

# Hierarchical Planning for Mobile Manipulation

**Jason Wolfe**

Computer Science Division  
University of California, Berkeley  
Berkeley, CA 94720  
jawolfe@cs.berkeley.edu

**Bhaskara Marthi**

Willow Garage, Inc.  
Menlo Park, CA 94025  
bhaskara@willowgarage.com

**Stuart Russell**

Computer Science Division  
University of California, Berkeley  
Berkeley, CA 94720  
russell@cs.berkeley.edu

Humans somehow manage to choose quite intelligently the 20 trillion primitive motor commands that constitute a life. It has long been thought that hierarchical structure in behavior is essential in managing this complexity. Structure exists at many levels, ranging from small (hundred-step?) motor programs for typing characters and saying phonemes up to large (billion-step?) actions such as writing an ICAPS paper, getting a good faculty position, and so on.

Hierarchical planning attempts to capture this source of power. It has a rich history of contributions going back to the seminal work of Tate (1977). The basic idea is to supply a planner with a set of high-level actions (HLAs) in addition to the primitive actions. Each HLA admits one or more *refinements* into sequences of (possibly high-level) actions that implement it. Given a hierarchy, a planner can start with the designated top-level action  $Act$ , and repeatedly replace HLAs with their immediate refinements until a fully *primitive refinement* is found that reaches the goal (with low cost).

For example, consider the task of tidying up a room by putting away objects in a set of target regions. In one possible hierarchy for this task,  $Act$  has recursive refinements  $MoveToGoal(o), Act$  ranging over all objects  $o$  that are not yet in their goal positions, or just the empty refinement if this set is empty.  $MoveToGoal(o)$  refines in turn to  $GoPick(o), GoPlace(o, p)$ , ranging over positions  $p$  in the goal region of  $o$ . These HLAs can be refined further, to generate the appropriate sequence of base, arm, torso, and gripper primitives to effect the appropriate pick or place operation. At the bottom of the hierarchy, primitive actions (e.g., for arm or base movements) can be modeled by calling out to external solvers such as rapidly-exploring random trees (RRTs). This hierarchy specifies the general structure of a solution, but leaves open details such as the ordering of the pick-and-place operations, specific target locations for each object, base positions, and arm trajectories, which the planner should fill in to produce a concrete plan that accomplishes the goal as quickly as possible.

Planning at multiple levels of abstraction has long been a staple of the robotics community. For instance, Shakey the robot used STRIPS for high-level task planning, then called out to separate low-level planning/control algorithms to execute each of the planned actions (Fikes and Nilsson 1971). This hard separation of levels, where a high-level plan is chosen before considering low-level details, greatly

simplifies the task planning problem. However, the resulting plans may be inefficient or even infeasible due to missed lower-level synergies and conflicts. For example, suppose a one-armed robot is to move (1) a cookbook from the counter to the shelf, and (2) a remote control from the shelf to the coffee table. The task planner might schedule (2) before (1), unaware that the tasks could be done much more efficiently in the opposite order. Or, worse, it might schedule (1) before (2) despite the fact that putting the book away would block the remote, rendering the high-level plan infeasible.

An alternative strategy, which we advocate, is to interleave planning at all levels of abstraction. Since lower-level interactions are accounted for, the resulting solutions are guaranteed to be feasible and of high quality. Unfortunately, however, the corresponding space of potential kinematic solutions is far too large to search exhaustively in a reasonable amount of time. Our ongoing research projects aim to compress and prune this space in several ways, making the approach tractable for real robotics problems.

First, in recent work (Wolfe, Marthi, and Russell 2010) we have designed and implemented a planning system for room-cleaning tasks as described above, and tested it on a prototype PR2 robot constructed by Willow Garage, Inc (Wyrobek et al. 2008). In this work, we assume the state of the world is known (approximately), and consider the resulting decision problem. In particular, we search for the best possible (a.k.a. *hierarchically optimal*) solution generated by a finite version of the hierarchy described above, where continuous choices (e.g., object goal positions) are made discrete by sampling a finite set of refinements for each HLA. As we sample refinements more and more densely, the quality of this solution is guaranteed to approach that of the best plan allowed by the original (un-sampled) hierarchy.

We also implemented, for each primitive action, a function to execute it on the PR2. The execution primitives were responsible for implementing perceptual feedback and returning a success flag, which was used to implement a simple executive that retried failed actions. A video of the PR2 executing a 4-object plan is available at [http://www.ros.org/wiki/Papers/ICAPS2010\\_Wolfe](http://www.ros.org/wiki/Papers/ICAPS2010_Wolfe).

