Incremental Plan Aggregation for Generating Policies in MDPs

May 14th, 2010
AAMAS 2010, Toronto, Canada
Markov Decision Processes

Markov Decision Process: definition

\[ M = \langle S, A, \text{app}, Pr, R \rangle: \]

- \( S \) and \( A \): finite sets of states and actions
- \( \text{app}(s) \): set of all actions applicable in \( s \)
- \( Pr(s, a, s') \): probability of the state transition \( s \xrightarrow{a} s' \) such that \( a \in \text{app}(s) \)
- \( R(s, a, s') \): reward of the state transition \( s \xrightarrow{a} s' \) such that \( a \in \text{app}(s) \)
Markov Decision Process (cont.)

MDP planning problem

\[ \mathcal{P} = (M, s_0, G, \rho) : \]

- \( M = (S, A, \text{app}, \text{Pr}, R) : \) an MDP
- \( s_0 \in S : \) the initial state
- \( G \subseteq S : \) set of goal states
- \( 0 < \rho \leq 1 : \) probability threshold
Markov Decision Process (cont.)

MDP planning problem

\[ \mathcal{P} = (M, s_0, G, \rho): \]

- \( M = (S, A, \text{app}, Pr, R) \): an MDP
- \( s_0 \in S \): the initial state
- \( G \subseteq S \): set of goal states
- \( 0 < \rho \leq 1 \): probability threshold

Solution \( \pi \) to an MDP planning problem

- \( \pi \): partial function \( S_\pi \rightarrow A \) for some set \( S_\pi \subseteq S \)
- \( \Omega(s_0, \pi) \): probability of reaching from \( s_0 \) a state \( s \notin \pi \)
- \( \pi \) is solution of \( \mathcal{P} = (M, s_0, G, \rho) \) iff \( \Omega(s_0, \pi) < \rho \)
Forward heuristic search methods for MDPs

Known initial state $\mathcal{I}$
Forward heuristic search methods for MDPs

Function $S \rightarrow A$ indicating all applicable actions in an already instantiated state
Forward heuristic search methods for MDPs

Expand action $a_6$ of state $\mathcal{I}$
Forward heuristic search methods for MDPs

Expansion function $S \times A \rightarrow \mathcal{P}_r (2^{S \times R})$ indicating all probabilistic successor states and rewards from a state and applying a given action.
Forward heuristic search methods for MDPs

And so on from newly instantiated states...
Forward heuristic search methods for MDPs

And so on from newly instantiated states...
Forward heuristic search methods for MDPs

tip-nodes: non-expanded instantiated states
Forward heuristic search methods for MDPs (cont.)

- Use of a heuristic to choose the best non-expanded instantiated states to expand next
- Heuristic: means to guide the search towards the goals or the highest rewards with cheap computations
- Forward heuristic search algorithms:
  - (L)RTDP [Barto et al. 1995, Bonet & Geffner 2003]
  - LAO* [Hansen & Zilberstein 2001]
  - FPG [Buffet & Aberdeen 2007]
  - ...

- Why not using a deterministic (classical) planner as a heuristic to expand states in the graph?
Overview of RFF

Main idea

- Call a deterministic planner from many probabilistic reachable states
- Aggregate plans into a policy with a bounded probability of reaching its fringe at execution: $\Omega(s_0, \pi) \leq \rho$

Main steps

1. Determinize the MDP planning problem
2. Generate a beam of unconditional plans with a deterministic planner
3. Aggregate all generated plans in a coherent partial policy $\pi$ for the MDP
4. Compute the probability $\Omega(s_0, \pi)$ to reach the fringe of the partial policy $\pi$ starting in $s_0$ and successively applying $\pi$
5. If $\Omega(s_0, \pi) > \rho$, then:
   1. Generate new intermediate goals for the deterministic planner
   2. Goto 2.
RFF: illustration of the algorithm

Initial input: initial state(s) \( I \) + goal state(s) \( G \) + expansion function \( S \times A \rightarrow \mathcal{P}_r(2^{S \times \mathbb{R}}) \)
Generate an initial trajectory plan from \( \mathcal{I} \) to \( \mathcal{G} \) with a deterministic planner (FF) \( \Rightarrow \) initial policy
RFF: illustration of the algorithm

