UDACITY - Artificial Intelligence Nanodegree Program

Project 2:Build a Forward Planning Agent

(AHMET ÇAĞRI AKCA)

Tables and Charts of problems

Table 1 - Air Cargo Problem 1

Algorithm	Actions	Expansions	Goal Tests	New Nodes	Plan Length	Elapsed Time
1. breadth_first_search	20	43	56	178	6	0,018326
2. depth_first_graph_search	20	21	22	84	20	0,007792
3. uniform_cost_search	20	60	62	240	6	0,023910
4. greedy_best_first_graph_search h_unmet_goals	20	7	9	29	6	0,004491
5. greedy_best_first_graph_search h_pg_levelsum	20	6	8	28	6	0,167809
6. greedy_best_first_graph_search h_pg_maxlevel	20	6	8	24	6	0,141760
7. greedy_best_first_graph_search h_pg_setlevel	20	6	8	28	6	0,382838
8. astar_search h_unmet_goals	20	50	52	206	6	0,025504
9. astar_search h_pg_levelsum	20	28	30	122	6	0,296376
10. astar_search h_pg_maxlevel	20	43	45	180	6	0,305936
11. astar_search h_pg_setlevel	20	33	35	138	6	0,606172

Table 2 - Air Cargo Problem 2

Algorithm	Actions	Expansions	Goal Tests	New Nodes	Plan Length	Elapsed Time
1. breadth_first_search	72	3343	4609	30503	9	0,277335
2. depth_first_graph_search	72	624	625	5602	619	0,491092
3. uniform_cost_search	72	5154	5156	46618	9	0,550483
4. greedy_best_first_graph_search h_unmet_goals	72	17	19	170	9	0,029929
5. greedy_best_first_graph_search h_pg_levelsum	72	9	11	86	9	0,530363
6. greedy_best_first_graph_search h_pg_maxlevel	72	27	29	249	9	0,688341
7. greedy_best_first_graph_search h_pg_setlevel	72	9	11	84	9	1,380927
8. astar_search h_unmet_goals	72	2467	2469	22522	9	0,669368
9. astar_search h_pg_levelsum	72	357	359	3426	9	4,705144
10. astar_search h_pg_maxlevel	72	2887	2889	26594	9	24,824134
11. astar_search h_pg_setlevel	72	1037	1039	9605	9	53,510848

