Build a Forward-Planning Agent Project

Mohammed Almaghrabi

Introduction:

The objective of this report is to study and analyze multiple search algorithms that used in the planning graph problem in terms of search time, number of expanded nodes and the optimally of the solution as a function of domain size. The algorithms are executed on 4 different variants of the air cargo problem. All 11 algorithms are executed for the first two problem, and the following algorithms were eliminated from 3th and 4th problem:

- depth_first_graph_search
- astar_search with h_pg_maxlevel
- astar_search with h_pg_setlevel

They were expected to take too much time to finish.

Actions vs Expanded Nodes:

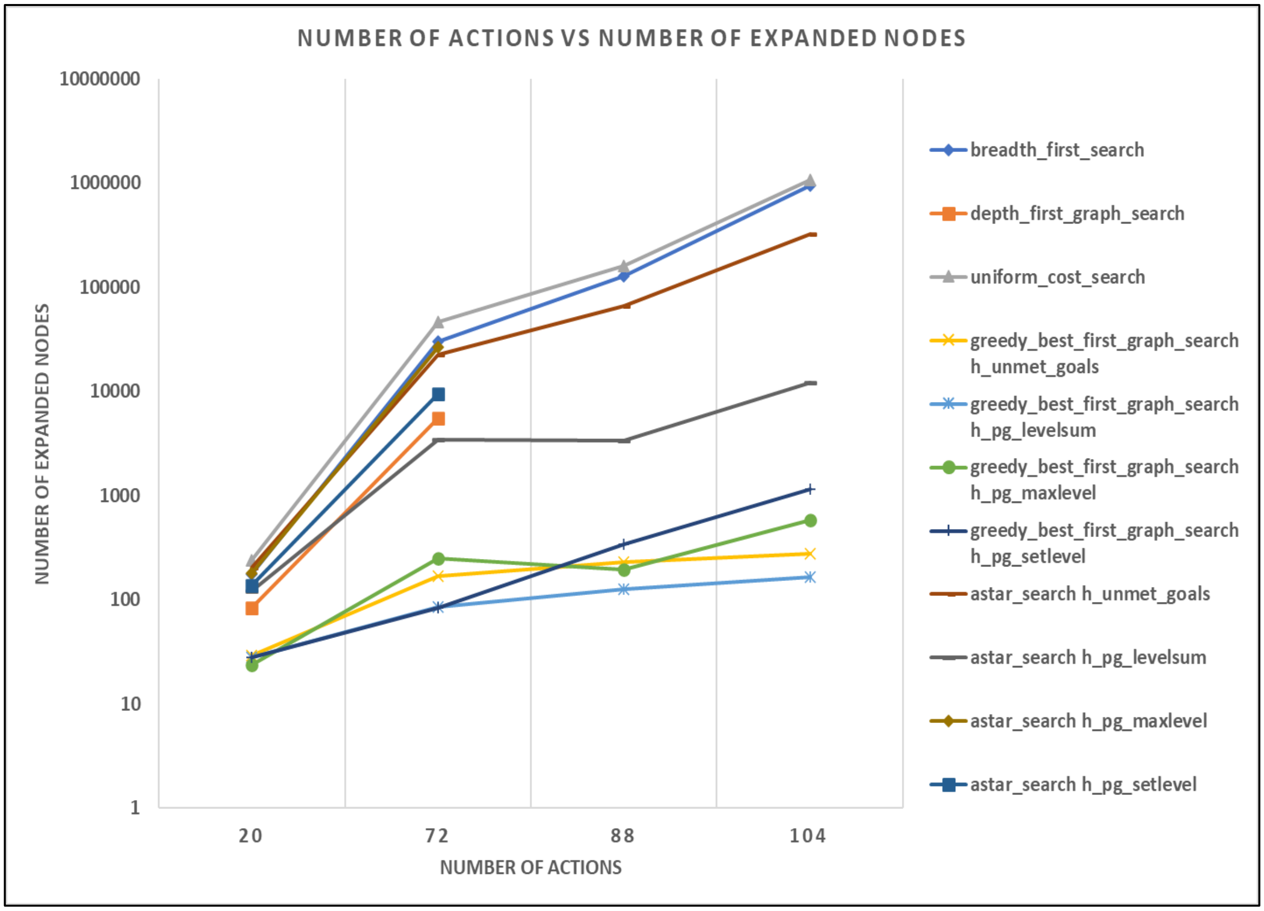

Figure 1, Problem 1, 2, 3 and 4 were executed on 11 search algorithms

In the Figure 1, the chart shows the number of actions need to plan and unload the cargos to one or multiple cities against the number of expanded nodes of different algorithms. Problem 1, 2, 3 and 4 have 20, 72, 88 and 104 number of actions, respectively. As presented, a positive linear relationship exists between the number of actions and the number of expanded nodes. As the number of possible actions in the problem increases, the number of expanded nodes increases as well. Uniform cost search is having the maximum number of expanded nodes in all the problems. Breadth first search almost have the same number of expanded nodes as uniform cost search. This should be taken into consideration if there are memory constraints. Greedy best search is having a low number of expanded nodes in all of the problems. Level sum heuristics seems that it decreases the number of expanded nodes quite well as it has the minimum value across all the other heuristics in both A* and greedy best first search. Although all actions have the same cost in this problem, uniform cost search expanded more nodes than breadth first search as they processed the nodes in different order.

