Modern Planning Techniques

John Levine
AI Applications Institute
Division of Informatics
University of Edinburgh

Contents

Introduction

- What is Planning?
- Some Connections to NLP
- Modern Planning
- Assumptions for Planning
- Classification of Planners

What is Planning?

"The solution of a problem – any problem – consists in discovering how to transform an existing state of affairs into one that has not yet come into being."

– Margaret Donaldson, "Children's Minds"

Some Connections to NLP

- Language as part of rational behaviour
- Planning in the generation of language
- Allowing planning systems to communicate plans

Modern Planning

- Much progress in the last 6-8 years
- Planners which search are more efficient
- New knowledge-based techniques
- New approaches to learning in planning
- New approaches to optimisation
- The AIPS competitions and benchmarks

Assumptions for Planning

- No interfering agents in the world
- No interfering external events
- All actions are deterministic
- Planner has complete knowledge
- All world states are sets of facts
- No fuzziness or uncertainty
- Actions are quantum transitions

Classification of Planners

- Linear planners
- Parallel time-step planners
- Partial-order planners

- Search-based planners
- Knowledge-based planners
- Planners which learn

SAT Planning: Blackbox

- Kautz and Selman, 1998
- Blackbox = SATPlan + Graphplan
- Parallel time-step planner
- Creates a planning graph
- Converts to CNF for SAT solving
- Various SAT solvers

Blackbox: Performance

- MUCH better than SATPlan or Graphplan
- Extremely good at logistics problems
- Not so brilliant for other benchmarks
- Can encode domain-dependent knowledge
- Tends to run out of memory due to the size of the planning graph

Heuristic Planning: FF

- Hoffmann and Nebel, 2001
- Linear planner
- Son of HSP (Bonet and Geffner, 1998)
- FF uses enforced hill-climbing
- Heuristic from relaxed problem, $\mathcal{P}'$
- Solve $\mathcal{P}'$ using graphplan

FF: Performance

- Fastest search-based planner at AIPS 2000
- Creates over-long plans
- Incomplete strategy $\longrightarrow$ best-first search
- Very good at all competition domains
- Not good at all domains

Search-Based Planning: Summary

- MUCH better than it used to be
- Very active research area
- Restrictive assumptions
- Somewhat artificial benchmarks
- Loses out to knowledge-based planners
- "I don't plan – I just do."

Policy-Based Planning

- TLPlan (Baccus and Kabanza, 1998)
- SHOP (Nau *et al.*, 1999)
- TALPlanner (Doherty and Kvarnström, 2000)
- All these are linear planners
- Plan using $\langle state, goal \rangle \rightarrow action$ rules:
  1. If you can grab a well-placed block, do it.
  2. Put any non-well-placed block on the table.
- Policy set = planning algorithm

Policy Sets: Performance

- Complete and very fast
- MUCH faster than search-based planners
- Creates non-optimal (over-long) plans
- Need to encode the policy sets by hand
- Learning policies:
  - Khardon (1999)
  - Martín and Geffner (2000)

Optimisation of Plans

- ANY plan? Or the best plan?
- Optimizing policy-generated plans:
- Ambité and Knoblock (1998)
  - Uses hand-encoded rewrite rules
- Westerberg and Levine (2001)
  - Genetic optmisation
  - Domain independent technique
  - Also works as a post-processor

O-Plan Optimal

- O-Plan (Currie and Tate, 1991) is an expansion-based HTN planner
  - choice of schema, S_i, for high-level action A_k
  - choice of resources, $R_i = \{r_0, r_1 \ldots r_n\}$ to use for S_i
- Partial-order planner
- O-Plan Optimal (Ruscio, Levine, Kingston, Kothari):
  - Simplified version of O-Plan
  - All choices encoded as a genotype
  - Final scheduled plan is the phenotype
  - Try to find the optimal plan

Knowledge in Planning

1. Decision theoretic planning
2. Deterministic policy rules
3. Probabilistic policy rules
4. Shortcuts, e.g. HTN schemas
5. Domain specific heuristics $h(s_i, g)$
6. Search control rules

All of these can be learnt...