Probabilistic inference for computing optimal policies in MDPs

Marc Toussaint **Amos Storkey**
School of Informatics, University of Edinburgh
Edinburgh EH1 2QL, Scotland, UK
mtoussai@inf.ed.ac.uk, amos@storkey.org

Abstract

We investigate how the problem of planning in a stochastic environment can be translated into a problem of inference. Previous work on planning by probabilistic inference was limited in that a total time T has to be fixed and that the computed policy is not optimal w.r.t. expected rewards. The generative model we propose considers the total time T as a random variable and we show equivalence to maximizing the expected future return for arbitrary reward functions. Optimal policies are computed via Expectation-Maximization.

1 Introduction

The problems of planning in stochastic environments and inference in Markovian models are closely related, in particular in view of the challenges both of them face: e.g., coping with very large state spaces spanned by multiple state variables, or realizing planning (or inference) in continuous state spaces. Both fields developed techniques to address these problems. They include, in the field of planning, Generalized Prioritized Sweeping [1], Factored Markov Decision Processes [3, 9, 7], or Abstract Hidden Markov Models [8]. On the other hand, in the field of probabilistic inference, techniques for approximate inference on factorial latent representations (e.g., Factorial Hidden Markov Models [6]) and an enormous amount of work on approximate inference in continuous state spaces does exist (ranging from particle filters to, e.g., Assumed Density Filtering; see, e.g., [10] for an overview).

In view of these similarities one may ask whether approaches to probabilistic inference can *exactly* be transferred to the problem of planning, in other words, whether one can translate the problem of planning exactly into a problem of inference. Clearly, the aim of this is to connect both fields more strongly but eventually also to apply, e.g., efficient methods of probabilistic inference directly in the realm of planning. Currently, most approaches to planning are based on estimating value functions, which is conceptually very different to inferring a posterior over actions.

Bui et al. [4] have used inference on Abstract Hidden Markov Models for policy recognition, i.e., for reasoning about executed behaviors, but do not address the problem of computing optimal policies from such inference. Attias [2] recently proposed a framework which suggests a straight-forward way to translate the problem of planning to a problem

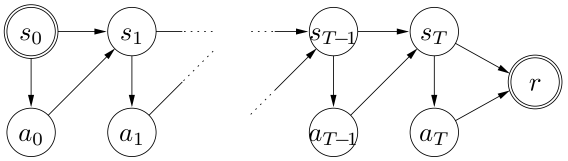

Figure 1: A single MDP of finite time T emits rewards only from the last state and action s_T and a_T. Our full model is the mixture of finite-time MDPs given a time prior $P(T)$.

of inference: A Markovian state-action model is assumed, which is conditioned on a start state $s_0 = A$ and a goal state $s_T = B$. Here, however, the total time T has to be fixed *ad hoc*—which is a first problem with the approach since the original planning problem does not fix such a time. Then standard inference techniques are used to infer a posterior over the actions taken on paths from s_0 to s_T. Action selection is realized by choosing the maximum posterior actions. However—and this is the second problem—it turns out that this maximum posterior action is not optimal in the traditional sense, i.e., it does not provide a policy that maximizes the probability to reach the goal as optimal policies computed by means of a value function do. Our work takes [2] as a starting point, but solves these two problems:

(1) We treat the total time T as a random variable, which has not to be fixed ad hoc. Instead, a prior $P(T)$ is specified which can be chosen uniform (to solve the infinite horizon scenario) or exponentially decaying (to solve the discounted future return scenario) or specified as a time window (to solve time-constrained scenarios).

(2) We guarantee that the computed policies are optimal in the traditional sense of optimizing the expected future reward for arbitrary reward functions. This is achieved by formulating a certain reward likelihood and treating the policy as a *parameter* of the Markovian model which is optimized by Expectation-Maximization.

