite grid of points. He used this grid to construct upper and lower bounds, generate suboptimal nonstationary and stationary policies, and determine a bound on the value loss, relative to optimal, for using these policies. We now briefly discuss Lovejoy's approach for determining the bounds and the suboptimal design.

Lovejoy constructed the lower bound functions $\{v_n^L\}$ as follows. Let Γ_n^L be such that $v_n^L(x) = \max\{x\gamma\colon \gamma \in \Gamma_n^L\}$. Analogous to our discussion of the SS algorithm, define $g_n : Z \times X \times A \to \Gamma_n$ as

$$xP(z, a)[g_n(z, x, a) - \gamma] \geq 0 \qquad \forall\, \gamma \in \Gamma_n.$$

Let $a' \in A$ be such that $x'[\alpha_n(a') - \alpha_n(a)] \geq 0$ for all $a \in A$, where

$$\alpha_n(a) = r(a) + \beta \sum_z P(z, a) g_n(z, x', a)$$

for $x' \in X'$. Then $\alpha_n(a')$ becomes the member of Γ_{n+1}^L associated with action a' and state x'. This process is repeated for all $x' \in X'$. Importantly, observe that ${}^{\#}\Gamma_n^L \leq {}^{\#}X'$ for all n. It is straightforward to show that $v_n^L \leq v_n$ for all n. Lovejoy also showed that the obvious strategy resulting from this lower bound generates expected value functions bounded below by $\{v_n^L\}$.

With respect to an operations count analysis for determining Γ_{n+1}^L given Γ_n^L, we note that:

(1) ${}^{\#}X'{}^{\#}Z\,{}^{\#}A\,{}^{\#}\Gamma_n^L\,{}^{\#}S$ operations (multiplications and additions) are required to determine $x'P(z, a)\gamma$ for all $\gamma \in \Gamma_n^L$, $z \in Z$, $x' \in X'$, and $a \in A$, where we assume that for all z, x', and a, $\beta x'P(z, a)$ is constructed a priori (requiring ${}^{\#}X'{}^{\#}Z\,{}^{\#}A({}^{\#}S)^2$ operations).

(2) ${}^{\#}X'{}^{\#}Z\,{}^{\#}A({}^{\#}\Gamma_n^L - 1)$ comparisons are required to determine $g_n(z, x', a)$ for all z, x', and a.

(3) ${}^{\#}X'{}^{\#}A({}^{\#}S + {}^{\#}Z)$ operations are required to determine $x'\alpha(a)$ for all x' and a.

(4) ${}^{\#}X'({}^{\#}A - 1)$ comparisons are required to determine $\alpha(a')$, and hence Γ_{n+1}^L for all x'.

Thus, determination of Γ_{n+1}^L given Γ_n^L requires ${}^\#X'{}^\#A({}^\#Z{}^\#\Gamma_n^L{}^\#S + {}^\#Z)$ operations and ${}^\#X'[{}^\#Z{}^\#A({}^\#\Gamma_n^L - 1) + ({}^\#A - 1)]$ comparisons. Since surely ${}^\#S \le {}^\#\Gamma_n^L \le {}^\#X'$, the number of operations is bounded below by ${}^\#A{}^\#Z({}^\#S)^3$.

With respect to the upper bound, Lovejoy considered the grid $X' = \{(1/M)m : m \in I^{\#S}, \sum_i m_i = M\}$, where $I^{\#S}$ is the ${}^\#S$-dimensional vectors having nonnegative integer components and M is a design parameter. It appears that ${}^\#X' = (M + {}^\#S - 1)!/(M!({}^\#S - 1)!)$; thus, the grid becomes finer as M becomes larger. Lovejoy presented a procedure, in part due to Freudenthal, for describing any point in X in terms of points in X' and some easily determined barycentric coordinates. The upper bound is based on the evaluation of Hv on X' and on a piecewise linear approximation of Hv for other values in X.

Once upper and lower bounds are known, then action elimination can be used to identify an optimal action at as many points in X as possible. Impressive numerical results are presented in [10].

The following completely observed MDP appears to capture much of the process presented by Lovejoy for constructing an upper bound on v^*. For each $x' \in X'$, $z \in Z$, and $a \in A$, let $\{\mu_i(z, x', a)\ i = 1, \ldots, {}^\#S + 1\} \subseteq X'$ be the set of elements and $\{w_i(z, x', a), i = 1, \ldots, {}^\#S + 1\}$ be the set of barycentric coordinates such that $\lambda(z, x', a) = \sum_i w_i(z, x', a)\mu_i(z, x', a)$. Lovejoy presents techniques for constructing $\{\mu_i\}$ and $\{w_i\}$. Then there exists a unique, real-valued function on X', v', that serves as the fixed point of the operator H', where

$$[H'v](x') = \max\{x'r(a) + \beta \sum_z \sigma(z, x', a) \sum_i w_i(z, x', a) v[\mu_i(z, x', a)] : a \in A\}.$$

For general $x \in X$, let $\{x_i\} \subseteq X'$ and $\{\xi_i\}$ be such that $x = \sum_i \xi_i x_i$. Then let $v'(x) = \sum_i \xi_i v'(x_i)$. It is easily shown that $v^* \le v'$.