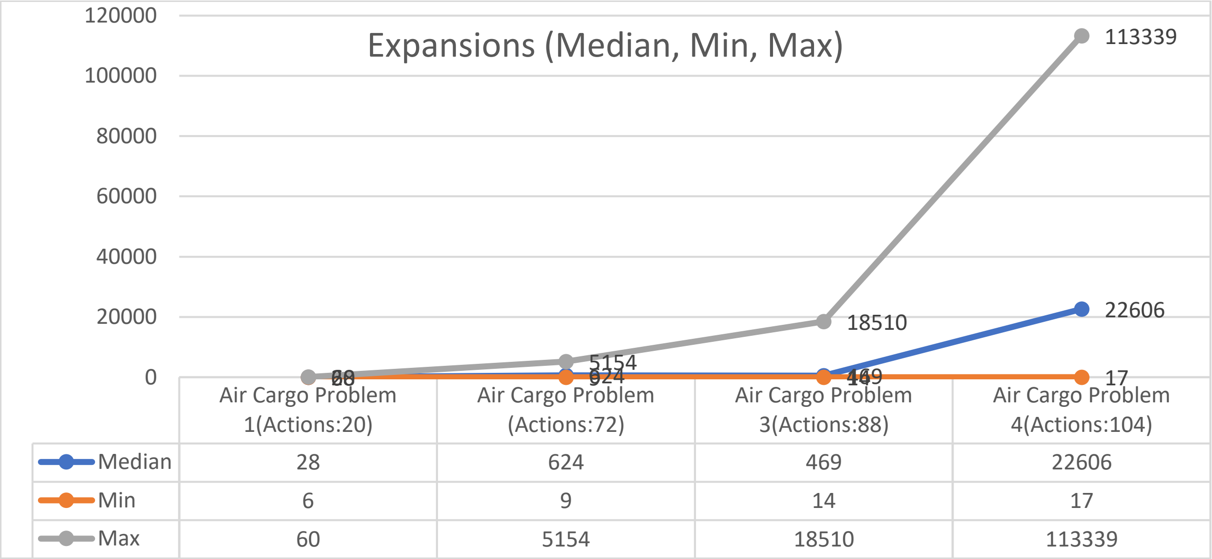
Table 3 - Air Cargo Problem 3

Algorithm	Actions	Expansions	Goal Tests	New Nodes	Plan Length	Elapsed Time
1. breadth_first_search	88	14663	18098	129625	12	0,804248
2. depth_first_graph_search	88	408	409	3364	392	0,250365
3. uniform_cost_search	88	18510	18612	161936	12	1,308845
4. greedy_best_first_graph_search h_unmet_goals	88	25	27	230	15	0,035850
5. greedy_best_first_graph_search h_pg_levelsum	88	14	16	126	14	0,915068
6. greedy_best_first_graph_search h_pg_maxlevel	88	21	23	195	13	0,971827
7. greedy_best_first_graph_search h_pg_setlevel	88	35	37	345	17	4,113836
8. astar_search h_unmet_goals	88	7388	7390	65711	12	1,017751
9. astar_search h_pg_levelsum	88	469	371	3403	12	8,587879
10. astar_search h_pg_maxlevel	88	9580	9582	86312	12	152,922367
11. astar_search h_pg_setlevel	88	3.423	3425	31596	12	280,617203

Table 4 - Air Cargo Problem 4

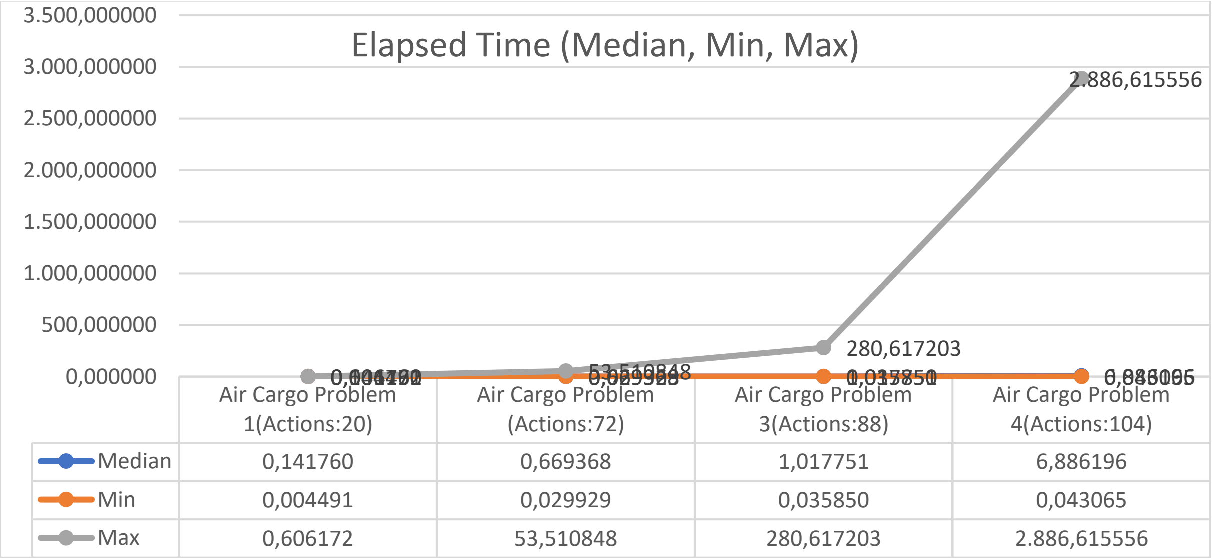
Algorithm	Actions	Expansions	Goal Tests	New Nodes	Plan Length	Elapsed Time
1. breadth_first_search	104	99736	114953	944130	14	4,216335
2. depth_first_graph_search	104	25174	25175	228849	24132	683,574191
3. uniform_cost_search	104	113339	113341	1066413	14	6,886196
4. greedy_best_first_graph_search h_unmet_goals	104	29	31	280	18	0,043065
5. greedy_best_first_graph_search h_pg_levelsum	104	17	19	165	17	1,149227
6. greedy_best_first_graph_search h_pg_maxlevel	104	56	58	580	17	1,789240
7. greedy_best_first_graph_search h_pg_setlevel	104	107	109	1164	14	14,930886
8. astar_search h_unmet_goals	104	34330	34332	328509	14	4,131556
9. astar_search h_pg_levelsum	104	1208	1210	12210	15	42,369601
10. astar_search h_pg_maxlevel	104	62077	62079	599376	14	1435,042835
11. astar_search h_pg_setlevel	104	22606	22608	224229	14	2886,615556

