Learning Relational Decision Trees for Guiding Heuristic Planning

Tomás de la Rosa, Sergio Jiménez and Daniel Borrajo

Planning and Learning Group (PLG)
Departamento de Informática
Universidad Carlos III de Madrid

September 17, 2008

Outline

Heuristic Planning

Advantages

- One of the top approaches in AI Planning
- Heuristic function correctly leads the search in most classical planning benchmarks
- Used for other planning paradigms

Issues

- Scalability: Node evaluation is expensive.
- In some benchmarks heuristic function is not good enough.

Heuristic Planning

Advantages

- One of the top approaches in AI Planning
- Heuristic function correctly leads the search in most classical planning benchmarks
- Used for other planning paradigms

Issues

- Scalability: Node evaluation is expensive.
- In some benchmarks heuristic function is not good enough.

Learning for Heuristic Planning

Learning Opportunities

- Avoiding node evaluations
- Developing more accurate heuristics

Other Learning Approaches

- Macros [Botea et al., 2005, Coles and Smith, 2007, Newton et al., 2007]
- Cases [De la Rosa et al., 2007]
- Heuristic Functions [Yoon et al., 2006, Xu et al., 2007]
- General Policies [Khardon, 1999, Martin and Geffner, 2004, Yoon et al., 2007]

Learning for Heuristic Planning

Learning Opportunities

- Avoiding node evaluations
- Developing more accurate heuristics

Other Learning Approaches

- Macros [Botea et al., 2005, Coles and Smith, 2007, Newton et al., 2007]
- Cases [De la Rosa et al., 2007]
- Heuristic Functions [Yoon et al., 2006, Xu et al., 2007]
- General Policies [Khardon, 1999, Martin and Geffner, 2004, Yoon et al., 2007]

Learning Helpful Context Policies

- Learning an action policy in form of relational decision trees
- Using a relational classifier (TILDE)
- The target is which instantiated action to select among the applicable candidates

Learning Helpful Context Policies

Learning Phases

- Generation of Learning Examples
- Action Classification
- Binding Classification

Learning Examples

Helpful Context

- Set of Helfpul Actions
- Target Goals (goals remaining in the problem)
- Problem Static Facts
- *Executed Action*

Learning Examples

Helpful Context Example

% Static Predicates of problem
static_fact_calibration_target(sat_prob,instrument0,star0).
static_fact_supports(sat_prob,instrument0,infrared2).
static_fact_supports(sat_prob,instrument0,spectrograph1).
static_fact_on_board(sat_prob,instrument0,satellite0).

% Example sat_E1
selected(sat_e1,sat_prob,switch_on).
candidate_turn_to(sat_e1,sat_prob,satellite0,phenomenon3,star0).
candidate_turn_to(sat_e1,sat_prob,satellite0,phenomenon4,star0).
candidate_switch_on(sat_e1,sat_prob,instrument0,satellite0).
target_goal_have_image(sat_e1,sat_prob,phenomenon3,infrared2).
target_goal_have_image(sat_e1,sat_prob,phenomenon4,infrared2).

Learning Examples

Why not the state?

- Easier matching
- Helpful action encode information about goals
- Better recursive predicate handling

Learning Examples

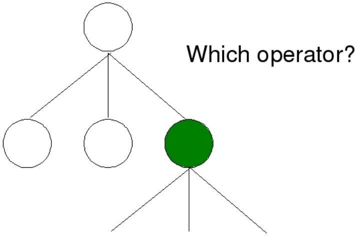
Generating Examples

- Training problems solved with EHC and refined with DfBnB
- From each node of a best-cost solution
  - An example of helpful context for action classification
  - Examples of helpful context for binding classification

Action Classification

Action Examples

- Helpful Context
- The class is the selected operator
- *Static predicates are shared by all problem examples*



