A POMDP Approximation Algorithm that Anticipates the Need to Observe

Valentina Bayer BAYER@CS.ORST.EDU
Thomas Dietterich TGD@CS.ORST.EDU
Department of Computer Science, Oregon State University, Corvallis, OR 97331 USA

Abstract

This paper introduces the *even-odd POMDP*, an approximation to POMDPs in which the world is assumed to be fully observable every other time step. The even-odd POMDP can be converted into an equivalent MDP, the 2MDP, whose value function, V^*_{2MDP}, can be combined online with a 2-step lookahead search to provide a good POMDP policy. We prove that this gives an approximation to the POMDP's optimal value function that is at least as good as methods based on the optimal value function of the underlying MDP. We present experimental evidence that the method gives better policies, and we show that it can find a good policy for a POMDP with 10,000 states and observations.

1. Introduction

The Partially Observable Markov Decision Problem (POMDP) is the most general model of a single agent interacting with a partially observable environment. Consequently, solving POMDPs is a central goal of artificial intelligence. However, the great generality of the model also means that no single method can be expected to solve all POMDPs effectively and efficiently. Indeed, the problem of finding optimal solutions for POMDPs is PSPACE-hard (Papadimitriou & Tsitsiklis, 1987). The best exact algorithms for POMDPs can solve problems with around 100 states and a small number of possible observations (Cassandra, Littman, & Zhang, 1997).

In response to this state of affairs, several authors have explored methods for the *approximate* solution of POMDPs (Parr & Russell, 1995; Littman, Cassandra, & Kaelbling, 1995; Rodríguez, Parr, & Koller, 2000). Some of these methods involve first solving the underlying MDP to find the value function V^*_{MDP} of the optimal policy assuming that the states of the environment are fully observable. This value function can then be employed to construct approximately optimal policies for the original POMDP.

In some domains, the history of sensor readings and actions is sufficient to determine the true state of the environment. Such domains can be solved by computing the current state, and then using V^*_{MDP} to choose actions. The necessary sensor and environment models can be provided by the programmer or learned from the environment. An important discovery here was that a complete model of the environment and of the sensors is not needed. It is only necessary to make "utile distinctions"—that is, to model only those aspects of the environment required for representing the value functions of good policies (McCallum, 1995).

Unfortunately, there are many domains that are inherently uncertain: even when the agent has a complete and correct model of the environment, the agent can be "lost"—that is, the agent may have a large amount of uncertainty about the current state of the environment. In such domains, the optimal policy may involve "avoiding getting lost" (i.e., avoiding states where actions become highly uncertain) and also "acting when lost" (i.e., taking actions to gather information or to act robustly in the absence of information).

As an example of the need to avoid getting lost, consider a robot that has a choice of two different hallways to traverse. One hallway is completely dark and provides no visual landmarks. The other hallway is brightly-lit and has many visual landmarks. Even when the second hallway requires traveling a longer distance to reach the goal, it may still be the optimal choice, because the robot avoids getting lost in the dark hallway (and colliding with obstacles).

An example of the need to act when lost is the task of disease diagnosis. Given a patient's initial symptoms, a physician may be uncertain about the true state of the patient (i.e., the disease). The physician must have a policy for how to act (i.e., by performing tests and

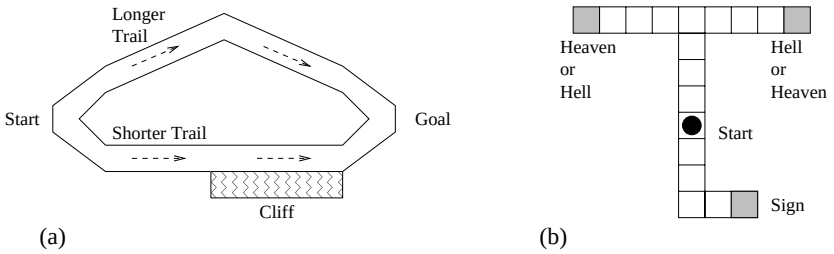

Figure 1. POMDPs illustrating (a) delayed need to observe and (b) delayed opportunity to observe.

prescribing therapies) under this uncertainty.

Fortunately, in cases where the effects of actions are immediately apparent, both of these problems can be solved by performing a shallow lookahead search, evaluating the leaf nodes in this search tree using V^*_{MDP}, and backing-up these values to choose the best action to perform. For the robot, a shallow lookahead search reveals that the robot rapidly becomes uncertain of its position. The expected value of the resulting positions according to V^*_{MDP} is poor, so the robot prefers the well-lit hallway. In the medical diagnosis case, many successful diagnostic systems have been based on a one-step value-of-information (VOI) calculation (Howard, 1966). The physician considers the expected utility of choosing a therapy immediately versus the expected utility of performing one test and then choosing the therapy after the test results are known. Greedy VOI often works extremely well, both for choosing the best test to perform and for deciding when to stop testing and recommend a therapy.

