Heuristic Analysis for AIND Planning Project

-Radhika Pasari

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))
```

Optimal Plan-
Load(C2, P2, JFK)
Load(C1, P1, SFO)
Fly(P2, JFK, SFO)
Unload(C2, P2, SFO)
Fly(P1, SFO, JFK)
Unload(C1, P1, JFK)

Uninformed search comparison –

Search	Optimal	Plan Length	Time elapsed	Expansions	Goal tests
BFS	Yes	6	0.025 s	43	56
DFS	No	12	0.008 s	12	13
Greedy	Yes	6	0.003 s	7	9

Here, we see that breadth first and greedy search give optimal results but greedy search first fares better in both time and memory requirements.

A Star search with different heuristics comparison -

Search	Optimal	Plan Length	Time elapsed	Expansions	Goal Tests
A* with ignore preconditions heuristic	Yes	6	0.04 s	41	43
A* with level sum heuristic	Yes	6	0.859 s	11	13

Both the heuristics are optimal. But on one hand, ignore preconditions one is faster whereas level sum heuristic uses much lesser space.

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))
```

Optimal Plan –
Load(C2, P2, JFK)
Load(C1, P1, SFO)
Load(C3, P3, ATL)
Fly(P2, JFK, SFO)
Unload(C2, P2, SFO)
Fly(P1, SFO, JFK)
Unload(C1, P1, JFK)
Fly(P3, ATL, SFO)
Unload(C3, P3, SFO)

Uninformed search comparison –

Search	Optimal	Plan Length	Time elapsed	Expansions	Goal tests
BFS	Yes	9	10.58 s	3343	4609
DFS	No	575	2.45 s	582	583
Greedy	No	10	0.042 s	18	20

For this problem only breadth first search is optimal. But one thing to note is that greedy first search is almost optimal and is considerably faster and uses much lesser space the breadth first search and hence can be used in cases where sub optimality is admissible but we have tighter space, time resources.

A Star searches with different heuristics comparison -

Search	Optimal	Plan Length	Time elapsed	Expansions	Goal Tests
A* with ignore preconditions heuristic	Yes	9	3.01 s	1311	1313
A* with level sum heuristic	Yes	9	133.91 s	74	76

For this problem, also we have a similar result where both heuristics are optimal but ignore preconditions is faster whereas level sum uses much lesser space.

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))
```

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

Uninformed searched comparisons –

Search	Optimal	Plan Length	Time elapsed	Expansions	Goal tests
BFS	Yes	12	87.17 s	14663	18098
DFS	No	596	2.62 s	627	628
Greedy	No	22	13.35	5615	5617

Here also we see that only breadth first search is optimal but again although greedy first is sub optimal it is much faster.

A Star searches with different heuristics comparison -

Search	Optimal	Plan Length	Time elapsed	Expansions	Goal Tests
A* with ignore preconditions heuristic	Yes	12	13.55 s	5040	5042
A* with level sum heuristic	Yes	12	896.65 s	312	314

For this problem, also we have a similar result where both heuristics are optimal but ignore preconditions is faster whereas level sum uses much lesser space.

Conclusion:

For uninformed searches, breadth first search is the only optimal search but it can get very slow and utilize a lot of memory. So, in very complex and big problems if sub optimal solutions are admissible greedy first search might be a better option since it is having better time and space complexity.

As problems become more complex introduction of heuristics improves performance while maintaining optimality. Hence, for bigger, more complex problems addition of intelligent heuristics can make them solvable in reasonable time and give us an optimal solution. Among the two heuristics we have, the first one (ignore preconditions) is considerably faster and gives us best time in all our searches. But in case space needs are more important than time spent the second heuristic is a better choice. Overall, as problem complexity increases we need heuristics to give us optimal solutions in admissible time whereas for simpler problems devising a heuristic may not be required.