Heuristic Analysis – Project 3

Sashank Santhanam

Provide an optimal plan for Problems 1, 2, and 3.

Compare and contrast non-heuristic search result metrics (optimality, time elapsed, number of node expansions) for Problems 1,2, and 3. Include breadth-first, depth-first, and at least one other uninformed non-heuristic search in your comparison; Your third choice of non-heuristic search may be skipped for Problem 3 if it takes longer than 10 minutes to run, but a note in this case should be included.

Analysis

Problem1:

Methods	Plan Length	Expansion	Goal Tests	New Nodes	Time
Breadth First Search	6	43	56	180	0.376
Breadth First Tree Search	6	1458	1459	5960	1.177
Depth first graph search	12	12	13	48	0.0177
Depth Limited Search	50	101	271	404	0.12
Uniform Cost Search	6	55	57	224	0.0464
Recursive BFS	6	4229	4230	17029	3.59
Greedy BFS	6	7	9	28	0.006
A* with H1	6	55	57	224	0.04
A* with ignore preconditions	6	41	43	170	0.04
A* with h_pg_levelsum	6	11	13	50	1.05

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)

Problem2 -

Methods	Plan Length	Expansion	Goal Tests	New Nodes	Time
Breadth First Search	9	3401	4672	31049	17.76
Breadth First Tree Search	Abort	Abort	Abort	Abort	Abort
Depth first graph search	346	350	351	3142	1.87
Depth Limited Search	Abort	Abort	Abort	Abort	Abort
Uniform Cost Search	9	4853	4855	44041	15.15
Recursive BFS	Abort	Abort	Abort	Abort	Abort
Greedy BFS	21	998	1000	8982	2.95
A* with H1	9	4953	4855	44041	14.43
A* with ignore preconditions	9	1450	1452	13303	5.15
A* with h_pg_levelsum	9	86	88	841	240

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

Problem3 -

Methods	Plan Length	Expansion	Goal Tests	New Nodes	Time
Breadth First Search	12	14663	18098	129631	164.9
Breadth First Tree Search	Abort	Abort	Abort	Abort	Abort
Depth first graph search	596	617	628	5176	4.292
Depth Limited Search	Abort	Abort	Abort	Abort	Abort
Uniform Cost Search	12	18223	18225	159618	65.1786
Recursive BFS	Abort	Abort	Abort	Abort	Abort
Greedy BFS	22	5578	5580	49150	19.77
A* with H1	12	18223	18225	159618	63.45
A* with ignore preconditions	12	5040	5042	44944	20.63
A* with h_pg_levelsum	Abort	Abort	Abort	Abort	Abort

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)

The above 3 tables shows the plan length, Expansion, Goal test, New Nodes and the time taken to complete the problem. The optimal plan length for the **problem 1 is 6, problem 2 is 9 and problem 3 is 12.** Comparing problems 1, 2 and 3 we find that Breadth First Search, Uniform Cost Search, A * with H1 and Ignore preconditions are the algorithms across the 3 problems that provide the optimal solution of 6, 9 and 12 respectively.

Comparing the 4 algorithms with regards to the problem 1, **we find that the A* with ignore preconditions performs marginally better than the other 3 algorithms in terms of new nodes, goal tests and expansions. The performance over time was similar.**

Comparing the 4 algorithms with regards to the problem 2, **we find that the A* with ignore preconditions performs significantly better than the other 3 algorithms in terms of new nodes, goal tests and expansions. A* with ignore pre conditions also is much faster than the other algorithms , So in this case I would prefer A* with ignore preconditions.**

Comparing the 4 algorithms with regards to the problem 2, **we find that the A* with ignore preconditions performs significantly better than the other 3 algorithms in terms of new nodes, goal tests and expansions. A* with ignore pre conditions was 3 times faster than the next closest ones of A* with H1 and Uniform Cost Search**

Across conditions of time, expansions, new nodes and goal tests we find the DFS is optimal but DFS does not provide the Optimal plan. With regards to DFS there is a possibility of the search being stuck in a path for a long time and this causes the algorithms to not be optimal. we find that the A* with ignore preconditions perform much better as it is the simplest way to relax a problem and it makes sures that every action will always be applicable and any literal can be reached in one step[1]. Level sum heuristic depends a lot on the domain of the problem. From the problems given to us, Level sum takes more than 10 mins to run the 3rd problem. So the goal of any problem is to have the best time and optimality. A* with ignore pre conditions provides with the fastest time to get the optimal solution.

References :

1. Stuart Russell and Peter Norvig, Artificial Intelligence A modern approach 2nd edition, Chapter 11.