Bridging the Gap with Angelic Semantics

Bhaskara Marthi BHASKARA@WILLOWGARAGE.COM
Willow Garage Inc.

Stuart Russell RUSSELL@CS.BERKELEY.EDU
Computer Science Division, University of California, Berkeley, CA 94720

Jason Wolfe* JAWOLFE@CS.BERKELEY.EDU
Computer Science Division, University of California, Berkeley, CA 94720

Abstract

We describe *angelic semantics*, a framework for hierarchical planning. In this framework, high level actions are described using bounds on their reachable sets. By reasoning about reachable sets, plans can often be pruned or committed to before searching over their refinements. We summarize earlier results showing large potential speedups on discrete problems, and argue that angelic semantics provides a promising framework for combining the various levels of decision making for robotics problems.

1. Introduction

Real-world planning problems often involve decision-making across multiple timescales and levels of abstraction. Consider, for example, a standard robotic mobile manipulation platform, which is given the task of tidying up a room. Some of the decisions that have to be made by the robot are:

- Task sequencing: which object to move next and where to move it
- Base position selection: where to move the base
- Navigation planning: how to move the base to this location
- Grasp point selection: where to grip the object
- Preshape selection: what the configuration of the end effector is before closing the gripper
- Inverse kinematics: a configuration of the arm that allows this end effector configuration to be achieved
- Arm motion planning: how to move the arm into this configuration
- Trajectory following: what torque commands will make the arm follow the required trajectory

These problems vary widely in their temporal scope, in their state representations, and in the techniques used to solve them. For example, a current decision system may use: for the highest level, a combinatorial technique such as reduction to an approximate TSP; for grasp point selection, a standard heuristic scoring function (Miller and Allen, 2004); for motion planning, a sampling-based cspace planner such as an RRT (Lavalle, 2006); for trajectory following, an optimal control technique such as differential dynamic programming (Tassa et al., 2007).

It is therefore challenging to get these levels to communicate. In practice, the flow of information may be described as ”top-down plus heuristics”. Thus, a high level decision may be made about which order to move the objects in, given a rough estimate of how long it will take to move each object to its position. Similarly, grasp point selection may be done using a fast estimate of how likely the given end effector configuration is to be feasible (Berenson et al., 2007).

Such a top-down combination is likely to work best when the levels are fairly independent, or when there exist accurate but efficient heuristic estimates of the lower levels. Once the levels become coupled in nontrivial ways, however, such a top-down scheme becomes less attractive. One problem is that, due to the lack of bottom-up information flow, much time may be wasted on unpromising high level plans. If we are

$*$. The authors appear in alphabetical order.

more aggressive about pruning away or committing to plans at the high level, it is hard to give completeness or (approximate) optimality guarantees of the overall planning system.

In this position paper, we argue for a richer flow of information between levels. In our framework, higher levels maintain and reason about bounds on the reachable sets of their lower-level refinements. These bounds often allow pruning away or committing to plans at the high level, while maintaining completeness and optimality guarantees. The technical material is based on previous work (Marthi et al., 2007, 2008). We are currently working on applying the techniques to tabletop manipulation problems.

2. Hierarchical Planning

We consider deterministic, fully observable, discrete-time environments. Such problems are described by a state space S, an action set A, a transition function $T : S \times A \rightarrow S$, a cost function $u : S \rightarrow R \cup \{\infty\}$, an initial state s_0, and a terminal state s_t.

As argued above, it is natural to divide planning problems into various levels of decision making. We formalize such decompositions using *action hierarchies*. Formally, a hierarchy consists of:

- A set of *high level actions* (HLAs) H;
- A designated top-level action $h_0 \in H$;
- For each $h \in H$, a set of *immediate refinements*, each of which is a sequence of (high-level or primitive) actions

Given an action sequence $\vec{a}$, we may repeatedly replace a high level action in it by one of its immediate refinements. The set of sequences obtained in this way are known as the *refinements* of $\vec{a}$. An action sequence is said to be consistent with the hierarchy if it is a primitive refinement of the top level action. The hierarchical optimal planning problem is to find an action sequence that minimizes total cost among those consistent with the hierarchy. The hierarchical satisficing problem is to find any sequence that achieves cost less than or equal to some threshold, or returns failure if no such sequence exists.

3. Angelic Semantics

A hierarchy may on its own speed up planning by structuring the search space. But even given a hierarchy, much time may be waste pursuing bad high level plans, or on searching at the high level after a satisfactory solution has already been found (in the satisficing case). To avoid this, we need *transition models* of the high level actions.

