AIND Project 3: Air Cargo Planning / Josef Schodl / 10/2017

Heuristic Analysis

Uninformed Search

In the following tables are the results of test runs of uninformed searches on all three air cargo problem sets of different complexity (all tests performed on the same computer).
Uninformed search methods are **Breadth First Search (BFS), Depth First Search (DFS)** and **Uniform Cost Search (UCS).**

Problem 1	BFS	DFS	UCS
#node expansions	43	12	55
#goal tests	56	13	57
#new nodes	180	48	224
time(s)	0.032	0.009	0.039
optimal	Yes	No	Yes

Problem 2	BFS	DFS	UCS
#node expansions	3,401	350	4,761
#goal tests	4,672	351	4,763
#new nodes	31,049	3,142	43,206
time(s)	11.91	1.30	9.92
optimal	Yes	No	Yes

Problem 3	BFS	DFS	UCS
#node expansions	14,491	1,948	17,783
#goal tests	17,947	1,949	17,785
#new nodes	128,184	16,253	155,920
time(s)	87.79	17.43	44.74
optimal	Yes	No	Yes

BFS is the slowest search on problems 2 and 3, slightly outperforming UCS on problem 1. Memory requirement is second highest, nevertheless BFS is an optimal search, because it examines all nodes one level at a time.

DFS is fastest search requiring least memory compared to both other searches, though it is no optimal search, because it expands nodes as deeply as possible one path at a time and ends on finding a goal state, while not considering if a node is better than the other, leading to unsatisfying results.

UCS is faster than BFS while using most memory, but still represents an optimal search as it expands nodes with lowest cost first.

Informed Search

In the following tables are the results of test runs of informed searches on all three air cargo problem sets using **A* searches** (all tests performed on the same computer).

Problem 1	A*(h-1)	A*(ignore-precon)	A*(level-sum)
#node expansions	55	41	11
#goal tests	57	43	13
#new nodes	224	170	50
time(s)	0.038	0.040	0.983
optimal	Yes	Yes	Yes

Problem 2	A*(h-1)	A*(ignore-precon)	A*(level-sum)
#node expansions	4,761	1,450	86
#goal tests	4,763	1,452	88
#new nodes	43,206	13,303	841
time(s)	9.98	3.67	199.27
optimal	Yes	Yes	Yes

Problem 3	A*(h-1)	A*(ignore-precon)	A*(level-sum)
#node expansions	17,783	5,003	311
#goal tests	17,785	5,005	313
#new nodes	155,920	44,586	2,863
time(s)	43.75	14.26	770.29
optimal	Yes	Yes	Yes

A* searches are all optimal.

A* search with h-1 heuristic is fastest on problem 1 and second fastest on problems 2 and 3, but has highest memory requirement on all problem sets.

A* search with ignore-preconditions heuristic is second fastest on problem 1 and fastest on problem 2 and 3 while memory requirement is second highest on all problem sets.

A* search with level-sum heuristic is slowest, but with by far lowest memory requirements. Especially the differences in #node expansions and execution time are remarkable. This is an indication that level-sum heuristic is an excellent estimator, but also very computing intensive.

Comparison Uninformed / Informed Searches

BFS finds optimal solution quicker than all A* searches on problem set 1, while A* search with ignore-preconditions heuristic finds quickest optimal solutions for problem sets 2 and 3.

These test results suggest that uniformed searches can be more efficient on simple problems, while informed searches outperform uninformed searches as problem complexity increases. As for memory requirements, informed searches expand fewer nodes than uninformed searches, depending on used heuristics.

Finally, BFS seems to be a good choice on simple problem sets, while A* search with ignore-preconditions heuristic looks like best choice on more complex problem sets. In case low memory requirements are more important than time, A* search with level-sum heuristic would be preferable choice across all problem set complexity levels.

Optimal Planning

One optimal planning solution for each air cargo problem is as follows:

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)

Problem 2:

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

Problem 3:

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