A POMDP Approximation Algorithm that Anticipates the Need to Observe

Valentina Bayer Zubek and Thomas Dietterich

Department of Computer Science, Oregon State University, Corvallis, OR 97331 USA
{bayer, tgd}@cs.orst.edu

Abstract. This paper introduces the *even-odd POMDP*, an approximation to POMDPs (Partially Observable Markov Decision Problems) in which the world is assumed to be fully observable every other time step. This approximation works well for problems with a *delayed need to observe*. 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 finds a good policy for a POMDP with 10,000 states and observations.

1 Introduction

The Partially Observable Markov Decision Problem (POMDP) is a general model of a single agent interacting with a partially observable environment. POMDP applications include quality control, autonomous robots, weapon allocation and medical diagnosis. Consequently, solving POMDPs is a central goal of artificial intelligence. However, the great generality of the model 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 undecidable [9].

Researchers have explored three main approaches to POMDP approximation. Any POMDP can be converted into a Markov Decision Problem (MDP) in belief space (see below), and one approach is to approximate this belief space MDP via value function approximation, factored decomposition, discretization of the belief state, or a combination of these [8, 11, 12, 3]. The second approach is based on solving the underlying MDP (i.e., the same POMDP but with a fully-observable state). This approach computes the optimal value function V^*_{MDP} and then applies it online to construct approximately-optimal policies for the POMDP [4]. Our paper presents a new method related to this approach. The third approach attempts to directly construct a finite-state controller that implements a good POMDP policy [6, 10].

What makes POMDP's difficult? The fundamental problem is that the agent may become "lost"–that is, the agent may have a large amount of uncertainty about the current state of the environment. Consequently, the optimal policy may involve "avoiding getting lost" and also "acting when lost".

As an example of the need to avoid getting lost (avoiding states where actions become highly uncertain), 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 prescribing therapies) under this uncertainty.

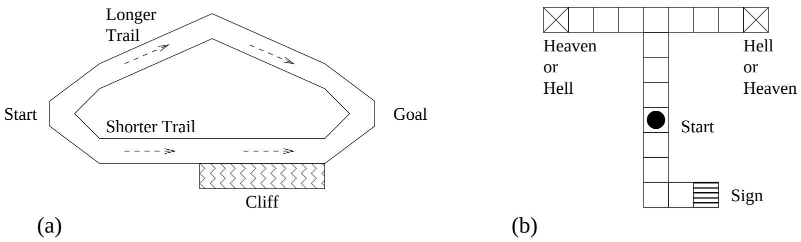

Fig. 1. POMDPs illustrating (a) delayed need to observe and (b) delayed opportunity to observe

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 [7]. 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.

The problem in Figure 1(b), due to S. Thrun, 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. Taxonomy of difficult POMDPs

	Avoiding getting lost (need to observe)	Acting when lost (opportunity to observe)
Immediate	Two hallways	Disease diagnosis
Delayed	Skier	Heaven-hell

This paper presents a new 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 [5], 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.

This paper is organized as follows. Section 2 introduces our notations. Section 3 introduces and theoretically justifies our approximation algorithm. Section 4 presents three experimental studies showing the strengths and weaknesses of the 2MDP approximation. 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).

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

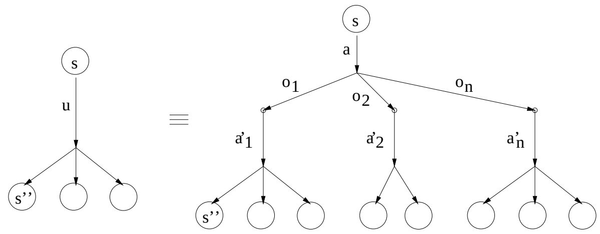

Fig. 2. 2MDP has actions of the form $\langle a, a'_1, a'_2, \ldots, a'_n \rangle$

The "Bellman backup operator" for this 2MDP is

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

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 $V = h_{2MDP}V$ [2].

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 is 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 first show that $h_{2MDP}V^*_{MDP}(s) \leq V^*_{MDP}(s)$. Consider applying h_{2MDP} to V^*_{MDP}. If a^* is the action that achieves the maximum in $\max_a$, then

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

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, so it becomes $V^*_{MDP}(s')$:

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

$V^*_{MDP}(s')$ does not depend on o, so $\sum_o P_{obs}(o|s',a^*)$ becomes 1:

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

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 $s \in S$. (see [1] for proof.)*

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,*

$$V^*_{POMDP}(b) \leq LA(2)V^*_{2MDP}(b) \leq LA(2)V^*_{MDP}(b) \leq LA(1)V^*_{MDP}(b) \ .$$

(see [1] for proof.)

