Probabilistic Plan Graph Heuristic for Probabilistic Planning

Yolanda E-Martín[1] and **María D. R-Moreno**[1] and **David E. Smith**[2]

[1] Departamento de Automática. Universidad de Alcalá.
Carretera Madrid-Barcelona, Km. 33,600. 28871 Alcalá de Henares (Madrid), Spain.
{yolanda, mdolores}@aut.uah.es
[2] Intelligent Systems Division. NASA Ames Research Center
Moffett Field, CA 94035-1000
david.smith@nasa.gov

Abstract

This work focuses on developing domain-independent heuristics for probabilistic planning problems characterized by full observability and non-deterministic effects of actions that are expressed by probability distributions. The approach is to first search for a high probability deterministic plan using a classical planner. A novel probabilistic plan graph heuristic is used to guide the search towards high probability plans. The resulting plans can be used in a system that handles unexpected outcomes by runtime replanning. The plans can also be incrementally augmented with contingency branches for the most critical action outcomes.

This abstract will describe the steps that we have taken in completing the above work and the obtained results.

Introduction

The success of planning graph heuristics in classical planners like FF (J. Hoffmann and Bernhard Nebel 2001) or HSP (Bonet and Geffner 2001), has influenced research on heuristic estimators to deal with probabilistic planning problems. A few probabilistic planners such as FF-rePlan (S. Yoon, A. Fern and B. Givan 2007) and mGPT (B. Bonet and H. Geffner 2005) use heuristic functions based on relaxed plans to guide a classical planner in the search for a deterministic plan. However, other probabilistic planners use plan graphs to compute estimates of the probability that propositions can be achieved and actions can be performed (A. Blum and J. Langford 1999) (I. Little and S. Thiébaux 2006). This information can be used to guide the probabilistic planner towards the most likely plan for achieving the goals.

In this extended abstract, we describe an approach to computing more accurate estimates of probability based on work of D. Bryce and D. E. Smith (2006). We proceed by introducing our probabilistic plan graph estimator and the probabilistic relaxed plan extraction procedure. We then do an empirical study of the techniques within our planner and compare with some other probabilistic planners.

Probabilistic Plan Graph Heuristic

We assume that we are given a probabilistic planning problem represented in PPDDL (H. L. S. Younes, M. L. Littman, D. Weissman and J. Asmuth 2005), where action outcomes are represented by a probability distribution. These distributions are used to build a probabilistic plan graph by propagating probability information forward through the graph. As our purpose here is to get better probability estimates, we introduce the term interaction (*I*), which captures the degree of dependence (positive or negative) between pairs of propositions and actions in the plan graph (D. Bryce and D. E. Smith 2006). Formally, the interaction, *I* between two propositions or two actions is defined as:

$$I(p,q) = \frac{P(p \wedge q)}{P(p)P(q)} \tag{1}$$

$I(p,q)$ is therefore a positive number ranging between zero and $1/\max(P(p), P(q))$. In general:

- $I < 1$ means that two propositions or actions interfere with each other - that is, the probability of establishing both is less than the product of the probabilities for establishing them independently. In the extreme case, $I = 0$, the propositions or actions are mutually exclusive.
- $I = 1$ means that two propositions or two actions are independent.
- $I > 1$ means that two propositions or two actions are synergistic.

The computation of probability and interaction information begins at level zero of the plan graph and proceeds sequentially to higher levels. For level zero we assume 1) the probability of the propositions of this level is 1 because the initial state is fully known and 2) the interaction between pair of propositions is 1, that is, the propositions are independent. With these assumptions, we start the propagation by computing the probability of the actions at level zero.

In general, for an action at level l with preconditions $x_1, ..., x_n$, the probability is calculated as follows:

$$Pr(a) = \prod_{i=1..n} \left[Pr(x_i) \prod_{j=1..i-1} I(x_i, x_j) \right] \tag{2}$$

The interaction between two actions a and b at level l, with a set of preconditions $prec_a$ and $prec_b$ is[1]:

$$\begin{aligned} I(a,b) &= \frac{Pr(prec_a \wedge prec_b)}{Pr(a)Pr(b)} \\ &= \prod_{i \in prec_a, j \in prec_b} I(i,j) \end{aligned} \tag{3}$$

For a proposition *x* at level *l*, achieved by the actions *Ach(x)* at the preceding level, the probability is calculated as:

$$Pr(x) = \max_{a \in Ach(x)} \left[Pr(a)Pr(x|a) \right] \tag{4}$$

In order to compute the interaction between two propositions at a level, we need to consider all possible ways of achieving those propositions at the previous level. Suppose that *Ach(x)* and *Ach(y)* are the sets of actions that achieve propositions *x* and *y* at level *l*. The interaction between *x* and *y* is then:

$$I(x,y) = \max_{a \in Ach(x), b \in Ach(y)} \left[\frac{Pr(a)Pr(b)I(a,b)Pr(x|a)Pr(y|b)}{Pr(x)Pr(y)} \right] \tag{5}$$

