

Project 2: Classical Planning

Miguel Tasende

May 26, 2018

1 Introduction

2 Problems 1 and 2

The first two problems were solved using each of the search algorithms. The results can be seen on the table below.

problem		1				2			
index	search algorithm	actions	expansions	time_s	path_length	actions	expansions	time_s	path_length
1	breadth_first_search	20	43	0.021	6	72	3343	0.383	9
2	depth_first_graph_search	20	21	0.007	20	72	624	0.551	619
3	uniform_cost_search	20	60	0.022	6	72	5154	0.628	9
4	greedy_best_first_graph_search with h_unmet_goals	20	7	0.002	6	72	17	0.013	9
5	greedy_best_first_graph_search with h_pg_levelsum	20	6	0.57	6	72	9	2.401	9
6	greedy_best_first_graph_search with h_pg_maxlevel	20	6	0.18	6	72	27	2.976	9
7	greedy_best_first_graph_search with h_pg_setlevel	20	6	0.615	6	72	27	9.611	9
8	astar_search with h_unmet_goals	20	50	0.018	6	72	2467	0.776	9
9	astar_search with h_pg_levelsum	20	28	0.388	6	72	357	49.888	9
10	astar_search with h_pg_maxlevel	20	43	0.308	6	72	2887	267.257	9
11	astar_search with h_pg_setlevel	20	42	0.776	6	72	1987	680.743	9

Figure 1: Results for the first two problems, with all the search algorithms

It was noticed that the Depth First algorithm was resulting in extremely long paths, and considering this is an Air Cargo problem, it is reasonable to expect that optimum or near-optimum solutions are searched for. Therefore, the Depth First algorithm won't be considered in the rest of the project.

A* with "maxlevel" and "setlevel" heuristics are taking a lot of time to find a solution, but they will be considered for the other problems, to have more data (it is expected that running all the remaining combinations may take about 10 hours of processing with pypy3).

3 Problems 3 and 4

Problems 3 and 4 were solved with all the algorithms except Depth First. The results are below.

4 Aggregated Results

Some tables and charts are shown below, aggregating the results for all problems, and all search algorithms in different ways.

5 Results Analysis

- 5.1 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?
- 5.2 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)
- 5.3 Which algorithm or algorithms would be most appropriate for planning problems where it is important to find only optimal plans?

problem		1				2			
	search_algorithm	actions	expansions	time_s	path_length	actions	expansions	time_s	path_length
index									
1	breadth_first_search	20	43	0.021	6	72	3343	0.383	9
2	depth_first_graph_search	20	21	0.007	20	72	624	0.551	619
3	uniform_cost_search	20	60	0.022	6	72	5154	0.628	9
4	greedy_best_first_graph_search with h_unmet_goals	20	7	0.002	6	72	17	0.013	9
5	greedy_best_first_graph_search with h_pg_levelsum	20	6	0.57	6	72	9	2.401	9
6	greedy_best_first_graph_search with h_pg_maxlevel	20	6	0.18	6	72	27	2.976	9
7	greedy_best_first_graph_search with h_pg_setlevel	20	6	0.615	6	72	27	9.611	9
8	astar_search with h_unmet_goals	20	50	0.018	6	72	2467	0.776	9
9	astar_search with h_pg_levelsum	20	28	0.388	6	72	357	49.888	9
10	astar_search with h_pg_maxlevel	20	43	0.308	6	72	2887	267.257	9
11	astar_search with h_pg_setlevel	20	42	0.776	6	72	1987	680.743	9

Figure 2: CAPTION HERE!