AIND Heuristic Analysis

AIND Project 3

Summary

On the whole, it seems there is a tendency that heuristic-based search methods are somehow better.

First, in terms of breadth-first search, as explained in AIMA text, P.91 Figure 3.4.7 and P.108 3.7 Summary, we can see its feature of exponential space complexity as the problem becomes more difficult(Figure 4). As in Figure 2 and Figure 3, breadth_first_tree_search could not finish its searching even within 30 minutes, whereas most of other search methods were able to finish their jobs within 10 minutes.

On the other hand, it seems that space complexity of depth_first_graph_search is more like linear as in the same summary in the textbook(Figure 5). In relation, each Time Elapsed of this search method is excellent; however, just as explained in the same summary above of the textbook, this search method has a crucial disadvantage that is not always to get completed or able to reach an optimal plan, which let us not choose results as optimal.

Third, about uniform_cost_search, as in the explanation of the textbook's summary, its feature is expressed as "expands the node with lowest path cost, and is optimal for general step costs". In fact, seeing the result as in Figure 1 to Figure 3, its results are comparatively better especially in Plan Length and Time Elapsed among other non-heuristic search methods.

Compared with these non-heuristic search methods, heuristic search methods seem to have better results as planning problems become more difficult. Watching only the results of Plan Length and Time Elapsed which are considered to be more important measurements, astar_search h1, astar_search h_ignore_preconditions, astar_search h_pg_level_sum are thought to be more effective. But with other measurements, astar_search_h1 is inferior to the other two methods because it needs more space.

Between the remaining two, astar_search h_ignore_preconditions has an excellent score in Time Elapsed, while it needs more cost in Expansions, Goal Tests, and New Nodes. This may stem from its feature that it may search nodes which lead to multiple goals(AIMA text P.376 10.2.3). About astar_search h_pg_levelsum, it needs more Time Elapsed, but it seems to need less space with its search effectiveness.

Optimal Plans

1. Program 1

the result of greedy_best_first_graph_search h_1

Load(C1, P1, SFO)
Load(C2, P2, JFK)
Fly(P1, SFO, JFK)
Fly(P2, JFK, SFO)
Unload(C1, P1, JFK)
Unload(C2, P2, SFO)

Plan length is one of the shortest and in any other aspects there needed the smallest cost.
Ref. Figure 1, the green row

2. Program 2

the result of astar_search h_ignore_preconditions

Load(C1, P1, SFO)
Fly(P1, SFO, JFK)
Unload(C1, P1, JFK)
Load(C2, P2, JFK)
Fly(P2, JFK, SFO)
Unload(C2, P2, SFO)
Load(C3, P3, ATL)
Fly(P3, ATL, SFO)
Unload(C3, P3, SFO)

Indeed there needed more Expansions, Goal Tests, and New Nodes compared with those of astar_search h_pg_levelsum, depth_first_graph_search, and greedy_best_first_graph_search h_1, Time Elapsed is the smallest among whose Plan lengths are only 9.
Ref. Figure 2, the green row

3. Program 3

the result of astar_search h_ignore_preconditions

Load(C1, P1, SFO)
Fly(P1, SFO, ATL)
Load(C3, P1, ATL)
Fly(P1, ATL, JFK)
Unload(C1, P1, JFK)
Load(C2, P2, JFK)
Fly(P2, JFK, ORD)
Load(C4, P2, ORD)
Fly(P2, ORD, SFO)
Unload(C2, P2, SFO)
Unload(C3, P1, JFK)
Unload(C4, P2, SFO)

Again, though some measurements are comparatively higher than some of the others, Plan Length is only 12 and its Time Elapsed is the smallest among those whose Plan Length is 12 as well.
Ref. Figure 3, the green row

Figure 1

Program 1						
No	Name	Expansions	Goal Tests	New Nodes	Plan Length	Time Elapsed approx.
1	breadth_first_search	43	56	180	6	0.02683
2	breadth_first_tree_search	1458	1459	5960	6	0.81514
3	depth_first_graph_search	12	13	48	12	0.00777
4	depth_limited_search	101	271	414	50	0.07479
5	uniform_cost_search	55	57	224	6	0.03148
6	recursive_best_first_search h_1	4229	4230	17029	6	2.32310
7	greedy_best_first_graph_search h_1	7	9	28	6	0.00553
8	astar_search h_1	55	57	224	6	0.03323
9	astar_search h_ignore_preconditions	41	43	170	6	0.03239
10	astar_search h_pg_levelsum	11	13	50	6	0.75647

Figures

Figure 2

Program 2						
No	Name	Expansions	Goal Tests	New Nodes	Plan Length	Time Elapsed approx.
1	breadth_first_search	3401	4672	31049	9	12.56060
2	breadth_first_tree_search	-	-	-	-	-
3	depth_first_graph_search	350	351	3142	346	1.32355
4	depth_limited_search	254020	2344879	2345254	50	913.44675
5	uniform_cost_search	4852	4854	44030	9	10.61929
6	recursive_best_first_search h_1	-	-	-	-	-
7	greedy_best_first_graph_search h_1	990	992	8910	17	2.13099
8	astar_search h_1	4852	4854	44030	9	10.85181
9	astar_search h_ignore_preconditions	1450	1452	13303	9	3.79097
10	astar_search h_pg_levelsum	86	88	841	9	158.30557

