Approximation Algorithms for Solving Cost Observable Markov Decision Processes

Ph.D. Proposal
Department of Computer Science
Oregon State University

Valentina Bayer

Fall, 1998

1 Introduction

Partial observability is a result of noisy or imperfect sensors that are not able to reveal the real state of the world. For example, consider a robot moving in a building. If its sensors detect only walls, then rooms with similar configurations will look the same. People have to deal with partial observability, too. If there is a truck in front of his/her car, the driver will have a limited visibility of the road ahead.

The problems that suffer from partial observability have been modelled as Partially Observable Markov Decision Processes (POMDPs). They have been studied by researchers in Operations Research and Artificial Intelligence for the past 30 years. Nevertheless, solving for the optimal solution or for close approximations to the optimal solution is known to be at least NP-hard ([26]). Current algorithms are very expensive and do not scale well.

Many applications can be modelled as POMDPs: quality control, autonomous robots, weapon allocation, medical diagnosis ([4]). In medical diagnosis, for example, the internal state of the patient is never known with certitude. Actions available to the physician are: administer laboratory tests, prescribe medicine, perform surgery. Costs are attached to actions and tradeoffs must be made between the health-risks/costs of actions and the accuracy of their resulting observations.

Problem definition

The specific problem addressed in this proposal is the development of good approximation algorithms for solving problems that have partial observability. The model we propose associates costs with obtaining information about the current state. We want to predict when and how much it is necessary to observe. We want to use our Cost Observable Markov Decision Process (COMDP) model to find good solutions for real-world problems.

The proposal is organized as follows: Section 2 introduces the MDP model and Section 3, its extension to account for partial observability (POMDP). Section 4 reviews the POMDP literature. Section 5 introduces the COMDP model and a current approximation method to solve it. The last section outlines the future work.

2 Markov Decision Processes (MDPs)

A (discrete) Markov Decision Process ([29]) is a model of interaction between an agent and the world, where the agent always knows what state of the world it is currently in. We say that an MDP is *fully observable*.

It can be formally described as a tuple $\langle S, A, P(S|S,A), R(S|S,A)\rangle$, where:

- S = the set of states of the world (we assume we are able to model every state in which the world might be)
- A = the set of actions
- $P(S|S,A)$ = the transition probabilities
- $R(S|S,A)$ = the immediate reward of the actions

An action can have a deterministic or a stochastic effect. If it is deterministic, we know for sure what the next state is. If it is stochastic, there may be more than one possible resulting state. We write $P(s_{t+1}|s_t, a_t)$ for the probability of ending in state s_{t+1} at time $t+1$, after performing action a_t in state s_t at time t. Note that $\sum_{s_{t+1}\in S} P(s_{t+1}|s_t, a_t) = 1$ (so this is a probability distribution over the next states).

An action has an immediate reward (or cost). Here, we will consider only negative rewards. We write $R(s_{t+1}|s_t, a_t)$ for the reward associated with the transition from s_t to s_{t+1}, after performing action a_t.

The reason we wrote only the current state s_t and action a_t as a condition for the next state s_{t+1} is the *Markov property*. It says that the next state (and reward) depends only on the current state and action, and not on the history of past states and actions:

$$P(s_{t+1}|s_t, a_t, s_{t-1}, a_{t-1}, \ldots, s_0, a_0) = P(s_{t+1}|s_t, a_t).$$

A "candidate solution" to an MDP is called a policy. A policy is a mapping from states to actions $\pi : S \to A$, and chooses an action to take for each state. The goal is to maximize some utility function (such as the expected sum of rewards). An optimal policy is an optimal solution to the MDP and chooses the best action to be performed in each state.

MDPs can be formulated over a finite horizon (when the agent reaches the goal state(s) after a finite number of steps) or over an infinite horizon (when

the agent has an infinite lifetime). The *utility function* can be expressed as the expected sum of discounted rewards:

$$E\left[\sum_{t=0}^{k-1}\gamma^t R(s_{t+1}|s_t,a_t)\right]$$

for a finite horizon of k steps, where $0<\gamma<1$ is a discount factor. This is the utility for a fixed policy that is choosing the actions a_t, and the expectation is taken with respect to the randomness in the effects of the actions.

The *value function* of an MDP associates with each state a value that is the expected reward of following an optimal policy from that state. *Value iteration* and *policy iteration* algorithms solve MDPs and find their optimal value functions and optimal policies.

