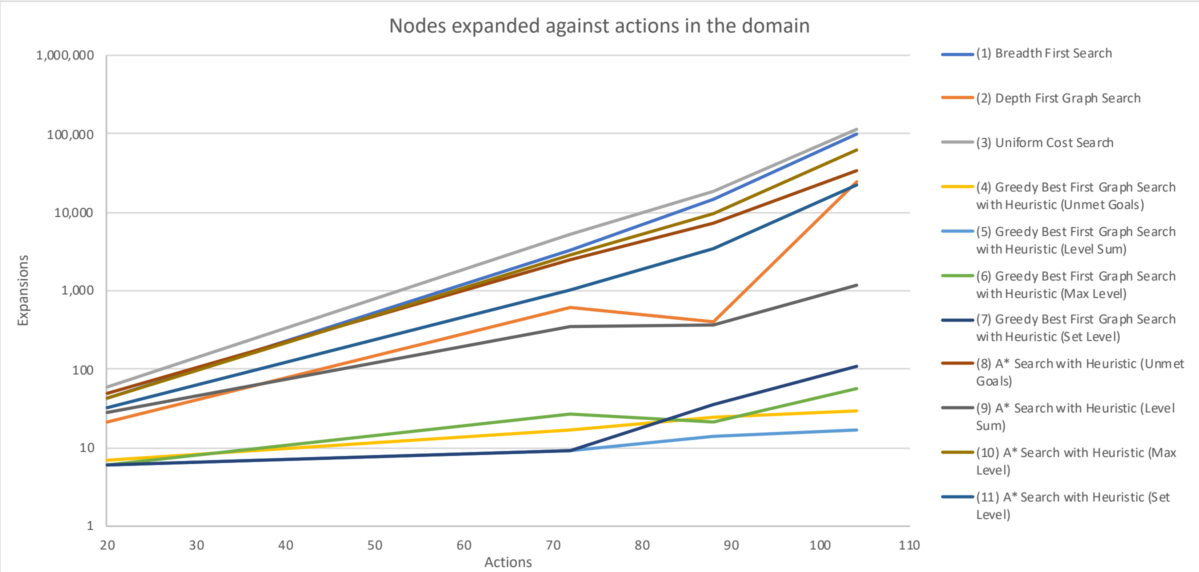
Project: Build a Forward Planning Agent

Robert Paech, 24 Oct 2021

Use a table or chart to analyse the number of nodes expanded against number of actions in the domain

