

## Air Cargo Planning

Problem 1

Search type	Expansions	Goal Tests	New Nodes	Plan Length	Time	Optimal
breadth for search	43	56	180	6	0.081	Yes
depth first graph s.	21	22	84	20	0.045	No
uniform cost search	55	57	224	6	0.088	Yes
A* ignore precondition.	41	43	170	6	0.096	Yes
A* h_pg levelsum	11	13	50	6	1.162	Yes

Problem 2

Search type	Expansions	Goal Tests	New Nodes	Plan Length	Time	Optimal
breadth for search	3343	4603	30107	9	18.135	Yes
depth first graph s.	1811	1812	15303	831	19.891	No
uniform cost search	4852	4854	43302	9	26.402	Yes
A* ignore precondition.	1450	1452	13222	9	9.483	Yes
A* h_pg levelsum	112	114	1061	9	170.13	Yes

Problem 3

Search type	Expansions	Goal Tests	New Nodes	Plan Length	Time	Optimal
breadth for search	14663	18098	129631	12	93.856	Yes
depth first graph s.	408	409	3364	392	4.125	No
uniform cost search	18235	18237	159716	12	118.93	Yes
A* ignore precondition.	5040	5042	44944	12	37.108	Yes
A* h_pg levelsum	318	320	2934	12	518.23	Yes

**One of optimal solution for Problem 1**

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

**One of optimal solution for Problem 2**

Load(C1, P1, SFO)  
Load(C2, P2, JFK)  
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)

**One of optimal solution for Problem 3**

Load(C1, P1, SFO)  
Load(C2, P2, JFK)  
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)

## Conclusion:

**Breadth for search** always finds optimal problem-solving plan for the reasonable amount of time. It's one of the best candidates for the first try of finding the solution of the search problem.

**Depth first graph search** very inconsistent with searching time. Sometimes it can find solution very fast (like in Problem 3). But sometimes it can spend almost as much time as breadth for search. It rarely can find the optimal solution because it not consider which node is better, and just explore it as deep as possible. So it's not a good candidate to find the optimal solution, but can be lightning fast with some problems.

**Uniform cost search** initially chooses the path with the lowest cost. It found optimal solution spending a little more time than breadth for search. Another good candidate for first search type to try.

**A\* ignore preconditions** can find optimal solution very quickly. The more complex the problem the more it outperforme breadth for search and uniform cost search in spend time.

**A\* h\_pg levelsum** found optimal solution but spend way more time than any other search type. Possibly because chosen heuristic is too complex.