It is easy to show that determining $H'v$ for a given real-valued function on X', v, requires

$${}^\#X'{}^\#A{}^\#Z({}^\#S + 1) = {}^\#A{}^\#Z({}^\#S + 1)(M + {}^\#S - 1)!/(M!({}^\#S - 1)!)$$

operations. We observe that this number has ${}^\#A{}^\#Z({}^\#S ** (M + 1))/M!$ as a lower bound.

9. Finite-memory suboptimal design

Assume that there is a process $\{\Theta_t, t = 0, 1, \ldots\}$ such that an optimal strategy can be obtained which is dependent on the data process only through $\{\Theta_t, t = 0, 1, \ldots\}$. That is, assume it is sufficient for optimality for $a(t)$ to depend only on $\Theta_t[d(t)]$. Then, Θ_t sufficiently summarizes the information useful for action selection that is contained in $d(t)$. We refer to such a process $\{\Theta_t, t = 0, 1, \ldots\}$ as a *sufficient statistic* (a more formal treatment of which can be found in [3]). We seek sufficient statistics for two reasons. First, a sufficient statistic can accommodate optimality. Second,

the sufficient statistic may represent the information contained in the data process that is useful for action selection in a computationally desirable manner. We remark that the data process is trivially a sufficient statistic, and we recall that the information process is a sufficient statistic possessing some numerically appealing features (e.g. the state space of the information process is stage invariant, whereas the state space of the data process grows geometrically in stage).

Platzman [14, 15] and White and Scherer [25] have considered a third sufficient statistic in the development of bounds and a suboptimal design for the POMDP. This third sufficient statistic $\{y(t), t = 0, 1, \ldots\}$ combines the data and information processes in the following manner:

$$y(t) = \{x(t-m), z_t^m, a_{t-1}^m\},$$

where

$$z_t^m = \{z(t), \ldots, z(t-m+1)\},$$

$$a_{t-1}^m = \{a(t-1), \ldots, a(t-m)\},$$

and where $m = t$ if $t < M$ and $m = M$ otherwise for design parameter M. In the development of the results to follow, M will designate the maximum number of the most recent observations and actions on which decisions will be based. Note that $x(t) = \lambda^m[z_t^m, x(t-m), a_{t-1}^m)]$, where

$$\lambda^m(z_t^m, x, a_{t-1}^m) = xP(z_t^m, a_{t-1}^m)/xP(z_t^m, a_{t-1}^m)\mathbf{1}$$

and

$$P(z_t^m, a_{t-1}^m) = P[z(t-m+1), a(t-m)] \times \ldots \times P[z(t), a(t-1)].$$

Thus, since $\{x(t), t = 0, 1, \ldots\}$ can be constructed from $\{y(t), t = 0, 1, \ldots\}$, $\{y(t), t = 0, 1, \ldots\}$ is a sufficient statistic.

We now present the development found in [25]. Let V^m be the set of all bounded, real-valued functions on $Z^m \times A^m$, $z^m = \{z(m), \ldots, z(1)\}$, and $a^m = \{a(m-1), \ldots, a(0)\}$. Define a policy as a mapping $\delta: Z^m \times A^m \to A$. Let $\|\cdot\|^m$ represent the supremum norm on V^m, and note that $(V^m, \|\cdot\|^m)$ is a Banach space. For $m < M$, let $\xi^m = \{z, z(m), \ldots, z(1)\}$ and $\alpha^m = \{a, a(m-1), \ldots, a(0)\}$, and let $\xi^M = \{z, z(M), \ldots, z(1)\}$ and $\alpha^M = \{a, a(M-1), \ldots, a(1)\}$. White and Scherer assumed that for all $(z, a) \in Z \times A$ and all $i \in S$, $\sum_j p_{ij}(z, a) \neq 0$, which ensures that the vector $\tilde{P}_i(z^m, a^m)$ is well defined for all i, where the jth element of $\tilde{P}_i(z^m, a^m)$ is $p_{ij}(z^m, a^m)/\sum_k p_{ik}(z^m, a^m)$ and where $p_{ij}(z^m, a^m)$ is the ijth element of the matrix $P(z^m, a^m)$.

