Report
Udacity AIND Project 2

Number of nodes expanded against number of actions in the domain

Problem	Air Cargo 1	Air Cargo 2	Air Cargo 3	Air Cargo 4	
Domain Size (actions)	20	72	88	104	
breadth_first_search	43	3343	14663	N/A	Expansions
depth_first_graph_search	21	624	408	N/A	
uniform_cost_search	60	5154	18510	113339	
greedy_best_first_graph_search h_unmet_goals	7	17	25	29	
greedy_best_first_graph_search h_pg_levelsum	6	9	14	17	
greedy_best_first_graph_search h_pg_maxlevel	6	27	21	N/A	
greedy_best_first_graph_search h_pg_setlevel	6	9	35	N/A	

astar_search h_unmet_goals	50	2467	7388	34330	
astar_search h_pg_levelsum	28	357	369	1208	
astar_search h_pg_maxlevel	43	2887	N/A	N/A	
astar_search h_pg_setlevel	33	1037	N/A	N/A	

Discussion

The main observation is that the number of expansions increases proportionally with the problem size: More actions results in more expansions.
Another observation is that greedy best first search tends to have the smallest number of expansions, while uniform cost search (aka Dijkstra’s Algorithm) has the biggest number.

Search time against the number of actions in the domain

Problem	Air Cargo 1	Air Cargo 2	Air Cargo 3	Air Cargo 4	
Domain Size (actions)	20	72	88	104	
breadth_first_search	0.006189597	2.010661678	10.58016986	95.00472258	Time Elapsed (Sec)
depth_first_graph_search	0.003374292	3.054074663	1.15805556499998	N/A	
uniform_cost_search	0.010100119	3.31699229	14.13545691	115.5051989	
greedy_best_first_graph_search h_unmet_goals	0.001647887	0.021276094	0.037961981	0.060654578	
greedy_best_first_graph_search h_pg_levelsum	0.308950642	6.888271806	15.51751293	27.9937368	
greedy_best_first_graph_search h_pg_maxlevel	0.232442059	13.79834977	18.93006415	N/A	
greedy_best_first_graph_search h_pg_setlevel	1.056358447	23.07740593	110.945302	N/A	

astar_search h_unmet_goals	0.009606304	2.208372651	8.48107668	55.1522493	
astar_search h_pg_levelsum	0.782495772	180.6400217	279.7768834	1565.205938	
astar_search h_pg_maxlevel	0.820162003	1037.538144	N/A	N/A	
astar_search h_pg_setlevel	2.758181196	1772.61183	N/A	N/A	

Discussion

The main observation is that the search time increases proportionally with the problem size: More actions results in a longer search time, which is expected.
We also observe that greedy best first search with unmet goals heuristic has the shortest search time.

Length of the plans returned by each algorithm on all search problems

Problem	Air Cargo 1	Air Cargo 2	Air Cargo 3	Air Cargo 4	
Domain Size (actions)	20	72	88	104	
breadth_first_search	6	9	12	14	Plan Length
depth_first_graph_search	20	619	392	N/A	
uniform_cost_search	6	9	12	14	
greedy_best_first_graph_search h_unmet_goals	6	9	15	18	
greedy_best_first_graph_search h_pg_levelsum	6	9	14	17	
greedy_best_first_graph_search h_pg_maxlevel	6	9	13	N/A	
greedy_best_first_graph_search h_pg_setlevel	6	9	17	N/A	
astar_search h_unmet_goals	6	9	12	14	

astar_search h_pg_levelsum	6	9	12	15	
astar_search h_pg_maxlevel	6	9	12	N/A	
astar_search h_pg_setlevel	6	9	12	N/A	

Discussion

DFS has the worst performance in terms of plan length. It's known that it does not guarantee returning the optimal solution, as opposed to BFS which will always return the optimal solution. Greedy best first search is sub-optimal but not as bad as DFS and still very close to the optimal solutions.

In a very restricted domain and needs to operate in real time

In that case greedy best first search with the unmet-goals heuristic will be appropriate since it has the shortest search time, so that it can meet the real time requirement.

In very large domains

In this case, the algorithm has to be efficient so that the plan lengths for delivery routes are as small as possible. The only constraint about search time is to provide the result within one day. A* search with unmet-goals heuristic will be appropriate as it meets the above requirements.

Find only optimal plans

In that case we must use BFS or uniform-cost search, because they are guaranteed to find the optimal solution every time.