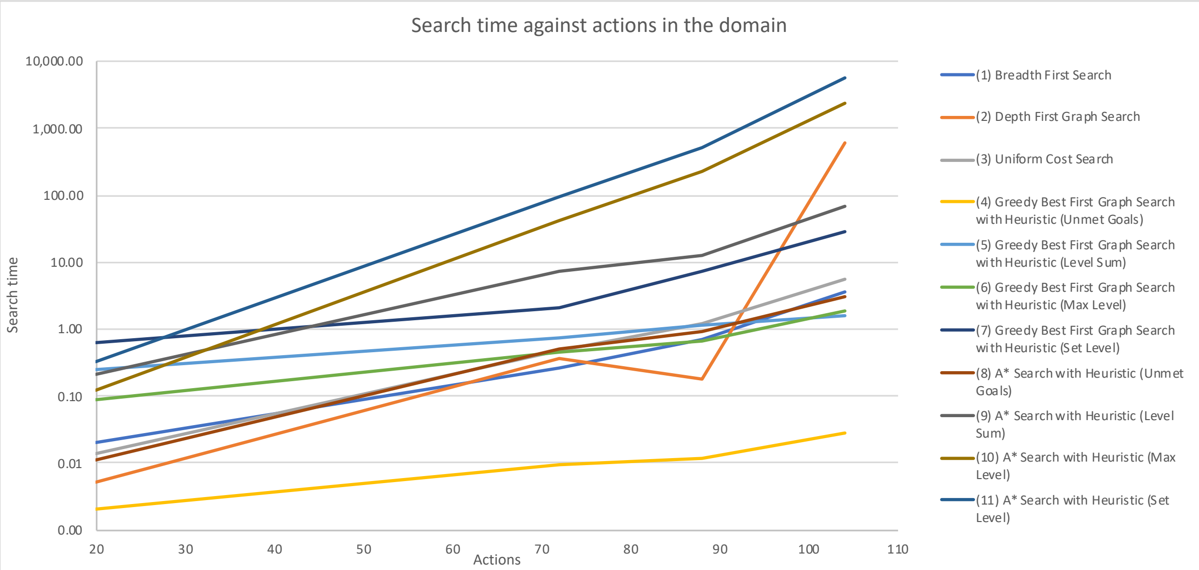


In terms of search complexity, all search algorithms appear to exhibit exponential growth relative to the number of actions.

All greedy best first graph search algorithms exhibit the best performance, with the level sum heuristic consistently at the top and unmet goals showing consistently good results (albeit across a small sample set of problems). The worst three performing algorithms are uniform cost search, breadth first search and the A* search with the max level heuristic.

The depth first graph search exhibits an unexpected improvement with problem 3 (88 actions) over problem 2 (72 actions); however, this is likely due to the specific nature of problem 3 and the goals being found early in the search. The sample set of four problems is small and a larger set of problems would help to minimise outliers like this.

Use a table or chart to analyse the search time against the number of actions in the domain

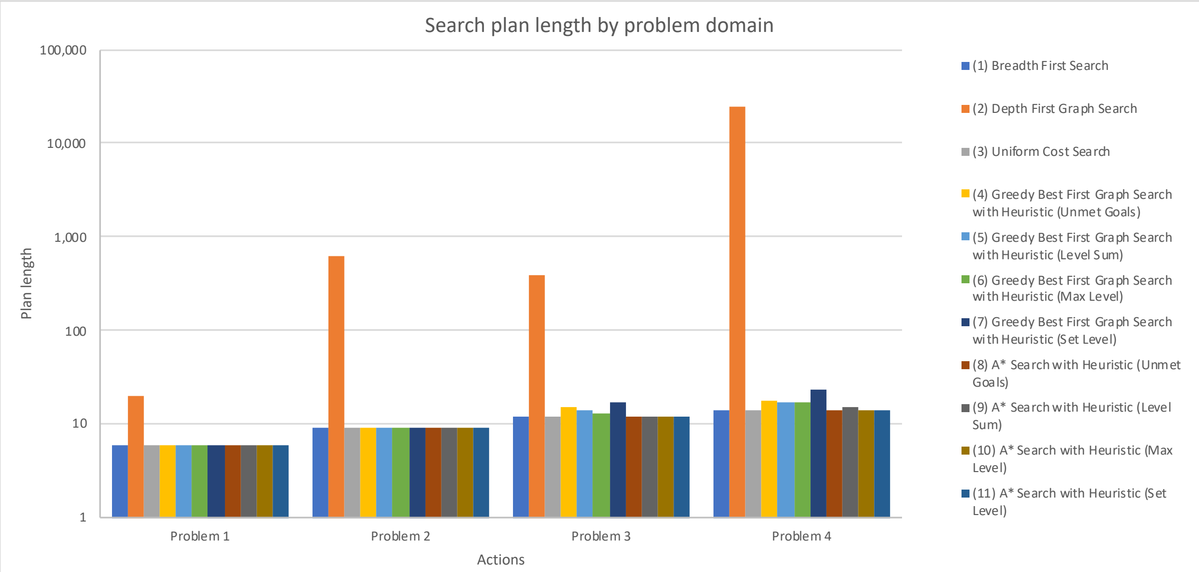


In terms of search duration, all search algorithms again appear to exhibit exponential growth relative to the number of actions.

The greedy best first search with unmet goals algorithm clearly exhibits the best performance across the four problems in the same set. The greedy best first search algorithms with the level sum and max level heuristics are worse, but show a relatively slow growth rate in time with increasing number of actions.

