Planning as Heuristic Search
(Invited Lecture)

Héctor Geffner
ICREA and Universitat Pompeu Fabra
Barcelona, Spain
Most basic approach to planning: reduces the search for plans to a search for a path in a directed graph.

In progression or forward search, the nodes in the graph are sets of atoms representing states in \( S(P) \), and

- root node is \( I \) (the initial state)
- goal nodes are the sets of atoms \( \sigma \) such that \( G \subseteq \sigma \) (goal states)
- edges \( s \rightarrow s' \) iff \( s' = s - \text{Del}(a) + \text{Add}(a) \) and \( \text{Pre}(a) \subseteq s \)
- edge costs \( c(s, s') = c(a, s) \)

This graph also called the forward or progression search space is the simplest search space (isomorphic to \( S'(P) \)) but not the only one.

Other search spaces also used: regression space, plan space, etc.
In regression or backwards search, the nodes in the graph are still sets of atoms but they represent sets of states in $S(P)$, and

- **root node** $\sigma_0$ is the goal $G$
- **goal nodes** are the sets of atoms $\sigma$ such that $\sigma \subseteq I$
- **edge** $\sigma \rightarrow \sigma'$ iff $\sigma' = \sigma - \text{Add}(a) + \text{Pre}(a)$, provided $\text{Add}(a) \cap \sigma \neq \emptyset$, and $\text{Del}(a) \cap \sigma' = \emptyset$
- **edge costs** $c(\sigma, \sigma') = 1$

**Correctness/Completeness:** The paths connecting the initial node to a goal node in this graph (regression space) encode the plans for $P$ that map the initial state into a goal state (in reverse)
Computation: How to search for paths in these graphs?

- **Blind search/Brute force algorithms**
  - Goal plays **passive** role in the search
    - e.g., *Depth First Search (DFS)*, *Breadth-first search (BrFS)*, *Uniform Cost (Dijkstra)*, *Iterative Deepening (ID)*

- **Informed/Heuristic Search Algorithms**
  - Search uses a function $h(s)$ that estimates ‘distance’ (cost) from state $s$ to $S_G$ to guide search
    - e.g., *A*\(^*\), *IDA*\(^*\), *Hill Climbing*, *Best First Search (BFS)*, *Branch & Bound*

The latter can be much more effective: the question is **how to get informed heuristic functions $h(s)$ quickly and automatically** . . .
Heuristics for Classical Planning: the Delete Relaxation

- **Heuristics functions** derived as optimal cost function of relaxed problems (Pearl 83)

- A common relaxation \( P^+ \) obtained from Strips P by dropping the delete-lists

- If \( c^*(P) \) is the optimal cost of \( P \), then heuristic \( h^+(P) \) defined as

\[
h^+(P) \overset{\text{def}}{=} c^*(P^+)
\]

- Heuristic \( h^+ \) intractable but easy to approximate
  - computing optimal plan for \( P^+ \) is intractable
  - computing a non-optimal plan for \( P^+ \) (relaxed plan) easy (FF)

- Approximations of \( h^+ \) yield heuristics as used in modern planners HSP, FF, . . .
Additive Heuristic in HSP

• For all atoms $p$:

$$g(p; s) \overset{\text{def}}{=} \begin{cases} 0 & \text{if } p \in s, \text{ else} \\ \min_{a \in O(p)} [1 + g(Prec(a); s)] & \end{cases}$$

• For sets of atoms $C$, assume also independence:

$$g(C; s) \overset{\text{def}}{=} \sum_{r \in C} g(r; s)$$

• Resulting heuristic function $h_{\text{add}}(s)$:

$$h_{\text{add}}(s) \overset{\text{def}}{=} g(\text{Goals}; s)$$

Heuristic not admissible but informative and fast
Max Heuristic

- For all atoms $p$:

$$g(p; s) \stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } p \in s, \text{ else} \\ \min_{a \in O(p)} [1 + g(Precedes(a); s)] & \end{cases}$$

- For sets of atoms $C$, replace sum by max

$$g(C; s) \stackrel{\text{def}}{=} \max_{r \in C} g(r; s)$$

- Resulting heuristic function $h_{\text{max}}(s)$:

$$h_{\text{max}}(s) \stackrel{\text{def}}{=} g(\text{Goals}; s)$$

Heuristic admissible but not very informative . . .
Max Heuristic, Planning Graphs, and FF

- Build reachability graph $P_0$, $A_0$, $P_1$, $A_1$, …
  
  $P_0 = \{p \in s\}$
  
  $A_i = \{a \in O \mid \text{Prec}(a) \subseteq P_i\}$
  
  $P_{i+1} = P_i \cup \{p \in \text{Add}(a) \mid a \in A_i\}$

- **Theorem:** $h_{\text{max}}(s) = \min i$ such that $G \subseteq P_i$

- More informed but inadmissible $h$ if recursively collect actions that add the goals in graph and their preconditions (not in $s$) and count them (FF)

- Collected actions $\pi$ form a plan for relaxation $P^+$; called **relaxed plan**, so $h_{\text{FF}}(s) = |\pi|$
Example

• Initial Situation $I = \{q\} = s_0$

• Actions:
  - $a: Pre(a) = \{q\} ; \ Add(a) = \{p\} ; \ Del(a) = \{q\}$
  - $b: Pre(a) = \{q\} ; \ Add(a) = \{r\} ; \ Del(a) = \{\}$
  - $c: Pre(a) = \{r\} ; \ Add(a) = \{q\} ; \ Del(a) = \{r\}$

• Goal: $G = \{p, q\}$

**Exercise:** determine optimal $h^*(s_0), h_{max}(s_0), h_{FF}(s_0), h_{add}(s_0)$
A more informed but admissible $h$: Graphplan's $h_G$

- More informed but admissible $h_G$ computed in Graphplan by keeping track of mutex pairs; pairs that cannot be simultaneously achieved in $i$ steps:
  - action pair mutex at $i$ if actions interfere or preconds mutex at $i$
  - atom pair mutex at $i+1$ if supporting action pairs mutex at $i$

- A set of atoms $S$ is mutex at $i$ if it contains a mutex pair at $i$

- Graphplan also adds ‘dummy actions’ NO-OP($p$) for each $p$ with $Prec = Add = \{p\}$ that ‘carry’ $p$ from layer to layer

- Result: build layered graph as below while identifying mutexes as above

\[
\begin{align*}
P_0 &= \{p \in s\} \\
A_i &= \{a \in O \mid Prec(a) \subseteq P_i \text{ and not mutex at } i\} \\
P_{i+1} &= \{p \in Add(a) \mid a \in A_i\}
\end{align*}
\]

Define: $h_G(s) \overset{\text{def}}{=} \min i \text{ s.t. } G \subseteq P_i \text{ and } G \text{ not mutex at } i$
Some Heuristic Search Planners

- **Graphplan, 1995:** makes plan graph heuristic $h_G$ more informed by keeping track of **mutex** relations among pairs of atoms and actions in layered graph. It uses this heuristic in a IDA* **regression search** from the goal.

- **HSP, 1998:** it does a **WA**\* **progression search** guided by **additive heuristic** computed from scratch for every visited state $s$.

- **FF: 2000** it does a hill-climbing **progression search** guided by heuristic given by the number of actions in ’relaxed plan’ (set of actions responsible for presence of goals in lowest layer).