The (optimal) value function V^* is the solution of the Bellman equations (one equation for every state):

$$V(s)=\max_a \sum_{s'\in S} P(s'|s,a)\times(R(s'|s,a)+\gamma V(s')).$$

Given the optimal value function, the optimal policy π^* is computed as:

$$\pi^*(s)=\arg\max_a \sum_{s'\in S} P(s'|s,a)\times(R(s'|s,a)+\gamma V^*(s')).$$

Both value iteration and policy iteration require a model of the probability of transitions and a model of the immediate rewards. When these models are not available, reinforcement-learning approaches can be used ([17] and [32])). The model-based approaches learn a model by interacting with the environment (such as Dyna and prioritized sweeping), and use the model to find the optimal policy. The model-free ones learn a controller (a policy) without learning a model, as is the case with $Sarsa(\lambda)$ and Q-learning.

3 Partially Observable Markov Decision Processes (POMDPs)

The following is an analogy taken from Monahan ([25]): "Howard described movement in an MDP as a frog in a pond jumping from lily pad to lily pad. ...we can view the setting of a POMDP as a fog shrouded lily pond. The frog is no longer certain about which pad it is currently on. Before jumping, the frog can obtain information about its current location".

In the POMDP model, the agent does not know the real state of the world; instead it can perceive it through some observations. The observations can be probabilistic, so an observation model will specify the probability of each observation for every state in the model.

Typical applications of POMDPs are robot navigation in noisy environments and management of a sick patient in the face of imperfect information about the patient's actual state.

A formal model of a POMDP is a tuple $\langle S, A, O, P(S|S,A), P(O|S,A), R(S|S,A)\rangle$ where $S, A, P(S|S,A), R(S|S,A)$ are the same as in the MDP definition,

- O = the set of observations
- $P(O|S,A)$ = the observation probabilities; we write $P(o_t|s_t, a_{t-1})$ for the probability of observing o_t in state s_t at time t, after executing a_{t-1}

At time t, the agent has a certain belief $P(s_t)$ of being in state s_t, such that $\sum_{s_t \in S} P(s_t) = 1$ (some of the probabilities may be zero). This distribution is called the *belief state* at time t. How do we maintain a belief state? One method is to remember the initial belief state and the entire history of actions and observations until the present. In some problems, the history of the last K steps (actions and observations) may be sufficient.

It turns out that simply maintaining a probability distribution over all of the underlying MDP's states (i.e a belief state) provides us with the same information as if we maintained the complete history. So the belief state is a *sufficient statistic* for the history, and the Markov property holds for belief states:

$$P(b_{t+1}|b_t, a_t, o_{t+1}, b_{t-1}, a_{t-1}, o_t, \ldots, b_0, a_0, o_1) = P(b_{t+1}|b_t, a_t, o_{t+1})$$

where we denote the belief state at time t as b_t.

Therefore a (discrete) POMDP can be converted into a continuous-state MDP (or a belief MDP) over the belief space. If an agent adopts the optimal policy for the belief MDP, the resulting behavior will be optimal for the POMDP. Exact solutions for belief MDPs are mentioned in the next section.

The belief state is updated after performing an action and receiving an observation.

The solution to a POMDP is a policy mapping from belief states to actions. How do we represent a policy, since there are uncountably many belief states? For a fixed horizon, there is a finite number of policies that can be followed, starting in a belief state b. Given the value function V_p associated with a policy p, the value of a belief state b for policy p is the weighted sum of the individual state values $V_p(b) = \sum_{s \in S} P(s)V_p(s)$. If we denote the vector of $V_p(s)$ as α_p, then $V_p(b) = b \cdot \alpha_p$ and the optimal value of belief state b is $V(b) = \max_p b \cdot \alpha_p$.

Each policy induces a value function that is linear in belief state b, and the value function V is the upper surface of those functions (see Figure 1). So V is piecewise linear and convex and only a finite number of α vectors are needed to represent it for any fixed horizon. Some of the vectors are completely dominated by others, so V consists only of the useful ones, i.e. vectors that are the best over a region of the belief space.

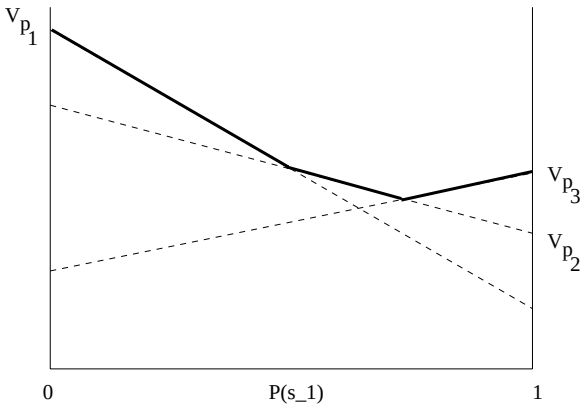

Figure 1: The value function over a finite horizon is piecewise linear and convex; this example shows the value function of a POMDP with 2 states.

