Heuristic Analysis

Planning Agent
By
Kiran Niranjan

Test Results

Air Cargo Problem 1

Search	Expansions	Goal Tests	New Nodes	Plan Length	Time Elapsed
breadth_first_search	43	56	180	6	0.045719
breadth_first_tree_search	1458	1459	5960	6	1.374822
depth_first_graph_search	12	13	48	12	0.010507
depth_limited_search	101	271	414	50	0.132278
uniform_cost_search	55	57	224	6	0.05467

h_ignore_preconditions	41	43	170	6	0.0548513
h_pg_levelsum	11	13	50	6	0.91161

Result (Breadth First Search)
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

Search	Expansions	Goal Tests	New Nodes	Plan Length	Time Elapsed
breadth_first_search	2307	3364	17183	9	7.094582
breadth_first_tree_search	Taking Long	-	-	-	-
depth_first_graph_search	146	147	481	40	0.216714
depth_limited_search	132336	1007122	1007500	50	674.0974
uniform_cost_search	3588	3590	26114	9	10.39556

h_ignore_preconditions	1078	1080	8295	9	3.863343
h_pg_levelsum	225	227	1483	9	120.98303

Result (Breadth First Search)
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

Search	Expansions	Goal Tests	New Nodes	Plan Length	Time Elapsed
breadth_first_search	14663	18098	129631	12	75.31844
breadth_first_tree_search	Taking Long	-	-	-	-
depth_first_graph_search	627	628	5176	596	5.14673
uniform_cost_search	18151	18153	159038	12	79.4731
h_ignore_preconditions	5038	5040	44926	12	25.29524
h_pg_levelsum	225	227	1483	9	120.98303

Result (Breadth First Search)
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)

Breadth First Search

Breadth-First search is like traversing a tree where each node is a state which may a be a potential candidate for solution. It expands nodes from the root of the tree and then generates one level of the tree at a time until a solution is found.

Breadth first search will never get trapped exploring the useless path forever.

Depth First Search

searches deeper into the problem space. Breadth-first search always generates successor of the deepest unexpanded node. It uses last-in first-out stack for keeping the unexpanded nodes

depth-first search finds solution without exploring much in a path then the time and space it takes will be very less.

Ignore preconditions

Ignore preconditions Heuristic drops all preconditions from operations any goal condition can be achieved in one step

Heuristic Search Analysis and Conclusion

From the results we see from both the Heuristic and Non-Heuristic search algorithm. We can see that the level sum heuristic expands less nodes and takes less goal test to create a significantly less node compare to the ignore heuristic.

compression between A* Search with the ignore precondition heuristic search and the breadth first search A* search executes faster than breadth first search and it uses significantly less resources so A* Search can be chosen based on the type of problem.

So this concludes that A* search can be optimistic heuristic to choose as it executes faster which is the most important thing to consider as it makes much difference

Useful links

1. http://ai.stanford.edu/~nilsson/OnlinePubs-Nils/PublishedPapers/strips.pdf
2. https://www.cs.cmu.edu/~avrim/Papers/graphplan.pdf
3. https://bonetblai.github.io/reports/aips98-competition.pdf
4. https://en.wikipedia.org/wiki/Artificial_intelligence