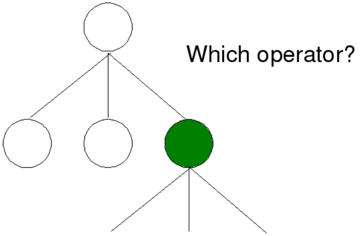
Output

- **Action decision tree**

Action Classification

Action Examples

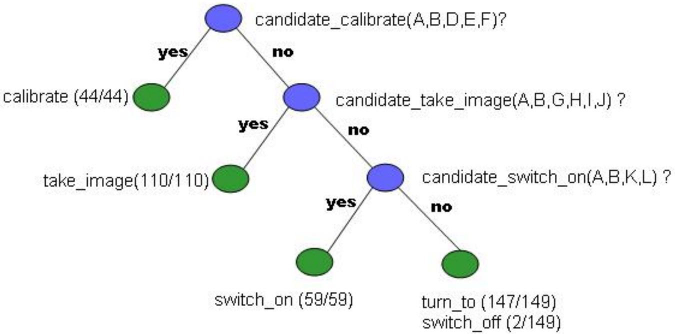
- Helpful Context
- The class is the selected operator
- *Static predicates are shared by all problem examples*



Output

- **Action decision tree**

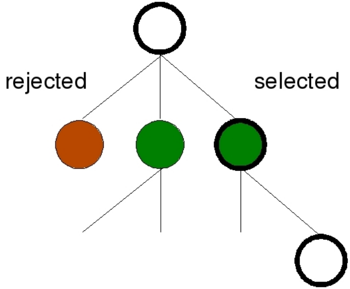
Action Decision Tree



Binding Classification

Binding Examples

- Helpful Context
- The positive classes are the operator bindings in one of the best-cost solutions
- The negative classes are the operator bindings not present in a best-cost solution



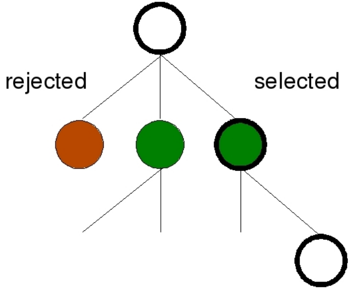
Output

- One binding decision tree **for each operator in the domain**

Binding Classification

Binding Examples

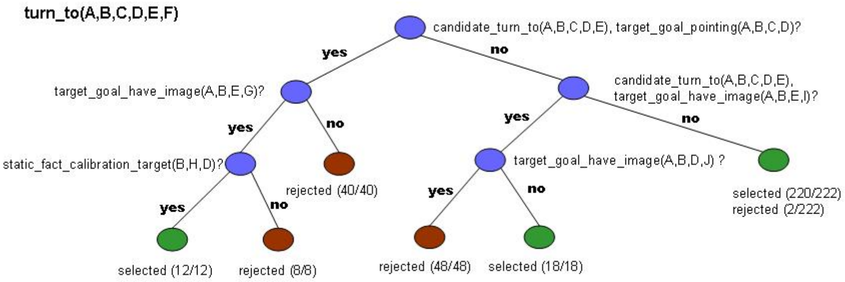
- Helpful Context
- The positive classes are the operator bindings in one of the best-cost solutions
- The negative classes are the operator bindings not present in a best-cost solution



Output

- One binding decision tree **for each operator in the domain**

Binding Decision Tree



Planning with Helpful Context Policy

Helpful Context Depth-first Search

At each node

- Helpful Context computation
- Candidate ordering
  - by the action tree leaf matching current context
  - by the selected/rejected ratio of the binding tree leaf

A backtrack-free search is the execution of the Helpful Context Policy

The best case $\Rightarrow$ $(Plan\_length + 1)$ heuristic evaluations

Planning with Helpful Context Policy

Helpful Context Depth-first Search

At each node

- Helpful Context computation
- Candidate ordering
  - by the action tree leaf matching current context
  - by the selected/rejected ratio of the binding tree leaf

A backtrack-free search is the execution of the Helpful Context Policy

The best case $\Rightarrow$ $(Plan\_length + 1)$ heuristic evaluations

Planning with Helpful Context Policy

Sorted EHC

At each node

- Helpful Context computation
- Candidate ordering for heuristic evaluation

- Node evaluation reduction, but the search still relies on heuristic function performance

