HEURISTIC_ANALYSIS

Project 3: Implement a planning search

FEBUARY 7TH, 2018

SUBMITED BY: MOHAMED AYMAN NAGUIB

Contents

Problem 1

Results:

As shown in Table 1, Breadth first search, breadth first tree search, uniform cost search, recursive best first search and greedy best first graph search provided the optimal plan length of 6, but greedy best first graph search has provided the plan in the minimum number of time (0.27 seconds) and the minimum number of node expansions (7 nodes). Therefore, greedy best first search was by far the "best-optimal" searching technique.

Table 1: Problem 1 Non-Heuristic Comparison

Search Technique	Expansions	Goal Tests	New Nodes	Plan Length	Time	Optimality
Breadth First Search	43	56	180	6	2.157	Yes
Breadth First Tree Search	1458	1459	5960	6	68.7188	Yes
Depth first graph search	12	13	48	12	0.49	No
Depth Limited Search	101	271	414	50	5.867	No
Uniform Cost Search	55	57	224	6	2.493	Yes
Recursive best first search	4229	4230	17029	6	195.9388	Yes
Greedy best first graph search	7	9	28	6	0.27	Yes

As shown in Table 2, the 3 techniques provided the optimal solution of 6, but regarding time needed to finish the task, A* search h_ignore_preconditions was the fastest, although A* search h_pg_levelsum has finished the same task with a bit more time but with less expansions of nodes (Only 11 nodes compared to the 55 nodes and 41 nodes by the other techniques).

Table 2: Problem 1 Heuristic Comparison

Search Technique	Expansions	Goal Tests	New Nodes	Plan Length	Time	Optimality
A* search h_1	55	57	224	6	1.984	Yes
A* search h_ignore_preconditions	41	43	170	6	1.8649	Yes
A* search h_pg_levelsum	11	13	50	6	2.03	Yes

Optimal Plan:

Load(C1, P1, SFO)

Load(C2, P2, JFK)

Fly(P1, SFO, JFK)

Fly(P2, JFK, SFO)

Unload(C1, P1, JFK)

Unload(C2, P2, SFO)

Problem 2

Results:

As shown in Table 3, only breadth first search has achieved the optimal plan length of length 9 and it did it in 350 seconds, were depth first graph search and greedy best first graph search were faster but they did not achieve the optimality of breadth first search.

Table 3: Problem 2 non heuristic comparison

Search Technique	Expansions	Goal Tests	New Nodes	Plan Length	Time	Optimality
Breadth First Search	3343	4609	30509	9	350.552	Yes
Breadth First Tree Search					>600	
Depth first graph search	582	583	5211	575	127.12	No
Depth Limited Search					>600	
Uniform Cost Search					>600	
Recursive best first search					>600	
Greedy best first graph search	550	552	4950	21	118.106	No

As shown in Table 4, the 3 techniques achieved the optimal solution, but A*search h_pg_levelsum achieved it in the lowest number of expansions (86 nodes) and A* search h_ignore_preconditions achieved it in the lowest time (178 seconds).

Table 4: Problem 2 Heuristic Comparison

Search Technique	Expansions	Goal Tests	New Nodes	Plan Length	Time	Optimality
A* search h_1	4761	4763	43206	9	233.325	Yes
A* search h_ignore_preconditions	1450	1452	13303	9	178.045	Yes
A* search h_pg_levelsum	86	88	841	9	204.878	Yes

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)

Problem 3

Results:

As shown below in Table 5, all the techniques exceeded the 10 min limit, but breadth first search was relatively the fastest and achieved the optimal plan length (12) with 14663 nodes expanded.

Table 5: Problem 3 non heuristic comparison

Search Technique	Expansions	Goal Tests	New Nodes	Plan Length	Time	Optimality
Breadth First Search	14663	18098	129631	12	719.889	Yes
Breadth First Tree Search					>600	
Depth first graph search					>600	
Depth Limited Search					>600	
Uniform Cost Search					>600	
Recursive best first search					>600	
Greedy best first graph search					>600	

As shown in Table 6, the 3 techniques achieved the optimal plan length (12), but A* search h_pg_levelsum did it in the lowest number of expansions(325 nodes) and A*search h_ignore_preconditions did it in the lowest time (175 seconds).

Table 6: Problem 3 heuristic comparison

Search Technique	Expansions	Goal Tests	New Nodes	Plan Length	Time	Optimality
A* search h_1	18235	18237	159716	12	1155.045	Yes
A* search h_ignore_preconditions	5040	5042	44944	12	175.8988	Yes
A* search h_pg_levelsum	325	327	3002	12	1367.046	Yes

Optimal Solution:

Load(C2, P2, JFK)

Fly(P2, JFK, ORD)

Load(C4, P2, ORD)

Fly(P2, ORD, SFO)

Unload(C4, P2, SFO)

Load(C1, P1, SFO)

Fly(P1, SFO, ATL)

Load(C3, P1, ATL)

Fly(P1, ATL, JFK)

Unload(C3, P1, JFK)

Unload(C2, P2, SFO)

Unload(C1, P1, JFK)

Conclusion:

The best heuristic used given in consideration that it is the fastest was the A* h_ignore_precondtions, in the 3 problems used in it has finished **the fastest as observed in the videos it relaxes the problem constraints as it ignores preconditions required for an action to be executed to makes the problem easier in order to estimate the minimum number of actions that must be carried out from the current state in order to satisfy all of the goal conditions (in terms of time or expansions) [1].** On the other hand if the main concern is the number of expansions and that finishing a bit late is not a limitation then A* h_pg_levesum is the best heuristic as it finishes the search with lowest number of expansions by far. In problem 1 the heuristics did not do better than the non-heuristic search techniques as the problem size was not that big, although in the rest of the problems the heuristics did by far better than the non-heuristics as they either time out or provide a non-optimal solution. In my opinion the best heuristic used in these problems was the ignore_precondtions as it provided the optimum solution for all the problems in the minimum time and the number of expansions was not that enormous compared to the other non-heuristic techniques.

References:

[1] AIND Lesson 11 Planning