Building a Forward Planning Agent

Author: Ray Phan

Date: June 18, 2021

Section 1 - Analyze the search complexity as a function of domain size, search algorithm, and heuristic.

Table 1: Number of nodes expanded against number of actions the domain

Notation:

- ACP #x: Air Cargo Problem #x, where `x = {1, 2, 3, 4}`
- NoE: Number of Expansions

Number of actions for each ACP

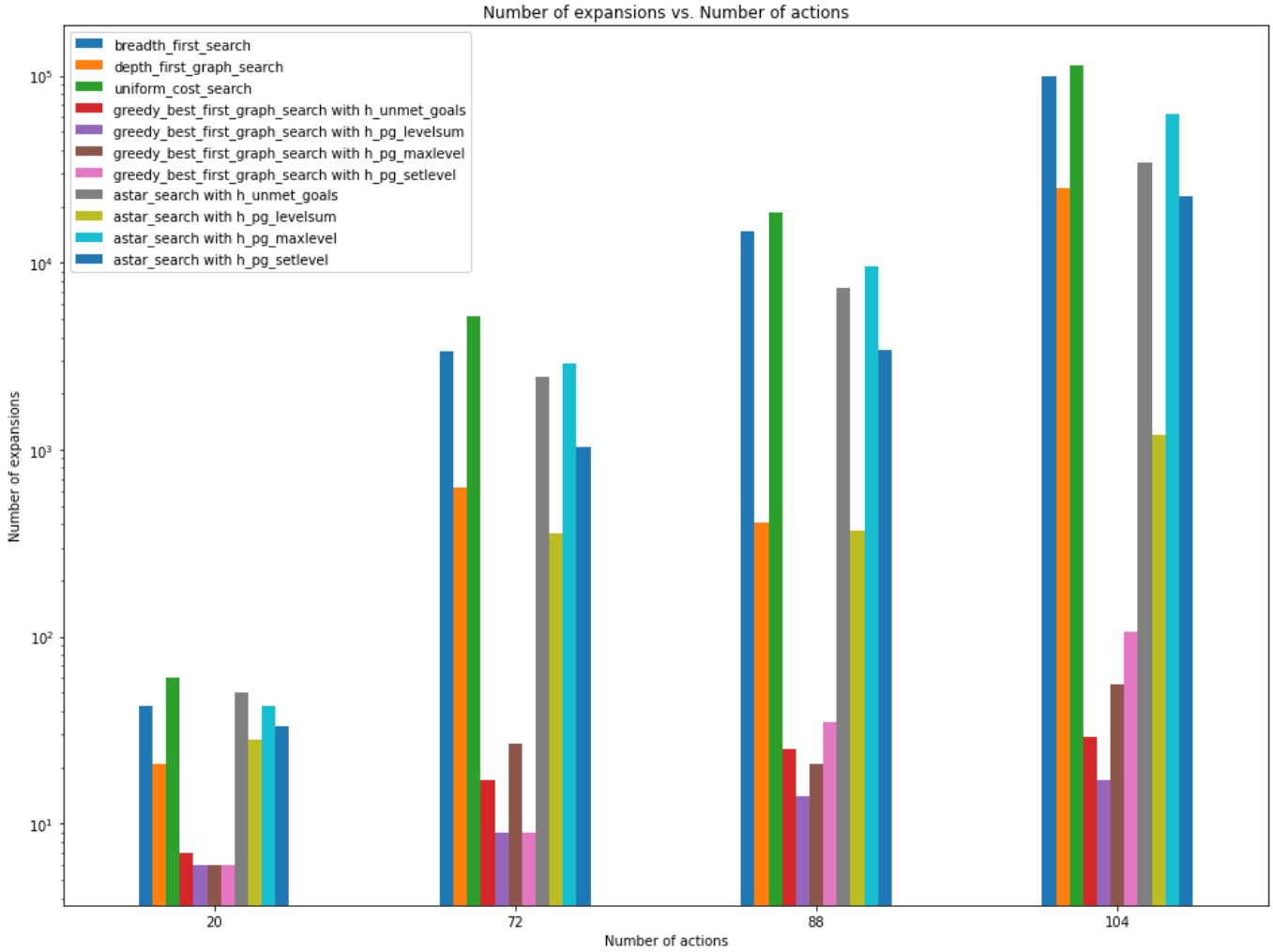
- ACP #1: 20
- ACP #2: 72
- ACP #3: 88
- ACP #4: 104