4 Literature Review

4.1 History: Operations Research and Artificial Intelligence

POMDPs were formulated by A. Drake in 1962 ([11]) and have been in the attention of researchers in Operations Research (see Monahan's and Lovejoy's surveys [25] and [21]). About ten years ago people in the Artificial Intelligence community approached the field, and in the recent years there have been a number of PhD theses about POMDPs ([24], [18], [15], [3] and [12]).

4.2 Piecewise Linear Convexity

Smallwood and Sondik formulated the optimal control problem for POMDPs over a finite horizon ([30]). The paper demonstrates, for a finite horizon, that the optimal value function is piecewise linear and convex over the state probabilities of the underlying Markov process. The authors showed that the current belief state is a sufficient statistic for the past history of actions and observations of a POMDP.

Sondik also explored the POMDPs over an infinite horizon, the discounted case ([31]). In general, infinite horizon POMDPs' optimal value functions are not piecewise linear. However, they can be approximated arbitrarily closely by a finite horizon value function for a sufficiently long horizon.

4.3 Exact Solutions

Exact solutions for (finite horizon) POMDPs compute a set of linear functions (vectors) defining the optimal value function. Each vector corresponds to a policy that is optimal in some region of the belief space. There are different ways to compute the optimal set of vectors:

- generate all possible linear functions first, and eliminate redundant ones afterwards, as in Monahan's enumeration algorithm.

- generate useful linear functions by evaluating and checking a finite number of points of the belief state space, as in Sondik's method ([30]) and Cheng's linear support algorithm ([7]). Improved algorithms include the witness algorithm ([5] and [16]) and Zhang's incremental pruning algorithm ([6]).

It is possible to represent a policy as a finite state controller, where the nodes are actions and the arcs are labelled with observations. At run-time, the initial belief state is used to choose the starting node. At every step, the action specified by the current node is taken, and depending on the observation received, a transition to a new node is made; the process continues. This representation has the advantage of not having to maintain a belief state at run time.

Existing exact algorithms can take an exponential amount of space and time to compute the policy, even if the policy itself does not require an exponential size representation.

4.4 Complexity Issues

Papadimitriou and Tsitsiklis showed that, for both finite and infinite horizon MDPs, computing the optimal policy is in P ([27]). They also showed that finding optimal policies for POMDPs is PSPACE-complete (PSPACE is the class of problems that can be solved in a polynomial amount of space; $NP \subseteq PSPACE$).

Mundhenk, Goldsmith, Lusena and Allender analyzed the complexity of POMDPs and the hardness of their approximate results ([26]), such as:

- The stationary (i.e. time independent) policy existence problem for POMDPs is NP-complete. The policy existence problem asks whether there exists a policy whose expected reward is greater or equal to a given value.
- The optimal stationary policy for POMDPs can be $\epsilon-$approximated for any $\epsilon < 1$ if and only if $P = NP$.

4.5 Approximate Solutions

The simplest approximate methods for solving POMDPs include:

- Most Likely State: the system assumes that it is in the state with the highest occupation probability and executes the action given by the optimal MDP policy for that state.
- Action Voting: states vote for their MDP optimal actions in proportion to their occupation probabilities.

- Q-MDP: estimates the Q value for a (belief state, action) as a linear function of the MDP's Q values for (states,actions).

Nevertheless, they all fail in situations where there is a lot of uncertainty in the belief state (see [3] for comparison results).

Parr and Russell introduced an approximate method for determining infinite horizon policies for POMDPs, called SPOVA ([28]). They used a gradient descent search and a continuous, differentiable representation of the value function ("soft" max), but they only tested it on very small problems.

Loch and Singh showed that $Sarsa(\lambda)$ (that is, the eligibility traces version of Sarsa) works well on small POMDPs that have good memoryless policies or low-order-memory-based policies ([19]). In their paper, $Sarsa(\lambda)$ learns a control policy while treating the immediate observation and a history of the past K observations as the current state of the system.