The most difficult POMDPs are those where the consequences of actions are not immediately apparent. Consider the "skier problem" in Figure 1(a). It involves a skiing robot that starts at a known location at the top of the mountain and must choose which trail to take. The upper trail is very safe—so safe that the robot can ski this route with its eyes closed, because the trail goes through a bowl-shape valley that naturally steers the robot down the center. The lower trail is initially just as safe as the upper one, but then it takes the skier along the side of a cliff. Here, the skiing robot must constantly observe its position to avoid falling off the cliff. Each time the robot uses its vision system, it consumes battery power, so the robot wants to minimize sensing. If this problem is solved while ignoring the costs of observation (i.e., computing V^*_{MDP}), the optimal policy will take the lower trail, because it is shorter. However, when the cost of observation is included, the upper policy is better. At the start state there is no apparent difference between the two paths, and a shallow lookahead search combined with V^*_{MDP} will choose the cliff trail. The key difficulty is that there is a *delayed need to observe* (or equivalently, a delayed risk of getting lost), and the shallow lookahead search cannot overcome this delay.

Table 1. Taxonomy of Difficult POMDPs

	Avoiding getting lost	Acting when lost
Immediate	Two hallways	Disease diagnosis
Delayed	Skier	Heaven-hell

The problem in Figure 1(b) is the "heaven and hell" domain, in which there are two terminal states, "heaven" and "hell", at the opposite ends of the top hallway. In the initial state, the robot knows its position, but it does not know which terminal state is heaven and which is hell, because the two possibilities are equally likely. There is one way of finding out this information: the robot can walk *down* the hallway, turn left, and read a sign that indicates where heaven is. So the optimal POMDP policy is to go down, read the sign, then go up and turn toward heaven. Now consider computing V^*_{MDP}. In the underlying MDP, there is no uncertainty about which terminal state is heaven, so the MDP optimal policy is to go up and turn appropriately. A shallow lookahead search using V^*_{MDP} will therefore go upwards and then turn arbitrarily either left or right. The difficulty here is that there is a *delayed opportunity to observe.*

Table 1 summarizes these four examples. "Immediate" and "Delayed" refer to the need or opportunity to observe.

This paper presents a new algorithm for approximate solution of POMDPs that works well when there is a delayed need to observe (lower left box, Table 1). The core idea is to define a new POMDP, the even-odd POMDP, in which the full state of the environment is observable at all times t where t is even. When t is odd, the environment returns the same information as in the original POMDP. Following an observation of Hansen (1994), we show that the even-odd POMDP can be converted into an equivalent MDP (the 2MDP)

with different actions and rewards. Let V^*_{2MDP} be the optimal value function for the 2MDP. Then we get an improved approximation to the optimal POMDP value function by performing a shallow lookahead search and evaluating the leaf states using V^*_{2MDP}.

The 2MDP will incorporate the costs of observation in cases where those costs become immediately apparent at some point in the future. For example, in the skier domain, as the skier approaches the cliff, it becomes immediately apparent (i.e., to a 2-step lookahead search) that there is a need to observe. Hence, the V^*_{2MDP} will include those observations—but only at times when t is odd! As the 2MDP is solved, these underestimated observation costs will be propagated backwards along the temporal sequence so that in the starting state at the top of the mountain, the robot skier will be able to make the optimal decision to take the upper trail.

The even-odd POMDP is not a complete solution to the problem of delayed need to observe, but we will see that it produces substantial improvements in some interesting synthetic problems. Furthermore, in domains where the observation costs are not immediately apparent, one can define a k-step MDP that observes the true state only once every k steps. As long as the need to observe reveals itself within k steps, the observation cost will be captured and propagated backwards through the state space.

This paper is organized as follows. Section 2 introduces our notations. Section 3 introduces the even-odd POMDP, the 2MDP, and shows that V^*_{2MDP} is a better approximation to the optimal value function of the POMDP, V^*_{POMDP}, than is V^*_{MDP}, the value function of the underlying MDP. It also proves that this improvement is maintained when these value functions are combined with a 2-step lookahead search. Section 4 presents three experimental studies showing the strengths and weaknesses of the 2MDP approximation. It demonstrates that the 2MDP approximation can solve a large POMDP. Conclusions are presented in Section 5.

2. POMDP Notations

A POMDP is a tuple $\langle S,\ A,\ O,\ P_{tr}(S|S,A),\ P_{obs}(O|S,A),\ R(S|S,A),\gamma\rangle$ where S is the set of states in the world, A is the set of actions, $P_{tr}(s_{t+1}|s_t,a_t)$ is the probability of moving to state s_{t+1} at time $t+1$, after performing action a_t in state s_t at time t, $R(s_{t+1}|s_t,a_t)$ is the expected immediate reward for performing action a_t in s_t causing a transition to s_{t+1}, O is the set of observations, $P_{obs}(o_t|s_t,a_{t-1})$ is the probability of observing o_t in state s_t at time t, after executing a_{t-1}, and γ is the discount factor.