Our approach is based on *reachable sets*. The reachable set of a high level action h from a state s is defined to be the set of states s' such that, for some primitive refinement $\vec{a}$ of h, $T(s, \vec{a}) = s'$. The angelic description of high level action h is a function that takes in a state s and returns the reachable set.

Angelic descriptions have several attractive properties. First, they logically follow from the primitive action descriptions and hierarchy, as opposed to being purely heuristic. Thus, they provide a good foundation for algorithms that synthesize or learn descriptions. Second, they are compositional: the reachable set of a sequence of actions is constructed from the individual angelic descriptions using set union and function composition. Finally, they support high level reasoning. Indeed, a high level sequence has a succeeding primitive refinement iff its reachable set intersects the goal.

If we could efficiently compute angelic descriptions of any high level action, we would be able to solve PSPACE complete problems. Thus, we consider bounds on HLAs. An *optimistic description* is a pointwise superset of the exact angelic description. Optimistic descriptions support proofs that a given high level sequence cannot possibly achieve the goal, and therefore allow pruning at the high level. Similarly, a *pessimistic description* is a pointwise subset of the exact description. Pessimistic descriptions support proofs that a given high level sequence definitely can achieve the goal, and therefore allow committing to plans at the high level.

Reachable sets as described only allow reasoning about goal achievement. They can, however, be extended to the cost-based case by replacing sets with *valuations*, or real-valued functions on the state space (Marthi et al., 2008).

4. Algorithms

We have developed several algorithms that take advantage of angelic descriptions. At a high level, these algorithms can all be viewed as doing some variation of the following:

1. Iterate over high level sequences;
2. If the next sequence cannot possibly achieve the goal, discard it;
3. If it can definitely achieve the goal:
   (a) If it is primitive, return it;
   (b) Otherwise, commit to it;
4. Else, add its refinements to the set of active plans, and continue.

Step 2) uses optimistic descriptions, and step 3) uses pessimistic descriptions. The algorithms include: AHA*, an optimal planner similar to A*; AHSS, a satisficing planner; AHLRTA*, an online planner similar to LRTA*, for the case when limited computation time is available.

5. Results

We have implemented our algorithms for the case of discrete domains, using an extension of STRIPS known as NCSTRIPS that supports angelic nondeterminism. Figure 1 is an example of the kinds of speedup that may be obtained. The results are on a domain known as the nav-switch problem, in which the low-level consists of navigation in a grid, and the high-level involves navigating to and flipping switches that increase or decrease the cost of certain moves. Hierarchical optimal planning is at least an order of magnitude faster than flat A*, and hierarchical satisficing planning allows a further speedup.

6. Related Work and Conclusions

The ideas of hierarchical decomposition and reasoning about reachable sets have appeared in several fields (Nau et al., 2004; Bhatia and Frazzoli, 2008; Plaku et al., 2007; Parr and Russell, 1998). We summarize the related work here; a more thorough overview can be found in (Marthi et al., 2008). Hierarchical task networks (Nau et al., 2004) have achieved impressive results on real world discrete planning tasks. In our language, an HTN corresponds to a particular type of hierarchy. HTNs additionally have annotations on actions, specifying conditions that must hold after an action is completed, but these annotations are treated as constraints on the planning process, rather than as independently verifiable facts about the action. We believe that the performance of HTNs can be further improved using angelic bounds.

There have also been several proposed hierarchical extensions to A* (Holte et al., 1996; Bulitko et al., 2007). These approaches differ from ours in that they consider hierarchical abstractions of the *state space* rather than actions.

In the robotics literature, (Plaku et al., 2007) and (Cambon et al., 2009) are two recent approaches that simultaneously do motion planning and task planning. These approaches currenty only support feasible rather than optimal planning, and focus on the case of a two-level hierarchy with a task and motion planning level.

We view the main contribution of this work as being a representation-independent unifying framework, and complete, hierarchically optimal planning algorithms. We believe the attractive theoretical properties of angelic descriptions mentioned above provide a firm foundation for future work, including nondeterministic and stochastic transition models, partially observable state, and the learning of hierarchies and descriptions. Until such learning algorithms exist, in the short term more experimentation is needed to determine the best way to structure hierarchies for a variety of domains.

