A Generalization of Automated Planning Using Dynamically Estimated Action Models – Dissertation Abstract

Eyal Weiss
The MAVERICK Group, Bar-Ilan University, Israel
Supervisor: Prof. Gal A. Kaminka
{weissey, galk}@cs.biu.ac.il

Abstract

Representing real-world planning problems is a major open subject. Standard planning modeling languages are fully declarative, making it challenging to use them for expressing complex mathematical functions, that are often required for describing the effects of actions. Recent approaches turn to external sources of information, such as simulators or black-box modules, to overcome such modeling limitations. This paper proposes a novel approach to represent and solve planning problems, by starting with partial declarative action models and incrementally refining them during planning by invoking domain-specific external modules. Since these might be computationally expensive, we provide the planner the ability to trade-off modeling uncertainty against computation time, to meet target plan accuracy. Results that were obtained for planning with dynamic estimation of action costs are sketched, and planned work, together with open challenges, are further detailed.

Introduction

AI planning is a mature research field that has undergone major developments over the years. While its roots built on purely symbolic and highly abstract problem formulation (Fikes and Nilsson 1971), it has gradually evolved to support richer declarative representations by using more detail, which is evident, e.g., in the various PDDL versions introduced (McDermott et al. 1998), cf. (Fox and Long 2003). In addition, current planning technology, that is tailored to solve problems expressed in terms of standard formulations (such as PDDL), is based on solid theory, sophisticated search algorithms (Lipovetzky and Geffner 2017), efficient domain-independent heuristics (see the description of many heuristics in the book (Ghallab, Nau, and Traverso 2016)) and data-dependent planner portfolios (Gerevini, Saetti, and Vallati 2014), where much of this work has been translated to optimized open-source software implementations (Helmert 2006). However, it is widely accepted that the adoption of AI planning technology outside the research community is not very common, which stands in sharp contrast to its maturity.

We believe that simplistic modeling is one of the key factors that inhibit widespread use. In particular, we claim that many real-world planning problems cannot be adequately represented solely using current modeling languages, and therefore they are not applicable to existing domain-independent planning technology. This argument has been raised before at various times and contexts (McCluskey 2003; Boddy 2003; Rintanen 2015), yet the solution that was consistently suggested—to make modeling languages more expressive—seems unlikely to be sufficient on its own. This is because the effects of some actions may only be described using complex mathematical functions, or even only known in black-box form. Hence, *it is our belief that as long as exclusively fully declarative models are used, there will still be planning problems out of reach*.

A recent trend advocates coupling of external sources of information to the planner, in order to overcome modeling limitations. Presently, there are two major lines of research taking this approach: planning with simulators (Francès et al. 2017), and domain-specific attempts—notably within the framework of Task and Motion Planning (a recent review is provided in (Garrett et al. 2021)). While these may well be appropriate for some applications, they do not offer a full solution to the gap in problem modeling. Indeed, the first relies on simulators that are not always available, and furthermore, it sacrifices much mathematical structure—inherent in declarative action models—rendering many known heuristics inapplicable, while the second is, as mentioned, domain-specific, and thus does not offer a high level of generality.

Motivated by the gap suggested, and the opportunity it presents, we focus our efforts on the following question.

Research Question *How can we leverage state-of-the-art domain-independent planning technology to tackle real-world problems that cannot be adequately represented using purely declarative models?*

Breaking the Barrier between Problem Modeling and Planning

Our proposal is to postpone part of the modeling to the planning phase, and to utilize external sources of information for model completion ad hoc. Since calling external modules during planning can be computationally expensive (similar to using heuristic functions), it is advantageous to provide the planner with the ability to make educated choices, to balance computation time against allowed uncertainty. Thus, *our vision is to start with a partial declarative model, and to incrementally refine it during planning only where it appears necessary for finding a plan that meets a target accuracy*.

While this idea is fairly high-level, we offer one concrete implementation based on the following specifications:

- Keep problem *structure* symbolic and abstract, by using declarative action models that initially only specify structural preconditions and effects (e.g. via predicates).
- Acquire *numeric* model parameters online, by letting the planner call domain-specific external modules (i.e., estimators) that provide information about their values.
- Define an acceptable accuracy for the sought-after plan, and allow the planner to control the model uncertainty, so it can trade-off accuracy vs. computation time.