The A* search with set level heuristic, followed by the A* search with max level heuristic are the two worst performers in terms of search duration. With the limited set of sample problems, it appears that the depth first search may be the overall worst performer when a problem has a large number of actions.

Use a table or chart to analyse the length of the plans returned by each algorithm on all search problems



In terms of plan length, except for depth for search, most algorithms perform reasonably well within a single problem domain.

For problems 1 and 2, except for depth for search, all algorithms exhibited the same plan length. In problem 3, breadth first search, uniform cost search and all four A* search algorithms performed the same. A similar result also occurred for problem 4, except for the A* search with the level sum heuristic, which performed slightly worse. In problems 3 and 4, all the greedy best first search algorithms exhibited larger plan lengths, with the level sum and unmet goals heuristics showing the worst performance in the group.

The depth first search was the overall worst performer, showing significant degradation in plan length with increasing number of actions.

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?

The greedy best first search algorithm with the unmet goals heuristic suits this domain. It demonstrates the lowest and most consistent time duration over a reasonable number of actions.

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)

The greedy best first graph search algorithms, with the level sum and max level heuristics, would best suit this problem domain. They both exhibit relatively low time and space complexity, while also providing a plan length that is close to optimal.

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

The following algorithms appear to all find optimal plans across the four problems:

- breadth first search
- uniform cost search
- A* search with unmet goals heuristic
- A* search with max level heuristic
- A* search with set level heuristic

It is difficult to choose a more concise set of algorithms as, when considering the other criteria (time and space complexity) and each algorithm appears to perform better in one criterion and worse in another.

Appendix – Summary of results

Problem 1

Search number	Actions	Expansions	Goal Tests	New Nodes	Plan length	Search duration
1	20	43	56	178	6	0.020
2	20	21	22	84	20	0.005
3	20	60	62	240	6	0.014
4	20	7	9	29	6	0.002
5	20	6	8	28	6	0.253
6	20	6	8	24	6	0.088
7	20	6	8	28	6	0.617
8	20	50	52	206	6	0.011
9	20	28	30	122	6	0.214
10	20	43	45	180	6	0.122
11	20	33	35	138	6	0.335

Problem 2

Search number	Actions	Expansions	Goal Tests	New Nodes	Plan length	Search duration
1	72	3343	4609	30503	9	0.265
2	72	624	625	5602	619	0.373
3	72	5154	5156	46618	9	0.486
4	72	17	19	170	9	0.009
5	72	9	11	86	9	0.731
6	72	27	29	249	9	0.452
7	72	9	11	84	9	2.067
8	72	2467	2469	22522	9	0.500
9	72	357	359	3426	9	7.398
10	72	2887	2889	26594	9	42.871
11	72	1037	1039	9605	9	92.857

Problem 3

Search number	Actions	Expansions	Goal Tests	New Nodes	Plan length	Search duration
1	88	14663	18098	129625	12	0.691
2	88	408	409	3364	392	0.176
3	88	18510	18512	161936	12	1.205
4	88	25	27	230	15	0.011
5	88	14	16	126	14	1.139
6	88	21	23	195	13	0.674
7	88	35	37	345	17	7.367
8	88	7388	7390	65711	12	0.899
9	88	369	371	3403	12	12.824
10	88	9580	9582	86312	12	229.268
11	88	3423	3425	31596	12	506.524

Problem 4

Search number	Actions	Expansions	Goal Tests	New Nodes	Plan length	Search duration
1	104	99736	114953	944130	14	3.591
2	104	25174	25175	228849	24132	616.263
3	104	113339	113341	1066413	14	5.469
4	104	29	31	280	18	0.028
5	104	17	19	165	17	1.588
6	104	56	58	580	17	1.837
7	104	107	109	1164	23	29.024
8	104	34330	34332	328509	14	3.105
9	104	1208	1210	12210	15	70.086
10	104	62077	62079	599376	14	2305.438
11	104	22606	22608	224229	14	5760.521