Chart 1 - Chart of nodes expanded



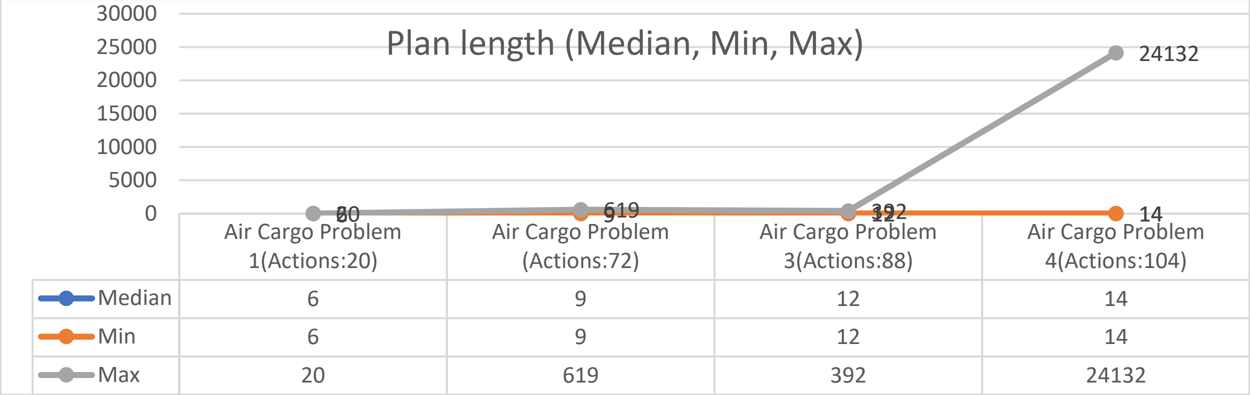
	Air Cargo Problem 1(Actions:20)	Air Cargo Problem (Actions:72)	Air Cargo Problem 3(Actions:88)	Air Cargo Problem 4(Actions:104)
Median	28	624	469	22606
Min	6	9	14	17
Max	60	5154	18510	113339

Chart 2 - Chart of the search time



	Air Cargo Problem 1(Actions:20)	Air Cargo Problem (Actions:72)	Air Cargo Problem 3(Actions:88)	Air Cargo Problem 4(Actions:104)
Median	0,141760	0,669368	1,017751	6,886196
Min	0,004491	0,029929	0,035850	0,043065
Max	0,606172	53,510848	280,617203	2.886,615556

Chart 3 - Chart of the length of the plans



	Air Cargo Problem 1(Actions:20)	Air Cargo Problem (Actions:72)	Air Cargo Problem 3(Actions:88)	Air Cargo Problem 4(Actions:104)
Median	6	9	12	14
Min	6	9	12	14
Max	20	619	392	24132

In the report, Table 1-4 was created to examine the details of the processes. These tables are also listed for all cases.

When the graphics are examined, it is observed that as the number of actions in the problem increases, we observe that the results increase exponentially with the number of actions. This shows that as the problem space expands, it will be more difficult to reach the result. Expanded nodes, elapsed time and plan length increase as the number of actions increases, of course, will be seen on the chart.

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?***

Algorithm 4 (Greedy_best_first_graph_search h_unmet_goals) offers a quick solution to all problems. If we have restricted space and perhaps need to offer solutions in real time, it makes sense to use algorithm 4 for the fastest solution.

- ***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 very large domains, the number of actions will increase and, accordingly, the number of nodes will increase exponentially. To find a quick solution to this situation, it is necessary to choose fast algorithms. Therefore, greedy approach algorithms can be preferred. Even if the plan length is increased for these algorithms, it is reasonable to use our preference for these algorithms since the search time is critical. Based on these explanations, algorithm4 (greedy_best_first_graph_search h_unmet_goals), algorithm5(greedy_best_first_graph_search h_pg_levelsum), algorithm 6(greedy_best_first_graph_search h_pg_maxlevel) and A-star algorithm (astar_search h_unmet_goals), which we observe as close to speed, can be preferred.

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

If we do not have time limits and need an optimal solution, it would be correct to choose the a-star algorithms. It seems reasonable to choose algorithm 11(astar_search h_pg_setlevel) for the best solution.