Consequences The immediate implication of the suggested approach is an enhanced ability to represent and solve planning problems, as clearly every declarative representation (that so far was constructed prior to planning) can be completed incrementally by the planner, given appropriate external modules. The price paid is increased planning time, due to additional computational effort spent on refining the model. This trade-off is typical for problem generalization, as it entails solving a harder problem. The main challenge that arises is thus to develop computationally efficient planners, able to balance resource allocation between search effort and model refinement effort.

We believe that this suggestion provides several appealing properties. First, any kind of estimator can be used, so there are no restrictions on the type of data being processed during planning, nor on the mathematical operations it utilizes, and in particular it can be black-box. Second, state-of-the-art domain-independent planning techniques retain relevance, as the only difference in the problem formulation is the need to dynamically acquire numeric model parameters. Namely, exiting heuristics can still get the information they need to work, where the sole modification is that they take as inputs estimations of—instead of exact—numeric parameters. More broadly, *current domain-independent planners need to be extended, rather than replaced, in order to be applicable*. Lastly, model uncertainty can be systematically controlled to meet target plan accuracy, while offering significant potential savings on redundant modeling time. This might seem somewhat unusual, as modeling time is not typically a factor considered from the planning perspective, yet richer representations of planning problems could become prohibitive if fully compiled prior to planning. Indeed, consider the implications on the time required to construct a model, in case an estimator is applied for every ground action prior to planning. This is similar to applying numerous heuristics prior to the planning phase, which is clearly a waste of resources.

PhD Research Goals We have set two goals for the PhD period that follow our proposal into concrete setups. The first is to develop a framework that supports dynamic estimation of action costs, and the second is to develop an analogous framework for dynamic estimation of action effect probabilities. Achieving these goals require appropriate problem formulations, algorithms, software implementation and finally empirical validation. We highlight that the first line of work falls into the category of deterministic planning, where the second belongs to probabilistic planning. Hence, the expectation is that pursuing each goal will require a different toolset, yet the similarities can help carry lessons learned from one line of work to the other.

Lastly, we wish to clarify a distinction between our intended setup for probabilistic planning, and standard Markov Decision Process (MDP) with unknown probabilities. While the latter is typically approached via Reinforcement Learning, so that an agent seeks to find a policy by trial and error (and in particular, *by acting*), our setup focuses on *pure planning*, where probabilities can be gradually estimated by calling appropriate estimators.

Research Status

We first briefly describe some of our achievements so far, and then continue to detail what is planned next. We note that most of the research that was carried out relates to cost estimation, where probability estimation is largely left for future work. In addition, since the results obtained for the latter have not yet passed external inspection, we do not present them here.

Dynamic Action Cost Estimation

Our framework employs the basic assumption that every ground action can potentially have multiple cost estimators, with varying degrees of accuracy and different running times. In particular we assume that once called, each estimator returns lower and upper bounds for the true action cost. Note that this does not prevent knowledge of exact costs (where the bounds are simply equal), nor the usage of bound priors, that can be specified in the initial problem model (these can be thought of as estimators that have fast $O(1)$ run time). It is worth mentioning that an anytime algorithm that serves as a cost estimator can in fact represent different estimators, where each of them is just an invocation of the same one but provided different running times.

Relying on this assumption, we then define a deterministic planning problem where the goal of the planner is to find a plan that meets a target sub-optimality multiplier as fast as possible. I.e., it aims to efficiently find a plan π^ϵ that satisfies

$$c(\pi^\epsilon) \leq c^* \times \epsilon,$$

with c^* being the optimal cost and $\epsilon \geq 1$. We proved that an algorithm which utilizes lower and upper bounds of costs, instead of exact values, can solve this problem by relying on the ratio of the accumulated bounds for the action costs composing the plan.

