

Monitoring Policy Execution

Christian Fritz, Sheila A. McIlraith

Department of Computer Science,
University of Toronto,
Toronto, Ontario, Canada.
{fritz,sheila}@cs.toronto.edu

September 19, 2007

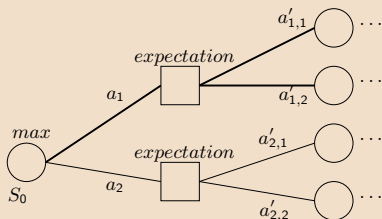
When executing plans in the real-world

- ▶ Things go wrong, because of
 - ▶ incorrect action models, exogenous events, or incomplete planning.
- ▶ Does a discrepancy affect the plan's (near-)optimality?
- ▶ Since on-line, need to answer this quickly, replanning too slow.
 - ▶ E.g. in RoboCup typically 10Hz sensor readings.
- ▶ In main conference: classical planning.
- ▶ Here: decision-theoretic planning.

Markov Decision Process (MDP)

- ▶ An MDP is a tuple $M = (\mathcal{S}, \mathcal{A}, T, R, C)$ where,
 - ▶ \mathcal{S} is a set of states,
 - ▶ \mathcal{A} a set of actions,
 - ▶ $T : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ a transition function s.t. $T(s, a, \cdot)$ a distribution,
 - ▶ $R : \mathcal{S} \rightarrow \mathbb{R}$ a reward function, and
 - ▶ $C : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$ a cost function.
- ▶ We represent the MDP *relationally*, states described by fluents.

Decision-Tree Search



Heuristic value function for leaves.

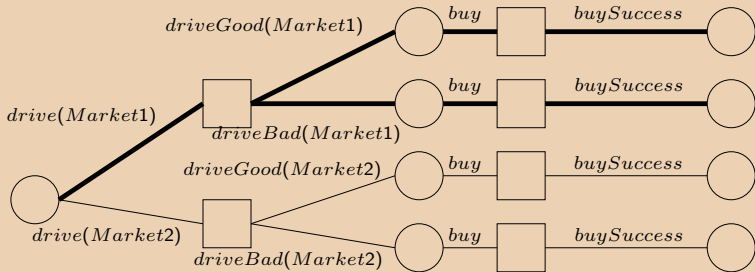
Policy

$$\pi = a_1;$$

if $a'_{1,1}$ **then** $\pi_{1,1}$

elseif $a'_{1,2}$ **then** $\pi_{1,2}$

Overview



Approach

- ▶ Annotate the search tree with relevant information.
- ▶ Patch the search tree upon discrepancies.

Result

- ▶ Can often resurrect a (near-)optimal policy.
- ▶ Order of magnitude speed-ups over replanning from scratch.

Regression:

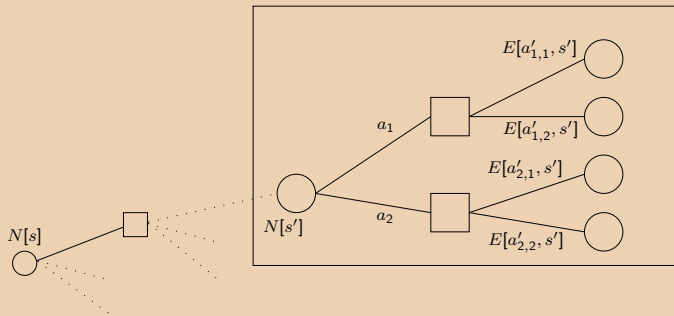
Given a property P and an action A , find a condition Q s.t. P holds after executing A iff Q holds now, assuming A is possible.

Example

- ▶ $P = "c = TotalCost"$
- ▶ $A = "driveTo(New York)"$
- ▶ $Q = "in Princeton \rightarrow c = TotalCost + 10$ and
in Boston $\rightarrow c = TotalCost + 30"$

Can be defined in many languages, e.g. Situation Calculus, STRIPS, ADL.

Monitoring Policy Execution



Off-line: annotate search tree

$$N[s'] = \begin{cases} \text{heuristic value function regressed to } s & \text{if leaf node} \\ \text{reward function regressed to } s & \text{otherwise} \end{cases}$$
$$E[a', s'] = \text{probability and cost of } a' \text{ regressed to } s$$

On-line: upon a discrepancy

- ▶ reevaluate conditions with affected fluents, and
- ▶ backup changed values.

Results

Theoretical Results

Theorem (Space Complexity)

$$|\textit{Annotation}| = O\left(\frac{n - \frac{1}{n}}{n-1}\right) \cdot |\textit{search tree}|$$

Theorem (Finite Horizon MDPs)

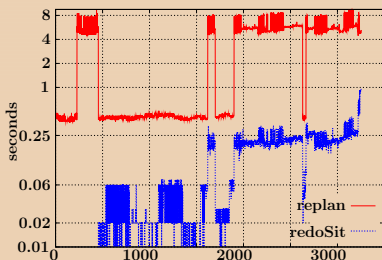
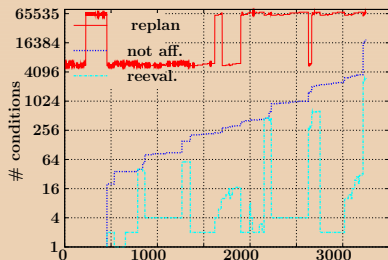
Can verify policy optimality.

Theorem (Sampling in Large and Continuous MDPs)

Can often resurrect near-optimal value function approximation.

Empirical Results

- ▶ We tested on a stochastic variant of the TPP domain:
 1. In each case, we performed decision tree search.
 2. Annotated the search tree.
 3. Perturbed the state systematically.
 4. Ran our algorithm and replanning from scratch.



- ▶ Many discrepancies are completely irrelevant.
- ▶ Others only affect a small subset of relevant fluents.
- ▶ Relevant vs. unique affected conditions: 9751.03 : 1

Conclusions

Our objective was..

- ▶ to resurrect a good value function approximation upon discrepancies,
- ▶ to minimize replanning effort and increase reactivity.

This is a real problem with relevance and significance, e.g., to RoboCup.

We have..

- ▶ extended a technique used for monitoring classical plans (to be presented at main conference) to decision tree search,
- ▶ proved that it is of neglectable space complexity,
- ▶ proved (near-)optimality results for two particular applications.

Empirical results show that..

- ▶ many discrepancies only affect a small part of the relevant conditions,
- ▶ our technique can be used to quickly find the only affected conditions,
- ▶ exploiting this knowledge can lead to computational savings of almost four orders of magnitude.