The primary scope of this paper is to gain considerable insight in this alternative way to solve MDPs and its relations to standard policy iteration. In fact, the backward propagation in the E-step can well be compared to value function approximation and the M-step to a policy update, whereas the forward propagation has no traditional analogue. Also, the inference framework provides us with posteriors over state-action sequences as well as, e.g., over the total time to reach a reward—such information is fundamentally not given by traditional value function approaches but can be exploited by heuristics to prune computations on state space regions that are very unlikely to be visited. The limited size of the paper prohibits to go ahead and address the probably most interesting applications: probabilistic inference techniques on factored and continuous spaces. But considering exact inference on discrete state spaces is here sufficient to introduce in depth the framework to apply probabilistic inference techniques to the problem of solving MDPs.

The next two sections introduce the generative model, show equivalence to maximizing expected future rewards, and describe inference to compute optimal policy. The experiment in Section 4 briefly demonstrates Probabilistic Inference Planning (PIP) on a stochastic maze and compares it to Prioritized Sweeping.

2 Generative model: a mixture of finite-time MDPs

A Markov-Decision Process (MDP) is given in terms of a probability $P(s_{t+1} \,|\, a_t, s_t)$ of transitioning from a state s_t to a state s_{t+1} when executing action a_t at a given time t. Further, let $P(a_t \,|\, s_t \,; \boldsymbol{\pi})$ be the probability of choosing an action a_t when in state s_t at a given time t. We assume that this probability is directly parameterized in terms of a *policy*

$\boldsymbol{\pi}$ such that $P(a_t = a \,|\, s_t = i \,; \boldsymbol{\pi}) = \pi_{ai}$, where the numbers $\pi_{ai} \in [0, 1]$ are normalized w.r.t. a. We will assume the MDP to be stationary, i.e., $P(s_{t+1} \,|\, a_t, s_t)$ is independent of t.

Figure 1 displays the graphical model of a *finite-time MDP* for a fixed total time T which is augmented with a binary reward variable r. For such a finite-time MDP we assume that rewards are emitted only from the final state and action, given by $P(r \,|\, a_T, s_T)$. This defines the joint probability over state-action sequences and reward as

$$\begin{aligned} &P_T(r, s_{0:T}, a_{0:T} \,; \boldsymbol{\pi}) \\ &\quad = P(r \,|\, a_T, s_T)\; P(a_0 \,|\, s_0 \,; \boldsymbol{\pi})\; P(s_0) \prod_{t=1}^{T} P(a_t \,|\, s_t \,; \boldsymbol{\pi})\; P(s_t \,|\, a_{t-1}, s_{t-1}) \;. \end{aligned} \tag{1}$$

We used the subscript T for the joint P_T to keep in mind that this is the joint for a finite-time MDP of given fixed total time T.

Solving a finite-time MDP in a fixed time scenario is an interesting case. However, in many scenarios it is not a priori clear how to choose a fixed total time T. We can cope with such cases by also treating T as a random variable. Hence, the full model we consider in this paper if the *mixture of finite-time MDPs* which defines the joint over state-action sequences, the total time, and reward as

$$P(r, s_{0:T}, a_{0:T}, T \,; \boldsymbol{\pi}) = P_T(r, s_{0:T}, a_{0:T} \,; \boldsymbol{\pi})\; P(T) \;. \tag{2}$$

Here, $P(T)$ is a prior over the total time, which has to be given as part of the problem formulation (see below). Note that each finite-time MDP shares the same transition probabilities and is parameterized by the same policy $\boldsymbol{\pi}$. Throughout the next two sections it will become more clear why we consider a mixture of finite-time MDPs instead of a single time-unlimited MDP where rewards could be emitted at any time.