Tables for Search Algorithms:

Please refer to the following tables for more information:

PR 1							
#	Search	Actions	Expansions	Goal Tests	New Nodes	Plan length	Time elapsed in seconds
1	breadth_first_search	20	43	56	178	6	0.0061302
2	depth_first_graph_search	20	21	22	84	20	0.0033851
3	uniform_cost_search	20	60	62	240	6	0.0101766
4	greedy_best_first_graph_search h_unmet_goals	20	7	9	29	6	0.001606
5	greedy_best_first_graph_search h_pg_levelsum	20	6	8	28	6	0.1646215
6	greedy_best_first_graph_search h_pg_maxlevel	20	6	8	24	6	0.1066779
7	greedy_best_first_graph_search h_pg_setlevel	20	6	8	28	6	0.4597198
8	astar_search h_unmet_goals	20	50	52	206	6	0.010438
9	astar_search h_pg_levelsum	20	28	30	122	6	0.3466078
10	astar_search h_pg_maxlevel	20	43	45	180	6	0.3931851
11	astar_search h_pg_setlevel	20	33	35	138	6	1.0420297

PR 2							
#	Search	Actions	Expansions	Goal Tests	New Nodes	Plan length	Time elapsed in seconds
1	breadth_first_search	72	3343	4609	30503	9	1.9881655
2	depth_first_graph_search	72	624	625	5602	619	2.9951805
3	uniform_cost_search	72	5154	5156	46618	9	3.2830171
4	greedy_best_first_graph_search h_unmet_goals	72	17	19	170	9	0.0190105
5	greedy_best_first_graph_search h_pg_levelsum	72	9	11	86	9	2.9405598
6	greedy_best_first_graph_search h_pg_maxlevel	72	27	29	249	9	4.7158528
7	greedy_best_first_graph_search h_pg_setlevel	72	9	11	84	9	10.951014
8	astar_search h_unmet_goals	72	2467	2469	22522	9	2.2028603
9	astar_search h_pg_levelsum	72	357	359	3426	9	81.42441
10	astar_search h_pg_maxlevel	72	2887	2889	26594	9	471.70041
11	astar_search h_pg_setlevel	72	1037	1039	9605	9	987.8224

