Implement a planning search

Heuristic analysis

AIND Nanodegree

Michail Kovanis

Optimal Plan for Problems 1, 2 & 3

Problem 1

```
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))
```

Plan for Problem 1

Plan length: 6

Load(C1, P1, SFO)

Fly(P1, SFO, JFK)

Unload(C1, P1, JFK)

Load(C2, P2, JFK)

Fly(P2, JFK, SFO)

Unload(C2, P2, SFO)

Problem 2

```
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))
```

Plan for Problem 2

Plan length: 9

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)

Problem 3

```
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))
```

Plan for Problem 3

Plan length: 12

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)

Comparison of Non-Heuristic Search

SEARCH	PLAN LENGTH	TIME ELAPSED	NODES EXPANDED
BFS	P1: 6	P1: 0.03s	P1: 180
	P2: 9	P2: 11.87s	P2: 30,509
	P3: 12	P3: 98.03s	P3: 129,631
DFS	P1: 20	P1: 0.01s	P1: 84
	P2: 624	P2: 3.07s	P2: 5,602
	P3: 392	P3: 1.54s	P3: 3,364
DLS	P1: 50	P1: 0.9s	P1: 414
	P2: 50	P2: 857.2s	P2: 2,054,119
	P3: N/A	P3: >1h	P3: N/A
UCS	P1: 6	P1: 0.04s	P1: 224
	P2: 9	P2: 10.88s	P2: 44,041
	P3: 12	P3: 48.25	P3: 159,038

Comparison of Different Heuristics for A* Search

HEURISTICS	PLAN LENGTH	TIME ELAPSED	NODES EXPANDED
H1	P1: 6	P1: 0.04s	P1: 224
	P2: 9	P2: 11.14s	P2: 44,044
	P3: 12	P3: 47.44s	P3: 159,038
IGNORE PRECONDITIONS	P1: 6	P1: 0.04s	P1: 170
	P2: 9	P2: 3.64s	P2: 13,303
	P3: 12	P3: 14.81s	P3: 44,926
LEVEL SUM	P1: 6	P1: 1s	P1: 170
	P2: 9	P2: 260s	P2: 13,303
	P3: 12	P3: 1,240s	P3: 44,926

Discussion

In this project, I implemented five different searches; breadth-first, depth-first, depth-limited, uniform cost, and A* to solve a planning problem. The A* search method was implemented using one trivial heuristic (equal to one at all times) and two real ones ("ignore preconditions", and "level sum"). Out of all searches only DFS and DLS did not manage to produce an optimal solution. A* search with the "ignore preconditions" heuristic provided the best results in terms of time for the second and third problems, while for the first it matched the performance of BFS, UCS, and A* with the trivial heuristic. A* with the "level sum" heuristic was the second worst in time performance, with the worst being DLS. In terms of number of expanded nodes, A* search (with both the non-trivial heuristics) performed the best out of all other searches, which provided an optimal plan. This result was especially apparent in the third problem, in which they expanded about one third of the nodes as compared to BFS, UCS and A* with the trivial heuristic. Moreover, due to the similarity of UCS and A* search with the trivial heuristic their performance was almost equal in all metrics.

In general, A* search performed mostly better than the rest of the search algorithms, depending on the heuristic. This result is due to the use of heuristics, which when well selected they greatly reduce the search space by trying to first expand nodes more likely to be closer to the solution. The difference in time performance between A* search that used the "ignore preconditions" heuristic and the one with the "level sum" heuristic is very high in the second and third problems. This may be explained by the fact that in order to estimate the "level sum" heuristic one needs to construct a planning graph at each step and this process costs a lot in terms of time (Russel & Norvig 2010). Finally, one can conclude that for the first problem, all searches and heuristics (apart from DFS and DLS) may be implemented, whereas for the second and third problem, A* search with the "ignore preconditions" heuristic performs significantly better than all the rest.

References

1. Russell S, Norvig P. AI a modern approach. Learning. 2010;2(3):4.