Planning with Helpful Context Policy

Sorted EHC

At each node

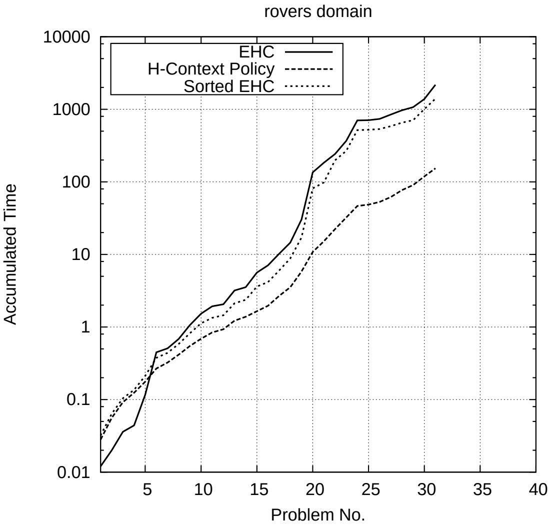
- Helpful Context computation
- Candidate ordering for heuristic evaluation

- Node evaluation reduction, but the search still relies on heuristic function performance

Experimental Results (problems solved)

Domain (problems)	EHC	H-Context Policy	Sorted-EHC
Blocksworld (103)	20	103	21
Miconic (150)	150	150	150
Logistics (79)	72	79	73
Zenotravel (20)	19	20	19
Satellite (36)	23	28	26
Rovers (40)	31	40	33
TPP (30)	19	30	19

Experimental Results



Conclusions

Representation

- Helpful Context as an alternative for representing the meta-state of the search
- Hierarchical information for planning benchmarks

Learning

- Control-knowledge acquisition as a classification task TILDE
- Generation of better training examples (DFBnB)

Planning

- Helpful Context DFS and Sorted EHC: additional means for embedding policies within search
- Reducing node evaluations for handling scalability problems

Conclusions

Representation

- Helpful Context as an alternative for representing the meta-state of the search
- Hierarchical information for planning benchmarks

Learning

- Control-knowledge acquisition as a classification task TILDE
- Generation of better training examples (DFBnB)

Planning

- Helpful Context DFS and Sorted EHC: additional means for embedding policies within search
- Reducing node evaluations for handling scalability problems

Conclusions

Representation

- Helpful Context as an alternative for representing the meta-state of the search
- Hierarchical information for planning benchmarks

Learning

- Control-knowledge acquisition as a classification task TILDE
- Generation of better training examples (DFBnB)

Planning

- Helpful Context DFS and Sorted EHC: additional means for embedding policies within search
- Reducing node evaluations for handling scalability problems

Current Work

Helpful Context Lookahead Search IPC Competitor

At each node expansion a lookahead node is included

- Helpful Context computation
- Applicable actions of the relaxed plan are sorted
  - by the action tree leaf matching current context (*still applicable*)
  - by the selected/rejected ratio of the binding tree leaf

Thanks

Botea, A., Enzenberger, M., Müller, M., and Schaeffer, J. (2005).
Macro-FF: Improving AI planning with automatically learned macro-operators.
JAIR, 24:581–621.

Coles, A. and Smith, A. (2007).
Marvin: A heuristic search planner with online macro-action learning.
JAIR, 28:119–156.

De la Rosa, T., García-Olaya, A., and Borrajo, D. (2007).
Using cases utility for heuristic planning improvement.
In *Proceedings of the 7th International Conference on CBR*, pages 137–148.

Khardon, R. (1999).
Learning action strategies for planning domains.
Artificial Intelligence, 113:125–148.

Martin, M. and Geffner, H. (2004).
Learning generalized policies from planning examples using concept languages.
Appl. Intell, 20:9–19.

Newton, M. A. H., Levine, J., Fox, M., and Long, D. (2007).
Learning macro-actions for arbitrary planners and domains.
In *ICAPS*.

Xu, Y., Fern, A., and Yoon, S. W. (2007).
Discriminative learning of beam-search heuristics for planning.
In *IJCAI 2007, Proceedings of the 20th IJCAI*, pages 2041–2046.

Yoon, S., Fern, A., and Givan, R. (2006).
Learning heuristic functions from relaxed plans.
In *ICAPS*.

Yoon, S., Fern, A., and Givan, R. (2007).
Using learned policies in heuristic-search planning.
In *Proceedings of the 20th IJCAI*.