Proceedings of the Nineteenth AAAI/SIGAI Doctoral Consortium

Compilation Based Approaches to Probabilistic Planning — Thesis Summary

Ran Taig
Department of Computer Science
Ben Gurion University of The Negev
Beer-Sheva, Israel 84105
taig@cs.bgu.ac.il

Motivation and Background

Models of planning under uncertainty, and in particular, MDPs and POMDPs have received much attention in the AI and Decision-Theoretic planning communities (Boutilier, Dean, and Hanks 1999; Kaelbling, Littman, and Cassandra 1998). These models allow for a richer and more realistic representation of real-world planning problems, but lead to increased complexity. Recently, a new approach for handling certain simple classes of planning under uncertainty was introduced (Palacios and Geffner 2009). This approach works by reducing problems of planning under uncertainty to classical planning problems. The main benefit of this technique is that it allows us to exploit techniques developed in classical planning, and in particular, effective and sophisticated methods for computing heuristic functions. So far, this technique has been shown to be effective for conformant and contingent planning (Albore, Palacios, and Geffner 2009; Shani and Brafman 2011). A related approach was very successful in handling MDPs in the FF-Replan planner (Yoon, Fern, and Givan 2007).

The goal of our research is to utilize the approach for probabilistic planning problems. That is, we'd like to be able to represent and reason about probabilities within non probabilistic planning frameworks. To start this research direction, we focus on the problem of conformant probabilistic planning (CPP) with deterministic actions. Although this problem is somewhat narrow, much like conformant planning, it provides a convenient initial step for exploring this research direction. We believe that our techniques can be extended to more general probabilistic planning problems. The best current CPP planner is *Probabilistic FF* (PFF) (Domshlak and Hoffmann 2007). Probabilistic-FF uses a time-stamped Bayesian Networks (BN) to describe probabilistic belief states. In most benchmarks, *PFF*'s results were improved by our results.

Conformant Probabilistic Planning (CPP)

We assume familiarity with the basic notation of classical planning domains via STRIPS with conditional effects: (V, A, I, G), corresponding to a set of *propositions*, *actions*, *initial world state*, and *goal*. A CP problem, (V, A, b_I, G), generalizes this framework, replacing the single initial state with a set of initially possible states, called the *initial belief state* b_I. A plan is an action sequence $\overline{a}$ such that $\overline{a}(w_I) \supseteq G$ for every $w_I \in b_I$. CPP extend CP by quantifying the uncertainty regarding b_I using a probability distribution b_I. In its most general form, CPP allows for stochastic actions, but we leave this to future work, and assume all actions are deterministic. CPP tasks are 5-tuples (V, A, b_I, G, θ). As before, G is a conjunction of propositions. b_I denotes a probability distribution over the world states, where $b_I(w)$ is the probability that w is the true initial world state. In many settings, achieving G with certainty is impossible. θ specifies the required *lower bound* on the probability of achieving G. A sequence of actions $\overline{a}$ is called a *plan* if the weight of the initial states from which $\overline{a}$ reaches the goal is at least θ.

My Work Until Now

During my Msc. and PhD. studies until now I've developed a few compilation based algorithms for CPP. Our initial results, published in (Brafman and Taig 2011) summarized my Msc. work. My recent results were published at (Taig and Brafman 2012) and (Taig and Brafman 2013). Our work in the last year is summarized in a paper accepted for *AAAI 14'*. Below is a concise summary of the work:

The translation approach - background and required modifications Our initial work is based on a modified version of the translation-based method by (Palacios and Geffner 2009). The essential idea behind the approach to is to reason by cases. The different cases correspond to different conditions on the initial state, or, equivalently, different sets of initial states. These sets of states, or conditions, are captured by tags. That is, a tag is identified with a subset of b_I. The simplest way for understanding tags is just considering one tag per each possible initial state (e.g an initial state w s.t $b_I(w) > 0$).

The set of propositions with new propositions of the form p/t, where t is one of the possible tags for p. p/t holds the current value of p given that the initial state satisfies the condition t. The value of each proposition p/t is known initially – it reflects the value of p in the initial states represented by t, and since we focus on deterministic actions only, then $p/t \vee \neg p/t$ is a tautology throughout. The actions are transformed accordingly to maintain our state of knowledge by propagating this conditioned knowledge, if the CPP had an action a which consist some conditional effect $C \rightarrow p$ then in the compiled problem we'll add conditional effect: $C/t \rightarrow p/t$ for every possible tag t for p, this propagates our conditioned knowledge throughout the planning process.

The resulting problem is a classical planning problem defined on a larger set of variables. The size of this set depends on the original set of variables and the number of tags we need to add. Hence, an efficient tag generation process is important. Some of our work focused on defining the theoretical properties of such a generation process or more accurately, how to modify PG's process to the probabilistic planning, please refer to our ICAPS paper for details, I cannot detail here due to lack of space. Based on the above, we developed a few compilation methods for CPP as follows:

Method 1: Compiling CPP into Metric Planning

Metric planning is an extension of the classical planning framework where some of the variables can get real values in addition to the finite domain variables. Our method, described in our *ADT'11* paper, utilized these variables to monitor the probabilities of each variable by adding a numerical variable Pr_p for each original variable p, this variable is initialized with the initial probability of p given by b_I. Actions are modified to update these variables w.r.t the conditioned knowledge on p. That is, if the compiled propositions p/t_1,p/t_2 are true in some stage we can deduce the current probability of p by calculating the joint initial probability of t_1 and t_2. In addition a special fluent Pr_g is added to monitor the current probability of the goal and actions are modified accordingly. The only goal is then just $Pr_g \geq \Theta$. The resulting compiled metric problem is then being fed to the state-of-the-art metric planner $Metric-FF$, we prove in the paper that the resulting plan, after simple modification, is a plan for the original CPP and the algorithm is sound. We also give conditions for completeness. Results for this algorithm were good but for a limited, simple, set of benchmarks. The main reasons are the inherited limitations of the underlying planner to handle the nature of our compiled problems in terms of size and many conditional effects.