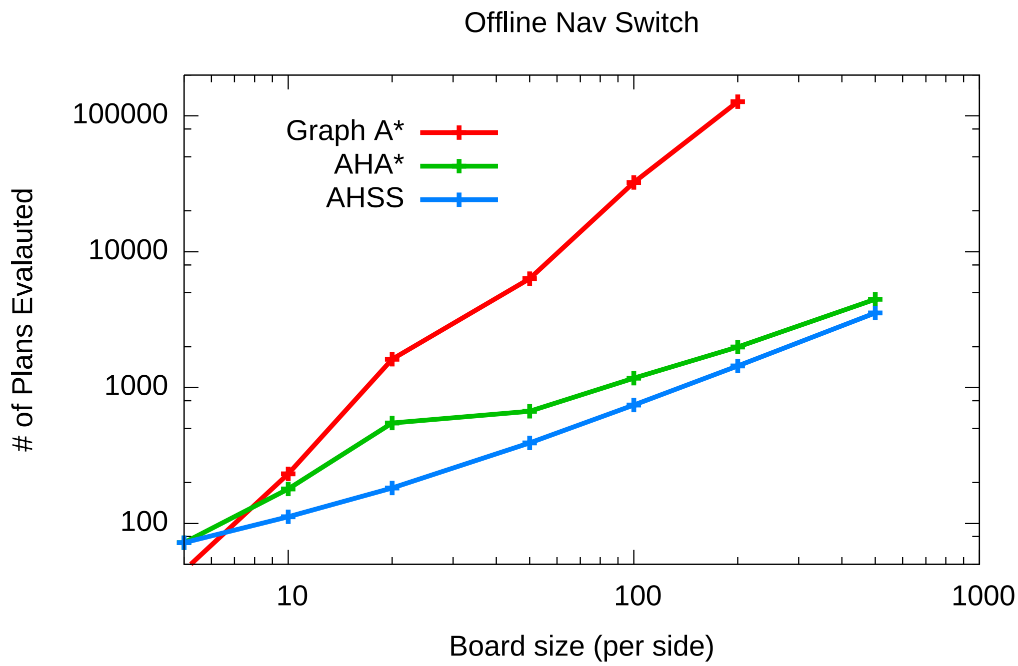

Figure 1: Number of plans evaluated (note log-scale) to find an (optimal) solution, on the nav-switch problem. The algorithms are (flat) graph A*, AHA*, and AHSS with threshold $\alpha{=}\infty$. Algorithms were terminated if they failed to return within 10^4 seconds (shown as "–").

References

Dmitry Berenson, Rosen Diankov, Koichi Nishiwaki, Satoshi Kagami, and James Kuffner. Grasp planning in complex scenes. In *IEEE RAS Conf. on Humanoid Robots*, 2007.

Amit Bhatia and Emilio Frazzoli. Sampling-based resolution-complete algorithms for safety falsification of linear systems. In *HSCC*, pages 606–609, 2008.

Vadim Bulitko, Nathan Sturtevant, Jieshan Lu, and Timothy Yau. Graph Abstraction in Real-time Heuristic Search. *JAIR*, 30:51–100, 2007.

Stéphane Cambon, Rachid Alami, and Fabien Gravot. A hybrid approach to intricate motion, manipulation and task planning. *Int. J. Rob. Res.*, 28(1):104–126, 2009. ISSN 0278-3649. doi: http://dx.doi.org/10.1177/0278364908097884.

R Holte, M Perez, R Zimmer, and A MacDonald. Hierarchical A*: Searching abstraction hierarchies efficiently. In *AAAI*, 1996.

Steven Lavalle. *Planning Algorithms*. Cambridge University Press, 2006.

Bhaskara Marthi, Stuart J. Russell, and Jason Wolfe. Angelic Semantics for High-Level Actions. In *ICAPS*, 2007.

Bhaskara Marthi, Stuart J. Russell, and Jason Wolfe. Angelic Hierarchical Planning: Optimal and Online Algorithms. In *ICAPS*, 2008.

Andrew Miller and Peter Allen. Graspit! a versatile simulator for robotic grasping. *IEEE Robotics and Automation Magazine*, 2004.

Dana Nau, Malik Ghallab, and Paolo Traverso. *Automated Planning: Theory & Practice*. Morgan Kaufmann, 2004. ISBN 1558608567.

Ronald Parr and Stuart Russell. Reinforcement Learning with Hierarchies of Machines. In *NIPS*, 1998.

Erion Plaku, Lydia E. Kavraki, and Moshe Y. Vardi. Discrete search leading continuous exploration for kinodynamic motion planning. In *Robotics: Science and Systems*, 2007.

Yuval Tassa, Tom Erez, and William D. Smart. Receding horizon differential dynamic programming. In *NIPS*, 2007.