FF: The Fast-Forward Planning System
by Jorg Hoffmann

Presented by Fairoz Nower Khan

Background of FF

- Most successful planner in Artificial Intelligence Planning and Scheduling (AIPS'00) planning systems competition
- Advanced successor of HSP
- Like HSP, FF relies on forward search through the state space

Differences with HSP

- Heuristics: Better heuristic evaluation considering positive interactions between facts
- Enforced hill climbing: local search strategy using systematic search to escape plateaus and local minima
- Pruning: identifies successors of a search node that may be most helpful in reaching the goal

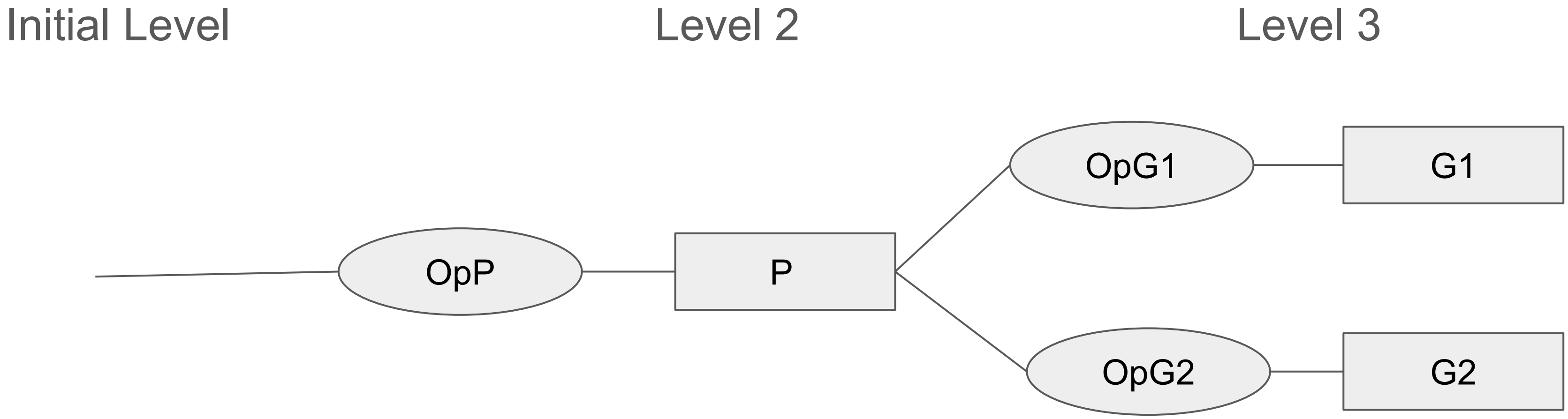
Relaxed Graphplan for Heuristic

- Expanding to larger states
- But without adding negative effects
- “Ignoring the delete lists”
- Actions can make things True but cannot make things False anymore

Relaxed Graphplan in FF

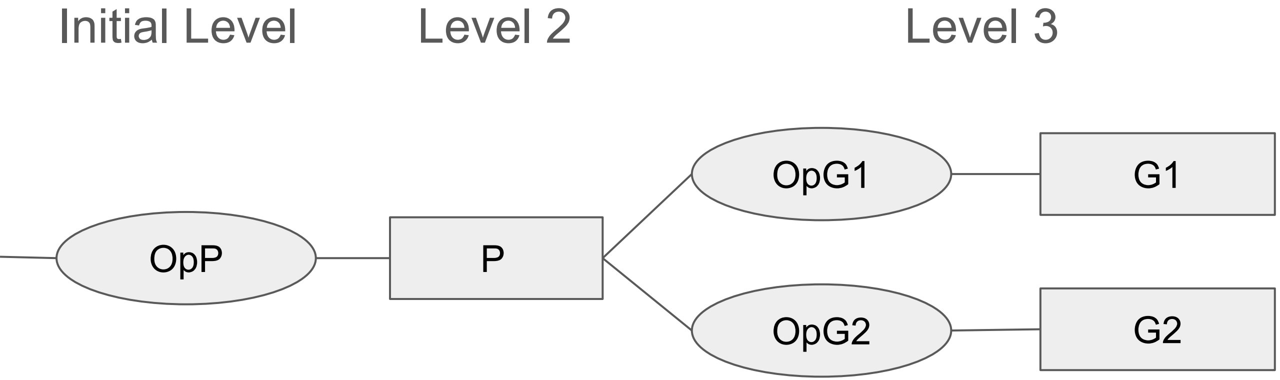
- FastForward uses a special version of Graphplan to compute a heuristic value
- Domain given to Graphplan contains no negative effects
- Thus, no mutual exclusion among actions or literals
- Thus, no need for backtracking
- Length of Graphplan’s solution to relaxed problem from a node is heuristic value for that node

How FF uses plan graphs in its heuristic



Initial State: Empty
Goals: {G1, G2}
Actions:
opG1: Preconditions P ⇒ Add G1
opG2: Preconditions P ⇒ Add G2
opP: No preconditions ⇒ Add P

How FF uses plan graphs in its heuristic



HSP Estimate
Cost of G1 = 2
Cost of G2 = 2
Cost of Goal = 2+2 = 4

Initial State: Empty
Goals: {G1, G2}
Actions:
opG1: Preconditions P ⇒ Add G1 *Shared*
opG2: Preconditions P ⇒ Add G2 *preconditions*
opP: No preconditions ⇒ Add P

Relaxed Graphplan Estimate
Selects opP only once, resulting in a plan containing only three actions

Search Strategy

- A hill-climbing algorithm is a greedy search strategy that moves to the neighbor with better heuristic
- Often finds a good local maxima, but not the optimal solution

Hill-Climbing in State Space

- HSP uses standard hill-climbing and a heuristic based on the relaxations as previously discussed
- When no successor is a better state, and goals are not met, make arbitrary choice
- No backtracking, so a bad choice can make the problem unsolvable

Enforced Hill-Climbing

- FF uses a slightly modified hill-climbing algorithm
- Instead of choosing the best successor, perform breadth-first search for the first strictly better descendent
- Less likely to randomly wander around plateaus

Helpful Actions

- For a state S, the set H(S) of helpful actions is defined as

  $H(S) := \{o | pre(o) \subseteq S, add(o) \cap G1 \nvDash \varnothing\}$

- H(S) is the set of helpful actions for a given search state S. These helpful actions are those whose preconditions are satisfied by the current state S, and their effects include at least one goal from the set G1.
- Restrict any state's successors to those generated by the first action set in its relaxed solution.

Helpful Actions

There are two rooms, A and B, and two balls,

which will be moved from room A to room B, using a robot. Say the robot is in room A and has picked up both balls. The relaxed solution that the heuristic extracts is

< {moveAB},
{ drop ball1 B left,

drop ball2 B right } >

Performance Evaluation

- Eight experiments were conducted by turning the three features of FF on or off.
- FF’s estimates improve run-time performance in about half of the domains across all switch alignments
- With enforced hill climbing in the background, FF’s estimates have clear advantages in terms of solution length
- Helpful actions strategy performs better in domains where a significant number of actions can be cut. Solutions are shorter.

Thank You!