Jovani Rico

August 7, 2017

Planning Search Heuristic Analysis

AIND Project 3

Optimal Plans

Problem 1	Problem 2	Problem 3
Load(C1, P1, SFO)	Load(C3, P3, ATL)	Load(C2, P2, JFK)
Fly(P1, SFO, JFK)	Fly(P3, ATL, SFO)	Fly(P2, JFK, ORD)
Unload(C1, P1, JFK)	Unload(C3, P3, SFO)	Load(C4, P2, ORD)
Load(C2, P2, JFK)	Load(C1, P1, SFO)	Fly(P2, ORD, SFO)
Fly(P2, JFK, SFO)	Fly(P1, SFO, JFK)	Unload(C4, P2, SFO)
Unload(C2, P2, SFO)	Unload(C1, P1, JFK)	Load(C1, P1, SFO)
	Load(C2, P2, JFK)	Fly(P1, SFO, ATL)
	Fly(P2, JFK, SFO)	Load(C3, P1, ATL)
	Unload(C2, P2, SFO)	Fly(P1, ATL, JFK)
		Unload(C3, P1, JFK)
		Unload(C1, P1, JFK)
		Unload(C2, P2, SFO)

The following tables show the results obtained after implementing the air cargo problems for this project with both uninformed non- heuristic search methods and heuristic based search.

Non-heuristic search planning methods

Air Cargo Problem 1 Results

Search	Expansions	Goal Tests	New Nodes	Plan length	Time	Optimal
Breadth First Search	43	56	180	6	0.0274	Yes
Depth first graph search	21	22	84	20	0.0130	No
Uniform cost search	55	57	224	6	0.0360	Yes

Air Cargo Problem 2 Results

Search	Expansions	Goal Tests	New Nodes	Plan length	Time	Optimal
Breadth First Search	3343	4609	30509	9	12.7549	Yes
Depth first graph search	624	625	5602	619	3.2338	No
Uniform cost search	4852	4854	44030	9	11.0236	Yes

Air Cargo Problem 3 Results

Search	Expansions	Goal Tests	New Nodes	Plan length	Time	Optimal
Breadth First Search	14663	18098	129631	12	100.1683	Yes
Depth first graph search	408	409	3364	392	1.6071	No
Uniform cost search	18235	18237	159716	12	47.9027	Yes

Observations

For this section Uniform cost search was chosen as a third algorithm. According to the table Depth first search was the fastest algorithm but also it was the one that the plan length was greater. This makes Depth first search not an optimal solution.

Breadth First Search and Uniform Cost Search, both of these algorithms reach the goal state with an optimal plan length, but Breadth First Search even if it takes a little more time to reach the goal in some problems, it creates less the nodes.

Heuristic search planning methods

Air Cargo Problem 1 Results Heuristic

Search	Expansions	Goal Tests	New Nodes	Plan length	Time	Optimal
A* search h_1	55	57	224	6	0.0352	Yes
A* search h_ignore_preconditions	41	43	170	6	0.0349	Yes
A* search h_pg_levelsum	11	13	50	6	0.9006	Yes

Air Cargo Problem 2 Results Heuristic

Search	Expansions	Goal Tests	New Nodes	Plan length	Time	Optimal
A* search h_1	4852	4854	44030	9	10.5989	Yes
A* search h_ignore_preconditions	1450	1452	13303	9	3.8560	Yes
A* search h_pg_levelsum	86	88	841	9	152.7983	Yes

Air Cargo Problem 3 Results Heuristic

Search	Expansions	Goal Tests	New Nodes	Plan length	Time	Optimal
A* search h_1	18235	18237	159716	12	47.6969	Yes
A* search h_ignore_preconditions	5040	5042	44944	12	15.2839	Yes
A* search h_pg_levelsum	318	320	2934	12	801.0605	Yes

Observations

For the heuristic algorithms, all o them reach the optimal plan length but each one with different expansion nodes and time. A* search H1 requires more expansion nodes, A* search Ignore Preconditions creates more new nodes but it is fastest one, A* search Level Sum requires less expansions but it is the lowest one. This last one performed poorly compare to the other two in terms of time.

Informed and Uninformed Search Strategies

The search strategies that generate optimal plans are all the three A* Search heuristics, Breadth First Search and Uniform Cost Search. As shown in the tables Depth Fist Search is faster and uses less memory than Uniform Cost Search and Breadth Fist Search, but it doesn't reach the optimal path. Breadth Fist Search uses less memory and creates less new nodes than the Uniform Cost Search.

For the informed search algorithms, A* Search Ignore Preconditions heuristic is the fastest and uses least memory. Therefore the choice is between Breadth First Search and A* Search Ignore Preconditions heuristic.

Conclusion

For uninformed (non-heuristic search), my pick is restricted to uniform_cost, when execution time is critical or breath_first when execution efficiency is critical (least memory used) because those two algorithms reach the goal in the optimal path.

For informed (heuristic search), I recommend A* search Ignore Preconditions when execution time is critical and A* search Level Sum when memory space efficiency is critical.

References

1. Stuart J. Russell, Peter Norvig (2010), Artificial Intelligence: A Modern Approach (3rd Edition).