Another approach to approximately solve POMDPs is to search in the policy space (all the above algorithms searched in value function space). Eric Hansen represented a policy explicitly as a finite state controller and used policy iteration to solve POMDPs for all belief states ([13]). His approach outperforms value iteration in solving infinite-horizon POMDPs.

Hansen also used heuristic search to solve POMDPs given the starting belief state. This had the advantage of focusing computation on regions of the belief space that are likely to be reached from the starting belief state. In general, the controller obtained with heuristic search is smaller than the controller computed by policy iteration (because that one optimizes the value of each possible belief state).

I also intend to study grid-based methods for solving POMDPs ([20], [2]) and upper/lower bounds of the POMDP value function ([14], [34]).

4.6 Scalability

Current methods for finding optimal or approximately optimal policies for POMDPs are very expensive and do not work for problems with more than a few hundred states. They are impractical for most real world applications.

One possible solution is to develop better approximation algorithms and to exploit the problem structure.

4.7 Factored Models

Craig Boutilier and David Poole represented POMDPs as dynamic Bayesian networks and used this model to structure the belief space ([1]). The value function is represented as a tree, where the branches correspond to different values for the state features. This idea seems better fitted for approximation algorithms where small differences between states could be ignored.

4.8 Learning Control Policies: Model-free Approaches

One approach to learn POMDP models is to extend techniques for learning hidden Markov models to learn POMDP models. Another approach is to learn a controller without learning a model (the model-free approach). The second method is briefly discussed in this section.

Whitehead and Ballard solved a restricted class of partially observable problems, by having their system learn to focus its attention on the relevant aspects of the domain ([35]). They introduced the term *perceptual aliasing* to denote situations where the agent's internal representation does not distinguish world states that perceptually look the same, but require different actions.

Chrisman extended the previous technique by allowing actions with stochastic effects and tasks that require memory. His *predictive distinctions* approach ([8]) learns a predictive model (i.e. a POMDP) by interacting with the world and discovering important distinctions. A single state in the model may correspond to several possible world states. When there are distinctions in the world not currently accounted for by the model, this will increase its number of states to fit the observed data.

While Chrisman's algorithm creates new states based on predicting perception, McCallum's *utile distinctions* algorithm creates new states if this helps predict reward (see [22]). Utility-based distinctions will build a state space as large as needed to perform the current task, while perception-based distinctions will build a state space as complex as the perceived world.

McCallum learns on-line a relevant history of perceptions and actions, and stores it as a branch in a suffix tree; a leaf node stores the policy for that history ([23]). This approach does not scale well for large number of observations, but his PhD thesis ([24]) presents an algorithm that incorporates perceptual distinctions in the suffix tree.

5 Cost-observable Markov Decision Processes (COMDPs)

In POMDPs, observations are received for free; we want to explicitly model how expensive it is to gather information. If the agent is pretty sure where it is or that it is doing the right thing without having to know its exact position, then it does not have to observe. But if it is confused, then it is better to observe. And the more it observes, the more it will have to pay.

To model this "cost-observability" we introduce the COMDP model, in which actions are of two kinds:

- *world actions* that change the state of the world, but return no observation information
- *observation (or sensing) actions* that return observation information, but the state of the world does not change while performing them

COMDPs are intended to model situations that arise in diagnosis and active vision where there are many observation actions that do not change the world and relatively few world-changing actions. We are particularly interested in problems where there are many alternative sensing actions (including, especially, no sensing at all) and where, if all observation actions are performed, the entire state of the world is observable (but presumably at very great cost). Hence, COMDPs can also be viewed as a form of fully observable MDPs where the agent must pay to receive state information (i.e., they are "cost-observable").

Our long-term goal is to model the acquisition of human visual observation strategies within an air-traffic control simulation.

5.1 Example

Imagine a skier that is faced with a choice of either going down a longer way, in a valley, or going along a shorter way, near a cliff. The valley is safe so the skier does not have to do any observation actions. The cliff way is very dangerous, such that the skier has to observe where he is at every step, making it very expensive. If the skier were not to observe, his belief state will be equally spread among relatively safe states and very dangerous states. He will have to behave cautiously and take expensive actions (as required by the

dangerous states), not benefiting at all from the safer states (where he can take cheaper actions). So if he takes the cliff way, he is forced to observe.

