heuristic_analysis

July 27, 2017

0.1 Part 3: Written Analysis

0.1.1 Provide an optimal plan for probelms 1,2,3

In [ ]:
```
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(C1, P1, JFK)
Unload(C3, P3, SFO)
Unload(C2, P2, SFO)

Problem 3:
Load(C2, P2, JFK)
Fly(P2, JFK, ORD)
Load(C4, P2, ORD)
Fly(P2, ORD, SFO)
Load(C1, P2, SFO)
Unload(C4, P2, SFO)
Unload(C2, P2, SFO)
Fly(P2, SFO, ATL)
Load(C3, P2, ATL)
Fly(P2, ATL, JFK)
Unload(C1, P2, JFK)
Unload(C3, P2, JFK)
```

0.1.2 Compare and contrast non-heuristic search result metrics (optimality, time elapsed, number of node expansions) for Problems 1,2, and 3

Problem 1:

Algorithm	Optimality	time	number of node expansion
breadth-first	6	0.05	43
depth-first	20	0.02	21
uniform	6	0.04	5

Problem 2:

Algorithm	Optimality	time	number of node expansion
breadth-first	9	12	2358
depth-first	84	0.2	107
uniform	9	9	3615

Problem 3:

Algorithm	Optimality	time	number of node expansion
breadth-first	12	148	13661
depth-first	876	5.3	1293
uniform	12	54	17014

Depth first seems to be the fastest search result metrics, but also the least optimal. Whereas breadth-first and uniform cost search seem to provide both an optimal solution, where uniform cost search is faster than breadth-first, but also has the highest number of expansions.

0.1.3 Compare and contrast heuristic search result metrics using A* with the "ignore preconditions" and "level-sum" heuristics for Problems 1, 2, and 3.

Problem 1:

Algorithm	Optimality	time	number of node expansion
A* ignore	6	0.04	41
A* levelsum	6	0.7	11

Problem 2:

Algorithm	Optimality	time	number of node expansion
A* ignore	9	2.8	1092
A* levelsum	9	67	252

Problem 3:

Algorithm	Optimality	time	number of node expansion
A* ignore	13	19	6554
A* levelsum	13	211	400

A star ignore seems to be the fastet search result metric, A levelsum the slowest. Number of node expansion is the highest for A star and the lowest fo A star levelsum. Interestingly A star has a lower optimality for the last problem than the other two.

0.1.4 What was the best heuristic used in these problems? Was it better than non-heuristic search planning methods for all problems? Why or why not?

The best heuristic in terms of computation time was A star ignore. Compared to the non heuristic ones, depth first is the fastest overall and has a reasonable number of node expansions for all problems.