heuristic_analysis.md

Heuristic Analysis

This document deals with the analysis of solutions to various planning problems provided, in terms of their number of goal states, plan length and time elapsed.

Air Cargo Problem 1

Problem 1 is an easy planning problem to solve. Each search algorithm was able to find an goal state in a short amount of time.

Algorithm	Expansions	Goal Tests	New Nodes	Plan Length	Elapsed Time (s)
breadth_first_search	43	56	180	6	0.032115
breadth_first_tree_search	1458	1459	5960	6	0.962014
depth_first_graph_search	12	13	48	12	0.008378
depth_limited_search	101	271	414	50	0.100066
uniform_cost_search	55	57	224	6	0.037079
recursive_best_first_search	4229	4230	17029	6	2.948969
greedy_best_first_graph_search	7	9	28	6	0.004461
astar_search_with_h1	55	57	224	6	0.037982
astar_search_with_h_ignore_preconditions	41	43	170	6	0.035665
astar_search_with_h_pg_levelsum	56	58	228	6	1.956162

- Greedy best first graph search took the shortest amount of time among the non-heuristic searches. It also had the least amount of expansions, goal tests, new nodes, plan length and time elapsed among the non-heuristic ones.
- In the case of heuristic search, A* search with ‘h_ignore_preconditions’ took the shortest amount of time, and had less expansions than A* with ‘h_1’ or 'h_pg_levelsum'

Optimal Plan:

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

Air Cargo Problem 2

Problem 2 is a harder problem than Problem 1.

Algorithm	Expansions	Goal Tests	New Nodes	Plan Length	Elapsed Time (s)
breadth_first_search	3343	4609	30509	9	17.097935
breadth_first_tree_search	NA	NA	NA	NA	Timeout
depth_first_graph_search	582	583	5211	575	3.058725
depth_limited_search	NA	NA	NA	NA	Timeout
uniform_cost_search	4853	4855	44041	9	12.842941
recursive_best_first_search	NA	NA	NA	NA	Timeout
greedy_best_first_graph_search	998	1000	8982	17	4.056419
astar_search_with_h1	4853	4855	44041	9	16.033788
astar_search_with_h_ignore_preconditions	1450	1452	13303	9	5.388868
astar_search_with_h_pg_levelsum	5575	5577	50355	9	1384.221903

- Timeout was encountered for Breadth first tree search, Depth limited search, and Recursive best first search.
- Again, greedy best first graph search outperforms the other non-heuristic searches in terms of time elapsed, though it had a higher plan length and more expansions compared to heuristic functions.
- Again, A* search with ‘h_ignore_preconditions’ had the least time elapsed compared to the other heuristic searches.

Optimal Plan:

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

Air Cargo Problem 3

Algorithm	Expansions	Goal Tests	New Nodes	Plan Length	Elapsed Time (s)
breadth_first_search	14663	18098	129631	12	124.148043
breadth_first_tree_search	NA	NA	NA	NA	Timeout
depth_first_graph_search	627	628	5176	596	3.797029
depth_limited_search	NA	NA	NA	NA	Timeout
uniform_cost_search	18223	18225	159618	12	67.956051
recursive_best_first_search	NA	NA	NA	NA	Timeout
greedy_best_first_graph_search	5578	5580	49150	22	24.628417
astar_search_with_h1	18223	18225	159618	12	75.887129

Algorithm	Expansions	Goal Tests	New Nodes	Plan Length	Elapsed Time (s)
astar_search_with_h_ignore_preconditions	5040	5042	44944	12	23.174203
astar_search_with_h_pg_levelsum	18955	18404	166171	12	4706.37442

- Again, timeout was encountered for Breadth First Tree Search, Depth Limited Search, and Recursive Best First Search.
- Though Depth first graph search has the least time elapsed in the non-heuristic searches, the plan length is huge compared to the others. Hence, again we choose the Greedy best first graph search algorithm as the optimal algorithm, as its plan length is less and the time elapsed is also less.
- Among the non-heuristic searches, the A* search with ‘h_ignore_preconditions’ had the least time elapsed compared to the other heuristic searches.

Optimal Plan:

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

Conclusions

- Overall, for more complex problems the non-heuristic functions do not perform as well as the heuristic ones in terms of plan length or time elapsed.
- Though the A* search with 'pg_levelsum' is efficient, it performs poorly in terms of time elapsed compared to our other algorithms.
- Chapter 10 of AIMA states: “An admissible heuristic can be derived by defining a relaxed problem that is easier to solve. The exact cost of a solution to this easier problem then becomes the heuristic for the original problem.” A* search with ignoring preconditions relaxes the problem and doesn't overestimate the goal, hence it performed better than 'h_1' or 'pg_levelsum'.

References

Russell, Stuart J., et al. Artificial Intelligence: a Modern Approach. Prentice Hall, 2016.