5.2 Mathematical Model

A formal model of a COMDP is a tuple

$$\langle S, A, C, O, P(S|S,A), P(O|S,C), R_A(S|S,A), R_C(C)\rangle, where$$

- S = the set of states of the world
- A = the set of actions
- C = the set of observation actions
- O = the set of observations
- $P(S|S,A)$ = the transition probabilities
- $P(O|S,C)$ = the observation probabilities
- $R_A(S|,S,A)$ = the immediate costs (rewards) of the actions
- $R_C(C)$ = the costs of the observation actions

At each time t, the agent chooses to perform a world action $a_t \in A$ and an observation action $c_{t+1} \in C$. The world action causes the world to make a transition from some state $s_t \in S$ to a new state $s_{t+1} \in S$ according to $P(s_{t+1}|s_t, a_t)$. Then the agent receives an observation $o_{t+1} \in O$ according to $P(o_{t+1}|s_{t+1}, c_{t+1})$ (see Figure 2). The agent also receives a scalar reward equal to

$$R(s_{t+1}|s_t, (a_t, c_{t+1})) = R_A(s_{t+1}|s_t, a_t) + R_C(c_{t+1}).$$

We assume that the costs of the observation actions depend only on the action, not on the state, so we write $R_C(c_{t+1})$ for the cost of the observation action performed at time $t+1$.

If the current belief state b_t is a vector of $P(s_t)$, then after performing world action a_t and observation action c_{t+1} and receiving observation o_{t+1}, the belief state can be updated as:

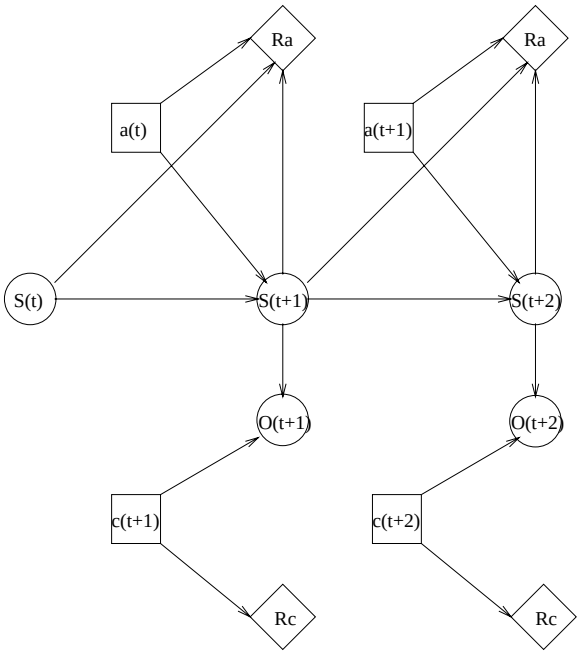

Figure 2: Decision diagram for a COMDP

$$
\begin{aligned}
P(s_{t+1}) &= P(s_{t+1}|b_t, a_t, c_{t+1}, o_{t+1}) \\
&= \frac{P(o_{t+1}|s_{t+1}, c_{t+1}) \times \sum_{s_t \in S} P(s_{t+1}|s_t, a_t) P(s_t)}{\sum_{s_{t+1} \in S} P(o_{t+1}|s_{t+1}, c_{t+1}) \times \sum_{s_t \in S} P(s_{t+1}|s_t, a_t) P(s_t)}.
\end{aligned}
$$

A COMDP policy is a mapping from belief states to world actions and observation actions. At time t, for the current belief state, the policy chooses a pair of (action, observation action) = (a_t, c_{t+1}).

5.3 Complexity Equivalence with POMDPs

Every POMDP can be transformed, in polynomial time, into a COMDP whose optimal policy can be mapped back into an optimal policy for the POMDP. For each COMDP we can construct a POMDP in polynomial time.

Let M_1 be the underlying MDP for the COMDP (the states are fully observable, at no cost), with the reward function $R_1(s_{t+1}|s_t, a_t) := R_A(s_{t+1}|s_t, a_t)$.
Let $k := 0$
repeat

- $k := k + 1$
- find the value function V_k of MDP M_k
- for each state s_t, perform a lookahead search to choose the best action a_t, observation action c_{t+1}, and subsequent action a_{t+1}. At time $t + 2$, we will use the value $V_k(s_{t+2})$ from MDP_k.
- define M_{k+1} to be the same as M_k except that the reward function R_{k+1} is changed to reflect the cost of c_{t+1} (this is incorporated into the cost of all actions that enter state s_t):

$$R_{k+1}(s_t|s_{t-1}, a_{t-1}) := R_A(s_t|s_{t-1}, a_{t-1}) + R_C(c_{t+1}), \forall s_t, s_{t-1}$$

until $M_{k+1} = M_k$
Define $\hat{V} := V_K$, where M_K is the last MDP constructed by this algorithm.