Equivalence between reward likelihood and expected future return. Let us first recall the definition of the expected future return. Traditionally one considers a reward function $\mathcal{R}_{ai} = \mathrm{E}\,\{r_t \,|\, a_t{=}a, s_t{=}i\}$ describing the expected reward when taking action k in state i. Assuming that the reward expectations are in the interval $[0, 1]$ (otherwise we rescale) allows us to consider the actual reward value to be binary, $r_t \in \{0, 1\}$, $\mathcal{R}_{ai} = P(r_t{=}1 \,|\, a_t{=}a, s_t{=}i)$. A standard optimization objective is the discounted sum $R = \sum_{t=0}^{\infty} \gamma^t\, r_t = r_0 + \gamma\, r_1 + \gamma^2\, r_2 + \cdots$ of future rewards, where $\gamma \in [0, 1]$ is the discount factor. Note that R is a random variable just as all the rewards r_t are. The expected future return given that the current state $s_0 = i$ is

$$V^{\pi}(i) = \mathrm{E}\,\{R \,|\, s_0{=}i \,; \boldsymbol{\pi}\} = \sum_{t=0}^{\infty} \gamma^t\; \mathrm{E}\,\{r_t \,|\, s_0{=}i \,; \boldsymbol{\pi}\} \tag{3}$$

Going back to our model, the reward likelihood of a *single* MDP of finite-time T is

$$L^{\pi}_T(i) = P_T(r{=}1 \,|\, s_0{=}i \,; \boldsymbol{\pi}) = \mathrm{E}_T\,\{r \,|\, s_0{=}i \,; \boldsymbol{\pi}\} \;. \tag{4}$$

Note that the expectation term $\mathrm{E}_T\,\{r \,|\, s_0{=}i \,; \boldsymbol{\pi}\}$ here is exactly the same as the terms $\mathrm{E}\,\{r_t \,|\, s_0{=}i \,; \boldsymbol{\pi}\}$ for $t = T$ in equation (3) for the value function: we are taking the expectation w.r.t. a full probabilistic forward-sweep through the MDP, from time 0 to time T, given the policy $\boldsymbol{\pi}$. (Recall that all MDPs share the same transition probabilities.) Accordingly, for the mixture of finite-time MDPs, the reward likelihood is

$$L^{\pi}(i) = P(r{=}1 \,|\, s_0 = i \,; \boldsymbol{\pi}) = \sum_T L^{\pi}_T(i)\; P(T) = \sum_T P(T)\; \mathrm{E}_T\,\{r \,|\, s_0{=}i \,; \boldsymbol{\pi}\} \;. \tag{5}$$

Hence, choosing the discount time prior $P(T) = \gamma^T/(1-\gamma)$, we have

$$L^{\pi}(i) = \frac{1}{1-\gamma}\; V^{\pi}(i) \tag{6}$$

and we showed that maximizing the expected discounted future reward $V^\pi(i)$ w.r.t. the policy $\boldsymbol{\pi}$ for a given start state i is equivalent to maximizing the reward likelihood $L^\pi(i)$ w.r.t. the model parameters $\boldsymbol{\pi}$. Beyond that, choosing other time priors allows us to optimize problems where the time of receiving rewards or reaching the goal matters other than just by discount: (1) Fixing T to a value T^* by setting $P(T) = \delta_{T=T^*}$ corresponds to the task of being at the goal in exactly T^* time steps. (2) Choosing a window prior ($P(T) = \text{const.} \iff T_m \le T \le T_M$, $P(T) = 0$ otherwise) leads to policies that are lazy (or security oriented) for states in the goal neighborhood while being greedily goal-directed (even at the cost of risky transitions) for states critically remote from the goal.

3 An EM-algorithm for computing the optimal policy

Formulating the objective function in terms of a likelihood of observed variables (s_0 and r) allows us to apply Expectation-Maximization to find optimal parameters (the policy $\boldsymbol{\pi}$) of our model. All action and state variables (except for s_0) are hidden variables. The E-step will compute posteriors over state-action sequences conditioned on observations $s_0 = A$ and $r = 1$, while the M-step adapts the parameters (the policy). Conceptually, the E-step in this Markovian model is straight-forward. However, the special structure of the finite-time MDPs will allow for certain simplifications and save us from performing separate inference sweeps in all finite-time MDPs.