This lead to the development of ASEC (which stands for A^* with Synchronous Estimations of Costs) that implements this idea. ASEC serves as our principal algorithm for solving such problem instances, and we have been able to prove that it is sound, and incomplete in general, but is complete under special circumstances. Next, we developed a post-search procedure and an iterative framework, which both build on ASEC to obtain improved results. We further showed that applying a particular strategy for using ASEC within the iterative framework renders the resulting algorithm complete.

We implemented ASEC and its extensions by modifying and extending Fast Downward, and then empirically tested its performance on problems generated from planning competition benchmarks, which were added synthetic estimators. Our findings provide strong empirical evidence that ASEC outperforms alternatives w.r.t. run time, while typically meeting the target bound. These results, along with detailed analysis and another variant of ASEC, are summarized in a paper that is currently under submission process.

Planned Work We have three more objectives that we plan to pursue: 1. In the near future we intend to test various strategies for using ASEC within the iterative framework, since we have reason to believe that applying a data-dependent approach could yield run time improvements (at least in some cases). 2. The empirical results we obtained suggest that supporting a cache mechanism (for the estimated values) could provide considerable savings. Furthermore, it appears to make it simpler to develop an asynchronous version for ASEC, which might be more efficient. Hence, we plan to put it to test. 3. Lastly, we are considering to make stronger assumptions by adding meta-information about the estimators (such as expected run time), so that the planner could make more educated choices.

Challenges and Future Work

We suggest several interesting possibilities for future research. First, our work is clearly just the first offspring, and there may exist better algorithms to be discovered that solve the problems we suggested. Second, model uncertainty can be quantified using various statistical measures, leading to divergent problem setups, e.g., utilizing the Probably Approximately Correct (PAC) framework, or using standard deviations for setting a target bound on the plan cost. Third, a considerable challenge that arises from our proposed research is to connect actual external computational modules to the planner. Namely, in order to test the suggested ideas on real-world examples, one has to embark on a significant software development project, as each domain has its own relevant estimators and their unique APIs. This also makes it harder to compare different algorithms, as synthetic data (generated by synthetic estimators) might fail to reveal practical pain points. On the other hand, we believe this also presents an opportunity to increase the exposure of existing planning tools outside the research community.

References

Boddy, M. S. 2003. Imperfect match: PDDL 2.1 and real applications. *Journal of Artificial Intelligence Research*, 20: 133–137.

Fikes, R. E.; and Nilsson, N. J. 1971. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial intelligence*, 2(3-4): 189–208.

Fox, M.; and Long, D. 2003. PDDL 2.1: An extension to PDDL for expressing temporal planning domains. *Journal of artificial intelligence research*, 20: 61–124.

Francès, G.; Ramiŕez, M.; Lipovetzky, N.; and Geffner, H. 2017. Purely Declarative Action Descriptions are Overrated: Classical Planning with Simulators. In *IJCAI*.

Garrett, C. R.; Chitnis, R.; Holladay, R.; Kim, B.; Silver, T.; Kaelbling, L. P.; and Lozano-Pérez, T. 2021. Integrated task and motion planning. *Annual review of control, robotics, and autonomous systems*, 4: 265–293.

Gerevini, A.; Saetti, A.; and Vallati, M. 2014. Planning through automatic portfolio configuration: The pbp approach. *Journal of Artificial Intelligence Research*, 50: 639–696.

Ghallab, M.; Nau, D.; and Traverso, P. 2016. *Automated planning and acting*. Cambridge University Press.

Helmert, M. 2006. The Fast Downward Planning System. *Journal of Artificial Intelligence Research*, 26: 191–246.

Lipovetzky, N.; and Geffner, H. 2017. Best-first width search: Exploration and exploitation in classical planning. In *Thirty-First AAAI Conference on Artificial Intelligence*.

McCluskey, T. L. 2003. PDDL: A Language with a Purpose?

McDermott, D.; Ghallab, M.; Howe, A.; Knoblock, C.; Ram, A.; Veloso, M.; Weld, D.; and Wilkins, D. 1998. PDDL-the planning domain definition language. Technical Report CVC TR-98-003/DCS TR-1165, Yale Center for Computational Vision and Control.

Rintanen, J. 2015. Impact of modeling languages on the theory and practice in planning research. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.