We can now define operators useful for generating upper and lower bounds on v^* and suboptimal designs. Let U_δ^m and U^m be such that for $m < M$, $U_\delta^m: V^{m+1} \to V^m$ and $U^m: V^{m+1} \to V^m$. Let $U_\delta^M: V^M \to V^M$ and $U^M: V^M \to V^M$. Let δ be such that $a = \delta(z^m, a^m)$. Then,

$$[U_\delta^m v](z^m, a^m) = \max_{i \in S} \{\tilde{P}_i(z^m, a^m)[r(a) + \beta \sum_z [P(z, a)\mathbf{1}]v(\xi^{m}, \alpha^m)]\},$$

$$U^m v = \sup_\delta U_\delta^m v.$$

Define L_δ^m and L^m identically, except replace "max" with "min" in the definition of L_δ^m. It is easily shown that U_δ^m, U^m, L_δ^m, and L^m are contraction operators on V^m with modulus β. Motivation for the definitions of these operators is given in [25]. Let u^M and l^M be the fixed points of U^M and L^M, and for $m < M$, let $\{u^m\}$ satisfy $u^m = U^m u^{m+1}$ and $l^m = L^m l^{m+1}$. Results presented in [25] include:

(1) Upper and lower bounds on v^*. For all $m \leq M$ and $(z^m, a^m) \in Z^m \times A^m$,

$$l^m(z^m, a^m) \leq v^*[\lambda^m(z^m, x, a^m)] \leq u^m(z^m, a^m),$$

for all $x \in X$.

(2) A sufficient condition for these bounds to be tight. If $\tilde{P}(z^m, a^m)$ has rank 1 for all $(z^m, a^m) \in Z^m \times A^m$ for some $m \leq M$, then for all n, $m \leq n \leq M$, and $(z^n, a^n) \in Z^n \times A^n$, $l^n(z^n, a^n) = u^n(z^n, a^n)$.

(3) A priori bounds on $\| u^M - l^M \| / \| l^M \|$.

(4) A Bertsekas-type extrapolation for accelerating the determination of l^M and u^M.

(5) A lower bound on the suboptimal strategy π induced by the determination of $\{l^m\}$. For all $(z^m, a^m) \in Z^m \times A^m$, $m \leq M$, $l^m(z^m, a^m) \leq v_m^\pi[\lambda^m(z^m, x, a^m)]$, for all $x \in X$.

(6) A guarantee that larger M produces tighter bounds.

An operations count analysis shows that determination of $L^M v$, given v, requires ${}^{\#}S(({}^{\#}A\ {}^{\#}Z)\ **\ (M+1))$ operations. Therefore, on the basis of operations per iteration, we conclude that for small action and observation spaces and large state spaces, use of the L^M operator for suboptimal design determination is preferred to the use of the Lovejoy lower bound procedure, and that for large action and observation spaces and small state spaces, we should prefer use of the Lovejoy lower bound procedure to use of the L^M operator.

10. Heuristic search – A direction for future research

Dynamic programming has thus far served as the basis for determining the solution of the POMDP. Another related basis for solution is heuristic search, a subarea of artificial intelligence [13].

There are two intriguing aspects of heuristic search. First, heuristic search procedures generally assume the existence of a heuristic function, which (if admissible)

represents an optimistic estimate (e.g. an upper bound) of the optimal value function and is used to guide the search. Results due to Astrom [1,2] (see also White and Harrington [23]) guarantee that a simple extension of the solution of the completely observed MDP represents an easily generated upper bound on the optimal value function of the POMDP. Second, heuristic search procedures guided by simply generated heuristic functions can significantly out-perform dynamic programming; e.g. note the discussion found in [13] concerned with the application of the heuristic search procedure A^* applied to the 8-puzzle.

Lark and White [9] have presented a heuristic search based procedure for solving the finite horizon, completely unobserved MDP (CUMDP). The CUMDP is a POMDP that assumes the $P(z, a)$ are independent of z, for all a. Interest in the CUMDP is due to the fact that a finite horizon version of the CUMDP can be cast as a locally finite, finite depth OR-graph. The specific heuristic search procedure applied is a multiobjective generalization of A^* called MPA^* [22]. (A^* is an informed, best-first search procedure for finding an optimal path through an OR-graph from a given start node to a given set of terminal nodes based on a scalar criterion.) Preliminary numerical results show that this heuristic search algorithm can compare quite favorably to the SS algorithm. Extending this line of research to the more general POMDP, which requires the use of an AND/OR-graph, appears to be a promising topic for future research.