Several features of the planner contribute to its ability to (relatively) quickly find high-quality solutions. First, our hierarchical problem formulation focuses effort on plans

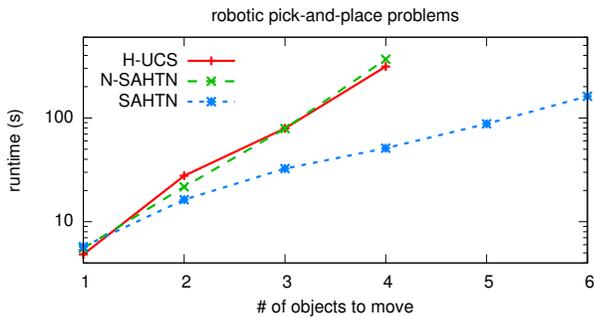


Figure 1: Runtimes (averaged over three runs) for SAHTN and two competing algorithms (which shared our hierarchical problem formulation) on pick-and-place tasks, as a function of the number of objects to be moved.

that are likely to work, while leaving open enough options that optimality is (hopefully) not compromised much. Second, it uses a novel hierarchical search algorithm called SAHTN (State-Abstracted Hierarchical Task Network planner), which uses *subtask-specific irrelevance* to dramatically speed up search within this hierarchy (see Figure 1). For example, the best way to move the arm to pick up object 23 depends only on the position of the base, arm, object 23, and nearby objects, and the results of such planning can be reused every time this subproblem occurs in the search space.

One could hope for more, however. Humans routinely commit to high-level plans such as “take out the trash” and “run for president” without preplanning the motor commands involved. This ability to commit to high-level plans (without sacrificing completeness or optimality) before refining them further is called the *downward refinement property* (Bacchus and Yang 1991). Because it lacks this elusive property, SAHTN must refine a high-level plan all the way down to the level of primitive motor commands to determine its feasibility and cost, leading to prohibitively slow runtimes for tasks involving more than 5 or so objects.

In a separate line of previous work, we proposed an “angelic semantics” for HLAs (Marthi, Russell, and Wolfe 2007; 2008) that provides a principled definition for what it means to *do* an HLA. The basic idea is to, along with the structure of the hierarchy, provide a planning algorithm with *approximate models* that specify upper and lower bounds on the sets of states that can be reached (and corresponding costs) by each HLA. When chained together, these models support proofs that a given high-level plan does or does not have a primitive refinement that (optimally) reaches a goal state. The resulting planning algorithms automatically possess the downward refinement property, and gain significant speedups from their ability to prune and commit to high-level plans without further refinement (see Figure 2).

We are currently working on combining these two lines of work, producing an enhanced version of SAHTN that can leverage angelic approximate models to reduce computation time and enable scaling to (much) larger problems. For example, angelic models for `Act` may enable pruning obviously suboptimal top-level task orderings without refining them further, and models for `Pick` may rule out infeasible

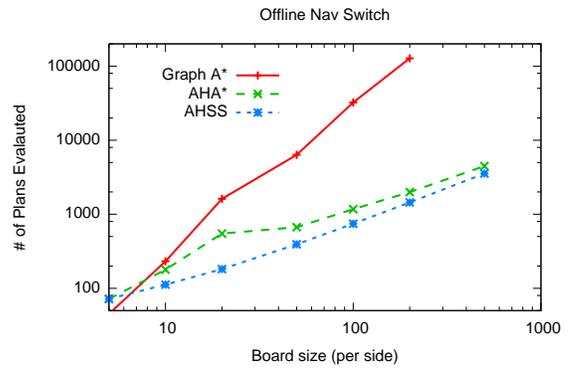


Figure 2: Performance of angelic algorithms AHA\* and AHSS versus A\*, on “nav switch” problems.

grasps before wasting tens of milliseconds calling an external arm planner. If we are willing to accept slightly suboptimal solutions, angelic models will also allow committing to high-level plans that are provably “good enough”, and discarding all other plans without refining them further. The efficiency of these new algorithms may approach that of the “hard multi-level” algorithms described earlier, while retaining the guaranteed feasibility and high quality of the plans found by an exhaustive algorithm like SAHTN.

In our previous work on angelic planning, we also considered HLAs in the *online* setting, wherein an agent performs a limited lookahead prior to selecting each action. In this setting, angelic models enable *hierarchical* lookahead with HLAs, which can bring back to the present value information from far into the future. Put simply, it’s better to evaluate the possible outcomes of taking out the trash first, than the possible outcomes of moving joint 3 to angle 2.71 first. Applying these ideas to robotics problems will enable faster planning algorithms with longer decision horizons, and perhaps lead to agents that can more easily adapt online to unexpected changes in the world or tasks at hand.

## References

- Bacchus, F., and Yang, Q. 1991. The downward refinement property. In *Proc. IJCAI '91*, 262–292.
- Fikes, R., and Nilsson, N. J. 1971. STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving. *Artif. Intell.* 2:189–208.
- Marthi, B.; Russell, S. J.; and Wolfe, J. 2007. Angelic Semantics for High-Level Actions. In *ICAPS*.
- Marthi, B.; Russell, S. J.; and Wolfe, J. 2008. Angelic Hierarchical Planning: Optimal and Online Algorithms. In *ICAPS*.
- Tate, A. 1977. Generating project networks. In *IJCAI*.
- Wolfe, J.; Marthi, B.; and Russell, S. J. 2010. Combined Task and Motion Planning for Mobile Manipulation. In *ICAPS (to appear)*.
- Wyrobek, K. A.; Berger, E. H.; der Loos, H. F. M. V.; and Salisbury, J. K. 2008. Towards a personal robotics development platform: Rationale and design of an intrinsically safe personal robot. In *ICRA*, 2165–2170.