Two Extensions

Extension 1: State Reuse

- Observation 1: The first three actions in Weak Plan 2 yield the same state (call it s) as the first four actions in Weak Plan 1.

  ![Diagram of Weak Plan 1]

  Weak Plan 1:
  1. PICK-UP b_1, PUT-ON-BLOCK b_1, PICK-UP-FROM-TABLE b_2, PUT-ON-BLOCK b_2
  2. PICK-UP b_3, PICK-UP-FROM-TABLE b_4, PUT-ON-BLOCK b_5

- Observation 2: Since s is already solved in Weak Plan 1, there is no need to try to find a path from s to g in Weak Plan 2.

  Given these observations, state reuse aims to improve efficiency by stopping the search as soon as a solved state is reached.

Extension 2: Goal Alternative

- Observation: To handle a failed effect (e.g., b_1 falling onto the table for action PICK-UP b_2), instead of establishing a path to the ultimate goal g (as in the Basic algorithm), we can try to establish a path to an intended effect of PICK-UP b_2, i.e., holding b_2.

  ![Diagram of Weak Plan 2]

  Weak Plan 2:
  1. PICK-UP b_1, PUT-ON-BLOCK b_1, PICK-UP-FROM-TABLE b_2, PUT-ON-BLOCK b_2
  2. PICK-UP b_3, PICK-UP-FROM-TABLE b_4, PUT-ON-BLOCK b_5

- The plan contains a single action, PICK-UP-FROM-TABLE b_2.
- Planning efficiency is improved and plan size is reduced!
- However, if a path to the intended effect cannot be found, we can then try to establish a path to the original goal g.

  This is the goal alternative heuristic: it aims to improve planning efficiency and reduce plan size by searching for an alternative, presumably closer goal, the intended effect of an action, and backing off to the original goal if needed.

Evaluation

- Goal: Evaluate FIP, which implements the Basic algorithm together with our two extensions, on problem instances from 4 domains in the IPC2008 FOND track.
- Blocksworld, faults, first-responders, forest
- Compared against two-state-of-the-art planners: Gamer & MBP

Results and Discussion

- FIP has a better problem coverage than Gamer & MBP [Table 1].
- Gamer & MBP cannot solve more than 10 problems in Blocksworld.
- FIP can solve all problems efficiently (cutoff time 1,200 seconds).
- FIP outperforms other planners w.r.t. CPU time t (expressed in seconds) and solution size s (expressed in the number of states in the solution policy) [Table 2].

---

Problem

- Find strong cyclic solutions to Fully-Observable Non-deterministic (FOND) planning problems

Related Concepts

- In nondeterministic planning
  - an action may generate multiple effects
- In fully-observable planning
  - the states of the world are fully observable

- More challenging than finding weak plans
  - Weak plans: only need to establish one path from the initial state to the goal state
  - Strong cyclic plans: need to establish one path from each state reachable from the initial state to the goal state

Example: Given initial state s_0 and goal g.
- The green path is a weak plan, since it is one path from s_0 to g.
- In strong cyclic planning, we also need to find a path from each red state to g.

Basic Strong Cyclic Algorithm

- 3 steps
  1. Generate a weak plan from s_0 to g.
  2. For each failed effect e, recursively find a weak plan from e to g.
  3. If a dead end is met (i.e., no path leads to g from it), then backtrack (i.e., disable the action that leads to the dead end and try another path)

Example: Blocksworld

- Initial state (s_0)

  ![Diagram of Blocksworld]

  - To generate a strong cyclic plan:
    - Step 1: Find a weak plan from s_0 to g
    - Weak Plan 1:
      1. PICK-UP b_1, PUT-ON-BLOCK b_1, PICK-UP-FROM-TABLE b_2, PUT-ON-BLOCK b_2
      2. PICK-UP b_3, PICK-UP-FROM-TABLE b_4, PUT-ON-BLOCK b_5

- Step 2: Since action PICK-UP b_2 b_4 may generate the failed effect of dropping b_2 onto the table, we generate a weak plan from this failed effect to g
  - Weak Plan 2:
    1. PICK-UP b_1, PUT-ON-BLOCK b_1, PICK-UP-FROM-TABLE b_2, PUT-ON-BLOCK b_2
    2. PICK-UP b_3, PICK-UP-FROM-TABLE b_4, PUT-ON-BLOCK b_5

- Step 3: Since no dead-ends are found, no backtrack is needed

This Basic algorithm is inefficient

- Certain states are repeatedly explored: the last two actions of Weak Plan 1 and Weak Plan 2 are identical.

Goal: Improve the efficiency of the Basic algorithm by proposing two extensions