E-step: forward-backward in all MDPs synchronously. Since we assume the transition probabilities to be stationary, we may use the simpler notations $p(j|a,i) = P(s_{t+1}\!=\!j \,|\, a_t\!=\!a, s_t\!=\!i)$ and $p(j|i\,;\boldsymbol{\pi}) = P(s_{t+1}\!=\!j \,|\, s_t\!=\!i\,;\boldsymbol{\pi}) = \sum_a p(j|a,i)\;\pi_{ai}$. Further, as a "seed" for backward propagation, we define

$$\begin{aligned}\bar\beta_i &= P(r\!=\!1 \,|\, s_T\!=\!i\,;\boldsymbol{\pi})\\ &= \sum_a P(r\!=\!1 \,|\, a_T\!=\!a, s_T\!=\!i)\; P(a_T\!=\!a \,|\, s_T\!=\!i\,;\boldsymbol{\pi}) = \sum_a \mathcal{R}_{ai}\;\pi_{ai}\;. \qquad (7)\end{aligned}$$

In the E-step, we consider a fixed given policy $\boldsymbol{\pi}$ and all the quantities we compute depend on $\boldsymbol{\pi}$ even if not explicitly annotated. For a *single* MDP of finite time T the forward and backward propagation computes

$$\alpha_i(0) = \delta_{i=A}\;, \quad \alpha_i(t) = P(s_t\!=\!i \,|\, s_0\!=\!A\,;\boldsymbol{\pi}) = \sum_j p(i|j\,;\boldsymbol{\pi})\;\alpha_j(t-1)\;, \qquad (8)$$

$$\beta_i(T) = \bar\beta_i\;, \quad \beta_i(t) = P(r\!=\!1 \,|\, s_t\!=\!i\,;\boldsymbol{\pi}) = \sum_j p(j|i\,;\boldsymbol{\pi})\;\beta_j(t+1)\;. \qquad (9)$$

The first observation is that all the α-quantities do in no way depend on T, i.e., they are valid for all MDPs of any finite time T. This is not true for the β-quantities. However, using a simple trick to index the β's backward in time (with the 'time-to-go' τ), we get

$$\beta_i(0) = \bar\beta_i\;, \quad \beta_i(\tau) = P(r\!=\!1 \,|\, s_{T-\tau}\!=\!i\,;\boldsymbol{\pi}) = \sum_j p(j|i\,;\boldsymbol{\pi})\;\beta_j(\tau-1)\;. \qquad (10)$$

Indexed in that way, all β-quantities do indeed not depend on T. For a specific MDP of finite time T, setting $\tau = T - t$ retrieves the β-quantities defined by (9).

This means that we can perform α- and β-propagation in parallel, incrementing t and τ synchronously, and can retrieve the α's and β's for all MDPs of any finite time T. Although we introduce a mixture of MDPs we only have to perform a single forward and backward sweep. This procedure is, from the point of view of ordinary Hidden Markov Models, quite unusual—it is possible because in our setup we only condition on the very first ($s_0 = A$) and very last state ($r = 1$). Apart from the implementation of discounting with the time prior, this is the main reason why we considered the mixture of finite-time MDPs in the first place instead of a single time-unbounded MDP that could emit rewards at any times.

During α- and β-propagation, we can compute the state posteriors, which are clearly not independent of the total time T. Conditioned on a specific T (i.e., w.r.t. the MDP of specific length T) the state posteriors can be derived form the calculated α- and β- quantities as

$$\gamma_i(t,\tau) = P(s_t\!=\!i \,|\, s_0\!=\!A, r\!=\!1, T\!=\!t+\tau\,;\boldsymbol{\pi}) = \frac{1}{Z(t,\tau)}\; \beta_i(\tau)\; \alpha_i(t)\ , \tag{11}$$

with the normalization

$$Z(t,\tau) = \sum_i \alpha_i(t)\; \beta_i(\tau) = \frac{P(s_0\!=\!A, r\!=\!1 \,|\, T\!=\!t+\tau\,;\boldsymbol{\pi})}{P(s_0\!=\!A)} = Z(t+\tau)\ . \tag{12}$$

