AIND – Planning

Heuristic analysis

Problem 1

Optimal Plan:

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

Result Metrics

Method	Optimal Plan attained	Length	Time Elapsed (seconds)	#Node Expansions
Breadth First Search	True	6	0.023	43
Depth First Graph Search	False	20	0.010	21
Uniform Cost Search	True	6	0.027	55
A* Search h_1	True	6	0.028	55
A* Search h_ignore_preconditions	True	6	0.027	41
A* Search h_pg_levelsum	True	6	0.509	11

(Method in green denotes optimal plan)

Problem 2

Optimal Plan:

Load(C3, P3, ATL)

Fly(P3, ATL, SFO)

Unload(C3, P3, SFO)

Load(C2, P2, JFK)

Fly(P2, JFK, SFO)

Unload(C2, P2, SFO)

Load(C1, P1, SFO)

Fly(P1, SFO, JFK)

Unload(C1, P1, JFK)

Result Metrics

Method	Optimal Plan attained	Length	Time Elapsed (seconds)	#Node Expansions
Breadth First Search	True	9	6.126	3343
Depth First Graph Search	False	619	2.056	624
Uniform Cost Search	True	9	9.116	4853
A* Search h_1	True	9	8.912	4853
A* Search h_ignore_preconditions	True	9	3.171	1450
A* Search h_pg_levelsum	True	9	45.467	86

Problem 3

Optimal Plan:

Load(C2, P2, JFK)

Fly(P2, JFK, ORD)

Load(C4, P2, ORD)

Fly(P2, ORD, SFO)

Unload(C4, P2, SFO)

Load(C1, P1, SFO)

Fly(P1, SFO, ATL)

Load(C3, P1, ATL)

Fly(P1, ATL, JFK)

Unload(C3, P1, JFK)

Unload(C2, P2, SFO)

Unload(C1, P1, JFK)

Result Metrics

Method	Optimal Plan attained	Length	Time Elapsed (seconds)	#Node Expansions
Breadth First Search	True	12	31.348	14663
Depth First Graph Search	False	619	2.056	624
Uniform Cost Search	True	12	38.177	18223
A* Search h_1	True	12	8.912	18223
A* Search h_ignore_preconditions	True	12	12.717	5040
A* Search h_pg_levelsum	True	12	222.123	315

Discussion

The three non-heuristic search strategies: breadth first search, depth first graph search and uniform cost search find a solution the cargo problems albeit with varying rates of success.

Depth first graph search is the slowest across the board for all three problems. Although it takes the least amount of time and memory, it does not attain the optimal plan. This is due to the fact that this algorithm does not care about the optimality of nodes, it just explores all the trees of nodes that go into the greatest depth.

Breadth first search is the most time and memory efficient of the three non-heuristic search strategies. It finds the optimal solution as it always considers the shortest path first.

As the problem becomes more complex, the heuristic search methods are clear favorites in terms of choice of algorithm. A* search with ‘h_ignore_preconditions’ performs the best for problems 2 and 3.

The A* search with ‘h_pg_levelsum’ performed poorly as compared to its counterpart as the heuristic was very complex. For smaller problems, breadth first search would be the way to go, but as the problem size increases, A* search with ‘h_ignore_preconditions’ should be considered.