UDACITY

< Back to Artificial Intelligence Nanodegree

Build a Forward Planning Agent

REVIEW

CODE REVIEW 4

HISTORY

Meets Specifications

Commendable Learning Attitude

Planning Graph Implementation

(AUTOGRADED) Student code passes all Project Assistant test cases for:

- `ActionLayer` mutual exclusion rules:
  - `_inconsistent_effects()`
  - `_interference()`
  - `_competing_needs()`
- `LiteralLayer` mutual exclusion rules:
  - `_inconsistent_support()`
  - `_negation()`

Correct! (Note: this rubric item was graded automatically.)

Heuristic Implementation

(AUTOGRADED) Student code passes all Project Assistant test cases for:
Correctly implemented

- `PlanningGraph` class heuristics:
  - `h_levelsum()`
  - `h_maxlevel()`
  - `h_setlevel()`

Correct! (Note: this rubric item was graded automatically.)

Experimental Results & Report

Report includes a table or chart to analyze the number of nodes expanded against number of actions in the domain.

- **The chart or table includes data for all search & heuristic combinations for air cargo problems 1 and 2**
- **The chart or table includes data at least one uninformed search, two heuristics with greedy best first search, and two heuristics with A* on air cargo problems 3 and 4**
- **Report includes at least a one paragraph discussion of these results that analyzes the growth trends as the problem size increases**

The report is fantastic. It has good tables and charts which analyze all the nodes expanded. The discussions too are really commendable.

Extra Tips

It is good to note that the state space for planning problem domains increases exponentially with the number of actions, so the number of nodes expanded will span several orders of magnitude and the run time increases tremendously for more complex problems. I guess you figured this out already. 😏

Report includes a table or chart to analyze the search time against the number of actions in the domain.

- **The chart or table includes data for all search & heuristic combinations for air cargo problems 1 and 2**
- **The chart or table includes data at least one uninformed search, two heuristics with greedy best first search, and two heuristics with A* on air cargo problems 3 and 4**
- **Report includes at least a one paragraph discussion of these results that analyzes the growth trends as the problem size increases**

Well done. The search time was analysed perfectly.

Report includes a table or chart to analyze the length of the plans returned by each algorithm on all search problems.

- **The chart or table includes data for all search & heuristic combinations for air cargo problems 1 and 2**
- **The chart or table includes data at least one uninformed search, two heuristics with greedy best first search, and two heuristics with A* on air cargo problems 3 and 4**

The tables and charts are really impressive.

Extra tips

Check the below links for extra guidiance on this point.

- Graph Search Algorithms
- Planning based on Graph
- Planning-Graph Techniques

Submission includes a short answer to each of the following questions. (A short answer should be at least 1-2 sentences at most a small paragraph.)

- **Which algorithm or algorithms would be most appropriate for planning in a very restricted domain (i.e., one that has only a few actions) and needs to operate in real time?**
- **Which algorithm or algorithms would be most appropriate for planning in very large domains (e.g., planning delivery routes for all UPS drivers in the U.S. on a given day)**
- **Which algorithm or algorithms would be most appropriate for planning problems where it is important to find only optimal plans?**

Very good answers to the questions. Keep up this creativity.

Extra Tips

- For planning in a very restricted and small problem domains, it is reasonable to use uninformed search (particularly UCS) if the plan needs to be optimal, or an informed algorithm with a fast (and therefore probably inadmissible) heuristic like greedy search with ignore preconditions (as long as the plan doesn't need to be optimal).
- In large domain planning, compromising on the search algorithm being optimal is acceptable. An example will be to use GBFS instead of A*. Also, the heuristic can be less admissible , that is choosing something like level sum over set level.
- Greedy search with Ignore Preconditions has the best scaling properties when planning in very large domains, however greedy searches with one of the planning graph heuristics may also be appropriate (the number of node expansions grows slowly with increasing problem complexity)

DOWNLOAD PROJECT

4 CODE REVIEW COMMENTS ❯

RETURN TO PATH