UDACITY

‹ Return to Classroom

# Build a Forward Planning Agent

REVIEW

CODE REVIEW

HISTORY

## Meets Specifications

Kudos,

The project and report show diligence and effort and address all the project rubrics very well. Congratulations on completing this challenging project and wish you good luck in the Udacity AI courses ahead.
Here are some links to very good materials that I believe will help increase the student's knowledge on Planning Search.

- This article gives an insight into what is are the Current Trends in Automated Planning
- Here another article will enhance your understanding of State Space Planning.

## 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

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

The report presents results obtained using simple and easy-to-read tables. The discussions are very accurate and straight to the point. Well done

Report includes a table or chart to analyze the length of the plans returned by each algorithm on all search problems 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 show good results obtained and discussions have been included in the report which gives an analysis of the growth trend as the problem size increases. Good work

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?

1. Since the greedy search algorithm gives the shortest times among them as well as the least number of nodes expanded and hence uses less memory, it should be a good choice in this scenario.
2. A good choice too, is the greedy search algorithm, even if it doesn't give the most optimal solution, the solution isn't too far off optimal, and it runs quicker and expands less nodes as problem size increases as compared to the others.
3. From the experiments, we can see that A star, breadth-first and uniform cost search all return the optimal plan, even as the problem size increases. If optimality is the only consideration, all of them are appropriate choices.

From the results of the set of experiments, we can see that informed search algorithms with the right heuristics can perform much better than uninformed search algorithms in terms of both speed and memory requirements, and still return optimal plans. from the experiments, the unmet goals heuristic is the most optimal for time, and the level sum heuristic is the most optimal for memory. Depending on the scenario complexity and computation requirements, we can tune the heuristic to suit our need.

DOWNLOAD PROJECT

RETURN TO PATH