Artificial Intelligence Nanodegree

Project: Build Forward Planning Agent

NGUYEN Duc Huy

1. Introduction

Planning is an important topic in AI because intelligent agents are expected to automatically plan their own actions in uncertain domains. Planning and scheduling systems are commonly used in automation and logistics operations, robotics and self-driving cars, and for aerospace applications like the Hubble telescope and NASA Mars rovers.

This project is split between implementation and analysis. First I will combine symbolic logic and classical search to implement an agent that performs progression search to solve planning problems. Then I will experiment with different search algorithms and heuristics, and use the results to answer questions about designing planning systems.

2. Problem definition

In this project, we consider four air cargo problems. The cargo problem instances have different numbers of airplanes, cargo items, and airports that increase the complexity of the domains.

```
Action(Fly(p, from,to),
        PRECOND: At(p, from) ∧ Plane(p) ∧ Airport(from) ∧ Airport(to)
        EFFECT: ¬At(p, from) ∧ At(p,to))
Action(Load(c, p, a),
        PRECOND: At(c, a) ∧ At(p, a) ∧ Cargo(c) ∧ Plane(p) ∧ Airport(a)
        EFFECT: ¬ At(c, a) ∧ In(c, p))
Action(Unload(c, p, a),
        PRECOND: In(c, p) ∧ At(p, a) ∧ Cargo(c) ∧ Plane(p) ∧ Airport(a)
        EFFECT: At(c, a) ∧ ¬ In(c, p))
```

We will run the search algorithms on these problems and analyze the results based on the following metrics:

- number of actions in the domain
- number of new node expansions
- time to complete the plan search

3. Results

The results are given as follows:

Problem 1: The optimal plan is of length of 6

Algorithm	Action	Expansion	Goal Tests	New nodes	Plan length	Runtime (s)
breadth_first_search	20	43	56	178	6	0.005
depth_first_graph_search	20	21	22	84	20	0.003
uniform_cost_search	20	60	62	240	6	0.008
greedy_best_first_graph_search h_unmet_goals	20	7	9	29	6	0.001
greedy_best_first_graph_search h_pg_levelsum	20	6	8	28	6	0.214
greedy_best_first_graph_search h_pg_maxlevel	20	6	8	24	6	0.167
greedy_best_first_graph_search h_pg_setlevel	20	6	8	28	6	0.306
astar_search h_unmet_goals	20	50	52	206	6	0.008
astar_search h_pg_levelsum	20	28	30	122	6	0.526
astar_search h_pg_maxlevel	20	43	45	180	6	0.547
astar_search h_pg_setleve	20	33	35	138	6	0.731

Problem 2: The optimal plan = 9

Algorithm	Action	Expansion	Goal Tests	New nodes	Plan length	Runtime (s)
breadth_first_search	72	3343	4609	30503	9	1.913
depth_first_graph_search	72	624	625	5602	619	2.772
uniform_cost_search	72	5154	5156	46618	9	3.192
greedy_best_first_graph_search h_unmet_goals	72	17	19	170	9	0.018
greedy_best_first_graph_search h_pg_levelsum	72	9	11	86	9	4.438
greedy_best_first_graph_search h_pg_maxlevel	72	27	29	249	9	9.001
greedy_best_first_graph_search h_pg_setlevel	72	9	11	84	9	7.242
astar_search h_unmet_goals	72	2467	2469	22522	9	2.218
astar_search h_pg_levelsum	72	357	359	3426	9	118.395
astar_search h_pg_maxlevel	72	2887	2889	26594	9	618.217
astar_search h_pg_setleve	72	1037	1039	9605	9	677.425

Problem 3: The optimal plan = 12

Algorithm	Action	Expansion	Goal Tests	New nodes	Plan length	Runtime (s)
breadth_first_search	88	14663	18098	129625	12	9.873
depth_first_graph_search	88	408	409	3364	392	1.082
uniform_cost_search	88	18510	18512	161936	12	14.486
greedy_best_first_graph_search h_unmet_goals	88	25	27	230	15	0.034
greedy_best_first_graph_search h_pg_levelsum	88	14	16	126	14	10.523
greedy_best_first_graph_search h_pg_maxlevel	88	21	23	195	13	12.512
greedy_best_first_graph_search h_pg_setlevel	88	88	35	37	345	42.279
astar_search h_unmet_goals	88	7388	7390	65711	12	8.568
astar_search h_pg_levelsum	88	369	371	3403	12	190.817

- **Note**: astar_search h_pg_maxlevel and astar_search h_pg_setlevel take too much time to solve the problem. So, we don't report the results of these algorithms.

Problem 4: The optimal plan = 14

Algorithm	Action	Expansion	Goal Tests	New nodes	Plan length	Runtime (s)
breadth_first_search	104	99736	114953	944130	14	90.1771
uniform_cost_search	104	113339	113341	1066413	14	112.231
greedy_best_first_graph_search h_unmet_goals	104	29	31	280	18	0.056
greedy_best_first_graph_search h_pg_levelsum	104	17	19	165	17	18.970
greedy_best_first_graph_search h_pg_maxlevel	104	56	58	580	17	45.867
greedy_best_first_graph_search h_pg_setlevel	104	107	109	1164	23	192.025
astar_search h_unmet_goals	104	34330	34332	328509	14	53.512

- **Note**: depth_first_graph_search , astar_search h_pg_levelsum, astar_search h_pg_maxlevel and astar_search h_pg_setlevel take too much time to solve the problem. So, we don't report the results of these algorithms.

4. Discussion

a. 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?

From table 3-4, the algorithm breadth_first_search and greedy_best_first_graph_search h_unmet_goals are the most appropriate for planning in a very restricted domain in real time because they can solve the problem quickly.

b. 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)

For very large domains, I think greedy_best_first_graph_search h_unmet_goals is the most appropriate (the problem 4 could be considered as a larger domain)

c. Which algorithm or algorithms would be most appropriate for planning problems where it is important to find only optimal plans?

From the tables 1-4, we can observe that breadth_first_search outperforms other algorithms. So, it would be the most appropriate for planning problem.