11. Conclusions

We have examined several recently developed procedures for improving the tractability of the POMDP. Several of these have been convergence accelerating variants of, and approximations to, the SS algorithm, attesting to the seminal importance of results found in [18–20]. Finite grid approximations of X have been reviewed. The finite-memory suboptimal design results reported in section 9 attempt to look at the POMDP from a different perspective than that found in [18], and the directions for future research presented in section 10 diverge from dynamic programming as the basis for solution determination. Attempts have been made to indicate which algorithms may be the most useful for various types of POMDPs. We hope that this survey will further stimulate research to improve the tractability, and hence the usefulness, of the POMDP.

References

[1] K.J. Astrom, Optimal control of Markov processes with incomplete state information, J. Math. Anal. Appl. 10(1965)174–205.

[2] K.J. Astrom, Optimal control of Markov processes with incomplete state information II. The convexity of the loss function, J. Math. Anal. Appl. 26(1969)403–406.

[3] D.P. Bertsekas, *Dynamic Programming: Deterministic and Stochastic Models* (Prentice Hall, Englewood Cliffs, NJ, 1987).

[4] H.T. Cheng, Algorithms for partially observed Markov decision processes, Ph.D. Dissertation, Commerce and Business Administration, University of British Columbia (1988).
[5] J.E. Eckles, Optimum replacement of stochastically failing systems, Ph.D. Dissertation, Dept. Eng.-Econ. Syst., Stanford University, Stanford, CA (1966).
[6] O. Hernández-Lerma and S.I. Marcus, Nonparametric adaptive control of discrete-time partially observable stochastic systems, Technical Report, Department of Electrical and Computer Engineering, University of Texas, Austin, TX (1987).
[7] J.S. Kakalik, Optimum policies for partially observable Markov systems, Technical Report 18, Operations Research Center, MIT, Cambridge, MA (1965).
[8] J.W. Lark, private communication (1989).
[9] J.W. Lark and C.C. White, A heuristic search approach for solving finite-horizon, completely unobserved Markov decision processes, in preparation (1989).
[10] W.S. Lovejoy, Computationally feasible bounds for partially observed Markov decision processes, Oper. Res. 39(1991)162–175.
[11] G. Monahan, A survey of partially observable Markov decision processes: Theory, models, and algorithms, Manag. Sci. 28(1982)1–16.
[12] A. Paz, *Introduction to Probabilistic Automata* (Academic Press, New York, 1971).
[13] J. Pearl, *Heuristics* (Addison–Wesley, Reading, MA, 1984).
[14] L.K. Platzman, Finite memory estimation and control of finite probabilistic systems, Ph.D. Dissertation (ESL-R1723), Dept. of Elec. Eng. and Comput. Sci., MIT, Cambridge, MA (1977).
[15] L.K. Platzman, Optimal infinite-horizon undiscounted control of finite probabilistic systems, SIAM J. Control Optim. 18(1980)362–380.
[16] M.L. Puterman and M.C. Shin, Action elimination procedures for modified policy iteration algorithms, Oper. Res. 30(1982)301–318.
[17] M.L. Puterman and M.C. Shin, Modified policy iteration algorithms for discounted Markov decision processes, Manag. Sci. 24(1978)1127–1138.
[18] R.D. Smallwood and E.J. Sondik, The optimal control of partially observable Markov processes over a finite horizon, Oper. Res. 21(1973)1071–1088.
[19] E.J. Sondik, The optimal control of partially observable Markov processes over the infinite horizon: Discounted costs, Oper. Res. 26(1978)282–304.
[20] E.J. Sondik, The optimal control of partially-observable Markov processes, Ph.D. Dissertation, Eng.-Econ. Syst., Stanford University, Stanford, CA (1971).
[21] E.J. Sondik and R. Mendelssohn, Information seeking in Markov decision processes, Southwest Fisheries Center Admin. Report H-79-13, National Marine Fisheries Service, Honolulu, HI (1979).
[22] B.S. Stewart, Heuristic search with general order relation, Ph.D. Dissertation, Department of Systems Engineering, University of Virginia, Charlottesville, VA (1988).
[23] C.C. White and D. Harrington, Application of Jensen's inequality for adaptive suboptimal design, J. Optim. Theory Appl. 32(1980)89–100.
[24] C.C. White and W.T. Scherer, Solution procedures for partially observed Markov decision processes, Oper. Res. 37(1989)791–797.
[25] C.C. White and W.T. Scherer, Finite-memory suboptimal design for partially observed Markov decision processes, submitted for publication (1989).
[26] C.C. White, L.C. Thomas and W.T. Scherer, Reward revision for discounted Markov decision processes, Oper. Res. 33(1985)1299–1315.
[27] C.C. White and D.J. White, Markov decision processes, Eur. J. Oper. Res. 39(1989)1–16.
[28] D.J. White, Further real applications of Markov decision processes, Interfaces 18(1988)55–61.
[29] D.J. White, Real applications of Markov decision processes, Interfaces 15(1985)7–83.