Figure 3 depicts this relationship for a 2-state finite-horizon POMDP. All value functions are piecewise linear and convex.

3.3 Even MDP Approximation Algorithm

We can easily compute V^*_{2MDP} offline via value iteration. 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 instead evaluated with V^*_{MDP}, we obtain the $LA(2)V^*_{MDP}$ policy.

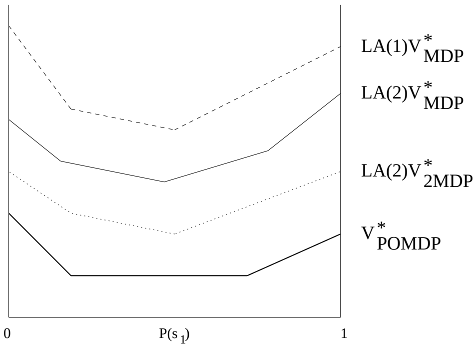

Fig. 3. Schematic diagram of the optimal POMDP value function and three approximations to it for a 2-state finite-horizon POMDP

4 Experimental Studies

4.1 Example with Delayed Need to Observe

In the first example (Figure 4, a), a skier is at the known start state on the left, and there are three trails leading down the mountain to circled absorbing states. The skier has four 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 deterministically tells the skier's location (reward -2).

First consider what happens if Trail 2 is closed. In this case, the 2MDP approximation chooses the optimal policy. The $LA(1)V^*_{MDP}$ and $LA(2)V^*_{MDP}$ policies 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.

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.

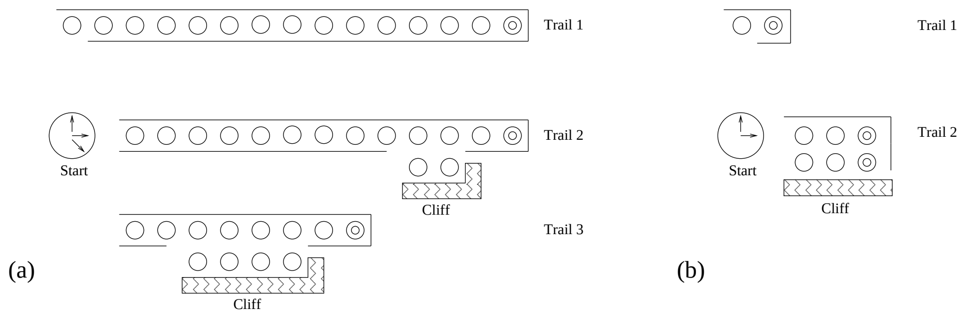

Fig. 4. (a)Three paths for skiing down a mountain. There is a delayed need to observe on Trails 2 and 3. (b) A second skier example

4.2 Example of Gradually Getting Lost

Figure 4 (b) 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 2MDP to choose an EO action during value iteration. So the 2MDP does not detect the need to observe. This illustrates a second weakness of the 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 A Large Example

Figure 5 presents a large maze problem with delayed need to observe. 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. The E, S, and SE actions do not return any observation information and have a reward of -1. 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 for POMDPs.

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

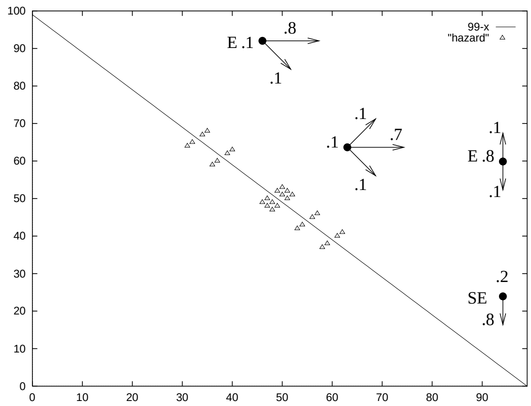

Fig. 5. POMDP with 10000 states, 10001 observations and 6 actions. Hazards are shown as triangles. The (.7,.1,.1,.1) transition probabilities of an action are shown schematically along with special cases for E and SE along the walls

agent can avoid hitting any hazards (with probability 1). 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} (e.g., linear Q-learning [8]) 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 6 shows the steady state occupancy probabilities for the two policies. 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. 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 for the two methods. The MDP approximation's average cost was 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 di-

