The HMDP Planner for Planning with Probabilities

Emil Keyder
Universitat Pompeu Fabra
Passeig de Circumvalació 8
08003 Barcelona Spain
emil.keyder@upf.edu

Héctor Geffner
ICREA & Universitat Pompeu Fabra
Passeig de Circumvalació 8
08003 Barcelona Spain
hector.geffner@upf.edu

Abstract

HMDPP is a probabilistic planner that computes two heuristic values for each state and chooses an action to apply through bounded heuristic search methods. One of these heuristics is the value of the well-known h_{add} heuristic on a relaxation of the underlying MDP in which probabilities are compiled into costs, and the other is computed from an iteratively generated pattern database based on the idea of mutex sets of literals, as computed by the h^2 heuristic.

Introduction

The general strategy we apply to probabilistic planning problems is heuristic evaluation and search. We define two heuristics for the evaluation of each state, which are then used in lexicographic ordering to choose the action to be applied from the current state.

The first heuristic is derived from a novel relaxation of the underlying Markov Decision Process (MDP) that accounts for probabilities in a clear and principled way, which we call the *self-loop* relaxation. Each probabilistic action in the original problem is mapped to a set of deterministic actions which are assigned costs depending on the cost of the original action and the probability associated with the outcome represented by the deterministic action in the original action. The heuristic value of the state is then defined as the value of a variation of the well-known additive heuristic (Bonet & Geffner 2001; Keyder & Geffner 2008) on this determinization of the problem. This approach is similar to the *all-outcomes-determinization* used by the most recent variant of FF-Replan (Yoon, Fern, & Givan 2007), yet the self-loop relaxation takes into account the probability associated with each outcome rather than treating outcomes as equally easy to obtain. Furthermore, heuristic values extracted here are used to guide a local action selection mechanism in the original problem rather than to guide the search for a complete plan in the relaxed problem.

FF-Replans performance on the probabilistic problems of the last two planning competitions can be explained by a number of factors, many of which have to do with the structure of the problems (Little & Thiebaux 2007). Yet as discussed in (Little & Thiebaux 2007), this approach is less powerful in a number of domains in which the probability of reaching a dead-end is non-negligible and care needs to be taken to construct a policy with high probability of success. Specifically, all of the probabilistic effects of an action must be taken into account and the delete effects of outcomes, which are ignored by heuristics that attempt to approximate the cost of the optimal plan in the delete relaxation h^+, must be considered. We therefore use a second, admissible heuristic that attempts to identify dead-end and/or high-risk patterns in states. This value is consulted first in the lexicographic ordering used in the lookahead step to rank actions, as we always prefer to take a path that minimizes the probability of reaching a dead end.

Self-loop MDPs

An MDP M is characterized by a set of states S, an initial state $s_0 \in S$, a set of goal states $G \subseteq S$, a set of actions A, a function giving the applicable actions at each state $A(s)$ for $s \in S$, a cost function $cost(a)$ for $a \in A$, and a transition function $p(.|s, a)$ for $s \in S$, $a \in A(s)$. We define a self-loop MDP as an MDP in which all actions have at most two probabilistic outcomes, and at most one of the outcomes changes the current state. An action may then either deterministically (with probability 1) lead to a different state, or lead with probability p to a different state and with probability $1 - p$ not change the current state. Formally, in a self-loop MDP the value of $p(s'|s, a)$ when $a \in A(s)$ is non-zero for at most one $s' \neq s$. Note that for this s', $p(s'|s, a) = 1 - p(s|s, a)$. In self-loop MDPs, uncertainty is reduced to the number of times that an action must be applied to obtain the desired outcome, as when it does not occur, the state does not change and the action can be repeated. This insight motivates the chain of reasoning below, in which we show that the value function for such an MDP is the same as that of a deterministic MDP with a modified cost function over the set of actions, or in other words, a classical planning problem with action costs.

Using the fact that each action has at most two outcomes, one of which is guaranteed to be a self-loop, the Bellman equations for the optimal value function $V^*(s)$ for MDPs of this type can be written as follows:

$$V^*(s) = min_{a \in A(s)}[cost(a) + \quad (1 - p(s'|s,a))V^*(s) + \\ p(s'|s,a)V^*(s')]$$

Denoting the optimal value obtained at at a state s when action a is applied as $Q^*(s,a)$, we can write:

$$\begin{aligned} Q^*(s,a) \quad &= cost(a) + \quad (1 - p(s'|s,a))\, Q^*(s,a) + \\ & \qquad p(s'|s,a)\, V^*(s') \\ &= \tfrac{cost(a)}{p(s'|s,a)} + \quad V^*(s') \end{aligned}$$

The optimal value function of the MDP can then be written as

$$\begin{aligned} V^*(s) \quad &= \quad min_{a \in A(s)}\, [Q^*(s,a)] \\ &= \quad min_{a \in A(s)} \left[\frac{cost(a)}{p(s'|s,a)} + V^*(s') \right] \end{aligned}$$

where the value function obtained is that of a deterministic model in which the cost of each action is set to $cost'(a) = cost(a)/p(s'|s,a)$, where $p(s'|s,a)$ is the probability of the non self-loop outcome of the action. Solving this deterministic problem optimally then gives the optimal value function for the associated self-loop MDP.

