A Beginner’s Introduction to Heuristic Search Planning

7. Going Further

Malte Helmert  Gabriele Röger

AAAI 2015 Tutorial

January 25, 2015
What else happens in heuristic planning?
Other current topics on **heuristics for classical planning**:

- making delete relaxations less relaxed
  - semi-relaxed planning (Keyder et al., ICAPS 2012)
  - red-black planning (e.g., Katz et al., AAAI 2013)

- heuristics based on linear programming
  (e.g., Pommerening et al., ICAPS 2014; AAAI 2015)

- heuristics vs. search vs. inference
  (e.g., Lipovetzky & Geffner, ECAI 2012)
Search algorithms mostly interesting for satisficing planning:

- escaping local minima and heuristic plateaus:
  - using multiple heuristics (e.g., Röger & Helmert, ICAPS 2010)
  - combining local and systematic search
    (e.g., Hoffmann & Nebel, JAIR 2001; Xie et al., AAAI 2014)
  - randomized systematic search (e.g., Valenzano et al., AAAI 2014)
  - random-walk algorithms (e.g., Nakhost & Müller, IJCAI 2013)

- search algorithms tailored to specific objectives:
  - anytime strategies for satisficing planning
    (Richter & Westphal, JAIR 2010)
  - many papers from Wheeler Ruml’s research group
    (e.g., Thayer et al., ICAPS 2012)
Search space pruning (mainly for optimal search):

- with symmetries (e.g., Pochter et al., IJCAI 2011)
- with redundant paths ("partial-order reduction"): (e.g. Wehrle et al., ICAPS 2013)
Invariant Synthesis

- **Invariant**: property of all reachable states
  (e.g., Rintanen, ECAI 2008)

- **Example**: mutually exclusive atoms
  - set of atoms of which at most one is true in each state
  - basis of translation from PDDL to SAS$^+$ (Helmert, AIJ 2009)
  - useful for strengthening heuristics, e.g., constrained PDBs (Haslum et al., AAAI 2005)
Preferred Operators

- heuristics often can identify **promising actions**
- for example **helpful action**: first action in a relaxed plan
  
  \[(\text{Hoffmann & Nebel, JAIR 2001})\]

- try these actions first during search

- often highly positive impact on overall performance

  \[(\text{Richter & Helmert, ICAPS 2009})\]
Portfolios

- **Idea**: solve tasks by running multiple (more or less) independent planning systems
- different strategies:
  - fixed schedule
  - select planners after task analysis

≣ results of IPC 2014 “classic tracks”
http://helios.hud.ac.uk/scommv/IPC-14/

≣ results of IPC 2014 “learning track”
http://www.cs.colostate.edu/~ipc2014/
What else happens in heuristic planning? ... in classical planning? ... in planning?

What else happens ... ... in classical planning?
Symbolic search algorithms:

- search processes sets of states at a time
- operators, state sets, heuristics etc. represented by binary decision diagrams (BDDs) (or related structures)
- planning systems:
  - Gamer (Edelkamp & Kissmann, IPC 2008)
  - SymBA* (Torralba et al., IPC 2014)
  - ...
- comprehensive treatment: Álvaro Torralba’s PhD thesis (Torralba, 2015)
SAT planning:
- create propositional formula that is satisfiable iff there is a plan with at most $k$ steps
- use off-the-shelf or specially tailored SAT solver to find plans
- e.g., Madagascar planner (Rintanen, IPC 2011 & 2014)
- many papers by Jussi Rintanen (e.g., Rintanen; AIJ 2012)
- related: property-directed reachability (Suda, JAIR 2014)
### Other Approaches

Many other approaches:

- partial-order causal link planning (POCL)
- compilation to (mixed) integer programming (IP/MIP)
- compilation to answer-set programming (ASP)
- compilation to quantified boolean formulae (QBF)

... but heuristic search, symbolic search and SAT are currently working best
What else happens in heuristic planning? ... in classical planning? ... in planning?

What else happens ... ... in planning?
More general kinds of planning include:

- **offline**: online planning; planning and execution
- **discrete**: continuous planning (e.g., real-time/hybrid systems)
- **deterministic**: FOND planning; probabilistic planning
- **single-agent**: multi-agent planning; general game playing; game-theoretic planning
- **fully-observable**: POND planning; conformant planning
- **sequential**: e.g., temporal planning

**Domain-dependent** planning problems in AI include:

- pathfinding, including grid-based and multi-agent (MAPF)
- continuous motion planning
Resources
Resources

- ICAPS conferences (International Conference for Automated Planning and Scheduling):
  http://www.icaps-conference.org/
  http://icaps15.icaps-conference.org/

- Planning Domain Definition Language (PDDL):
  http://ipc.informatik.uni-freiburg.de/PddlResources

- International Planning Competitions (IPC):
  http://ipc.icaps-conference.org/

- Fast Downward planning system:
  http://www.fast-downward.org/

- VAL plan validator:
  https://github.com/KCL-Planning/VAL
Thank you for your attention!