Figure 3: Approximation algorithm for solving a COMDP

This means that POMDPs and COMDPs have the same worst case complexity (and therefore a complete exact solution method for one problem will provide a complete exact solution method for the other problem). The POMDP $\leftrightarrow$ COMDP reductions are presented in the appendix.

What we want to do is approximately solve COMDPs and see what classes of the POMDPs these approximations are good for (this is an open problem).

5.4 Approximation Algorithm

To find an approximately optimal policy for a COMDP, we developed the following algorithm (see Figure 3).

The algorithm constructs a series of (fully observable) Markov Decision Processes $M_1, \ldots, M_K$. The MDPs are identical, except for the immediate

reward function that is changed to include the cost of observation actions. The algorithm converges when the rewards stop changing.

At each iteration, we perform a lookahead search from each state s_t, assuming the world is cost observable for one step and fully observable afterwards. The world action a_t spreads the belief over the resulting states, according to the model $P(s_{t+1}|s_t, a_t)$. Then we have to consider each observation action c_{t+1} and each possible observation o_{t+1}. We update the belief state using (a_t, c_{t+1}, o_{t+1}). Then another world action a_{t+1} is chosen and we assume the resulting state s_{t+2} is fully observable, so we can use its value as given by the current MDP. We are interested in the pair (a_t, c_{t+1}) that maximizes the expected return. Finally we modify the costs of all actions a_{t-1} that have transitions into the state s_t to include the cost of the observation action c_{t+1}. The next MDP will make state s_t more expensive to enter.

At run-time, we perform a lookahead search from the current belief state to find the best (action, observation action) to perform. In the process, we use the value function $\hat{V}$ and the modified rewards computed for the last MDP, M_K, and we marginalize out the uncertainty about the current (and resulting) states. After executing the (action, observation action) and receiving an observation, we update the belief state and so on.

Either the world action or the observation action (but not both) can be a no-operation, so it is possible to have many observation actions between two world actions, as well as many world actions between two observation actions.

A disadvantage of our algorithm is that its off-line computation is based only on belief states that result after the agent starts in a known state and takes a world action. These belief states are not very spread out, so they are less likely to encompass situations where the agent is highly confused.

The advantage of this approximation algorithm is that it can be used for COMDPs with a large number of states (tens of thousands of states), because the main computational effort is concentrated on solving the chain of MDPs. Existing algorithms for solving POMDPs do not scale for a large number of states.

For the skier example, the POMDP's optimal policy is to take the valley way and our COMDP approximation method is able to find it, too. Our algorithm will examine possible future states where expensive observation actions will be needed, and propagate this information backwards, so the skier avoids the cliff way altogether.

6 Future Work

6.1 Methods and Experiments

We plan to develop new approximation algorithms for COMDPs and use mathematical analysis to understand how these algorithms compare to each other. We will do experimental testing to evaluate the quality of the approximation algorithms and to compare their running times.

Once provable good results are obtained for small benchmark problems, we will switch our research to a real world application (the Air Traffic Control simulation or a medical diagnosis problem).

6.2 Long Term Goal

We want to apply the COMDP approximation algorithms to the problem of active visual perception in real-time problem-solving tasks such as air traffic control. Active vision refers to systems that not only sense, but also interact with the world during sensing, by focusing attention, processing selectively, choosing where to look, etc. ([9], [10], [33]).

We want to look at reinforcement learning algorithms for learning the COMDPs. During the early phases of learning, the controller can observe the entire state of the system and acquire an accurate model.

6.3 Schedule

- 1999:
  - work on approximation algorithms for COMDPs
  - do a comparative study of COMDP and POMDP's algorithms
  - write paper for NIPS
- 2000:
  - learn COMDPs using factored models
  - work on a real world application
  - write papers
- 2001: write thesis and defend

7 Appendix

7.1 POMDP to COMDP Reduction

The POMDP $= \langle S, A, O, P(S|S,A), P(O|S,A), R(S|S,A)\rangle$ is transformed into the COMDP $= \langle S', A', C, O', P'(S'|S',A'), P'(O'|S',C'), R_A(S'|S',A'), R_C(C)\rangle$,