PR 3							
#	Search	Actions	Expansions	Goal Tests	New Nodes	Plan length	Time elapsed in seconds
1	breadth_first_search	88	14663	18098	129625	12	10.183384
3	uniform_cost_search	88	18510	18512	161936	12	13.710472
4	greedy_best_first_graph_search h_unmet_goals	88	25	27	230	15	0.0304425
5	greedy_best_first_graph_search h_pg_levelsum	88	14	16	126	14	6.9744639
6	greedy_best_first_graph_search h_pg_maxlevel	88	21	23	195	13	6.9491598
7	greedy_best_first_graph_search h_pg_setlevel	88	35	37	345	17	59.013667
8	astar_search h_unmet_goals	88	7388	7390	65711	12	7.9583709
9	astar_search h_pg_levelsum	88	369	371	3403	12	150.42325

PR 4							
#	Search	Actions	Expansions	Goal Tests	New Nodes	Plan length	Time elapsed in seconds
1	breadth_first_search	104	99736	114953	944130	14	92.688023
3	uniform_cost_search	104	113339	113341	1066413	14	114.25189
4	greedy_best_first_graph_search h_unmet_goals	104	29	31	280	18	0.0599615
5	greedy_best_first_graph_search h_pg_levelsum	104	17	19	165	17	12.840629
6	greedy_best_first_graph_search h_pg_maxlevel	104	56	58	580	17	21.391837
7	greedy_best_first_graph_search h_pg_setlevel	104	107	109	1164	23	273.47908
8	astar_search h_unmet_goals	104	34330	34332	328509	14	53.949754
9	astar_search h_pg_levelsum	104	1208	1210	12210	15	846.6383

Actions vs Time Elapsed:

