Planning Under Uncertainty with Temporally Extended Goals

Alberto Camacho *
Department of Computer Science
University of Toronto. Canada.
acamacho@cs.toronto.edu

1 Introduction

In the last decade, we have seen an exponential increase in the number of devices connected to the Internet, with a commensurate explosion in the availability of data. New applications such as those related to smart cities exemplify the need for principled techniques for automated intelligent decision making based on available data. Many decision-making problems require reasoning in large and complex state spaces, sometimes under stringent time constraints. The nature of these problems suggests that planning approaches could be used to find solutions efficiently. Automated planning is the basis for addressing a diversity of problems beyond classical planning such as automated diagnosis, controller synthesis, and story understanding. Nevertheless, many planning paradigms make assumptions that do not hold in real-world settings.

Our work focuses on exploring planning paradigms that capture properties of real-world decision-making applications. These properties include the ability to model nondeterminism in the outcome of actions, the ability to deal with complex objectives that are temporally extended (in contrast to final-state goals) some of which may be necessary and other simply desirable to optimize for. Finally, we are interested in dealing with incomplete information. Addressing this class of problems presents challenges related to problem specification, modeling, and computationally efficient techniques for generating solutions.

Illustrative Example Consider the problem of designing a tourist route to visit a set of touristic attractions in London. The tour is subject to certain constraints. For example, an individual may feel it's mandatory for the tour to include the London Eye and the Houses of Parliament, and also desirable to visit the Maritime Museum and the Greenwich Observatory or other highly-rated attractions, if these can be included. The tour must be realizable via a combination of walking and public transit. If it is raining, then walking should be minimized. These are examples of temporally extended goals. Following from our example, aspects of the dynamics of the environment are not controllable by the agent, such as traffic, punctuality of public transport, and the weather. If the stochastic model for these events is available, we can quantify the *expected quality* of the plan according to a certain metric (e.g. probability of visiting the noted touristic attractions at the end of the journey) and attempt to produce plans that maximize this objective. When the stochastic model is not available, we may want to produce plans that are robust to *any* contingency (e.g. a plan that suggests visiting a museum, at any moment, if it starts to rain).

2 Progress to the Date

In our work to date, we have advanced the state of the art in planning problems with non-deterministic actions and temporally extended goals. In this section, we introduce the FOND and probabilistic planning models, and describe the high-level contributions of our work. We refer the reader to the respective publications for further details.

A *Fully Observable Non-Deterministic* (FOND) planning problem is a tuple $\mathcal{P} = \langle S, s_I, \mathcal{A}, F, S_\mathcal{G} \rangle$, where S is a finite set of *states*, $s_\mathcal{I} \in S$ is the *initial state*, $S_\mathcal{G} \subseteq S$ is a set of *goal states*, and $\mathcal{A}$ is a finite set of actions. For each action $a \in \mathcal{A}$, and state $s \in S$, the result of applying a in s is one of the states in the set $F(s, a) \subseteq S$. Solutions to FOND planning problems are *policies*, or mappings $\pi : S \to \mathcal{A}$ from states into actions. In concrete, *strong-cyclic* solutions are those that lead the agent to a goal state with complete guarantees (Cimatti et al. 2003).

A *probabilistic* planning problem is a tuple $\mathcal{P} = \langle S, s_I, \mathcal{A}, T, S_\mathcal{G} \rangle$. Different than the FOND model, for each action $a \in \mathcal{A}$, and pair of states $s, s' \in S$, $T(s, a, s')$ is the *transition probability* of reaching s' when a is applied in s. Solutions to probabilistic planning problems are policies. In goal-oriented probabilistic planning models such as MaxProb, solutions are policies that lead the agent to a goal state with maximal probability.

Execution of a policy π generates state-action sequences $s_0, a_0, s_1, a_1, \ldots$ where $s_0 = s_I$, $a_i = \pi(s_i)$, and $s_{i+1} \in F(s_i, a_i)$. When execution finishes in a goal state s_n, assumed to be absorbing, the sequence $P = a_0, a_1, \ldots, a_n$ is called a *plan*. For a probabilistic planning problem, the *likelihood* of P is $L_P = \Pi_{i=0}^{n-1} T(s_i, a, s_{i+1})$.

*The contributions presented in this paper reflect joint work with (in alphabetical order) Jorge Baier (jabaier@ing.puc.cl), Sheila McIlraith (sheila@cs.toronto.edu), Christian Muise (cjmuise@mit.edu), and Eleni Triantafilou (eleni@cs.toronto.edu).

