Build a Forward-Planning Agent

Results

The following table shows the results of running all algorithms for problems 1 and 2.

Table 1: Results for applying all algorithms on problems 1 and 2.

#	Algorithm	# of actions P1/P2	# of new node expansions P1/P2	Total time P1/P2
1	breadth_first_search	20/72	178/30503	0.00687/2.03841
2	depth_first_graph_search	20/72	84/5602	0.00351/3.11525
3	uniform_cost_search	20/72	240/5154	0.00998/3.37171
4	greedy_best_first_graph_search with h_unmet_goals	20/72	29/46618	0.00165/0.02115
5	greedy_best_first_graph_search with h_pg_levelsum	20/72	28/170	0.49210/11.63848
6	greedy_best_first_graph_search with h_pg_maxlevel	20/72	24/29	0.14683/6.84647
7	greedy_best_first_graph_search with h_pg_setlevel	20/72	28/84	1.34864/30.31048
8	astar_search with h_unmet_goals	20/72	206/22522	0.00941/2.31695
9	astar_search with h_pg_levelsum	20/72	122/3426	1.26211/298.64415
10	astar_search with h_pg_maxlevel	20/72	180/26594	0.51405/692.00727
11	astar_search with h_pg_setlevel	20/72	138/9605	3.56769/ 2443.90355

From the table above, algorithms 2, 3, 6, 7, 10, and 11 will be excluded from problems 3 and 4 experiments.

The following table shows running one uninformed search, two heuristics with greedy best-first search, and two heuristics with A*:

Table 2: Results for applying 1, 4, 5, 8, and 9 algorithms on problems 3 and 4.

#	Algorithm	# of actions P3/P4	# of new node expansions P3/P4	Total time P3/P4
1	breadth_first_search	88/104	129625/944130	10.84590/34.47811
4	greedy_best_first_graph_search with h_unmet_goals	88/104	230/280	0.050520/0.030178
5	greedy_best_first_graph_search with h_pg_levelsum	88/104	126/165	9.952331/18.65694
8	astar_search with h_unmet_goals	88/104	65711/328509	8.56944/56.24570
9	astar_search with h_pg_levelsum	88/104	3403/12210	481.16197/1054.180663

Questions and Answers

Q1. 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?

A1. From Table 1, I would suggest using **greedy_best_first_graph_search with h_unmet_goals** algorithm as it took a fraction of a second for planning a very restricted domain.

Q2. 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)

A2. From Table2, I believe algorithms **greedy_best_first_graph_search with h_unmet_goals** and **greedy_best_first_graph_search with h_pg_levelsum** execution time isn't too long.

Q3. Which algorithm or algorithms would be most appropriate for planning problems where it is important to find only optimal plans?

A3. The first three algorithms, namely: breadth_first_search, breadth_first_search, and uniform_cost_search.