A Markov decision process (MDP) is a simplification of the POMDP where the agent can observe the true state s of the environment after each action. Any POMDP can be converted into a continuous-state MDP called the belief MDP. The states in this MDP, called belief states, are probability distributions b such that $b(s_t) = P(s_t|a_0,o_1,\ldots,a_{t-1},o_t)$ is the agent's belief that the environment is in state s_t, given the entire action and observation history.

A policy π for an MDP is a mapping from states to actions. Hence, a policy for the belief MDP is a mapping from belief states to actions. The value function of a policy, $V^\pi(b) = E[\sum_{t=0}^{\infty}\gamma^t r_{t+1}]$, is the expected cumulative discounted reward of following policy π starting in belief state b. The optimal policy π^* maximizes $V^\pi(b)$ for all belief states. The value function of the optimal policy is denoted V^*.

We will say that a belief state b is "pure" if $b(s) = 1$ for some state s, and instead of $V(b)$ we will write $V(s)$.

3. The Even-Odd Approximation

3.1 Even-odd POMDP and Even MDP

Given a POMDP we can define a new POMDP, the even-odd POMDP, where everything is the same except that at even times t, the set of observations is the same as the set of states ($O = S$), and the observed state is the true underlying state ($P_{obs}(o|s,a) = 1$ iff $o = s$). Note that at even times, the belief state will be pure, but at odd times, the belief state may become "spread out." The optimal value function for the even-odd POMDP can be computed by converting it into an equivalent MDP, which we call the even MDP (abbreviated 2MDP). By "equivalent" we mean that the value function for the 2MDP is the same as the value function for the even-odd POMDP at even times t. At odd times, $V(b)$ is computed by performing a one-step lookahead search to reach an even time.

The 2MDP is constructed as follows. The states are the same as the even-odd POMDP's (world) states. Each action u in the 2MDP (Figure 2) is a tuple $\langle a, a'_1, a'_2, \ldots, a'_n\rangle$, where $n = |O|$. We will write $u[0] = a$ and $u[o_i] = a'_i$. An action u is executed in state s by first performing $a = u[0]$ in the even-odd POMDP. The agent will move to state s' with probability $P_{tr}(s'|s,a)$, and an observation o will be received with probability $P_{obs}(o|s',a)$. The agent then executes action $a' = u[o]$, which will cause a transition to state s'' with probability $P_{tr}(s''|s',u[o])$. This is the fully

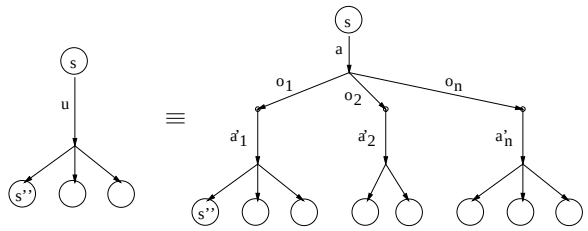

Figure 2. 2MDP has actions of the form $\langle a, a'_1, a'_2, \ldots, a'_n \rangle$.

observable result state in the 2MDP. The probability transition function is $P_{tr}(s''|s,u) = \sum_{s'} P_{tr}(s'|s,u[0]) \cdot \sum_o P_{obs}(o|s',u[0]) \cdot P_{tr}(s''|s',u[o])$. The immediate reward of executing action u in state s is $R(s,u) = \sum_{s'} P_{tr}(s'|s,u[0]) \cdot [R(s'|s,u[0]) + \gamma \sum_o P_{obs}(o|s',u[0]) \cdot \sum_{s''} P_{tr}(s''|s',u[o]) \cdot R(s''|s',u[o])]$. The discount factor is γ^2.

The "Bellman backup operator" for this 2MDP is

$$h_{2MDP}V(s) = \max_u R(s,u) + \sum_{s''} P_{tr}(s''|s,u) \cdot \gamma^2 V(s'').$$

By expanding the definitions, this can be simplified to

$$h_{2MDP}V(s) = \max_a R(s,a) + \sum_o \gamma \max_{a'} \sum_{s'} P_{tr}(s'|s,a) \cdot P_{obs}(o|s',a) \cdot \sum_{s''} P_{tr}(s''|s',a') \cdot (R(s''|s',a') + \gamma V(s'')),$$

where $R(s,a) = \sum_{s'} P_{tr}(s'|s,a) \cdot R(s'|s,a)$. Standard results tell us that h_{2MDP} is a max-norm contraction (under various conditions) and that it is monotonic (i.e., for any pair of value functions V_a and V_b, if for all s, $V_a(s) \leq V_b(s)$ then $h_{2MDP}V_a(s) \leq h_{2MDP}V_b(s)$). Furthermore, V^*_{2MDP} is the unique solution to the fixed-point equation $V = h_{2MDP}V$ (Bertsekas & Tsitsiklis, 1996).

