Air Cargo Planning Heuristic Analysis

Test Results

Air cargo planning search was run for each problem set and the results are given below. The optimal solutions are marked with yellow highlights.

The table below gives the result for problem set 1

Search Type	Expansions	Goal Tests	New Nodes	Plan Length	Elapsed Time (s)
Breadth First Search	43	56	180	6	0.02524
Breadth First Tree Search	1458	1459	5960	6	0.755732638
Depth First Graph Search	21	22	84	20	0.01363713
Depth Limited Search	101	271	414	50	0.07997
Uniform Cost Search	55	57	224	6	0.0276881
Recursive Best First Search h_1	4229	4230	17023	6	2.0938
Greedy Best **First** Graph Search h_1	7	9	28	6	0.0036352699
A* Search h_1	55	57	224	6	0.029
A* Search h_ignore_preconditions	41	43	170	6	0.03
A* Search h_pg_levelsum	11	13	50	6	0.974229

Below is the result for problem set 2. Search types Breadth First Tree Search, Depth Limited Search, Recursive Best First Search with h_1 and A* Search with h_pg_levelsum took too long to complete and had to be manually terminated.

Search Type	Expansions	Goal Tests	New Nodes	Plan Length	Elapsed Time (s)
Breadth First Search	3343	4609	30509	9	11.435519
Breadth First Tree Search	N/A	N/A	N/A	N/A	Too long
Depth First Graph Search	624	625	5602	619	2.82175
Depth Limited Search	N/A	N/A	N/A	N/A	Too Long
Uniform Cost Search	4853	4855	44041	9	9.174799
Recursive Best First Search h_1	N/A	N/A	N/A	N/A	Too Long

Greedy Best First Graph Search h_1	998	1000	8982	15	1.8444889
A* Search h_1	4853	4855	44041	9	9.22
A* Search h_ignore_preconditions	1450	1452	13303	9	3.34161512
A* Search h_pg_levelsum	N/A	N/A	N/A	N/A	Too Long

Below is the result for problem set 3. Search types Breadth First Tree Search, Depth Limited Search, Recursive Best First Search with h_1, A* Search with h_1 and A* Search with h_pg_levelsum took too long to complete and had to be manually terminated.

Search Type	**Expansions**	**Goal Tests**	**New Nodes**	**Plan Length**	**Elapsed Time (s)**
Breadth First Search	14663	18098	129631	12	82.59431
Breadth First Tree Search	N/A	N/A	N/A	N/A	Too Long
Depth First Graph Search	408	409	3364	392	1.5093
Depth Limited Search	N/A	N/A	N/A	N/A	Too Long
Uniform Cost Search	18223	18225	159618	12	40.8363
Recursive Best First Search h_1	N/A	N/A	N/A	N/A	Too Long
Greedy Best First Graph Search h_1	5578	5580	49150	22	12.41854
A* Search h_1	N/A	N/A	N/A	N/A	Too Long
A* Search h_ignore_preconditions	5040	5042	44944	12	13.745888
A* Search h_pg_levelsum	N/A	N/A	N/A	N/A	Too Long

Optimal Sequence

The table below gives optimal sequence for each problem

Problem	**Optimal Search Type**	**Sequence**
Problem Set 1	Greedy Best First Graph Search with 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)
Problem set 2	A* Search h_ignore_preconditions	Load(C3, P3, ATL) Fly(P3, ATL, SFO) Unload(C3, P3, SFO) Load(C2, P2, JFK) Fly(P2, JFK, SFO) Unload(C2, P2, SFO)

		Load(C1, P1, SFO) Fly(P1, SFO, JFK) Unload(C1, P1, JFK)
Problem Set 3	A* Search h_ignore_preconditions	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

For simpler case like problem 1, greedy best first graph search with h_1 gave the most optimal solution, with fewest expansions and taking least time, followed by breadth first search. As complexity increased in problem 2 and 3 A*Search with h_ignore_preconditions gave much better solution than breadth first search.

Depth first search can find solution quickly, but it lacks optimality, because it does not consider all nodes as mentioned in lesson 10 of video lessons about search comparison. In problem 3 we saw depth first search find solution fastest, because luckily fastest path was explored first. But it is not the optimal solution as it has very high plan length. Breadth first search generally always finds optimal solution, because it considers all nodes to find the shortest path as mentioned in lesson 10 of video lessons. Greedy best first search with h_1 is able to optimize breadth first search even further by using heuristic. When it comes to heuristic based search it should be noted that A* Search with h_ignore_precondition performs better in more complex scenarios, but A* Search with h_1 and A* Search with h_pg_levelsum don't, because they suffer from increasingly complex heuristic.