AIND Artificial Intelligence Nanodegree

Luiz Roma

Air Cargo Planning Heuristic Analysis

The problem is to develop a strategy algorithm that could solve the Air Cargo problem, which consists in planning a departure/arrival situations for planes considering being charged with a cargo located in some specific points (airports)

> In AI (Artificial Intelligence) researches, there are few algorithms for exploring the domain and there were applied in this project some of them: Breadth First Search, Depth First Search, Uniform Cost Search, Recursive Best First Search, Greedy Best First Graph and A* Search

There were 3 different problem (Air Cargo Problem 1, 2 and 3). The differences between them were the pre-conditions and post-conditions, it means, departure airport, cargo conditions, path to be done, delivery, arrival airport. Below the benchmark between all of them - optimal method for the problem is highlighted

AirCargo Problem 1

Search Type	Expansions	Goal Tests	New Nodes	Plan Length	Time (s)
breadth_first_search	43	56	180	6	0.0365
breadth_first_tree_search	1458	1459	5960	6	0.7835
depth_first_graph_search	12	13	48	12	0.0074
depth_limited_search	101	271	414	50	0.1040
uniform_cost_search	55	57	224	6	0.0319
recursive_best_first_search h_1	4229	4230	17029	6	2.2747
greedy_best_first_graph_search h_1	7	9	28	6	0.0059
astar_search h_1	55	57	224	6	0.0306
astar_search h_ignore_preconditions	41	43	170	6	0.0301
astar_search h_pg_levelsum	11	13	50	6	1.3491

Optimal sequence for Air Cargo Problem 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)
```

AirCargo Problem 2

Search Type	Expansions	Goal Tests	New Nodes	Plan Length	Time (s)
breadth_first_search	3401	4672	31049	9	14.5807
breadth_first_tree_search			*timeout*		
depth_first_graph_search	350	351	3142	346	1.5495
depth_limited_search	254020	2344879	2345254	50	1817.414
uniform_cost_search	4780	4782	43381	9	17.8209
recursive_best_first_search h_1			*timeout*		
greedy_best_first_graph_search h_1	598	600	5382	21	3.1722
astar_search h_1	4780	4782	43381	9	23.1549
astar_search h_ignore_preconditions	1450	1452	13303	9	8.4137
astar_search h_pg_levelsum	86	88	841	9	517.6186

Optimal sequence for Air Cargo Problem 2:

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

AirCargo Problem 3

Search Type	Expansions	Goal Tests	New Nodes	Plan Length	Time (s)
breadth_first_search	14629	18072	129356	12	138.0329
breadth_first_tree_search			*timeout*		
depth_first_graph_search	2269	2270	19021	2200	40.2590
depth_limited_search			*timeout*		
uniform_cost_search	17532	17534	153777	12	87.8299
recursive_best_first_search h_1			*timeout*		
greedy_best_first_graph_search h_1	4501	4503	39624	26	25.7369
astar_search h_1	17532	17534	153777	12	114.6828
astar_search h_ignore_preconditions	5022	5024	44764	12	24.3643
astar_search h_pg_levelsum			*timeout*		

Optimal sequence for Air Cargo Problem 3:

```
Load(C1, P1, SFO)
Fly(P1, SFO, ATL)
Load(C3, P1, ATL)
Fly(P1, ATL, JFK)
Unload(C1, P1, JFK)
Load(C2, P2, JFK)
Fly(P2, JFK, ORD)
Load(C4, P2, ORD)
Fly(P2, ORD, SFO)
Unload(C2, P2, SFO)
Unload(C3, P1, JFK)
Unload(C4, P2, SFO)
```

Excepting Recursive Best First Search and A* Search (Level Sum), all others had a pretty good result in terms of exploring the length of the plan and the time needed to execute the jobs

A* Search (Ignoring Preconditions) performed well among all Air Cargo Problems meaning that an Heuristic based approach can outperform belong all methods presented previously [1]

Among all results, however, Breadth First Search and Depth First Search are the most adapted methods to solve planning problems both fast and optimally - specially for problems with low complexity. When it increases, it might be worth to consider a Heuristic based approach such as A* Search

References:

1. **Stuart J. Russel and Peter Norvig** (2010). « Artificial Intelligence: A Modern Approach (3rd Edition) »