Hierarchical Planning

Ron Parr
CPS 570

Motivation for Hierarchical Planning

• Planning is hard

• Plans to achieve important/useful things can involve long sequences of actions
  – Finding these plans can be hard
  – Can be exponential in plan length

• Exhaustive search seems unnecessary
  – We don’t do that
  – Not all sequences of plans make sense

Hierarchical Trip Planning

• Plan to get from here to the west coast
• Don’t plan at the level of individual steps

• Experience tells us that most reasonable plans follow a template:
  – Walk to your car
  – Drive to the airport
  – Fly to the west coast
  – Take a taxi to your hotel
  – Walk to your hotel room

Hierarchy Restricts our Search

• We may find suboptimal plans because we fail to consider certain options, e.g., driving to a more distant airport to take a non-stop flight

• Depending upon how persistent we are, we may fail to find plans, e.g., if all flights from your favored airport are booked, do you consider other options?

• Hierarchical optimality finds the best plan consistent with our goal hierarchy, though not necessarily best overall
One View of Human Planning

• Humans never solve for very complicated plans
• They always solve for fairly small plans that result from decomposing problems into smaller problems
• Such decompositions may be learned or developed on our own

• Note: This is an introspective approach to gaining insight into intelligent behavior – not always a good idea

Views of Hierarchical Planning

• We can provide the planner with a hierarchical decomposition of the task to guide the planner

• We could ask the planner to discover such a task decomposition on its own – very hard

HTN Planning

• Hierarchical task networks are a language for providing task decompositions to a planner
• HTNs (and things like them) are the prior work upon which Marthi et al. are building

• Note: Papers often blend together representations and algorithms, but it’s good to remember that they can be teased apart. (Marthi et al. do a good job of teasing these apart.)

Limitations of Prior HTN Work (What are Marthi et al. Fixing?)

• Challenge in hierarchical planning: Dealing with how abstract actions/plans get refined

• Suppose you have the abstract plan of:
  – Drive to your airport
  – Fly to your destination city
  – Take a cab to your hotel

• Traditionally, plans should be guaranteed to succeed
• What if your hotel is in Venice?
Dealing with Plan Failure in HTN Planning

- At the planner level:
  - Fully refine plans before declaring success
  - Add “ground level” actions so that planner can fall back on planning the hard way if hierarchy fails
  - Add LOTS OF abstract actions
  - Downside: Successful use of HTN planning is often very much like programming, requires lots of effort an anticipation of possible refinement failures

- At run time:
  - Could decide to refine the plan on the fly
  - Risky, but that different from what people do?
  - Acceptable as a guarantee?

Angellic Semantics

- What if high level action preconditions were sufficient to ensure existence of refinement achieving desired effects?
  - Can plan at high level, yet
  - Refine plans at our leisure/as needed
  - Avoid wasting time discovering that high level plans can’t be refined

- Can introduce upper/lower bounds on cost of refinement to guide planning
- Can use bounds on costs to find not just a satisfactory plan, but a good one

Do you believe in angels?*

- Is it reasonable to hope that we can have HLA descriptions with angellic semantics?
- Maybe
  - Definitely seems plausible for domains in the paper
  - Could force us to create more actions that are specialized for different preconditions

- Ultimately faces the qualification problem, which is also a motivation for probabilistic reasoning

Some polls show that over 70% of Americans believe in angels, a classic example of an untestable hypothesis!

Concluding Thoughts

- Generally accepted that hierarchical planning is the “right” way to do planning
- Many approaches from something close to programming to fully automated abstraction
- Lack of common language makes comparisons difficult
- Success often achieved in specific domains through careful integration of domain expertise
- Angellic semantics may help relieve some of the “fiddliness” of hierarchical planning