Algorithm	NoE ACP #1	NoE ACP #2	NoE ACP #3	NoE ACP #4
`breadth_first_search`	43	3343	14663	99736
`depth_first_graph_search`	21	624	408	25174
`uniform_cost_search`	60	5154	18510	113339
`greedy_best_first_graph_search with h_unmet_goals`	7	17	25	29
`greedy_best_first_graph_search with h_pg_levelsum`	6	9	14	17
`greedy_best_first_graph_search with h_pg_maxlevel`	6	27	21	56
`greedy_best_first_graph_search with h_pg_setlevel`	6	9	35	107
`astar_search with h_unmet_goals`	50	2467	7388	34330
`astar_search with h_pg_levelsum`	28	357	369	1208
`astar_search with h_pg_maxlevel`	43	2887	9580	62077
`astar_search with h_pg_setlevel`	33	1037	3423	22606

Bar Chart #1: Number of nodes expanded against number of actions the domain

NOTE: The source used to create the bar chart can be found in the Jupyter notebook found in this assignment directory.

NOTE #2: The `y`-axis is logarithmically scaled to account for the large difference in counts between Air Cargo Problem #1 and Air Cargo Problem #4.



Analysis

We can see from the above table and bar chart that the number of expanded nodes increases with the number of available actions increases. Uniform cost search overall had the most number of expansions over all Air Cargo Problems. Greedy Best First Search regardless of the heuristic used had the minimum number of expanded nodes - most notably the level sum heuristic.

Section 2 - Analyze search time as a function of domain size, search algorithm, and heuristic.

Table 2: Search time against number of actions in the domain

Notation:

- ACP #x: Air Cargo Problem #x, where `x = {1, 2, 3, 4}`
- Time is in seconds - Reported up to 4 significant digits

Number of actions for each ACP

- ACP #1: 20
- ACP #2: 72
- ACP #3: 88
- ACP #4: 104

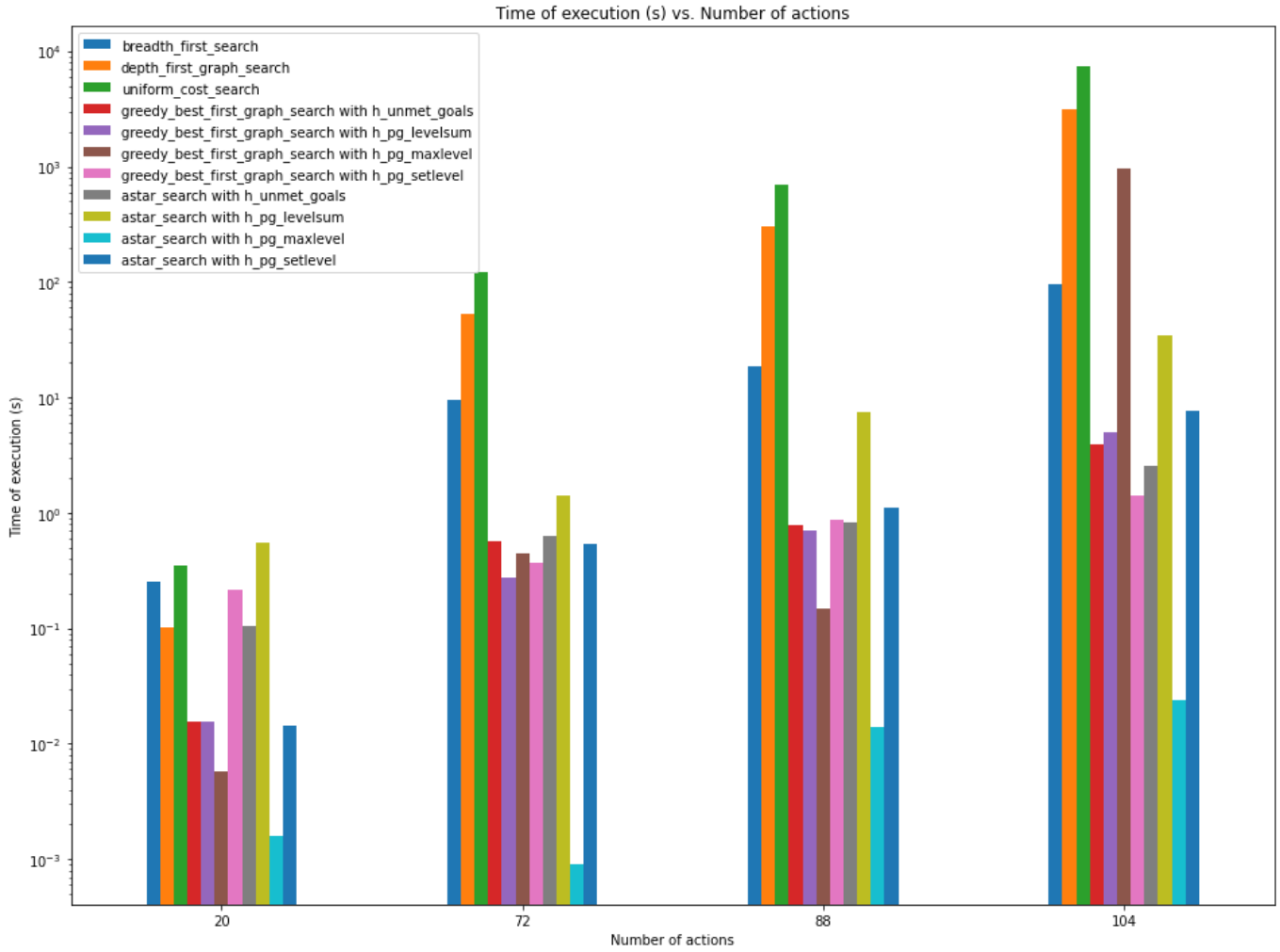
NOTE: The timings reported are using an alternative Python interpreter called PyPy. The main differences are that the interpreter is memory efficient and has a Just-In-Time (JIT) compiler to promote faster execution times. If you are using the original Python interpreter, the timings reported will not be the same.