3.2 Improved Approximation

Let V^*_{POMDP} be the optimal value function for the POMDP and V^*_{MDP} be the optimal value function for the underlying MDP. We show that V^*_{2MDP} is a better approximation to V^*_{POMDP} than V^*_{MDP}. First we prove that $V^*_{2MDP}(s) \leq V^*_{MDP}(s)$ for all states s. This, of course, makes sense, because the MDP optimal value function has perfect information about all the states, while the 2MDP only has perfect information about every other state. Then we apply a similar argument to show that $V^*_{POMDP}(s) \leq V^*_{2MDP}(s)$ for all $s \in S$. This will show that on pure belief states, V^*_{2MDP} is a better approximation to V^*_{POMDP}.

Theorem 1 *$V^*_{2MDP}(s) \leq V^*_{MDP}(s)$ for all $s \in S$.*

Proof: We begin by showing that $h_{2MDP}V^*_{MDP}(s) \leq V^*_{MDP}(s)$. Consider applying h_{2MDP} to V^*_{MDP}:

$$h_{2MDP}V^*_{MDP}(s) = \max_a R(s,a) + \sum_o \gamma \max_{a'} \sum_{s'} P_{tr}(s'|s,a)P_{obs}(o|s',a) \cdot \sum_{s''} P_{tr}(s''|s',a')(R(s''|s',a') + \gamma V^*_{MDP}(s'')).$$

If a^* is the action that achieves the maximum in $\max_a$,

$$h_{2MDP}V^*_{MDP}(s) = R(s,a^*) + \sum_o \gamma \max_{a'} \sum_{s'} P_{tr}(s'|s,a^*)P_{obs}(o|s',a^*) \cdot \sum_{s''} P_{tr}(s''|s',a')(R(s''|s',a') + \gamma V^*_{MDP}(s'')).$$

By applying the inequality $max_a \sum_s X(a,s) \leq \sum_s \max_a X(a,s)$ for a' and s', we can rewrite this as

$$h_{2MDP}V^*_{MDP}(s) \leq R(s,a^*) + \sum_o \gamma \sum_{s'} P_{tr}(s'|s,a^*)P_{obs}(o|s',a^*) \cdot \max_{a'} \sum_{s''} P_{tr}(s''|s',a')(R(s''|s',a') + \gamma V^*_{MDP}(s'')).$$

The last line is just the Bellman backup for the MDP:

$$h_{2MDP}V^*_{MDP}(s) \leq R(s,a^*) + \sum_o \gamma \sum_{s'} P_{tr}(s'|s,a^*)P_{obs}(o|s',a^*) \cdot V^*_{MDP}(s').$$

$V^*_{MDP}(s')$ does not depend on o, so $\sum_o P_{obs}(o|s',a^*)$ becomes 1 and drops out to give us

$$h_{2MDP}V^*_{MDP}(s) \leq R(s,a^*) + \sum_{s'} P_{tr}(s'|s,a^*)\gamma V^*_{MDP}(s').$$

The right hand side is a Bellman backup for a particular action a^*, so it is less than or equal to $V^*_{MDP}(s)$, which would be obtained by backing up the best action for the MDP. Hence, we obtain $h_{2MDP}V^*_{MDP}(s) \leq V^*_{MDP}(s)$ for all s. Because h_{2MDP} is monotonic, the inequality is true when we apply h_{2MDP} to both sides: $h^2_{2MDP}V^*_{MDP}(s) \leq h_{2MDP}V^*_{MDP}(s)$. By induction, $h^k_{2MDP}V^*_{MDP}(s) \leq V^*_{MDP}(s)$ for all k. $\lim_{k\to\infty} h^k_{2MDP}V^*_{MDP} = V^*_{2MDP} \leq V^*_{MDP}$. **Q.E.D.**

Theorem 2 *$V^*_{POMDP}(s) \leq V^*_{2MDP}(s)$ for all $s \in S$.*

Proof: By analogy with the 2MDP, we can define a 2^mMDP based on a 2^mPOMDP where the state is observable only every 2^m steps. The POMDP is the limit of this process as $m \to \infty$. We can also view the $2^{m+1}MDP$ as a 2MDP whose underlying MDP is a $2^m MDP$. Hence, we can apply Theorem 1, and conclude that $V^*_{2^{m+1}MDP} \leq V^*_{2^m MDP} \forall m \geq 1$. In the limit, $V^*_{POMDP}(s) \leq \ldots \leq V^*_{2^{m+1}MDP}(s) \leq V^*_{2^m MDP}(s) \leq \ldots \leq V^*_{2MDP}(s) \leq V^*_{MDP}(s) \forall s \in S$. **Q.E.D.**

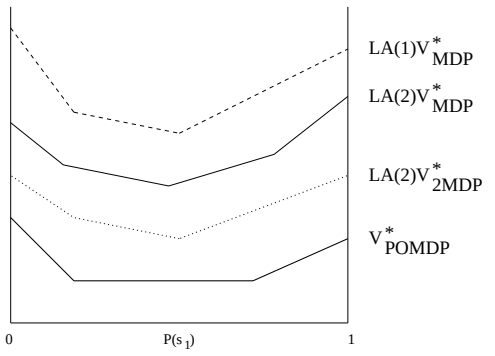

Figure 3. Schematic diagram of the optimal POMDP value function and three approximations to it for a 2-state finite-horizon POMDP.

