UDACITY: AIRBUS – Artificial Intelligence Nanodegree

Laurent Ferhi - 2022/03/20

All algorithms ran on my local machine using pypy3
Graphs have been made with TIBCO Spotfire

1. Air Cargo Problem 1

Nb	Search function	Actions	Expansions	Goal tests	New nodes	Plan length	Elapsed time (s)
1	breadth first search	20	43	56	178	6	0,0424
2	depth first graph search	20	21	22	84	20	0,0096
3	uniform cost search	20	60	62	240	6	0,0330
4	greedy best first graph search h_unmet goals	20	7	9	29	6	0,0045
5	greedy best first graph search h_pg levelsum	20	6	8	28	6	0,8313
6	greedy best first graph search h_pg maxlevel	20	6	8	24	6	0,2786
7	greedy best first graph search h_pg setlevel	20	6	8	28	6	0,6976
8	astar search h_unmet goals	20	50	52	206	6	0,0280
9	astar search h_pg levelsum	20	28	30	122	6	0,3769
10	astar search h_pg maxlevel	20	43	45	180	6	0,3269
11	astar search h_pg setlevel	20	33	35	138	6	0,5954

2. Air Cargo Problem 2

Nb	Search function	Actions	Expansions	Goal tests	New nodes	Plan length	Elapsed time (s)
1	breadth first search	72	3343	4609	30503	9	0,5864
2	depth first graph search	72	624	625	5602	619	0,8372
3	uniform cost search	72	5154	5156	46618	9	1,0048
4	greedy best first graph search h_unmet goals	72	17	19	170	9	0,0243
5	greedy best first graph search h_pg levelsum	72	9	11	86	9	2,6019
6	greedy best first graph search h_pg maxlevel	72	27	29	249	9	1,9741
7	greedy best first graph search h_pg setlevel	72	9	11	84	9	2,6021
8	astar search h_unmet goals	72	2467	2469	22522	9	1,1972
9	astar search h_pg levelsum	72	357	359	3426	9	26,2229
10	astar search h_pg maxlevel	72	2887	2889	26594	9	140,0579
11	astar search h_pg setlevel	72	1037	1039	9605	9	137,3126

DFS can be excluded due to its plan length which is far less optimal than all other algorithm.

3. Air Cargo Problem 3

Nb	Search function	Actions	Expansions	Goal tests	New nodes	Plan length	Elapsed time (s)
1	breadth first search	88	14663	18098	129625	12	1,5628
3	uniform cost search	88	18510	18512	161936	12	2,4186
4	greedy best first graph search h_unmet goals	88	25	27	230	15	0,0425
5	greedy best first graph search h_pg levelsum	88	14	16	126	14	3,5438
6	greedy best first graph search h_pg maxlevel	88	21	23	195	13	4,3777
7	greedy best first graph search h_pg setlevel	88	35	37	345	17	9,7366
8	astar search h_unmet goals	88	7388	7390	65711	12	2,3452
9	astar search h_pg levelsum	88	369	371	3403	12	37,0778
10	astar search h_pg maxlevel	88	9580	9582	86312	12	636,1481