Method 2: CPP as cost-bounded sub optimal planning problem

In cost bounded classical planning a classical planning problem is extended with a constant parameter $c \in \mathbb{R} > 0$. The task is to find a plan with cost $\leq c$ as fast as possible. In this setting the optimal plan cost and the distance of the resulting plan from optimal does not matter. This compilation was motivated by recent advances in the research of this field resulted in efficient cost bounded planners (Thayer et al. 2012).

The basic rational in the method we present now is the understanding that we can solve a CPP problem by identifying a set b' of initial states whose joint probability is greater or equal to θ, such that there exists a conformant plan for b'. This plan is a solution to the CPP problem, too. We let a classical planner to decide which states to ignore, and also generates a conformant plan for all other states. We must ensure that the joint probability of ignored states does not exceed $1-\theta$. Technically, this is done by introducing special actions that essentially tell the planner to ignore a state (or set of states). The cost of each such action is equal to the probability of the state(s) it allows us to ignore.

In the compiled problem we generate, knowledge is added by applying *merge* actions. Once a state has been "ignored" by an "ignore" action, special inference actions we add effectively ignore it, and deduce the information as if this state is not possible.

A solution to the generated problem will make assumptions whose cost does not exceed the bound. Hence, it will work on a sufficiently large set of initial states and thus will correspond to a valid plan for the original CPP. We refer the reader to our *ICAPS'13* paper for more details.

Method 3: Relevance Based compilation into Conformant planning

Our recent method, although lies on the same rational, suggests different approach. Our planner starts with a preprocessing relevance analysis phase that determines a promising set of initial states on which to focus. It then calls an off-the-shelf conformant planner to solve the resulting problem. This approach has a number of advantages. First, we can introduce specific, efficient relevance reasoning techniques for selecting the set of initial states, rather than depend on the heuristic function used by the planner. Second, we can benefit from various optimizations used by conformant planners that are unsound when applied to the original CPP. Finally, we have the freedom to select among different existing CP solvers. Consequently, the new planner dominates previous solvers on almost all domains and scales to instances that were not solved before. We proved the soundness of the method. This method is incomplete. We note that since CPP is at least as hard as CP, which is *PSPACE-COMPLETE*, ensuring completeness seems ill advised. We refer the reader to our *AAAI'14* paper for more details.

Research Plan

We intend to focus on the following main research thrusts: First, we hope to extend our compilation techniques to handle richer settings. As a first step, we will focus on CPP with stochastic actions beginning with simply consider deterministic version of these actions while verifying that the solution addresses a sufficiently large probability mass. Another possibility we consider is to create partition of the problem, based on the relevance analysis, which will allow to plan and reason about probabilities over a compact portion of the belief space. Later, we hope to handle stochastic observations, motivated by the work done at (Albore, Palacios, and Geffner 2009; Shani and Brafman 2011) where the approach was used to handle *contingent planning* with non-deterministic observations.

On the longer term, we'd like to improve the ability of classical planners to handle our compiled problems. Currently, we use the planners as black boxes. We've identified weaknesses of the planners w.r.t to the properties of our problems.

Acknowledgments The author was supported in part by ISF grant 933/13 and the Lynn and William Frankel Center for Computer Science.

References

Albore, A.; Palacios, H.; and Geffner, H. 2009. A translation-based approach to contingent planning. In *IJCAI*, 1623–1628.

Boutilier, C.; Dean, T.; and Hanks, S. 1999. Decision-theoretic planning: Structural assumptions and computational leverage. *J. Artif. Intell. Res. (JAIR)* 11:1–94.

Brafman, R. I., and Taig, R. 2011. A translation based approach to probabilistic conformant planning. In *ADT*.

Domshlak, C., and Hoffmann, J. 2007. Probabilistic planning via heuristic forward search and weighted model counting. *J. Artif. Intell. Res. (JAIR)* 30:565–620.

Kaelbling, L. P.; Littman, M. L.; and Cassandra, A. R. 1998. Planning and acting in partially observable stochastic domains. *Artif. Intell.* 101(1-2):99–134.

Palacios, H., and Geffner, H. 2009. Compiling uncertainty away in conformant planning problems with bounded width. *JAIR* 35:623–675.

Shani, G., and Brafman, R. I. 2011. Replanning in domains with partial information and sensing actions. In *IJCAI*, 2021–2026.

Taig, R., and Brafman, R. I. 2012. Using classical planners to solve conformant probabilistic planning problems. In *Problem Solving Using Classical Planners AAAI Technical Report WS-12-12,p. 65-71*.

Taig, R., and Brafman, R. I. 2013. Compiling conformant probabilistic planning problems into classical planning. In *ICAPS*.

Thayer, J.; Stern, R.; Felner, A.; and Ruml, W. 2012. Faster bounded-cost search using inadmissible estimates. In *Proceedings of the Twenty-Second International Conference on Automated Planning and Scheduling*.

Yoon, S.; Fern, A.; and Givan, R. 2007. Ff-replan: A baseline for probabilistic planning. In *ICAPS*, volume 7, 352–359.