Time posterior. It is interesting to realize that the normalization constant Z only depends on the sum $t+\tau$, i.e., we can define $Z(t+\tau) = Z(t,\tau)$. Further, $Z(t+\tau)$ is related to the likelihood $P(s_0\!=\!A, r\!=\!1 \,|\, T\!=\!t+\tau\,;\boldsymbol{\pi})$, i.e., the likelihood that the start state is A and the final state leads to reward if one assumes an MDP of specific length T. Using Bayes rule, this leads us to the *posterior over* T (in abbreviated notation),

$$P(T \,|\, s_0, r\,;\boldsymbol{\pi}) = \frac{P(s_0, r \,|\, T\,;\boldsymbol{\pi})}{P(s_0, r\,;\boldsymbol{\pi})}\; P(T) = \frac{Z(T)\; P(T)}{P(r \,|\, s_0\,;\boldsymbol{\pi})}\ . \tag{13}$$

An intuitive understanding of $Z(t+\tau)$ is the following: The α- and β-propagations could be grasped as diffusion processes emanating from $s_0 = A$ and from rewarded states (as given by $\bar\beta_i$), respectively. Assume we have iterated α- and β-propagation up to some t and τ, respectively. Then we can calculate the quantity $Z(t+\tau) = \sum_i \alpha_i(t)\; \beta_i(\tau)$ which is an inner product of the two distributions $\alpha_i(t)$ and $\beta_i(\tau)$ over the state space—for simplicity let us call it the *overlap* between the α- and β-distribution. If there is no overlap, $Z(t+\tau) = 0$, then $P(s_0, r \,|\, T = t+\tau\,;\boldsymbol{\pi})$ is also zero, which means that the Markov structure of specific finite time $T = t+\tau$ has zero likelihood of $s_0 = A$ and $r = 1$, which also means that there exists no path from A to rewards in $T = t+\tau$ time steps. Thus, also the time posterior $P(T\!=\!t+\tau \,|\, s_0, r\,;\boldsymbol{\pi}) \propto Z(t+\tau)$ is zero. More generally, the overlap of the α- and β- distribution after t and τ iterations, respectively, is proportional to the time posterior $P(T\!=\!t+\tau \,|\, s_0, r\,;\boldsymbol{\pi})$. Tracking this overlap during α- and β-propagation thus allows us to monitor the time posterior on the fly.

Action and state posteriors. Let us briefly derive the action and state posteriors that are relevant for the later discussion. For convenience, let

$$q_{ai}(t,\tau) = P(r\!=\!1 \,|\, a_t\!=\!a, s_t\!=\!i, T\!=\!t+\tau\,;\boldsymbol{\pi}) = \sum_j p(j|i,a)\; \beta_j(\tau-1) = q_{ai}(\tau)\ . \tag{14}$$

As expected, this quantity is independent of A and t because we conditioned on s_t and the history before time t becomes irrelevant in the Markovian setup. We may use the simpler notation $q_{ai}(\tau) = q_{ai}(t,\tau)$. Multiplying with the time prior and eliminating the total time we get the *action-conditioned reward likelihood*

$$P(r\!=\!1 \,|\, a_t\!=\!a, s_t\!=\!i\,;\boldsymbol{\pi}) = \frac{1}{C}\; \sum_\tau P(T\!=\!t+\tau)\; q_{ai}(\tau)\ , \tag{15}$$

where $C = \sum_{\tau'} P(T\!=\!t+\tau')$. Further, with Bayes rule we get the *action posterior*

$$P(a_t\!=\!a \,|\, s_t\!=\!i, r\!=\!1\,;\boldsymbol{\pi}) = \frac{\pi_{ai}}{C'}\; \sum_\tau P(T\!=\!t+\tau)\; q_{ai}(\tau)\ , \tag{16}$$