2.1 ProbPRP

In (Camacho, Muise, and McIlraith 2016) we present ProbPRP, a probabilistic planner that finds solutions to probabilistic planning problems where the objective is to attempt to maximize the probability of reaching a goal state. We formalize this class of problems and call it HighProb.

ProbPRP leverages core similarities between probabilistic and FOND planning, building on top of the state-of-the-art FOND planner PRP (Muise, McIlraith, and Beck 2012). The features present in PRP are of great value to ProbPRP. Namely, the *partial state* representation obtained via plan regression facilitates states entailment during the search process, and results in considerable improvements in the algorithm convergence. Besides, the compact representation of state results in smaller policies. The *deadend detection* mechanism prunes the search space effectively by means of forbidden state-action pairs (FSAPs) generated automatically during the search process, and guarantees optimality of the algorithm when deadends are avoidable.

ProbPRP extends the state-of-the-art FOND planner PRP (Muise, McIlraith, and Beck 2012) with techniques that leverage probabilistic information to produce high quality HighProb solutions. Some of these enhancements to ProbPRP are detailed below.

High-Likelihood Plan Exploration The search of plans is biased towards exploration of high-likelihood plans or, equivalently, plans with high log-likelihood $\log(L_P) = \Sigma_{i=0}^{n-1} \log(T(s_i, a, s_{i+1}))$. To this end, each transition is given associated cost $-\log(T(s_i, a, s_{i+1}))$, and a suboptimal search is performed to find plans with low cost. The search bias produces policies that have smaller expected plan length – orders of magnitude smaller in some instances.

Final FSAP-free Round A final search round is performed to extend the best incumbent policy found by the algorithm, this time with the FSAP mechanism disabled. This allows plan exploration in those policies that cannot reach the goal with complete guarantees. We observed the final FSAP-free round increments the probability of reaching a goal state up to 30% in the most beneficial cases.

Safety Belt Mechanism The current version of ProbPRP gradually disables the strong-cyclic detection (SCD) feature when it is not contributing to the solver's progress. If the SCD mechanism is consistently never used to detect states, then the (potentially costly) SCD computation is gradually disabled and used less over time.

Eliminating Non-Robust Plans The speed with which ProbPRP converges depends on the order in which weak plans are explored, and the nature of those plans. For example, when strong cyclic solutions exist, the policy will never include non-robust plans. One way to accelerate convergence is to eliminate those plans that are easily determined to be non-robust. In particular, when a weak plan is not robust and non-deterministically leads to a deadend, ProbPRP will eventually find it, compute the FSAPs, and start the search again.

Policy Optimization To reduce the number of state-action pairs in the solution found by ProbPRP, a *simulation* checks all the states reachable by the policy and discards the portion of the policy that is no longer used in the solution.

ProbPRP has two important merits. First, it overcomes scaling difficulties that previous offline algorithms experienced. And second, it offers superior optimality guarantees with respect to the previous state of the art in HighProb, the online planner RFF (Teichteil-Königsbuch, Kuter, and Infantes 2010). Despite being an offline algorithm, ProbPRP outperforms RFF in general and solutions are of better quality.

2.2 LTL-FOND Translations

In (Camacho et al. 2016) we address the problem of planning with non-deterministic actions and temporally extended goals. We assume goals are specified as LTL formulas (Pnueli 1977), and call the model LTL-FOND. More formally, a LTL-FOND planning problem is a tuple $\mathcal{P} = \langle S, s_I, \mathcal{A}, F, \varphi \rangle$, where S, s_I, $\mathcal{A}$, and F are defined as in FOND problems, and φ is an LTL formula. Solutions to a LTL-FOND problem are finite-state controllers whose executions generate state trajectories that satisfy φ.

LTL formulae can be interpreted over finite or infinite state trajectories. Solutions to different interpretations are not always equivalent. A number of techniques exist to solve planning with LTL goals, a subset in the presence of non-deterministic actions, and with finite and infinite LTL interpretations. A common approach is to compile the problem into one with a final-state goal, and solve the resulting problem with state-of-the-art planning technology (e.g. (Baier and McIlraith 2006; Patrizi, Lipovetzky, and Geffner 2013; Torres and Baier 2015)). Related work attempts to maximize reward in MDPs with finite LTL goals and preferences (e.g. (Lacerda, Parker, and Hawes 2015)), and in decision processes with non-markovian rewards (e.g. (Thiébaux et al. 2006)).