The Self-loop Relaxation

Here we define the *self-loop relaxation* M^{SL} of an arbitrary MDP M, obtained by modifying the set of actions A, the applicability function $A(s)$ and the transition function $p(.|s,a)$ of M. A', $A'(s)$, and $p'(.|s,a)$ in M^{SL} are then defined as follows:

$$\begin{aligned} A' \quad &= \quad \{a'_{s,s'} | a \in A(s) \wedge p(s'|s,a) \neq 0\} \\ A'(s) \quad &= \quad \{a'_{s,s'} | a \in A(s)\} \\ p'(s'|s, a'_{s,s''}) \quad &= \quad \begin{cases} p(s'|s,a) & \text{if } s' = s'' \\ 1 - p(s''|s,a) & \text{if } s = s' \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

Informally, for each outcome of an action in the original MDP with probability p, we create a new action that has the outcome with probability p, and with probability $1 - p$ has no effect. Though here we have discussed the relaxation in terms of the state space of the MDP, it can easily be expressed in terms of the factored representation used in PPDDL, by creating two probabilistic effects for each action: one that has the same add and delete lists as the probabilistic outcome that is represented by the action, and one that has no effect.

The Self-loop Relaxation and Cost-sensitive Heuristics

The self-loop relaxation and the equivalence between the value function of a self-loop MDP and that of a classical planning problem with costs suggest a method of obtaining heuristic estimates of the cost-to-go from any state in the original MDP. The general procedure is to apply the self-loop relaxation and obtain a heuristic estimate from the deterministic problem which shares the value function of the relaxed MDP, using any cost-sensitive heuristic from classical planning. Here we use the duplicate-eliminated version of the additive heuristic (Bonet & Geffner 2001), which is discussed in detail in (Keyder & Geffner 2008). We denote the heuristic estimate of a state s from the original MDP obtained in this way as $h^{SL}_{add}(s)$.

A Pattern Database Heuristic for MDPs

The drawback of decomposing an action with multiple possible effects into different actions is that the resulting relaxation ignores the undesired effects of actions. We therefore use a second mechanism in HMDPP to take the alternative effects of actions into account. Roughly, an abstract and computationally tractable MDP is defined by abstracting states into patterns, an approach similar to that taken by pattern database heuristics in classical planning, as in (Helmert, Haslum, & Hoffmann 2007; Haslum *et al.* 2007). The difference, however, is that the abstract problem is not a deterministic search problem in a pattern space, but rather an MDP. This MDP is then solved by value iteration, resulting in an admissible cost function for the original problem, which we denote $h_{PDB}(s)$. Details on this construction will be reported elsewhere.

Using h^{SL}_{add} and h_{PDB} Together

h^{SL}_{add} scales up well and often provides strong guidance towards the goal. In contrast, h_{PDB} does not scale up as well but can identify high-risk states that must be avoided. HMDPP integrates the two heuristics in a simple way: roughly, among the actions that minimize the expected value of h_{PDB}, those that also decrease the value of the self-loop heuristic are selected. We are currently exploring various look-ahead schemes that make use of this selection criterion for choosing the action to do in any given state.

Acknowledgements

We thank Felipe Trevizan, now at CMU, who helped us in the initial stages of this work; and Blai Bonet for making his mGPT MDP planner available.

References

Bonet, B., and Geffner, H. 2001. Planning as heuristic search. *Artificial Intelligence* 129(1–2):5–33.

Haslum, P.; Botea, A.; Helmert, M.; Bonet, B.; and Koenig, S. 2007. Domain-independent construction of pattern database heuristics for cost-optimal planning. In *AAAI*, 1007–1012.

Helmert, M.; Haslum, P.; and Hoffmann, J. 2007. Flexible abstraction heuristics for optimal sequential planning. In *Proc. ICAPS-2007*.

Keyder, E., and Geffner, H. 2008. Heuristics for planning with action costs revisited. In *18th European Conference on Artificial Intelligence (ECAI-08)*.

Little, I., and Thiebaux, S. 2007. Probabilistic planning vs replanning. In *ICAPS 2007 Workshop on International Planning Competition: Past, Present and Future*.

Yoon, S.; Fern, A.; and Givan, B. 2007. FF-replan: A baseline for probabilistic planning. In *Proc. ICAPS-07*.