Logan Griffin

HW & PA 2

7/8/2025

EN.665.645.8VL.SU25

	Actions	Expansions	Goal Tests	New Nodes	Time
Cargo 1					
1. breadth_first_search	20	43	56	178	0.0382496
2. depth_first_graph_search	20	21	22	84	0.0118321
3. uniform_cost_search	20	60	62	240	0.0467404
4. greedy_best_first_graph_search h_unmet_goals	20	7	9	29	0.0077352
5. greedy_best_first_graph_search h_pg_levelsum	20	6	8	28	0.3667935
6. greedy_best_first_graph_search h_pg_maxlevel	20	6	8	24	0.232288
7. greedy_best_first_graph_search h_pg_setlevel	20	6	8	28	0.7033793
8. astar_search h_unmet_goals	20	50	52	206	0.0466548
9. astar_search h_pg_levelsum	20	28	30	122	0.5531474
10. astar_search h_pg_maxlevel	20	43	45	180	0.578106
11. astar_search h_pg_setlevel	20	33	35	138	1.0320895
Cargo 2					
1. breadth_first_search	72	3343	4609	30503	0.4670989
2. depth_first_graph_search	72	624	625	5602	0.5996625
3. uniform_cost_search	72	5154	5156	46618	0.9565186
4. greedy_best_first_graph_search h_unmet_goals	72	17	19	170	0.0229262
5. greedy_best_first_graph_search h_pg_levelsum	72	9	11	86	1.0319351
6. greedy_best_first_graph_search h_pg_maxlevel	72	27	29	249	0.6413617
7. greedy_best_first_graph_search h_pg_setlevel	72	9	11	84	2.2131756
8. astar_search h_unmet_goals	72	2467	2469	22522	0.7242451
9. astar_search h_pg_levelsum	72	357	359	3426	8.9124212
10. astar_search h_pg_maxlevel	72	2887	2889	26594	55.2846328
11. astar_search h_pg_setlevel	72	1037	1039	9605	97.7314608
Cargo 3					
1. breadth_first_search	88	14663	18098	129625	1.2513373
2. depth_first_graph_search	88	408	409	3364	0.3117957
3. uniform_cost_search	88	18510	18512	161936	1.7870964
4. greedy_best_first_graph_search h_unmet_goals	88	25	27	230	0.0323817
5. greedy_best_first_graph_search h_pg_levelsum	88	14	16	126	1.8661325
6. greedy_best_first_graph_search h_pg_maxlevel	88	21	23	195	1.007868
7. greedy_best_first_graph_search h_pg_setlevel	88	35	37	345	8.4415516
8. astar_search h_unmet_goals	88	7388	7390	65711	1.7911402
9. astar_search h_pg_levelsum	88	369	371	3403	18.7819136
10. astar_search h_pg_maxlevel	88	9580	9582	86312	326.3290093
11. astar_search h_pg_setlevel	88	3423	3425	31596	558.8461914
Cargo 4					
1. breadth_first_search	104	99736	114953	944130	6.2789287
2. depth_first_graph_search					
3. uniform_cost_search	104	113339	113341	1066413	7.7313523
4. greedy_best_first_graph_search h_unmet_goals	104	29	31	280	0.0664338
5. greedy_best_first_graph_search h_pg_levelsum	104	17	19	165	1.6827447
6. greedy_best_first_graph_search h_pg_maxlevel	104	56	58	580	2.2664294
7. greedy_best_first_graph_search h_pg_setlevel	104	107	109	1164	25.4410146
8. astar_search h_unmet_goals	104	34330	34332	328509	4.9665722
9. astar_search h_pg_levelsum	104	1208	1210	12210	67.5912427
10. astar_search h_pg_maxlevel	104	62077	62079	599376	5128.411036
11. astar_search h_pg_setlevel	104	22606	22608	224229	7811.759831

The table above shows the data from running the search problem with the different algorithms. Based on the data, here are my answers to the posed questions:

- 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. 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?

   If a system needed to operate in real time, the calculation time would be the largest factor. On the easiest cargo plane problem, the fastest algorithms were depth first search, and the greedy search with the unmet goals heuristic. The depth first search worked well here because few actions means few branches, which optimizes that kind of search. The unmet goals heuristic also has very little overhead since it is quick and doesn't require many other calculations. Other than those two algorithms, the greedy searches with the other heuristics also performed well for the small problem size.

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)

   For very large domains, breadth first search works well because it takes into account all the branches and actions before picking one and exploring it all the way as depth first search does. Again, the greedy search + unmet goals heuristic provided good performance. The best algorithms here would be the options that combine speed with accuracy. Since greedy algorithms are usually faster than astar algorithms, using a greedy approach with an accurate heuristic, or astar with low-overhead heuristic seems like the best bet here. From the set of algorithms we tested, that could either be the greedy search + the set level heuristic, or the astar search with the unmet goals or levelsum heuristic.

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

   For planning problems where it is important to find only optimal plans, greedy searches with heuristics are probably off the table. The astar searches are more accurate in providing optimal solutions. From the astar searches, the unmet goals heuristic had the slowest run time for the most complex problem by far. This would probably be the algorithm chosen to find an optimal solution. The uniform cost search also returns an optimal path, and performed almost as well as the astar + unmet goals heuristic did for the 4th cargo problem. Overall, to find optimal plans, a fast run time can be sacrificed for accuracy.