These two theorems establish that V^*_{2MDP} is a better approximation to V^*_{POMDP} than V^*_{MDP} on pure belief states. We extend this result to arbitrary belief states b by considering a 2-step lookahead process. Let $LA(n)$ be an operator defined such that "$LA(n)V(b)$" estimates the value of belief state b by performing an n-step lookahead search and evaluating the fully observable leaf states using V. For example, $LA(1)$ can be written

$$LA(1)V(b) = \max_a \sum_s b(s) \sum_{s'} P_{tr}(s'|s,a)[R(s'|s,a) + \gamma V(s')].$$

Theorem 3 *For all belief states b, $V^*_{POMDP}(b) \leq LA(2)V^*_{2MDP}(b) \leq LA(2)V^*_{MDP}(b) \leq LA(1)V^*_{MDP}(b)$.*

Proof sketch: Because $V^*_{2MDP}(s) \leq V^*_{MDP}(s) \forall s$, we can use this at the leaves of the 2-step lookahead to prove that $LA(2)V^*_{2MDP}(b) \leq LA(2)V^*_{MDP}(b)$ for an arbitrary belief state b. The proof $LA(2)V^*_{MDP}(b) \leq LA(1)V^*_{MDP}(b)$ follows the pattern of the proof of Theorem 1. Using $LA(2)V^*_{2MDP}(b) \leq LA(1)V^*_{MDP}(b)$ and an argument similar to Theorem 2, we can show that $LA(2^{m+1})V^*_{2^{m+1}MDP}(b) \leq LA(2^m)V^*_{2^mMDP}(b)$ for all $m \geq 1$. Taking the limit as $m \to \infty$, we obtain $V^*_{POMDP}(b) \leq \ldots \leq LA(2^m)V^*_{2^mMDP}(b) \leq \ldots \leq LA(2)V^*_{2MDP}(b)$. **End of sketch**.

Figure 3 depicts the relationship between the value functions $V^*_{POMDP}(b) \leq LA(2)V^*_{2MDP}(b) \leq LA(2)V^*_{MDP}(b) \leq LA(1)V^*_{MDP}(b)$ for a finite horizon POMDP problem with 2 states. All four value functions are piecewise linear and convex. It is important to note that just because $LA(2)V^*_{2MDP}$ is a better approximation than $LA(2)V^*_{MDP}$, this does not guarantee that it will produce a better policy at run time. Nonetheless, it is usually the case that the more accurately we approximate the value function, the better the performance of a greedy policy that corresponds to that value function.

3.3 Even MDP Approximation Algorithm

We compute V^*_{2MDP} offline. To generate a policy for the original POMDP, we maintain a belief state, and at each time t we perform a 2-step lookahead (evaluating the leaf states with V^*_{2MDP}), and choose the action with the best backed-up value. This policy is called the $LA(2)V^*_{2MDP}$ policy. If the leaf states are evaluated with V^*_{MDP}, we obtain the $LA(2)V^*_{MDP}$ policy.

4. Experimental Studies

This section presents three experiments to demonstrate the strengths and weaknesses of the method.

4.1 Example with Delayed Need to Observe

In the first example (Figure 4), a skier is at the known start state on the left, and there are three trails leading down the mountain to circled absorbing states on the right. The skier has four possible actions: SE, N, E, and EO. SE is only available in the start state, and it deterministically takes the agent to the start of Trail 3 (reward -1). N deterministically moves the agent north one square (reward -4). Bumping against a "wall" leaves the state unchanged. E normally moves east with probability 0.5 and southeast with probability 0.5. The reward of E is normally -1, but -100 if the skier goes over the cliff. E moves east with probability 1 in the start state and in all states where there is no choice. The SE, N and E actions provide no observation information. The EO (east, observe) action behaves like E and it deterministically tells the skier its location (reward -2).

First consider what happens if Trail 2 is closed. In this case, the 2MDP approximation chooses the optimal policy. The policies corresponding to $LA(1)V^*_{MDP}$ and $LA(2)V^*_{MDP}$ will take Trail 3, whereas the optimal POMDP policy and the $LA(2)V^*_{2MDP}$ policy will take Trail 1. The 2MDP value function, V^*_{2MDP}, detects 2.5 out of 4 observations needed along Trail 3, but this is enough to make it choose the optimal path. Interestingly, because of the lookahead search, none of the policies goes over the cliff. The $LA(1)V^*_{MDP}$ policy never chooses to observe. Instead it relies on the N action to move away from the cliff. The $LA(2)V^*_{MDP}$ policy will observe just as much on Trail 3 as the POMDP optimal policy and the $LA(2)V^*_{2MDP}$ policy would if they were to take Trail 3. This shows that 2-step VOI computations are enough to permit $LA(2)V^*_{MDP}$ to act sensibly in this case.