Figure 3

Program 3						
No	Name	Expansions	Goal Tests	New Nodes	Plan Length	Time Elapsed approx.
1	breadth_first_search	14663	18098	129631	12	93.34923
2	breadth_first_tree_search	-	-	-	-	-
3	depth_first_graph_search	627	628	5176	596	2.93549
4	depth_limited_search	-	-	-	-	-
5	uniform_cost_search	18151	18153	159038	12	48.97772
6	recursive_best_first_search h_1	-	-	-	-	-
7	greedy_best_first_graph_search h_1	5398	5400	47665	26	13.73761
8	astar_search h_1	18151	18153	159038	12	49.69814
9	astar_search h_ignore_preconditions	5038	5040	44926	12	15.76047
10	astar_search h_pg_levelsum	314	316	2894	12	854.63447

Figure 4

breadth_first_search -> Exponential Space Complexity					
Program	Expansions	Goal Tests	New Nodes	Plan Length	Time Elapsed approx.
1	43	56	180	6	0.02683
2	3401	4672	31049	9	12.56060
3	14663	18098	129631	12	93.34923

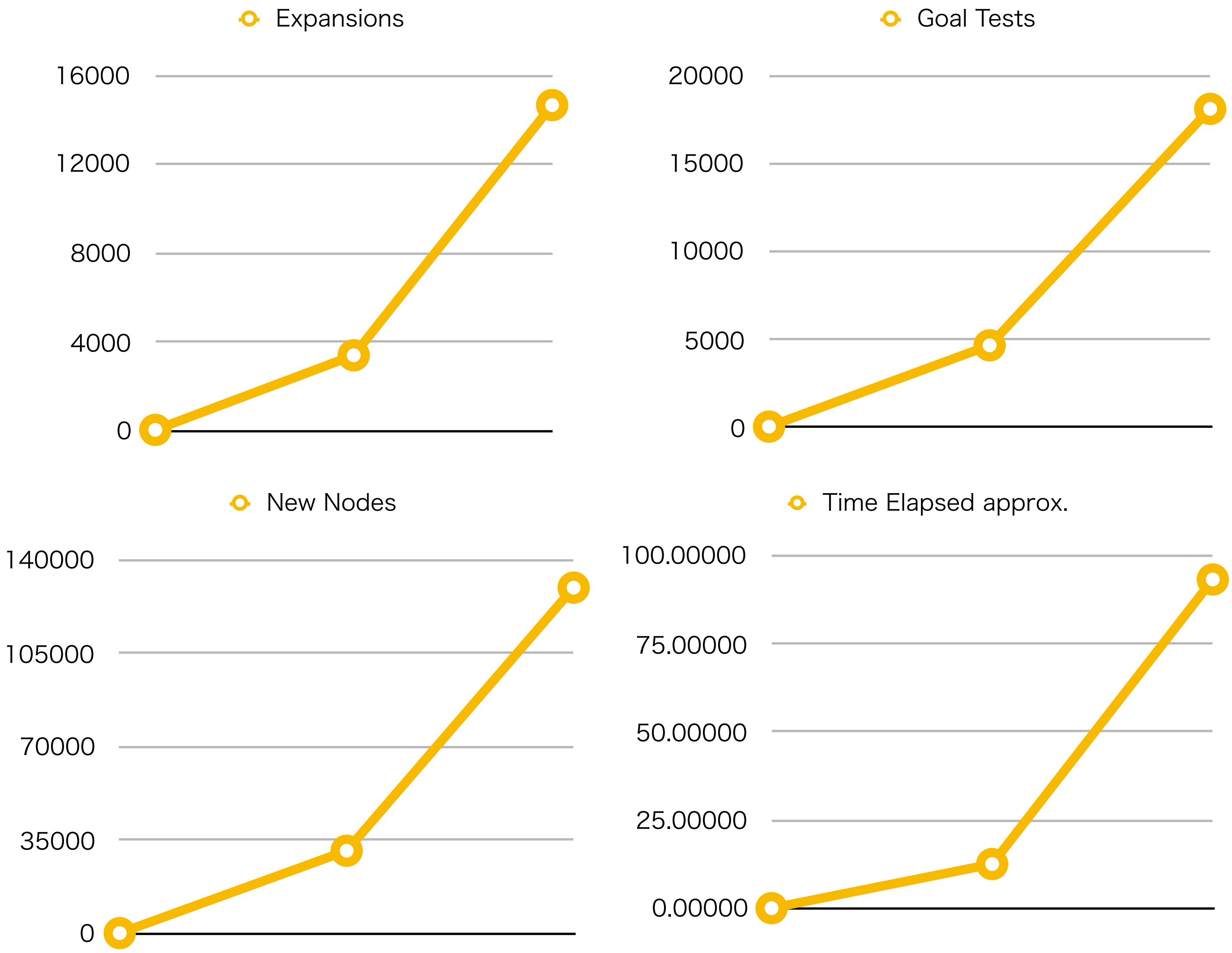

Figure 5

depth_first_graph_search -> Linear Space Complexity					
Program	Expansions	Goal Tests	New Nodes	Plan Length	Time Elapsed approx.
1	12	13	48	12	0.00777
2	350	351	3142	346	1.32355
3	627	628	5176	596	2.93549