Algorithm	ACP #1 Time	ACP #2 Time	ACP #3 Time	ACP #4 Time
breadth_first_search	0.2563	9.6307	18.7543	96.4861
depth_first_graph_search	0.1013	53.6832	307.8437	3159.5437
uniform_cost_search	0.3548	121.5738	693.2235	7389.6437
greedy_best_first_graph_search with h_unmet_goals	0.0156	0.5644	0.7893	3.9654
greedy_best_first_graph_search with h_pg_levelsum	0.0156	0.2765	0.6969	5.0149
greedy_best_first_graph_search with h_pg_maxlevel	0.0057	0.4504	0.1465	968.9437
greedy_best_first_graph_search with h_pg_setlevel	0.2184	0.3678	0.8634	1.4156
astar_search with h_unmet_goals	0.1039	0.6354	0.8361	2.5663
astar_search with h_pg_levelsum	0.5516	1.4267	7.4406	34.6134
astar_search with h_pg_maxlevel	0.0016	0.0009	0.0138	0.0237
astar_search with h_pg_setlevel	0.0145	0.5452	1.1093	7.6457

Bar Chart #2: Number of nodes expanded against number of actions in the domain

NOTE: The source used to create the bar chart can be found in the Jupyter notebook found in this assignment directory.

NOTE #2: The y-axis is logarithmically scaled to account for the large difference in counts between Air Cargo Problem #1 and Air Cargo Problem #4.



Analysis

As seen from the above table and bar chart, we can also verify that the search time increases when the number of available actions increases. Specifically, for Air Cargo Problem #1, we can see that the times were amongst the shortest across all algorithms. For Air Cargo Problem #4, this created the longest times across all algorithms. For each Air Cargo Problem, A* search with the max level heuristic generally had the shortest search times, whereas A* search with the level sum heuristic took the most time for Air Cargo Problem #1, and Uniform Cost Search took the most time for Air Cargo Problems #2, #3 and #4. In general, the uninformed search algorithms (Breadth-First Search, Depth-First Search and Uniform Cost Search) took more time to execute than the informed search algorithms (Greedy Best First Search and A*). The heuristics definitely have an effect on the search time for the informed search algorithms when it comes to the available action space. We notice that the search time grows as the available action space grows.

Section 3 - Analyze the optimality of solution as a function of domain size, search algorithm and heuristic

Table 3: Plan length against the number of actions in the domain

Notation:

- ACP #x: Air Cargo Problem #x, where `x = {1, 2, 3, 4}`
- PL: Plan Length

Number of actions for each ACP

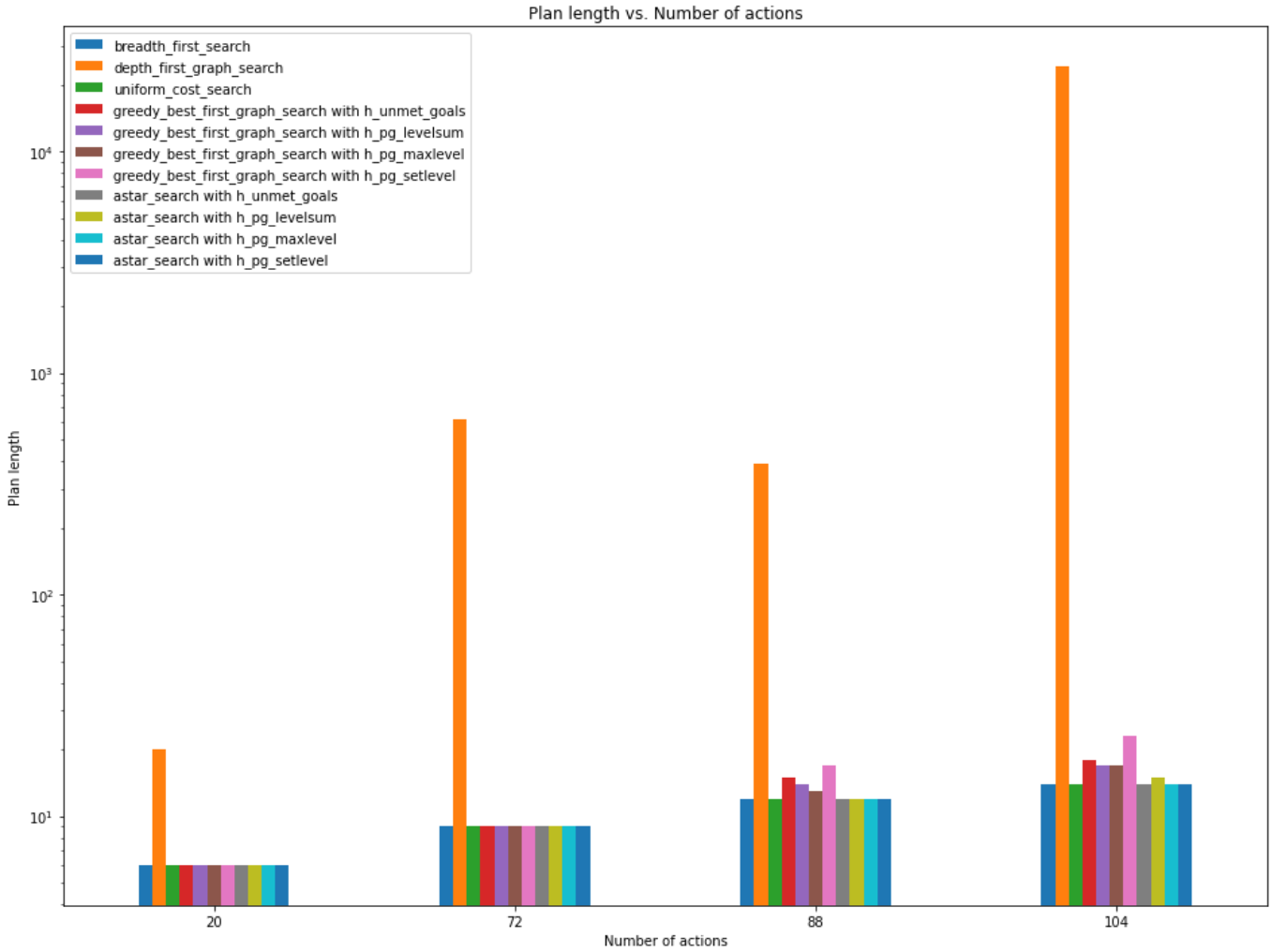
- ACP #1: 20
- ACP #2: 72
- ACP #3: 88
- ACP #4: 104

Algorithm	PL ACP #1	PL ACP #2	PL ACP #3	PL ACP #4
breadth_first_search	6	9	12	14
depth_first_graph_search	20	619	392	24132
uniform_cost_search	6	9	12	14
greedy_best_first_graph_search with h_unmet_goals	6	9	15	18
greedy_best_first_graph_search with h_pg_levelsum	6	9	14	17
greedy_best_first_graph_search with h_pg_maxlevel	6	9	13	17
greedy_best_first_graph_search with h_pg_setlevel	6	9	17	23
astar_search with h_unmet_goals	6	9	12	14
astar_search with h_pg_levelsum	6	9	12	15
astar_search with h_pg_maxlevel	6	9	12	14
astar_search with h_pg_setlevel	6	9	12	14

Bar Chart #3: Number of nodes expanded against number of actions the domain

NOTE: The source used to create the bar chart can be found in the Jupyter notebook found in this assignment directory.

NOTE #2: The y-axis is logarithmically scaled to account for the large difference in counts between Air Cargo Problem #1 and Air Cargo Problem #4.



Analysis

We can see from the above table and bar chart that the plan length increases with the number of available actions increasing. Depth-first search is the most problematic and produced the longest plan across all Air Cargo Problems. All of the other algorithms showed comparable plan lengths for Air Cargo Problems #1 and #2. For Air Cargo Problems #3 and #4, the increasing available action size demonstrates that Breadth-First Search, Uniform-Cost Search and A* produced the shortest plans.