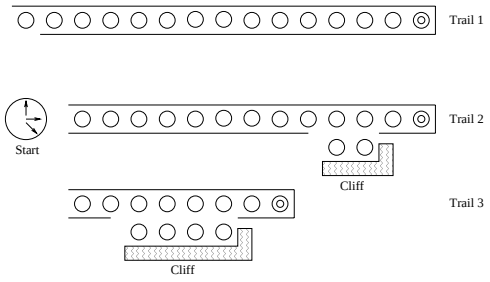

Figure 4. Three paths for skiing down a mountain. There is a delayed need to observe on Trails 2 and 3.

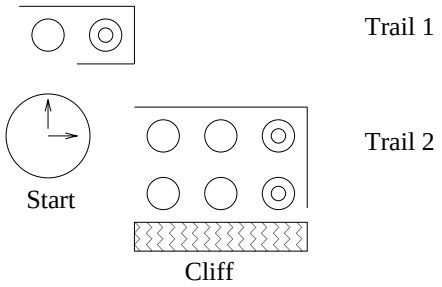

Figure 5. A second skier example: the MDP and 2MDP approximations take Trail 2, while the optimal POMDP policy takes Trail 1.

Now suppose we open Trail 2. In this case, the policies based on V^*_{MDP} still prefer Trail 3, and the optimal POMDP policy is still Trail 1. But the $LA(2)V^*_{2MDP}$ policy will choose Trail 2, because it detects too few of the necessary observations to prefer the optimal path (it only anticipates a need for 0.5 EO actions instead of the 2 EO actions that will actually be required). At run time, the policy does observe correctly.

Because the even MDP only needs to observe every other time step, it underestimates the observation costs of both Trail 2 and Trail 3.

4.2 Example of Gradually Getting Lost

Figure 5 shows a slightly different skiing problem. Here we have changed the dynamics so that the E action moves east with probability 0.9 and southeast with probability 0.1. The optimal POMDP policy is to take Trail 1, but all the approximations take Trail 2. The MDP approximations ($LA(1)V^*_{MDP}$ and $LA(2)V^*_{MDP}$) take Trail 2 for the same reasons as before: they cannot detect the need to observe. Unfortunately, the 2MDP approximation has the same problem: the probability 0.1 of moving southeast is not large enough to cause the 2-step lookahead to choose an EO action. So the 2MDP does not detect the need to observe. This illustrates a second weakness of the

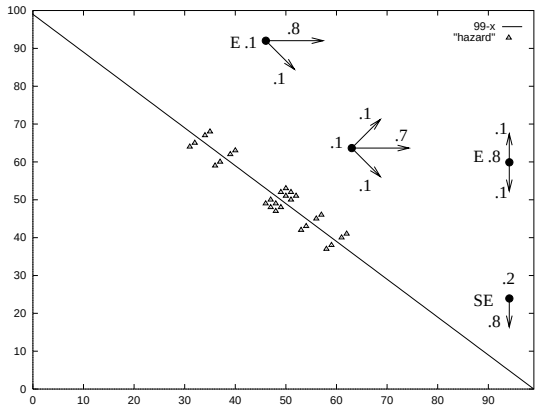

Figure 6. POMDP with 10000 states, 10001 observations and 6 actions. Hazards are shown as triangles. The (.7,.1,.1,.1) dynamics of an action are shown schematically along with special cases for E and SE.

2MDP approximation: the gradual accumulation of uncertainty. If uncertainty accumulates gradually, the 2MDP approximation will not detect the need to observe, and it will behave just like the MDP approximations. At execution time, the agent maintains a belief state, so it realizes when the uncertainty has accumulated, and it will choose to observe. So even in this case, it will usually avoid going over the cliff. The $LA(2)V^*_{MDP}$ and $LA(2)V^*_{2MDP}$ policies behave identically on this problem.

4.3 Large Example

To gain experience with a much larger problem, we designed the maze shown in Figure 6. The agent starts in the upper left corner of a 100×100 grid world, and it must reach an absorbing state in the lower right corner. Along the diagonal, there are several "hazard" states. Each time the agent enters a hazard state, it gets a reward of -1000, but the task does not terminate.