where $C' = P(r\!=\!1 \,|\, s_t\!=\!i\,;\boldsymbol{\pi})\; \sum_{\tau'} P(T\!=\!t+\tau')$ can be computed from normalization. Finally, in the experimental results we will display the posterior probability of visiting a certain state i. This is derived as

$$P(i \in s_{0:T} \,|\, s_0, r\!=\!1\,;\boldsymbol{\pi}) = \sum_T \frac{P(T)\; Z(T)}{P(r\!=\!1 \,|\, s_0\,;\boldsymbol{\pi})}\; \Big[1 - \prod_{t=0}^{T} \big[1 - \gamma_i(t, T-t)\big]\Big]\ . \tag{17}$$

M-step: the policy update. In the M-step, we generally assign values to the parameters $\boldsymbol{\pi}$ that maximize the expected log-likelihood, where expectation is taken w.r.t. the posterior over latent variables found in the E-step. Here this simply amounts to maximizing (15) w.r.t. a,

$$\pi_{ai} \leftarrow \delta_{a=a^*(i)} \;, \quad a^*(i) = \underset{a}{\operatorname{argmax}}\, P(r\!=\!1 \,|\, a_t\!=\!a, s_t\!=\!i) \;. \tag{18}$$

We will denote the number of EM-iterations by k.

Relation to policy iteration. The β-quantities computed during backward propagation are actually the value function for a single MDP of finite time T. More precisely, comparing (9) with the reward likelihood (4) for the MDP of finite time T and the definition (3) of the value function, we have $\beta_i(\tau) \propto \big(V^\pi(i)$ of the MDP of time $T = \tau\big)$. Accordingly, the full value function is the mixture of the β's, $V^\pi(i) \propto \sum_T P(T)\, \beta_i(T)$, when a discount time prior is chosen.

The quantities $q_{ai}(\tau)$ defined in (14) are equally related to the Q-function, in the sense that $Q^\pi(a,i) \propto \sum_T P(T)\, q_{ai}(T)$ for a discount time prior. Note that this is also the action-conditioned reward likelihood (15) and, interestingly, the action posterior (16) is proportional to $\pi_{ai}\, Q^\pi(a,i)$. Given this relation to the Q-function, we find that the M-step (18) is, for the choice of a discount time prior, the standard policy update in Policy Iteration [11], maximizing the Q-function w.r.t. the action a in state i. In contrast, the forward propagation and—related to that—the derived state and action posteriors have no traditional analogue. But knowing these posteriors is very useful, e.g., for pruning unnecessary computations as discussed in the next section.

Pruning computations. Assume that we fixed the maximum allowed time T by some upper limit T_M (e.g., by having a window prior or by deciding on a cutoff time heuristically, see below). Then there are potentially large regions of the state space on which we may prune computations, i.e., states i for which the posterior $\gamma_i(t,\tau) = 0$ for any t and τ with $t + \tau \le T_M$. Let us consider the α-propagation only (all statements apply conversely for the β-propagation). At iteration time t, let us define a set of states

$$S_\alpha(t) = \{ i \in S \,|\, \alpha_i(t) \neq 0 \;\wedge\; (t < T_M/2 \vee \beta_i(T_M - t) \neq 0) \} \;. \tag{19}$$

Further, given $\beta_i(\tau)\!=\!0 \Rightarrow \forall \tau' \!\le\! \tau \!:\! \beta_i(\tau') = 0$, it follows

$$\begin{aligned} i \in S_\alpha(t) &\Leftarrow\; \alpha_i(t) \neq 0 \;\wedge\; \beta_i(T_M - t) \neq 0 \\ &\Leftarrow\; \exists_{\tau' \le T_M - t} :\; \alpha_i(t) \neq 0 \;\wedge\; \beta_i(\tau') \neq 0 \\ &\iff \exists_{\tau : t+\tau \le T_M} \;:\; \gamma_i(t,\tau) \neq 0 \end{aligned} \tag{20}$$

