

# Research review: three developments in AI planning and search

## 1) Partial order planning

Partial order planning (POP) does not search the space of states, but the space of plans. In the most simple form, a plan is a totally ordered sequence of actions. In the context of the air cargo problems from our project, a plan to bring two cargo items c1 and c2 from SFO to JFK might look like this:

```
Load(c1, plane, SFO); Load(c2, plane, SFO);  
Fly(plane, SFO, JFK);  
Unload(c1, plane, JFK); Unload(c2, plane, JFK)
```

In this plan, some steps are interchangeable (it does not matter which cargo item is loaded first), and others are not (flying an empty plane will not bring the cargo anywhere). POP takes account of this by replacing the total ordering of the sequence by a partial ordering, e.g.

```
Load(c1, plane, SFO) < Fly(plane, SFO, JFK) and Load(c2, plane, SFO) < Fly(plane, SFO, JFK)
```

POP uses a concept called *causal link* (introduced in Tate (1977)) to remember which action was added to a plan in order to reach which subgoal. When a new action is added to a plan, we can check whether this action *threatens* a causal link. If this is the case, the partial ordering can be extended by specifying that the new action has to be executed before or after some other actions.

POP was the predominant planning approach from the mid-seventies until the mid-nineties of the last century. Then faster state-space search algorithms (see below) were developed, and POP went out of fashion. It was revived by Nguyen and Kambhampati (2001) who transferred the improvements in heuristics to POP.

## 2) State-space planning with heuristics

Forward search in the space of states gained a lot of popularity when good domain-independent heuristics were developed. The first such system was HSP (Bonet et al. 1997). The heuristic used by HSP estimates the distance from a state to the goal by considering a *relaxed problem*: the delete lists of actions are ignored. The heuristic assumes that the elements of the goal state are reached independently, i.e. that the distance to the goal state equals the sum of the distances to each element in the goal state.

Hoffmann's fast-forward planning system (Hoffmann 2001) is based on this approach, but makes the following improvements:

- A heuristic based on the Graphplan algorithm is used. This heuristic takes positive interactions into account and thus provides smaller, more accurate estimates.
- Search is faster because fewer states are considered. From a given state, the algorithm moves to a successor state with a better heuristic value. If none of the successors has a better heuristic value, all successors' successors are considered, etc. - until a state with a better heuristic is found. Hoffmann calls this *enforced hill climbing*.
- A concept of *helpful actions* is used. Actions that seem not helpful are not considered. If this does not lead to a solution, all actions are considered. This finds solutions faster in most cases.

### 3) Real world applications: acting instead of planning

Classical planning assumes a fixed set of states and deterministic actions that have preconditions and effects. Planning is about finding a sequence of actions to reach a specific goal state. For many real world applications, this approach is too theoretical. Ghallab et al. (2014) suggest to focus research more on *acting* instead of planning. Acting is the purpose of planning. Although planning is one important requirement for acting, it is not sufficient. The authors use two examples to illustrate their point: the management of a large harbor, and a robot whose purpose is to move objects in an open environment. For automated systems like these the following points are important:

- Hierarchies of components: one component receives tasks from the components above it. To execute this task the component can create or update plans, receive and analyze sensor data, or delegate subtasks to components below it.
- Continual planning: an actor in the real world needs to interact with its environment and update plans continually. For some events it is not possible or feasible to plan ahead of time. Instead the actor should adjust its plan.
- Uncertainty: in the real world an actor's environment is dynamic. Unexpected events may happen, or actions may not have the intended effect.

A system that takes these aspects into account needs to be more complex than the systems used in classical planning: It has to have a better understanding of its environment.

## References

- Bonet, B., Loerincs G., and Geffner, H.** (1997). *A robust and fast action selection mechanism for planning*. Proc. AAAI-97, 714-719.
- Ghallab, M., Nau, D., and Traverso, P.** (2014). *The actor's view of automated planning and acting: A position paper*. Artificial Intelligence 208.
- Hoffmann, J.** (2001). FF: *The fast-forward planning system*. AI Magazine 22(3), 57-62.
- Nguyen, X. and Kambhampati, S.** (2001). *Reviving partial order planning*. IJCAI-01, 459-464.
- Russell, S. J. and Norvig, P.** (2011). *Artificial Intelligence - A Modern Approach* (Third Edition), Chapters 10 & 11. Pearson.
- Tate, A.** (1977). *Generating project networks*. Proc. 5th Int. Joint Conf. on AI, 888-893.
- Weld, D. S.** (1994). *An introduction to least commitment planning*. AI Magazine 15(4), 27-61.