There are 6 actions: E, S, SE, EO, SO, and SEO. We describe the E action, as the others are analogous. The E action moves east with probability 0.7, northeast with probability 0.1, southeast with probability 0.1, and it stays in place with probability 0.1. If there is a wall to the north, E moves east with probability 0.8, southeast with probability 0.1, and stays in place with probability 0.1. A wall to the south is handled symmetrically. If there is a wall to the east, E moves north with probability 0.1, south with probability 0.1, and stays in place with probability 0.8. In all cases, the reward is -1. The E, S, and SE actions do not return any observation information. The EO, SO, and SEO actions have the same dynamics as E, S, and SE, but they return the exact location of the agent as well,

with a reward of -10. We believe this problem is too large to be solved by any of the exact algorithms.

We implemented the $LA(2)V^*_{MDP}$ and $LA(2)V^*_{2MDP}$ approximations. Value iteration required 30s to compute V^*_{MDP} and 444s to compute V^*_{2MDP}. To test the resulting online policies, we ran both for 100 trials.

The MDP optimal policy follows the diagonal towards the goal—there is enough space between the hazards to ensure that in a fully-observable world, the agent can avoid hitting any hazards (with high probability). When the $LA(2)V^*_{MDP}$ policy is executed online, it first performs a long series of SE actions (with no observation). This causes the belief state to spread out, and when the belief state starts to include some points near the hazards, it chooses to observe. It then exhibits two general behaviors. If it discovers that it is still near the diagonal, it continues to follow the diagonal, and it is forced to perform an average of 20 observations to avoid hitting the hazards. Otherwise, if it discovers that it has drifted away from the diagonal, then it follows a blind policy and goes "outside" the hazards. This actually leads to better performance, and online updating of V^*_{MDP} might yield improved performance in these cases. Over the 100 trials, the $LA(2)V^*_{MDP}$ policy never hit a hazard.

The even MDP determines that in states close to the hazards it is worth observing. V^*_{2MDP} includes these observation costs and propagates them back through the state space. Even if the true observation costs are underestimated, they are enough to make the 2MDP optimal policy go outside the hazards. When executed online, $LA(2)V^*_{2MDP}$ never performs any observations. It executes 30 SE steps, then it turns E for 15 steps, SE for 16 steps, then E, SE, E, SE, SE, E, followed by 40 SE actions, and finally it alternates single S actions with chains of SE actions. Over the 100 trials, this policy hit a hazard twice.

Figure 7 summarizes the behavior of the two policies by showing the steady state occupancy probabilities of each. The MDP approximation stays primarily on the diagonal. The distributions become concentrated near the hazards and near the start and end states. The 2MDP approximation follows the diagonal for a while and then moves E of it. (There is actually a tie, and it could have chosen to move S instead.) The probabilities become concentrated along the east wall north of the terminal state as the agent relies on the walls to "funnel" it into the terminal state.

Table 2 summarizes the total cost (= $-$reward) per trial received by the two methods. The MDP approximation gave an average cost of 239.52, whereas the 2MDP approximation's average cost was 160.34. The table reports a 95% bootstrap confidence interval, which shows that this difference is statistically significant. The $LA(2)V^*_{2MDP}$ policy correctly anticipated the need to observe along the diagonal, while $LA(2)V^*_{MDP}$ did not. The difference in performance is due to the extra cost of observing incurred by the $LA(2)V^*_{MDP}$ policy near the hazards.

Table 2. Cost ($-$reward) per trial, averaged over 100 trials

	min	max	mean	medn.	95%
MDP	136	398	239.52	271	[223,258.8]
2MDP	125	1136	160.34	140	[140.4,200.5]

We expect a continuous change in the policy obtained from the 2MDP approximation as the observation cost changes. If the observation cost decreases, the $LA(2)V^*_{2MDP}$ policy will move closer to the diagonal, and for zero observation cost it will be identical to the $LA(2)V^*_{MDP}$ policy. If the observation cost increases, the $LA(2)V^*_{2MDP}$ policy will move even further away from the hazards.

It is interesting to ask how close the 2MDP approximation comes to the optimal POMDP policy. From Theorem 2, we know that $V^*_{2MDP}(s_0) \geq V^*_{POMDP}(s_0)$ is an upper bound on the value of the optimal policy, where s_0 is the start state. In terms of cost, this is a lower bound, and in this problem it is 131.38. Hence, we can reasonably assume that the cost of the optimal POMDP policy is between 131.38 and 160.34 (the average cost of the 2MDP approximation). This shows that the 2MDP approximation is a large improvement over the MDP approximation.

5. Conclusions

The even MDP approximation provides a partial solution to the problem of avoiding getting lost when there is a delayed need to observe. While solving the 2MDP, a 2-step lookahead search detects cases where there is an immediately obvious need to observe. Value iteration then propagates the associated observation costs backward through the state space so that earlier states can detect a delayed need to observe. This makes V^*_{2MDP} more informed about future sensing in a POMDP framework than V^*_{MDP} is.