Thus, every state that is potentially visited at time t (for which $\exists_{\tau:t+\tau\le T_M} \,:\, \gamma_i(t,\tau) \neq 0$) is included in $S_\alpha(t)$. We will exclude all states $i \notin S_\alpha(t)$ from the propagation procedure and not deliver their messages. The constraint $t < T_M/2$ concerning the β's was inserted in the definition of $S_\alpha(t)$ only because of the feasibility of computing $S_\alpha(t)$ at iteration time t. Initializing $S_\alpha(0) = \{A\}$, we can compute $S_\alpha(t)$ recursively via

$$S_\alpha(t) = \Big[S_\alpha(t\!-\!1) \cup \mathsf{OUT}(S_\alpha(t\!-\!1)) \Big] \;\cap\; \left\{ \begin{array}{cc} S & t < T_M/2 \\ \{ i :\; \beta_i(T_M - t) \neq 0 \} & t \ge T_M/2 \end{array} \right. \;,$$

where $\mathsf{OUT}(S_\alpha(t\!-\!1))$ is the set of states which have non-zero-probability transitions from states in $S_\alpha(t\!-\!1)$.

When we choose a window time prior, then T_M is taken to be the upper limit of this window. For the discount prior, we can use a time cutoff for which we expect further contributions

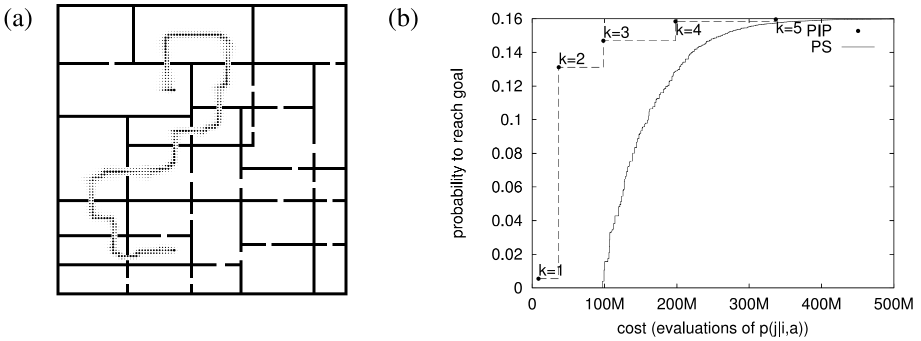

Figure 2: (a) State visiting probability calculated by PIP for some start and goal state. The radii of the dots are proportional to (17). (b) The probability to reach the goal (for PIP) and the value calculated for the start state (PS) against the cost of the planning algorithms (measured by evaluations of $p(j|i,a)$) for both start/goal configurations.

to be insignificant. The choice of this cutoff involves a payoff between computational cost and accuracy. Let T_0 be the minimum time for which $Z(T_0) \neq 0$. It is clear that the cutoff needs to be greater than T_0. In the experiment we will use an increasing schedule for the cutoff time, $T_M = (1 + k/2)\, T_0$, depending on the iteration k of the policy update.

4 Experiment

We tested the algorithm on a discrete maze of size 100×100 and compared it to Prioritized Sweeping (PS). Walls of the maze are considered to be trap states (leading to unsuccessful trials) and actions (north, south, east, west, stay) are highly noisy in the sense that with a probability of 0.5 they lead to random transitions. In the experiment we chose a uniform time prior (discount factor $\gamma = 1$) and iterated the policy update $k = 5$ times. Figure 2(a) displays the state visiting probabilities generated by our probabilistic inference planner (PIP) for a problem where rewards are given when some goal state B is reached. Computational costs are measured by the number of evaluations of the environment $p(j|i,a)$ needed during the planning procedure. Figure 2(b) displays the probability of reaching the goal $P(B|A)$ against these costs for the same two start/goal state configurations. Note that for PIP, we can give this information only after a complete inference sweep (i.e., for $k = 1, 2, ..$), which are the discrete dots in the graph. The graph also displays curves for Prioritized Sweeping, where the currently calculated value V_A of the start state (which converges to $P(B|A)$ for the optimal policy) is plotted against how often PS evaluated $p(j|i,a)$.