We present two different techniques for compiling LTL-FOND into FOND, each addressing both the case of finite LTL interpretations, and the case of infinite LTL interpretations. Remarkably, we are the first to solve the full spectrum of LTL FOND planning interpreted on infinite state trajectories. Equipped with strong-cyclic planner, PRP, our system proves competitive with other state-of-the-art algorithms for LTL FOND, with the advantage of being able to solve the full spectrum of LTL FOND problems.

Our translations leverage ideas from (Baier and McIlraith 2006; Torres and Baier 2015; Patrizi, Lipovetzky, and Geffner 2013), and use Non-deterministic Finite Automata (NFA) and Alternating Automata (AA) representations of the LTL formula to monitor progression, and strong-cyclic planning to synthesize solutions. The size of NFA-based translations is worst-case exponential in the size of the formula, and the size of AA-based translations is worst-case polynomial. Interestingly, PRP performance was better with NFA-based translations, with smaller policies and lower runtimes than with AA-based translations.

From Infinite LTL-FOND to FOND Our first approach uses Büchi Alternating Automata (BAA) representations of the LTL formula. The translation scheme alternates between different modes – *world* mode, and *synchronization* mode –, similar to the AA translation scheme for deterministic planning with finite LTL goals presented by Torres and Baier (2015). Remarkably, the technicalities in our BAA translations for infinite LTL-FOND are significantly different and incorporate non-trivial changes.

Our second approach uses NFA representations of the LTL formula. The dynamics of the translated problem is similar to the NFA translation scheme for deterministic planning with LTL goals presented by Baier and McIlraith (2006).

From Finite LTL-FOND to FOND Our translations for finite LTL-FOND extend the translations for deterministic planning with finite LTL goals presented by Baier and McIlraith (2006), and Torres and Baier (2015). More precisely, our translations first determinize the LTL-FOND problem. Then, one of the translations mentioned above is applied to the resulting deterministic problem with LTL goal. After this step, a deterministic problem with final-state goal is obtained. Finally, a FOND problem is obtained by creating non-deterministic actions from the deterministic actions that resulted from the determinization of the original problem.

3 Discussion and Future Work

The techniques we are developing are applicable to a diversity of real-world problems from the control of collections of smart-home devices, to applications in transportation planning and industrial process planning. A natural next step is to extend our recent work to address the class of probabilistic planning problems with LTL goals which we believe can be done via our existing translations and ProbPRP. We are also interested in extending our work to capture LTL preferences and rewards. Finally, we plan to explore extensions to our models to include both propositional and real-valued variables since such hybrid models are prevalent in many of the real-world applications we've encountered.

References

Baier, J., and McIlraith, S. 2006. Planning with first-order temporally extended goals using heuristic search. In *AAAI*, 788–795.

Camacho, A.; Triantafilou, E.; Baier, J. A.; Muise, C.; and McIlraith, S. A. 2016. LTL Synthesis for Non-Deterministic Systems on Finite and Infinite Traces. In *HSDIP*.

Camacho, A.; Muise, C.; and McIlraith, S. A. 2016. From fond to robust probabilistic planning: Computing compact policies that bypass avoidable deadends. In *ICAPS*.

Cimatti, A.; Pistore, M.; Roveri, M.; and Traverso, P. 2003. Weak, strong, and strong cyclic planning via symbolic model checking. *AIJ* 147:35–84.

Lacerda, B.; Parker, D.; and Hawes, N. 2015. Optimal Policy Generation for Partially Satisfiable Co-Safe LTL Specifications. *IJCAI* 1587–1593.

Muise, C.; McIlraith, S. A.; and Beck, J. C. 2012. Improved Non-deterministic Planning by Exploiting State Relevance. In *ICAPS*, 172–180.

Patrizi, F.; Lipovetzky, N.; and Geffner, H. 2013. Fair LTL synthesis for non-deterministic systems using strong cyclic planners. In *IJCAI*, 2343–2349.

Pnueli, A. 1977. The temporal logic of programs. In *FOCS*, 46–57.

Teichteil-Königsbuch, F.; Kuter, U.; and Infantes, G. 2010. Incremental plan aggregation for generating policies in MDPs. In *AAMAS*, volume 1, 1231–1238.

Thiébaux, S.; Gretton, C.; Slaney, J. K.; Price, D.; Kabanza, F.; et al. 2006. Decision-theoretic planning with non-markovian rewards. *JAIR* 25:17–74.

Torres, J., and Baier, J. A. 2015. Polynomial-time reformulations of LTL temporally extended goals into final-state goals. In *IJCAI*, 1696–1703.