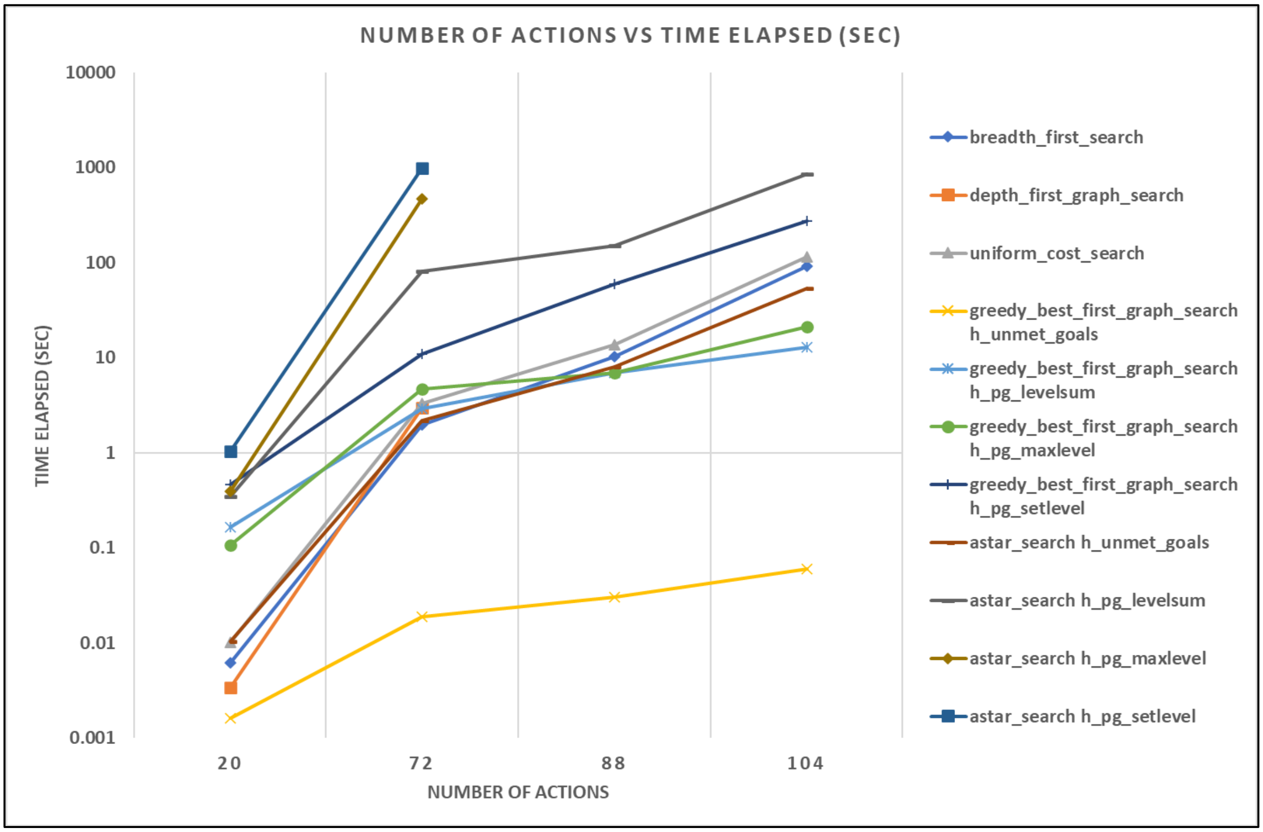

Figure 2, Time Elapsed vs Number of Action

In the Figure 2, the chart shows the number of actions need to plan and unload the cargos to one or multiple cities against the time needed to find the optimal planning route and unloading the cargos of different algorithms. Problem 1, 2, 3 and 4 have 20, 72, 88 and 104 number of actions, respectively. The search time is similar to the positive linear relationship between the number of action and expanded nodes. As the number of the expanded nodes increases, the search time as well as the number of actions increases. Greedy_best_first_graph_search with h_unmet_goals had minimum search time in all the problems. In general, Unmet goals heuristic seems to require less of the elapsed time. A* took more time than greedy best first as it expands more nodes. Uniform cost search took slightly more time than breadth first search because of its priority queue overhead and it processes the nodes in a different order.

Length of Plans:

#	Search	Problem 1	Problem 2	Problem 3	Problem 4
1	breadth_first_search	6	9	12	14
2	depth_first_graph_search	20	619	N/A	N/A
3	uniform_cost_search	6	9	12	14
4	greedy_best_first_graph_search h_unmet_goals	6	9	15	18
5	greedy_best_first_graph_search h_pg_levelsum	6	9	14	17
6	greedy_best_first_graph_search h_pg_maxlevel	6	9	13	17
7	greedy_best_first_graph_search h_pg_setlevel	6	9	17	23
8	astar_search h_unmet_goals	6	9	12	14
9	astar_search h_pg_levelsum	6	9	12	15
10	astar_search h_pg_maxlevel	6	9	N/A	N/A
11	astar_search h_pg_setlevel	6	9	N/A	N/A

The table above shows the length of the plan extracted for different algorithms. Astar_search with h_unmet_goals, breadth_first_search, and uniform_cost_search always got the minimum path length in all problems which is expected as they all optimal and has admissible heuristics (for informed ones). Greedy best first search is not optimal by definition that is why it did not get the best plan in problems 3 and 4; greedy_best_first_graph_search with h_pg_maxlevel was closest to the shortest plan length.

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?

greedy_best_first_graph_search with h_unmet_goals, breadth_first_search and depth_first_graph_search should be appropriate as they took short execution time in problems 1 and which have low number of actions.

depth_first_graph_search, although not being optimal, it should get a solution in a short 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)?

greedy_best_first_graph_search with h_pg_levelsum, greedy_best_first_graph_search with

h_pg_maxlevel, greedy_best_first_graph_search with h_unmet_goals as all of them took considerably

short execution time in problems 3 and 4.

If there is time constraints, the following algorithms are best fit:

greedy_best_first_graph_search h_unmet_goals, greedy_best_first_graph_search h_pg_levelsum

greedy_best_first_graph_search h_pg_maxlevel

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

breadth_first_search, uniform_cost_search (if actions have different costs), astar_search with h_unmet_goals, astar_search with h_pg_maxlevel and astar_search with h_pg_setlevel.

Greedy best first and depth first search are not appropriate as they are both do not optimal plans by

definition.

astar_search with h_pg_levelsum is not appropriate as level sum heuristic is not admissible