- $S' = S \times A \times \{1,2\}$; if action a_t results in state s_{t+1}, then the COMDP state at time $t+1$ is written (s_{t+1}, a_t, l_{t+1}), where the tag $l_{t+1} \in \{1,2\}$
- $A' = A \times \{1,2\}$
- $C = \{obs, no-op\}$
- $O' = O \times \{1,2\}$
- $P'((s_{t+1}, a_t, 1)|(s_t, a_{t-1}, l_t), (a_t, l_t)) = P'((s_{t+1}, a_t, 2)|(s_t, a_{t-1}, l_t), (a_t, l_t))$ $= \frac{1}{2}P(s_{t+1}|s_t, a_t)$, that is, when the action tag matches the state tag, a standard state transition occurs, but the resulting state tag is set at random (half the time to 1, half the time to 2)

  $P'((s_t, a_{t-1}, l_1)|(s_t, a_{t-1}, l_1), (a_t, l_2)) = 1$, i.e. the state does not change if the action tag does not match the state tag $l_1 \neq l_2$
- $P'((o_{t+1}, l_{t+1})|((s_{t+1}, a_t, l_{t+1}), obs)) = P(o_{t+1}|s_{t+1}, a_t)$, where $o_{t+1} \in O$, so 'obs' reveals the current tag with certainty, and also gives the observation associated with the previous action
- $R_A((s_{t+1}, a_t, 1)|(s_t, a_{t-1}, l_t), (a_t, l_t)) = R_A((s_{t+1}, a_t, 2)|(s_t, a_{t-1}, l_t), (a_t, l_t))$ $= R(s_{t+1}|s_t, a_t)$ and $R_A((s_t, a_{t-1}, l_1)|(s_t, a_{t-1}, l_1), (a_t, l_2)) = -\infty$, if $l_1 \neq l_2$, so if the action tag does not match the state tag, there is a huge penalty
- $R_C(c) = 0$

Every state, world action and observation in the COMDP has a tag, 1 or 2. If an action tag does not match the state tag, there is a huge penalty. When the tags match, a standard state transition occurs, but the tag for the resulting state is set at random (half the time to 1, half the time to 2). The observation 'obs' reveals the current tag with certainty, and also gives the observation associated with the previous action. Rewards are associated with the transitions as normal.

We must show than an optimal COMDP policy can be transformed into an optimal POMDP policy. For every belief state b_t, the optimal COMDP policy chooses (a_t, obs). But a_t is also the optimal action the POMDP chooses for belief state b_t.

7.2 COMDP to POMDP Reduction

The COMDP $= \langle S, A, C, O, P(S|S,A), P(O|S,C), R_A(S|S,A), R_C(C) \rangle$ is transformed into the POMDP $= \langle S', A', O', P'(S'|S',A'), P'(O'|S',A')), R'(S'|S',A') \rangle$, where

- $S' = S$
- $A' = A \times C$
- $O' = O$
- $P'(s_{t+1}|s_t, (a_t, c_{t+1})) = P(s_{t+1}|s_t, a_t)$
- $P'(o_{t+1}|s_{t+1}, (a_t, c_{t+1})) = P(o_{t+1}|s_{t+1}, c_{t+1})$
- $R'(s_{t+1}|s_t, (a_t, c_{t+1})) = R_A(s_{t+1}|s_t, a_t) + R_C(c_{t+1})$

The equivalent POMDP's set of actions is the cross product of the COMDP's set of actions and observation actions.

We must show than an optimal POMDP policy can be transformed into an optimal COMDP policy. The optimal POMDP policy will choose, for belief state b_t, the action (a_t, c_{t+1}). This is mapped back into the COMDP's (optimal) choice of the couple (a_t, c_{t+1}) for the belief state b_t.

References

[1] C. Boutilier and D. Poole. Computing optimal policies for partially observable decision processes using compact representations. *Proceedings of the Thirteenth National Conference on Artificial Intelligence, pages 1168-1175*, 1996.

[2] R.I. Brafman. A heuristic variable grid solution method for POMDPs. *Proceedings of the Fourteenth National Conference on Artificial Intelligence, pages 727-733*, 1997.

[3] A. Cassandra. *Exact and approximate algorithms for partially observable Markov Decision Processes.* PhD thesis, Brown University, 1998.

[4] A. Cassandra. A survey of POMDP applications. Technical report, Microelectronics and Computer Technology Corporation (MCC), 1998.

[5] A. Cassandra, L. P. Kaelbling, and M. Littman. Acting optimally in partially observable stochastic domains. *Proceedings of the Twelfth National Conference on Artificial Intelligence, pages 1023-1028*, 1994.

[6] A. Cassandra, M. Littman, and N. Zhang. Incremental pruning: A simple, fast, exact method for partially observable Markov decision processes. *Proceedings of the Thirteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-97), pages 54–61*, 1997.