Select and expand tip-nodes by considering all probabilistic effects of each state in the graph $\Rightarrow$ new tip-nodes
Policy reinforcement (optional): shortest stochastic path from $I$ to $G$ on expanded nodes by considering tip-nodes as dead-ends
Estimate the probability of reaching any tip-node with Monte-Carlo sampling: $P_{tn} = \frac{10}{21} \approx 0.476$
If $P_{tn} > \rho$: generate new trajectory plans from reachable tip-nodes with the deterministic planner, and merge them in the policy graph.
Computing the probability of reaching any tip-node

**Fixed-point approximate computation**

Let $\mathcal{T}$ be the current set of tip-nodes in the graph. Probability $P_{TN}$ of reaching any tip-node in $\mathcal{T}$ starting in $\mathcal{I}$ and successively applying $\pi$:

$$P_{TN} = \lim_{t \to +\infty} P_t(\mathcal{T}|\mathcal{I})$$

with

$$\begin{cases} P_0(\mathcal{T}|s) = \delta_{\mathcal{T}}(s) \\ P_t(\mathcal{T}|s) = \sum_{s' \in s.successors} P(s'|s, \pi(s))P_{t-1}(\mathcal{T}|s') \end{cases}$$

**Costly ⇒ statistic approximation with Monte-Carlo sampling**

**Statistic estimation with Monte-Carlo sampling**

$$P_{TN} = \frac{\text{Number of particles reaching a tip-node}}{\text{Number of thrown particles}}$$

$$= \sum_{n: \text{reachable tip-nodes}} \frac{\text{Number of particles arrived in } n}{\text{Number of thrown particles}}$$
### Deterministic relaxations of effects

**How to transform probabilistic effects into deterministic ones?**

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Pros/Cons (+/−)</th>
<th>Illustration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Most Probable Effect</td>
<td>+ few actions, find most probable paths</td>
<td><img src="image1.png" alt="Diagram" /></td>
</tr>
<tr>
<td></td>
<td>− may not find any path</td>
<td><img src="image2.png" alt="Diagram" /></td>
</tr>
<tr>
<td>1 effect = 1 action</td>
<td>+ non-zero probability of reaching $G$ if paths to $G$ exist</td>
<td><img src="image3.png" alt="Diagram" /></td>
</tr>
<tr>
<td></td>
<td>− lot of actions, perhaps unlikely paths</td>
<td><img src="image4.png" alt="Diagram" /></td>
</tr>
</tbody>
</table>

Motivations

Results

Conclusion
## Goal states of the deterministic planner

**How are the MDP goal states and the deterministic relaxed problem’s goals related?**

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Pros/Cons (+/−)</th>
<th>Illustration</th>
</tr>
</thead>
<tbody>
<tr>
<td>FF goals $=$ RFF goals</td>
<td>+ higher probability of reaching RFF goals, easier implementation</td>
<td><img src="image1" alt="Diagram" /></td>
</tr>
<tr>
<td></td>
<td>− explore more states, larger FF computation times</td>
<td></td>
</tr>
<tr>
<td>FF goals $= k$ states with highest value explored by RFF</td>
<td>+ explore less states, smaller FF computation times</td>
<td><img src="image2" alt="Diagram" /></td>
</tr>
<tr>
<td></td>
<td>− lower probability of reaching RFF goals, complex goals generation</td>
<td></td>
</tr>
</tbody>
</table>
Theoretical results

Theorem 1: RFF termination
For every MDP planning problem $\mathcal{P} = (M, s_0, G, \rho)$, RFF terminates in finite time (number of iterations).

Theorem 2: probability of success
Let $\mathcal{P} = (M, s_0, G, \rho)$ be an MDP planning problem. If there are no unsolvable states in $M$, then the probability of success of any solution found by RFF is higher than $1 - \rho$. 
Theoretical results (cont.)

MPO = Most Probable Outcome (effects determinization)
AO = All Outcomes (effects determinization)

**Theorem 3: soundness of RFF\textsubscript{MPO}**
For every MDP planning problem \( \mathcal{P} = (M, s_0, G, \rho) \), every solution that RFF\textsubscript{MPO} finds is correct.

**Theorem 4: soundness of RFF\textsubscript{AO}**
For every MDP planning problem \( \mathcal{P} = (M, s_0, G, \rho) \), every solution that RFF\textsubscript{AO} finds is correct.

