Heuristic Analysis AIND planning project

Optimal plans

P1

Problem:

```
Init(At(C1, SFO) □ At(C2, JFK)
    □ At(P1, SFO) □ At(P2, JFK)
    □ Cargo(C1) □ Cargo(C2)
    □ Plane(P1) □ Plane(P2)
    □ Airport(JFK) □ Airport(SFO))
Goal(At(C1, JFK) □ At(C2, SFO))
```

Solution:

Length: 6

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

P2

Problem:

```
Init(At(C1, SFO) □ At(C2, JFK) □ At(C3, ATL)
    □ At(P1, SFO) □ At(P2, JFK) □ At(P3, ATL)
    □ Cargo(C1) □ Cargo(C2) □ Cargo(C3)
    □ Plane(P1) □ Plane(P2) □ Plane(P3)
    □ Airport(JFK) □ Airport(SFO) □ Airport(ATL))
Goal(At(C1, JFK) □ At(C2, SFO) □ At(C3, SFO))
```

Solution:

length: 9

```
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)
```

P3

Problem:

```
Init(At(C1, SFO) □ At(C2, JFK) □ At(C3, ATL) □ At(C4, ORD)
    □ At(P1, SFO) □ At(P2, JFK)
    □ Cargo(C1) □ Cargo(C2) □ Cargo(C3) □ Cargo(C4)
    □ Plane(P1) □ Plane(P2)
    □ Airport(JFK) □ Airport(SFO) □ Airport(ATL) □ Airport(ORD))
Goal(At(C1, JFK) □ At(C3, JFK) □ At(C2, SFO) □ At(C4, SFO))
```

Solution:

Length: 12

```
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)
```

Performance

P1

Heuristic	Plan length	Expansions	Goal Tests	New Nodes	Time elapsed sec
1. breadth_first_search	6	43	56	180	0.041
2. breadth_first_tree_search	6	1458	1459	5960	1.093
3. depth_first_graph_search	20	21	22	84	0.017
4. depth_limited_search	50	101	271	414	0.107
5. uniform_cost_search	6	55	57	224	0.050
6. recursive_best_first_search_with_h_1	6	4229	4230	17023	3.329
7. greedy_best_first_graph_search_with_h_1	6	7	9	28	0.006
8. astar_search_with_h_1	6	55	57	224	0.046
9. astar_search_with_h_ignore_preconditions	6	41	43	170	0.039
10. astar_search_with_h_pg_levelsum	6	11	13	50	1.517

P2

Heuristic	Plan length	Expansions	Goal Tests	New Nodes	Time elapsed sec
1. breadth_first_search	9	3343	4609	30509	13.585
2. breadth_first_tree_search	--	--	--	--	timeout
3. depth_first_graph_search	619	624	625	5602	3.332
4. depth_limited_search	50	222719	2053741	2054119	905.32
5. uniform_cost_search	9	4853	4855	44041	40.349
6. recursive_best_first_search_with_h_1	--	--	--	--	timeout
7. greedy_best_first_graph_search_with_h_1	21	998	1000	8982	7.427
8. astar_search_with_h_1	9	4853	4855	44041	41.251
9. astar_search_with_h_ignore_preconditions	9	1506	1508	13820	11.96
10. astar_search_with_h_pg_levelsum	9	86	88	841	163.02

P3

Heuristic	Plan length	Expansions	Goal Tests	New Nodes	Time elapsed sec
1. breadth_first_search	12	14663	18098	129631	113.52
2. breadth_first_tree_search	--	--	--	--	timeout
3. depth_first_graph_search	392	408	409	3364	1.969
4. depth_limited_search	--	--	--	--	timeout
5. uniform_cost_search	12	18223	18225	159618	342.12
6. recursive_best_first_search_with_h_1	--	--	--	--	timeout
7. greedy_best_first_graph_search_with h_1	22	5578	5580	49150	96.17
8. astar_search_with_h_1	12	18223	18225	159618	340.055
9. astar_search_with_h_ignore_preconditions	12	5118	5120	45650	83.97
10. astar_search_with_h_pg_levelsum	--	--	--	--	timeout

Analysis

- Since `h_1` is just a constant, it doesn't have additional information about the states, so the first 8 searches are uninformed.
- Depth first graph search runs fast but the solution is not optimal.
- A* with level sum heuristics reduces significantly the search space (At least for p1 and p2) thanks to GRAPHPLAN and the level-sum algorithm. But the time consumption is large (In p3 didn't finish). Running the heuristics increases the time consumption of the algorithm.
- With A* and ignore preconditions heuristics we can see a search space reduction with more acceptable time consumption. This is because the heuristics is cheaper to calculate. Also the solution is optimal, so it looks like a pretty good choice in terms of search space reduction vs time consumption.
- Which one is the best? I would say it depends. Fos small problems a uninformed search like BF gives an optimal solution with low time consumption. Larger problems will need a costless

heuristic that decreases the search space but since it doesn't cost too much to calculate, finds the solution in a reasonable time.