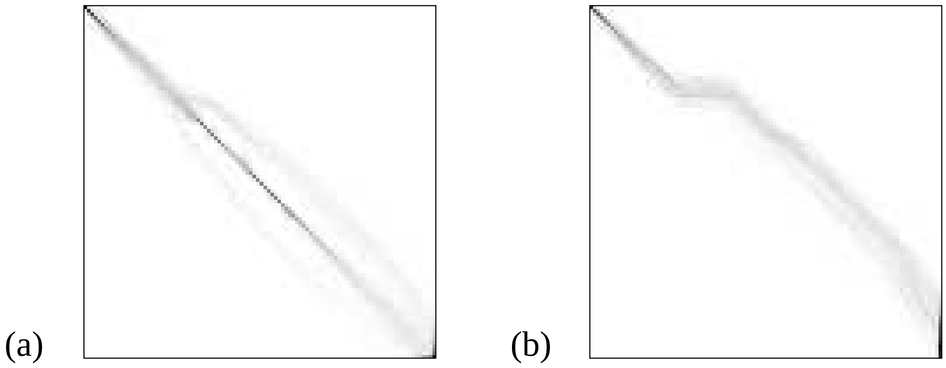

Fig. 6. Steady state occupation probabilities for (a) the MDP approximation $LA(2)V^*_{MDP}$ and (b) the 2MDP approximation $LA(2)V^*_{2MDP}$ over 100 trials

agonal, 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	median	95%
MDP	136	398	239.52	271	[223,258.8]
2MDP	125	1136	160.34	140	[140.4,200.5]

How close does the 2MDP approximation come 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 infer 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} much 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 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, and this implies that the second actions $u[o]$ of the 2MDP will never be observation actions. We are 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 large maze problem. In addition, the value function V^*_{2MDP} is an upper bound on the value of the optimal POMDP policy, which is 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. We also thank Michael Littman and Tony Cassandra for helpful advice.

References

1. Bayer, V., Dietterich, T.: A POMDP Approximation Algorithm that Anticipates the Need to Observe. Technical Report 00-30-01, Oregon State University, Dept. of Computer Science (2000)
2. Bertsekas, D. P. ,Tsitsiklis, J. N.: Neuro-Dynamic Programming. Athena Sci. (1996)
3. Bonet, B., Geffner, H.: Planning with Incomplete Information as Heuristic Search in Belief Space. AIPS 2000. AAAI Press/MIT Press (2000) 52–61
4. Cassandra, A. R., Kaelbling, L.P., Kurien, J. A.: Acting under Uncertainty: Discrete Bayesian Models for Mobil-Robot Navigation. IROS-96. IEEE (1996)
5. Hansen, E. A.: Cost-Effective Sensing During Plan Execution. AAAI-94. AAAI Press/MIT Press (1994) 1029–1035
6. Hansen, E. A.: Solving POMDPs by Searching in Policy Space. UAI-14. Morgan Kaufmann (1998) 211–219
7. Howard, R. A.: Information Value Theory. IEEE Trans. Sys. Sci. and Cyber., Vol. SSC-2 (1966) 22–26
8. Littman, M. L., Cassandra, A. R., Kaelbling, L.P.: Learning Policies for Partially Observable Environments: Scaling Up. ICML-95. Morgan Kaufmann (1995) 362–370
9. Madani, O., Hanks, S., Condon, A.: On the Undecidability of Probabilistic Planning and Infinite-Horizon POMDPs. AAAI-99. AAAI Press/MIT Press (1999) 541–548
10. McCallum, R. A.: Instance-based Utile Distinctions for Reinforcement Learning with Hidden State. ICML-95. Morgan Kaufmann (1995) 387–396
11. Parr, R., Russell, S.: Approximating Optimal Policies for Partially Observable Stochastic Domains. IJCAI-95. Morgan Kaufmann (1995) 1088–1094
12. Rodríguez, A., Parr, R., Koller, D.: Reinforcement Learning Using Approximate Belief States. NIPS-12, MIT Press (2000)