What’s Hot in Planning
Standing on the Shoulders of Classical Planners

Ronen I. Brafman

Department of Computer Science
Ben-Gurion University, Israel

25.7.2012 – AAAI’12
Focus of This Talk

Two Key Topics:

1. Four methods for generating heuristic functions for classical planning (very brief)
2. Using classical planners as black boxes to help solve richer planning models

There’s much more going on!
I assume you’ve taken a basic AI course and remember:

1. What classical planning is about;
2. That classical problems are described via a planning description language;
3. What heuristic search is;
Classical Planning

Diagram showing a series of states (S1 to S13) connected by arrows, representing the planning space and possible transitions between states.
Heuristic Search: the Key Solution Technique
Heuristic search: idea

- init
  - 32
  - 35
- 29
- goal
  - distance estimate
  - distance estimate
  - distance estimate
Four Common Methods for Obtaining Heuristic Estimates

1. **Abstraction**: solve a smaller problem

2. **Delete relaxation**: solve a simpler problem (Joerg Hoffmann’s Talk)

3. **Critical path**: focus on \((k)\) hardest precondition(s) to achieve

4. **Landmarks**: count milestones

   Landmark: a fact that must be satisfied at some point by all plans.
   Example: To attend AAAI in the US, you must have a VISA

   - Compute a set of landmarks \(L\)
   - \(h(s) = \#\) of landmarks in \(L\) not yet achieved on route to \(s\)

   (Richter, Helmert & Westphal 2008; Karpas & Domshlak 2009; Helmert & Domshlak 2009)
Four Common Methods for Obtaining Heuristic Estimates

1. Abstraction: solve a smaller problem

Project problem to a smaller state space; use distance between projected states.
Example: Original problem has 20 blocks; simplified problem ignores all but 5.

2. Delete relaxation: solve a simpler problem (Joerg Hoffmann’s Talk)

3. Critical path: focus on \((k)\) hardest precondition(s) to achieve

4. Landmarks: count milestones

Landmark: a fact that must be satisfied at some point by all plans.
Example: To attend AAAI in the US, you must have a VISA
Four Common Methods for Obtaining Heuristic Estimates

1. Abstraction: solve a smaller problem

2. Delete relaxation: solve a simpler problem (Joerg Hoffmann’s Talk)
   Use solution length for simplified version of the problem that ignores negative effects of actions
   Example: On(a, b) remains true after moving a from on-top of b to the table; or your gas tank remains full after you drive.
   (Bonet & Geffner, 1999; Hoffman & Nebel, 2001; Mirkis & Domshlak 2007; Keyder & Geffner 2008; Keyder, Hoffmann & Haslum 2012)

3. Critical path: focus on (k) hardest precondition(s) to achieve

4. Landmarks: count milestones
   Landmark: a fact that must be satisfied at some point by all plans.
   Example: To attend AAAI in the US, you must have a VISA
Four Common Methods for Obtaining Heuristic Estimates

1. Abstraction: solve a smaller problem

2. Delete relaxation: solve a simpler problem (Joerg Hoffmann’s Talk)

3. Critical path: focus on \((k)\) hardest precondition(s) to achieve

   - Cost estimate of achieving \(p_1 \land \cdots \land p_k = \max_{i=1, \ldots, k} \text{cost estimate}(p_i)\).

   - Cost estimate of achieving \(p = \min_a \text{achieves } p \) \(\text{cost}(a) + \text{cost estimate}(pre(a))\)

   (Haslum & Geffner, 2000; Haslum, Bonet & Geffner, 2005; Coles, Long, Fox & Smith, 2008)

4. Landmarks: count milestones

   Landmark: a fact that must be satisfied at some point by all plans.

   Example: To attend AAAI in the US, you must have a VISA
### Four Common Methods for Obtaining Heuristic Estimates

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Abstraction</td>
<td>Solve a smaller problem</td>
</tr>
<tr>
<td>2. Delete relaxation</td>
<td>Solve a simpler problem (Joerg Hoffmann’s Talk)</td>
</tr>
<tr>
<td>3. Critical path</td>
<td>Focus on (k) hardest precondition(s) to achieve</td>
</tr>
<tr>
<td>4. Landmarks</td>
<td>Count milestones. A fact that must be satisfied at some point by all plans.</td>
</tr>
<tr>
<td></td>
<td>Example: To attend AAAI in the US, you must have a VISA</td>
</tr>
</tbody>
</table>

- **Landmark**: a fact that must be satisfied at some point by all plans.
- **Example**: To attend AAAI in the US, you must have a VISA

- Compute a set of landmarks \(L\)
- \(h(s) = \) \# of landmarks in \(L\) not yet achieved on route to \(s\)

(Richter, Helmert & Westphal 2008; Karpas & Domshlak 2009; Helmert & Domshlak 2009)
Useful Applications
## Beyond Classical Planning

### Richer models people are working on

1. **Temporal Planning** (actions have duration)
2. **Metric Planning** (continuous variables)
3. **Planning with Preferences**
4. **Planning with Resource Constraints**
5. **Net-benefit Planning** (maximize net value of goals achieved)
6. **Generalized Planning** (complex control structures, such as loops)
7. **Multi-agent Planning**
8. **Planning Under Uncertainty:**
   - 1. Conformant Planning
   - 2. Contingent Planning
   - 3. Markov Decision Processes (MDPs)
   - 4. Partially Observable MDPs
   - 5. Conformant Probabilistic Planning (Fully Unobservable POMDPs)
Key Insights:

1. Classical planning offers a wealth of ideas for generating good solutions, fast.
2. We’d really like to exploit them to solve richer models.
3. Importing these ideas to each of the above non-classical formalisms is difficult, and often simply does not work.