**Theorem 5: completeness of RFF\textsubscript{AO}**
For every MDP planning problem \( \mathcal{P} = (M, s_0, G, \rho) \), if a solution exists, it is found by RFF\textsubscript{AO}, otherwise RFF\textsubscript{AO} returns failure.
# Experimental results: RFF won the IPPC 2008

<table>
<thead>
<tr>
<th>Team</th>
<th>Planners</th>
<th>Members</th>
<th>Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>FSP*–(RBH/RDH)</td>
<td>Florent Teichteil</td>
<td>forward heuristic search</td>
</tr>
<tr>
<td></td>
<td>RFF–(BG/PG)</td>
<td>Guillaume Infantes</td>
<td>graph-based, optimal</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(ONERA)</td>
<td>domain determinization</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ugur Kuter</td>
<td>graph-based, plans fusion</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(University of Maryland)</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>LPPFF</td>
<td>Rajesh Kalyanam</td>
<td>divide-and-conquer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Robert Givan</td>
<td>domain determination</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Purdue University)</td>
<td>deterministic subgoals</td>
</tr>
<tr>
<td>6</td>
<td>SEH</td>
<td>Jia-Hong Wu</td>
<td>domain determinization</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Rajesh Kalyanam</td>
<td>stochastic enforced hill-climbing</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Purdue University)</td>
<td>local MDPs to escape basins</td>
</tr>
<tr>
<td>9</td>
<td>HMDPP</td>
<td>Emil Keyder</td>
<td>domain determinization</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Universitat Pompeu Fabra)</td>
<td>self-loop relaxation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Hector Geffner</td>
<td>heuristic pattern database</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(ICREA &amp; UPF)</td>
<td>heuristic lexicographic</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>heuristic choice</td>
</tr>
<tr>
<td>11</td>
<td>FF-Replan</td>
<td>Sungwook Yoon</td>
<td>domain determinization</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Palo Alto Research Center)</td>
<td>plan &amp; repair</td>
</tr>
</tbody>
</table>

*Official results available in the competition booklet*
Varying the probabilistic threshold $\rho$

- Number of calls to RFF increases with $\rho$ (upper bound on the probability to replan)
- Percentage of problems solved: no general impact
- Quality of solutions: no general impact
- Total time (all calls to RFF per problem + simulation): no general impact

Explanation
If the policy fails, we have to recompute a policy (= call to RFF) from the failure state; and the failure probability of policies increases with $\rho$. 
RFF with different method for relaxations of MDPs: Most Probable Outcome (MPO), and All Outcomes (AO)

- **Percentage of problems solved:** MPO > AO
- **Quality of solutions:** MPO > AO
- **Total time (all calls to RFF per problem + simulation):** no general impact

**Explanation**
The action makespan of the deterministic domain is larger with AO than with MPO. Policies generated with MPO are more likely to fail than the ones generated with AO, but MPO solves far more problem instances than AO.
RFF using the goal-selection strategies ProblemGoals (PG), Random Goals (RG), and Best Goals (BG)

- **Percentage of problems solved**: RG is the best for blocksworld and boxworld, no general impact on other domains
- **Quality of solutions**: PG is the best for (ex-)blocksworld, no general impact on other domains
- **Total time**: RG is the best for boxworld and PG is the best for (ex-)blocksworld, no general impact on other domains

**Explanation**
The goal-selection strategy impacts the way states are explored, so that its consequences in terms of solution quality highly depends on the domain.
Conclusions

- **RFF** is a new MDP planner that uses a deterministic subplanner to generate policies that are robust to effect uncertainties. **RFF**:
  - determinizes the given MDP model into a classical planning problem;
  - generates partial policies off-line by producing solution plans to the classical planning problem and incrementally aggregating them into a policy;
  - uses sequential Monte-Carlo (MC) simulations of the partial policies before execution, in order to assess the probability of replanning for a policy.

- **RFF generates policies whose probability of success is below a given threshold**

- the deterministic planner can be viewed as a **heuristic to explore new states** in the graph

- special use-case: $\text{RFF}(\rho = 1) \equiv \text{FF-replan}$ [Yoon et al. 2007]
Future work

- Use different deterministic planners and compare how they affect the aggregated policy
- Use different plan aggregation techniques (between merged policy optimization and action rewrite)
- Use different goal selection strategies
- Use different determinization strategies
- “Aggressive” parallelization of calls to the deterministic planner using multi-core processors
- Is optimality achievable?
- Extensions:
  - Hybrid MDPs (discrete and continuous variables)
  - temporal planning: SMDPs and GSMDPs
  - partial observability: POMDPs
Questions?

The authors thank the anonymous reviewers for their helpful comments on the paper.

This research was supported by the French “Délégation Générale pour l'Armement” grant 07.60.031.00.470.75.01, DARPA’s Transfer Learning and Integrated Learning programs, and NSF grant IIS0412812. The opinions in this paper are those of the authors and do not necessarily reflect those of the funders.