Heuristic analysis

**Yun Ling**

This article is my review for AIND-planning project. Below the content, there are several search functions deal with three air cargo problem in different complexity.

The graphs presented below show us the solving the problems with both uninformed and informed search strategy. Some strategies are not chosen to be present due to infinity time elapsed. in other word, it cannot find plan in acceptable time.

**Air\_cargo\_problem\_1**

Optional plan:

Load(C1, P1, SFO)

Load(C2, P2, JFK)

Fly(P1, SFO, JFK)

Fly(P2, JFK, SFO)

Unload(C1, P1, JFK)

Unload(C2, P2, SFO)



The first problem is pretty simple. In non-heuristic search strategies, we can notice that only breadth first search gave us the optimal plan. But the greedy best first search uses the least expansions, goal tests and new nodes. It is also use the least time to give us an acceptable solution. In heuristic search strategies, all of them can give us the optimal plan. The “ignore preconditions” heuristics uses more expansions goal test and nodes to calculate but spends less time compared with “level sum” heuristics.

**Air\_cargo\_problem\_2**

Optional plan:

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(C2, P2, SFO)

Unload(C3, P3, SFO)



The first problem is a bit more complex. Although the optimal plan length just increase 6 to 9. All the resources like time or memory is enlarged significantly. And depth first search use the least expansions, goal tests and new nodes. But the shortage of depth first search is present. It is not an optimal method and execute unstable. It just can find a first solution it met. The first solution is not in a good place in searching graph, which caused 619 plan lengths. It is hard to accept such a solution based on just 9 steps for optimal plan. In non-heuristic search strategies, the greedy best research still performs a good result in time elapsed. It is obvious to see that the breadth first search and plain astar search need a lot of space to execute. Between the “ignore preconditions” and “level sum” heuristics. We can know the previous one need 10 percent time but 20 times on other resources.

**Air\_cargo\_problem\_2**

Optional plan:

Load(C1, P1, SFO)

Load(C2, P2, JFK)

Fly(P1, SFO, ATL)

Load(C3, P1, ATL)

Fly(P2, JFK, ORD)

Load(C4, P2, ORD)

Fly(P2, ORD, SFO)

Fly(P1, ATL, JFK)

Unload(C4, P2, SFO)

Unload(C3, P1, JFK)

Unload(C2, P2, SFO)

Unload(C1, P1, JFK)



The optimal plan length in problem 3 increased to 12, which is twice as large as first problem. But depth first just use 1 second to find a solution. The reason I think is the goal is near the first searching branch. “ignore preconditions” strategy still draw a good result in time elapsed while finding optimal solution. “levelsum” strategy use much less expansions, goal tests and new nodes to find as optimal solution compared with other heuristic strategy. While the time elapsed is much longer than any other search.

**Conclusions**

For heuristic strategies, they base one a good heuristic function. More precisely, the heuristic cannot overestimate the goal astar\_with\_h\_ignore\_preconditions performs well in all three problem because of the good searching feature the strategy had, that is, relaxes the problem with ignoring preconditions. But if we don’t care about the optimal solution and just focus on whether the plan is feasible, breadth first search will be the best choice.