Yet:

1. Goal oriented sequencing of actions is a fundamental computational problem at the heart of all planning problems.
2. Classical planners have reached a certain performance level that makes them attractive for addressing this problem.

So...
Let’s Stand on the Shoulders of Classical Planners
Using Classical Planners within Non-Classical Planners

Two Key Techniques:

1. Replanning: the classical problem is a simplification of the original problem
2. Compilation: the classical problem is equivalent to the original problem
Replanning

An online method for solving planning problems with some uncertainty

1. Make some assumptions → get a simpler model
2. Solve simpler model
3. Execute until your observation contradict your assumptions
4. Repeat (Replan)

An established technique:

- Underlies many closed loop controllers
- Used in motion planning under uncertainty (Zelinsky, 1992)
Stochastic Shortest Path (SSP)

Imagine a classical planning problem except:

- Actions have stochastic effects
- We get to observe the state following each action
- Special case of a Markov Decision Process (MDP)
FF-Replan

Replanning in SSP

1. Simplify: determinize the effect of actions to get a classical model
2. Solve
3. Execute plan until you observe an unexpected state =
   = effect was not the one you assumed in your classical model
4. Replan from new state
5. Repeat until you reach the goal
FF-Replan

**Performance**

- Base-line planner for IPC 2004 probabilistic planning track
- Won the first place and got some people quite pissed off...
- Very fast thanks to its underlying classical planner (FF)
FF-Replan

Some flaws:
- Choices are not well informed
- Ignores risks: an effect we ignored may trap as in a dead-end
- Ignores numbers: no evaluation of expected path length
- Clearly sub-optimal

Improvements
By selecting more sophisticated sampling/resampling, these problems can be addressed or mitigated!
- Make sure effects of different instance of an action differ
- Solve multiple determinization; aggregate results; and more...

(Yoon, Fern, Kambhampati & Givan, 2008; Keyder & Geffner, 2008; Kolobov, Mausam & Weld, 2010; Teichteil-Konigsbuch, Kuter & Infantes, 2010)
FF-Replan

Some flaws:
- Choices are not well informed
- Ignores risks: an effect we ignored may trap as in a dead-end
- Ignores numbers: no evaluation of expected path length
- Clearly sub-optimal

Improvements
By selecting more sophisticated sampling/resampling, these problems can be addressed or mitigated!
- Make sure effects of different instance of an action differ
- Solve multiple determinization; aggregate results; and more...
(Yoon, Fern, Kambhampati & Givan, 2008; Keyder & Geffner, 2008; Kolobov, Mausam & Weld, 2010; Teichteil-Konigsbuch, Kuter & Infantes, 2010)
Replanning

- Solving a simplified problem always carries some risk.
- Can we regain completeness? optimality?
Compilation

The compilation method

- Reduce a planning problem into a classical planning problem
- Solve using your favorite classical planner
- Transform solution back

Information is not lost, although problem size may grow
Compilation Example

Net Benefit Planning

- Setting: classical planning with action costs
- Add: a set of soft goals
- Each soft goal has a value
- Task: find a plan that maximizes net benefit, i.e., plan
  - Achieves the goal
  - Maximizes $\sum$\{value of soft goals achieved\} − $\sum$ action costs
- Special case: empty hard goal ($G = true$)
  - Find an action sequence that maximizes net benefit.
Suppose \textit{clean-car} is a soft goal with value 5. 

Add a special action \textit{make-clean-car}

- Preconditions: none
- Add: \textit{clean-car}
- Cost: 5

Change your goal from \( g \) to \( g \land \textit{clean-car} \)

Apply cost optimal planner

- A plan must achieve \textit{clean-car}
- It can use \textit{make-clean-car} with benefit(=5) \(-\) cost(=5) = 0; or,
- It can truly achieve \textit{clean-car} using real actions, if net benefit \( > 0 \)
Compilation

Conformant Planning – Palacios and Geffner (2009) Next talk!

- Probably the most interesting and important example, currently
- Conformant planning: find a classical plan that succeeds from a set of initial states
  - Captures uncertainty about the initial state
  - Simplistic, but useful building block for planning under uncertainty
- The compiled problem includes
  - New variables that keep track of the agent’s knowledge
  - Original actions modified to update these variables correctly
  - New inference actions added; update the knowledge state, not the world state

Other Examples

- Conformant probabilistic planning (Taig & Brafman, 2011)
- Contingent planning (Albore, Palacios & Geffner 2009; Bonet & Geffner 2011; Shani & Brafman, 2011, 2012)
- Domain knowledge and plan constraints (Edelkamp 2006; Bojar, 2008; Geffner, 2009)
Compilation

Conformant Planning – Palacios and Geffner (2009) Next talk!

Other Examples

- Conformant probabilistic planning (Taig & Brafman, 2011)
- Contingent planning (Albore, Palacios & Geffner 2009; Bonet & Geffner 2011; Shani & Brafman, 2011, 2012)
- Domain knowledge and plan constraints (Edelkamp 2006; Baier, Fritz, Bienvenu & McIlraith, 2008)
- Temporally extended goals (Cresswell & Codington, 2004; Edelkamp, 2006; Patrizi, Lipovetzky, De Giacomo & Geffner 2011)
Conclusions

- Domain independent solvers can solve large classical planning problems quickly.
- They rely on domain-independent methods for generating heuristic functions.
- ... and on many other interesting techniques I did not have time to discuss.
- It now makes sense to use such solvers as black boxes within solvers for more complex models.
- Expect to see many more applications of this idea in the future!
Thanks

- Carmel Domshlak
- Hector Geffner
- Malte Helmert
- Joerg Hoffmann
- Gabi Roeger
- Moshe Tennenholtz