The PIP algorithm takes considerable advantage of knowing the start state in this planning scenario: the forward propagation allows for the pruning and the early decision on cutoff times of the E-step as described above. It should thus not be a surprise and not overstated that PIP is significantly more efficient in this specific scenario. Certainly, some heuristic forward propagations could also be introduced for PS to achieve similar efficiency. Nonetheless, our approach provides a principled way of pruning by exploiting the computation of proper posteriors.

A detailed inspection of the policies computed by PIP and PS showed that are equal for states which have significantly non-zero state visiting probabilities. Other experiments with various γ gave the same result. An interesting observation was that the total time posterior can be multi-modal when there exist multiple, spatially segregated pathways from start to goal of different lengths. In those cases, the total time posterior reflects a topological property of the environment. Generally, these preliminary experiments only show the feasibility and efficiency of our approach. More interesting results can be expected when other inference methods will be applied to the problem of planning in the future.

5 Conclusion

In this paper we introduced a model that translates the problem of planning into a problem of probabilistic inference. The model is formulated as a mixture of finite-time MDPs treating the total time T as a random variable and we showed how a single synchronous forward and backward sweep realizes inference in all MDPs. Policies computed via Expectation-Maximization are shown to be optimal w.r.t. the expected future reward for arbitrary reward functions. Although the current presentation only addressed exact inference on discrete spaces it is clear that the approach equally allows to transfer inferences techniques on continuous or factorial spaces to the problem of planning. Currently, for instance, we are investigating how to apply standard sequential Monte Carlo methods (e.g., particle filters [5]) to the problem of optimal stochastic control in high-dimensional, continuous robotic scenarios.

Acknowledgments

The first author is grateful to the German Research Foundation (DFG) for the Emmy Noether fellowship TO 409/1-1. The second author thanks Microsoft Research, Cambridge for the fellowship that supported his work.

References

[1] D. Andre, N. Friedman, and R. Parr. Generalized Prioritized Sweeping. In *Advances in Neural Information Processing Systems*, 1997.

[2] H. Attias. Planning by probabilistic inference. In C. M. Bishop and B. J. Frey, editors, *Proc. of the 9th Int. Workshop on Artificial Intelligence and Statistics*, 2003.

[3] C. Boutilier, R. Dearden, and M. Goldszmidt. Exploiting structure in policy construction. In *Proc. of the 14th Int. Joint Conf. on Artificial Intelligence (IJCAI 1995)*, pages 1104–1111, 1995.

[4] H. Bui, S. Venkatesh, and G. West. Policy recognition in the abstract hidden markov models. *Journal of Artificial Intelligence Research*, 17:451–499, 2002.

[5] A. Doucet, N. de Freitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice*. Springer, Berlin, 2001.

[6] Z. Ghahramani and M. I. Jordan. Factorial hidden Markov models. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems, NIPS*, volume 8, pages 472–478. MIT Press, 1995.

[7] C. Guestrin, D. Koller, and R. Parr. Multiagent planning with factored MDPs, 2001.

[8] M. Hauskrecht, N. Meuleau, L. P. Kaelbling, T. Dean, and C. Boutilier. Hierarchical solution of Markov decision processes using macro-actions. In *Proc. of Uncertainty in Artificial Intelligence (UAI 1998)*, pages 220–229, 1998.

[9] D. Koller and R. Parr. Computing factored value functions for policies in structured MDPs. In *Proc. of the 16th Int. Joint Conf. on Artificial Intelligence (IJCAI)*, pages 1332–1339, 1999.

[10] T. Minka. A family of algorithms for approximate bayesian inference, 2001.

[11] R. Sutton and A. Barto. *Reinforcement Learning*. MIT Press, Cambridge, 1998.