4. Air Cargo Problem 4

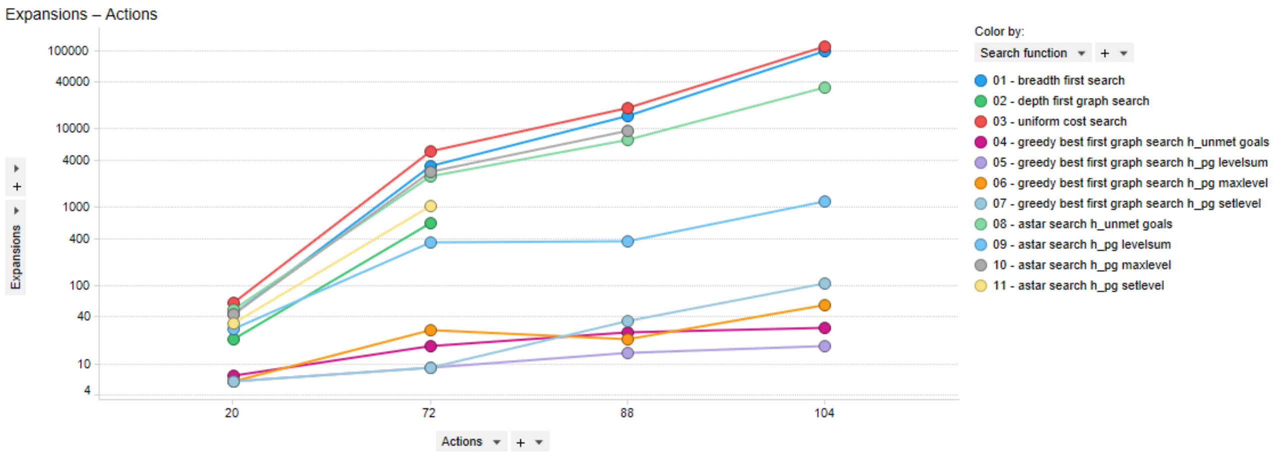
Nb	Search function	Actions	Expansions	Goal tests	New nodes	Plan length	Elapsed time (s)
1	breadth first search	104	99736	114953	944130	14	8,8291
3	uniform cost search	104	113339	113341	1066413	14	14,2938
4	greedy best first graph search h_unmet goals	104	29	31	280	18	0,0503
5	greedy best first graph search h_pg levelsum	104	17	19	165	17	5,2873
6	greedy best first graph search h_pg maxlevel	104	56	58	580	17	10,8176
7	greedy best first graph search h_pg setlevel	104	107	109	1164	23	38,5447
8	astar search h_unmet goals	104	34330	34332	328509	14	8,6031
9	astar search h_pg levelsum	104	1208	1210	12210	15	213,5832

Results Analysis:

- Node expansion:

As the number of actions grows, the number of expanded nodes increases :

Note : Expansions in log scale

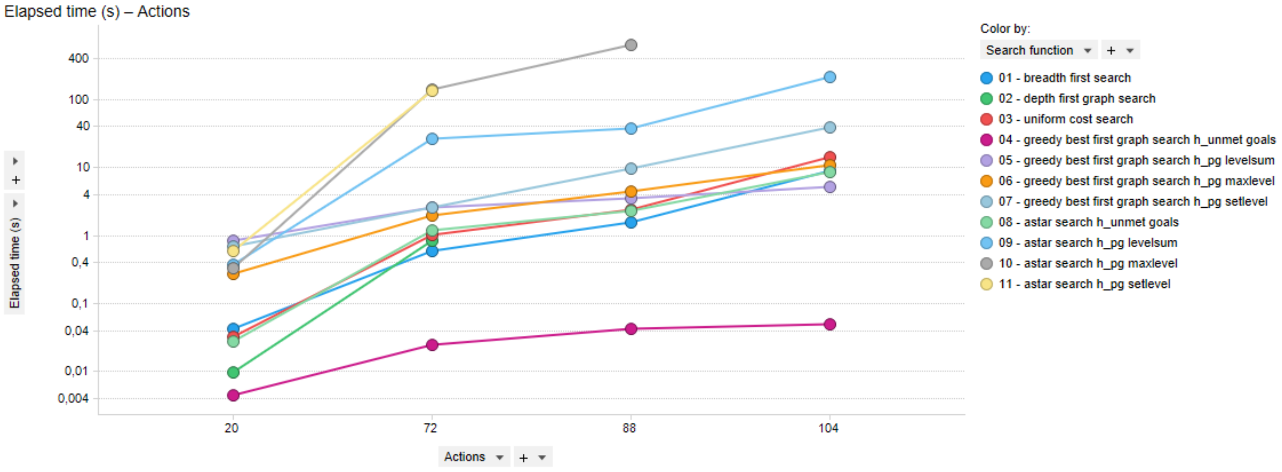


Greedy best first graph search algorithms have lower node expansion and also lower expansion increase with regards to number of actions (which makes them less greedy in terms of memory). In the other hand, BFS and A* methods expansion growth increases dramaticaly with action number.

- *Elapsed time:*

As the number of actions grows, the elapsed time increases:

Note : Elapsed time in log scale



For Greedy best first graph search algorithm with unmet golas, the elapsed time increases with actions almost linearily. Uninformed search and A* algorithm time versus action number tends to dramatically increase for a large domain.

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?*

In this case, we want to optimize time and with few actions, we are in a similar case than air cargo problem 1. The quickest algorithm to solve a few action problem is **greedy best first graph search h_unmet goals**.

- *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)?*

In this case, we want to find a an optimal solution for a large action number (similar to air cargo problem 4). **BFS** and **A* with unmet goals** give the most optimal plan an run in very reasonnable time.

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

If we want to find the optimal solution without worrying about time or memory usage, again **BFS** and **A* with unmet goals** seem to be the best solutions.