The plan graph and probability estimates are used to help guide a forward state-space planning search. For each state, a relaxed plan is constructed to estimate the probability of achieving the goals from that state. The construction of the relaxed plan makes use of the probability and interaction information to make better choices for actions. In particular, to achieve a particular goal at a level, the relaxed plan construction algorithm chooses the action that will maximize the probability of achieving the goal, given all the other action choices that have been made at that level (a greedy algorithm).

Experimental Results

Our approach has been implemented in the PIPSS system (Yolanda E-Martin, M. D. R-Moreno and Bonifacio Castaño 2010). PIPSS emerges from the union between a heuristic search planner and a scheduling system. Since PIPSS is deterministic, the probabilistic domain is converted into a deterministic one (S. Jimenez, A. Coles and A. Smith 2006).

PIPSS has been tested with several domains of the IPPC-06 probabilistic track, and the results have been compared with those shown in (S. Yoon, A. Fern and B. Givan 2007).

When running the experiments we observed that the probabilities of some actions in the last layers of the probabilistic plan graph of some problems were zero or close to zero. Consequently, in the construction of the relaxed plan, these actions will never be chosen. But sometimes they can generate a goal proposition or they are essential to achieve the most likely plan. So in order to solve the problem, we have computed the plan graph using costs instead of probabilities using $-Ln(p)$ (S. Jimenez, A. Coles and A. Smith 2006).

Table 1 shows the percentage of successful trials for each domain. For all the planners, the number of trials per problem is 30, with 15 minutes for each one. However, with PIPSS we have just performed a single trial because at the moment, our system does not deal with unexpected states. Despite this, PIPSS gets better results in two of the tested domains, and high rates of success in the rest. For this reason, we think that when we provide PIPSS with the replanning capacity, the success rate will increase.

Table 1: Percentage of Successful Problems

	Planners						
Domains	FFR_a	FOALP	sfDP	FPG	Paragraph	FFR_s	PIPSS
blocksworld	86.22	100	29.11	62.89	0	76.8	53.33
ex-blocksworld	51.56	24.22	31.33	42.67	30.74	51.56	93.33
elevators	93.33	100	0	76.22	0	93	100
tire	82.22	81.56	0.00	74.89	91.14	69	100
zeno	100	0	6.67	26.89	6.67	7	33.33

Acknowledgments

This work has been funded by the Junta de Comunidades de Castilla-La Mancha project PEII11-0079-8929.

References

A. Blum and J. Langford. 1999. Probabilistic Planning in the Graphplan Framework. *In Proceedings of ECP-99*.

B. Bonet and H. Geffner. 2005. mGPT: A Probabilistic Planner Based on Heuristic Search. *JAIR* 24:933–944.

Bonet, B., and Geffner, H. 2001. Planning as Heuristic Search. *JAIR* 129:5–33.

D. Bryce and D. E. Smith. 2006. Using Interaction to Compute Better Probability Estimates in Plan Graphs. *In Proceedings of ICAPS-06 Workshop on Planning Under Uncertainty and Execution Control for Autonomous Systems*.

H. L. S. Younes, M. L. Littman, D. Weissman and J. Asmuth. 2005. The First Probabilistic Track of the International Planning Competition. *JAIR* 24:841–887.

I. Little and S. Thiébaux. 2006. Concurrent Probabilistic Planning in the Graphplan Framework. *In Proceedings of ICAPS-06 Workshop on Planning Under Uncertainty and Execution Control for Autonomous Systems*.

J. Hoffmann and Bernhard Nebel. 2001. The FF Planning System: Fast Plan Generation Through Heuristic Search. *JAIR* 14:253–302.

S. Jimenez, A. Coles and A. Smith. 2006. Planning in Probabilistic Domains using a Deterministic Numeric Planner. *In Proceedings of PLANSIG-06*.

S. Yoon, A. Fern and B. Givan. 2007. FF-Replan: A Baseline for Probabilistic Planning. *In Proceedings of ICAPS-07*.

Yolanda E-Martin, M. D. R-Moreno and Bonifacio Castaño. 2010. PIPSS*: a System based on Temporal Estimates. *In Proceedings of SGAI-10*.

[1] Before performing the calculation of the interaction between a pair of operators, we need to determine if the operators are mutex by inconsistent effect or interference. In both cases, the interaction is zero.