There are two limitations to the method which result in an underestimate of the true observation costs. First, if uncertainty accumulates gradually, the 2-step lookahead will not choose to observe, because the belief state after the first step will not be sufficiently diffused to make an observation worthwhile. One solution to this problem is to use a k-step lookahead, which will capture the costs of sensing that are apparent within

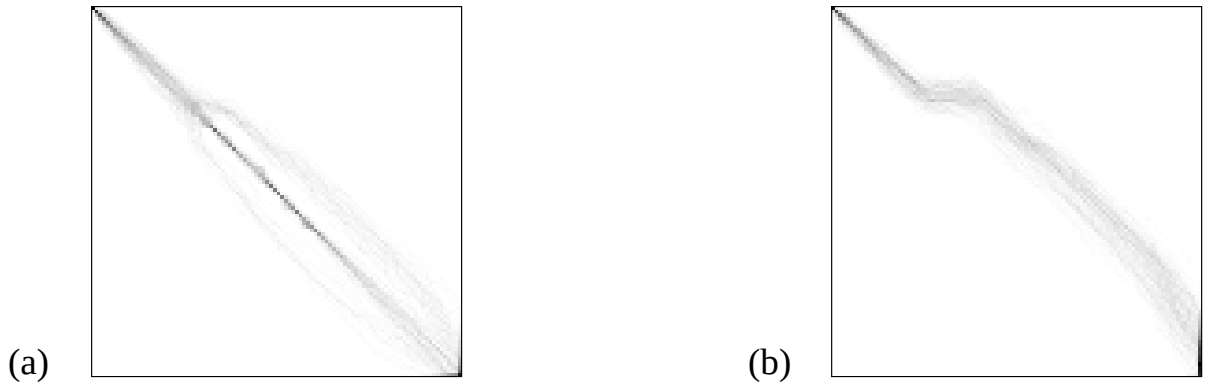

Figure 7. Steady state occupation probabilities for (a) the MDP approximation $LA(2)V^*_{MDP}$ and (b) the 2MDP approximation $LA(2)V^*_{2MDP}$ over 100 trials. The policy in (b) consistently avoids the hazards.

k steps. But the computational cost of this solution grows exponentially with k.

The second limitation arises from the fact that the 2MDP only needs to observe at odd times t. This is because at the end of the 2-step lookahead the states are assumed to be fully observable. A consequence of this is that the second actions $u[o]$ of the 2MDP will never be observation actions. We are currently exploring modifications to the 2MDP approximation that estimate observation costs at every step.

Despite these limitations, the 2MDP method performed well on the first skier example and on a very large maze problem. In addition, the value function V^*_{2MDP} can provide an upper bound on the value of the optimal POMDP policy, which can be useful for evaluating hand-derived policies.

Because the 2MDP method incorporates a 2-step value-of-information computation, it is also suitable for solving problems, such as medical diagnosis, where there is an immediate opportunity to observe, and the two hallways problem, where there is an immediate need to observe (see Table 1). However, the 2MDP method does not provide any solution to problems where there is a delayed opportunity to observe, such as the "heaven and hell" problem.

The 2MDP method can be applied to very large POMDPs because it only requires solving a Markov decision problem, and existing reinforcement learning algorithms can solve very large MDPs. This makes the 2MDP method the first scalable POMDP approximation that can anticipate the need to observe.

Acknowledgements

This research was supported by AFOSR F49620-9810375 and NSF 9626584-IRI. We also thank Michael Littman and Tony Cassandra for helpful advice.

References

Bertsekas, D. P., & Tsitsiklis, J. N. (1996). *Neuro-Dynamic Programming.* Athena Scientific.

Cassandra, A. R., Littman, M. L., & Zhang, N. L. (1997). Incremental pruning: A simple, fast, exact method for partially observable markov decision processes. In *In UAI-97*, pp. 54–61.

Hansen, E. A. (1994). Cost-effective sensing during plan execution. In *AAAI-94*, pp. 1029–1035 Cambridge, MA. AAAI Press/MIT Press.

Howard, R. A. (1966). Information value theory. *IEEE Trans. Sys. Sci. and Cyber., SSC-2*, 22–26.

Littman, M. L., Cassandra, A., & Kaelbling, L. P. (1995). Learning policies for partially observable environments: Scaling up. In *ICML-95*, pp. 362–370 San Francisco, CA. Morgan Kaufmann.

McCallum, R. A. (1995). Instance-based utile distinctions for reinforcement learning with hidden state. In *ICML-95*, pp. 387–396 San Francisco, CA. Morgan Kaufmann.

Papadimitriou, C. H., & Tsitsiklis, J. N. (1987). The complexity of Markov decision processes. *Math. Op. Rsrch., 12*(3), 441–450.

Parr, R., & Russell, S. (1995). Approximating optimal policies for partially observable stochastic domains. In *IJCAI-95*, pp. 1088–1094 San Francisco, CA. Morgan Kaufmann.

Rodríguez, A., Parr, R., & Koller, D. (2000). Reinforcement learning using approximate belief states. In *NIPS-12*, Cambridge, MA. MIT Press.