[7] H.T. Cheng. *Algorithms for partially observable Markov decision processes.* PhD thesis, University of British Columbia, 1988.

[8] L. Chrisman. Reinforcement learning with perceptual aliasing: the perceptual distinctions approach. *Proceedings of the Tenth National Conference on Artificial Intelligence, pages 183-188*, 1992.

[9] T. Darell. Reinforcement learning of active recognition behaviors. Technical report, Interval Research, 1997.

[10] A.J. Davison and D.W. Murray. Mobile robot localisation using active vision. *Proceedings of the European Conference on Computer Vision*, 1998.

[11] A. Drake. *Observation of a Markov process through a noisy channel.* PhD thesis, Massachusetts Institute of Technology, 1962.

[12] E.A. Hansen. *Finite-memory control of partially observable systems.* PhD thesis, University of Massachusetts Amherst, 1998.

[13] E.A. Hansen. Solving POMDPs by searching in policy space. *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, pages 211-219*, 1998.

[14] M. Hauskrecht. Incremental methods for computing bounds in partially observable Markov decision processes. *Proceedings of AAAI-97, pages 734-739*, 1997.

[15] M. Hauskrecht. *Planning and control in stochastic domains with imperfect information.* PhD thesis, Massachusetts Institute of Technology, 1997.

[16] L. P. Kaelbling, M. Littman, and A. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence, 101: 1-2, pages 99-134*, 1998.

[17] L.P. Kaelbling, M.L. Littman, and A.W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research, Vol. 4, pages 237-285*, 1996.

[18] M.L. Littman. *Algorithms for sequential decision making.* PhD thesis, Brown University, 1996.

[19] J. Loch and S. Singh. Using eligibility traces to find the best memoryless policy in partially observable Markov decision processes. *Proceedings of the Fifteenth International Conference on Machine Learning*, 1998.

[20] W.S. Lovejoy. Computationally feasible bounds for partially observed Markov decision processes. *Operations Research, Vol 39, No. 1, pages 162-175*, 1991.

[21] W.S. Lovejoy. A survey of algorithmic methods for partially observable Markov decision processes. *Annals of Operations Research, Vol 28, pages 47-66*, 1991.

[22] A. McCallum. Overcoming incomplete perception with utile distinction memory. *Proceedings of the Tenth International Conference on Machine Learning*, 1993.

[23] A. McCallum. Instance-based utile distinctions for reinforcement learning with hidden state. *Proceedings of the Twelfth International Conference on Machine Learning*, 1995.

[24] A McCallum. *Reinforcement learning with selective perception and hidden state*. PhD thesis, University of Rochester, 1996.

[25] G.E. Monahan. A survey of partially observable Markov decision processes: Theory, models and algorithms. *Management Science, Vol 28, No. 1, pages 1-16*, 1982.

[26] M. Mundhenk, J. Goldsmith, C. Lusena, and E. Allender. Encyclopaedia of complexity results for finite-horizon Markov decision processes. Technical report, University of Kentucky, 1997.

[27] C.H. Papadimitriou and J.N. Tsitsiklis. The complexity of Markov decision processes. *Mathematics of Operations Research, Vol 12, Number 3, pages 441-450*, 1987.

[28] R. Parr and S. Russell. Approximating optimal policies for partially observable stochastic domains. *Proceedings of the International Joint Conference on Artificial Intelligence*, 1995.

[29] M.L. Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, New York, 1994.

[30] R.D. Smallwood and E.J. Sondik. The optimal control of partially observable Markov processes over a finite horizon. *Operations Research, Vol. 21, pages 1071-1088*, 1973.

[31] E.J. Sondik. The optimal control of partially observable Markov processes over the infinite horizon: discounted costs. *Operations Research, Vol. 26, No. 2, pages 282-304*, 1978.

[32] R.S. Sutton and A.G. Barto. *Reinforcement Learning (An Introduction)*. The MIT Press, 1998.

[33] M.J. Swain and M. Stricker. Promising directions in active vision. *International Journal of Computer Vision, Vol. 11, No. 2, pages 109-126*, 1993.

[34] R. Washington. BI-POMDP: Bounded, incremental partially-observable Markov-model planning. *Proceedings of the 4th European Conference on Planning*, 1997.

[35] S.D. Whitehead and D.H. Ballard. Learning to perceive and act by trial and error. *Machine Learning, Vol 7, No. 1, pages 45-83*, 1991.