Planning Search Heuristic Analysis
Karolina Sala

For this project, we created a planning search agent to solve deterministic logistics planning problem for an Air Cargo transport system. The goal of this analysis is to collect the results and find an optimal solution for each air cargo problem. The following tables show the results. In case of some problems in canceled data collection because their execution time exceeded 10 minutes.

An optimal plans for Problems 1, 2, and 3:
This table describes an optimal solutions to solve each of the air cargo problems.

	Initial state and goal	Optimal solution
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))	Load(C1, P1, SFO) Load(C2, P2, JFK) Fly(P2, JFK, SFO) Unload(C2, P2, SFO) Fly(P1, SFO, JFK) Unload(C1, P1, JFK)
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))	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)
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))	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)

Non-Heuristic search strategies:

Problem 1:

Search Type	Plan Length	Expansions	Goal Tests	New Nodes	Time (s)	Optimal
Breadth First	6	43	56	180	0.029	true
Breadth First Tree	6	1458	1459	5960	0.909	true
Depth First Graph	20	21	22	84	0.014	false
Depth Limited	50	101	271	414	0.922	false
Uniform Cost	6	55	57	224	0.034	true
Recursive Best First	6	4229	4230	17023	2.680	true
Greedy Best First Graph	6	7	9	28	0.029	true

Breadth first search, breadth first tree search, uniform cost search, recursive best first search and greedy best first graph search returned the optimal solution. Depth first graph search was the fastest solution and more optimal than other. Depth limited search was the least optimal solutions because using more additional steps than other solutions.

Problem 2:

Search Type	Plan Length	Expansions	Goal Tests	New Nodes	Time (s)	Optimal
Breadth First	9	3401	4672	31049	13.034	true
Breadth First Tree	> 10 min					
Depth First Graph	1138	1192	1193	10606	7.627	false
Depth Limited	> 10 min					
Uniform Cost	9	4761	4763	43206	11.168	true
Recursive Best First	> 10 min					
Greedy Best First Graph	9	550	552	4950	1.296	true

Breadth first search, uniform cost search and greedy best first graph search returned the optimal solution. Depth limited search was the least optimal solutions. The longest and most expansive was breadth first search.

Problem 3:

Search Type	Plan Length	Expansions	Goal Tests	New Nodes	Time (s)	Optimal
Breadth First	12	14491	17947	128184	96.607	true
Breadth First Tree	> 10 min					
Depth First Graph	2014	2099	2100	17558	20.855	false
Depth Limited	> 10 min					
Uniform Cost	12	17783	17785	155920	49.818	true
Recursive Best First	> 10 min					
Greedy Best First Graph	22	4031	4033	35794	11.311	false

Breadth first search and uniform cost search returned the optimal solution. Depth first graph search was the least optimal solutions. The most expensive and longest solution was breadth first search.

Heuristic search strategies:

Problem 1:

Search Type	Plan Length	Expansions	Goal Tests	New Nodes	Time (s)	Optimal
A* search	6	55	57	224	0.046	true
A* search with ignore preconditions heuristic	6	41	43	170	0.044	true
A* with level sum heuristic	6	11	13	50	0.579	true

Problem 2:

Search Type	Plan Length	Expansions	Goal Tests	New Nodes	Time (s)	Optimal
A* search	9	4761	4763	43206	11.210	true
A* search with ignore preconditions heuristic	9	1450	1452	13303	4.034	true
A* with level sum heuristic	9	86	88	841	69.622	true

Problem 3:

Search Type	Plan Length	Expansions	Goal Tests	New Nodes	Time (s)	Optimal
A* search	12	17782	17785	155920	48.764	true
A* search with ignore preconditions heuristic	12	5003	5005	44586	15.510	true
A* with level sum heuristic	> 10 min					

Each A* search found the optimal solution. A* Search with Ignore Preconditions heuristic is the fastest and the least expensive solution. It would be the best choice for Air Cargo problem.

Conclusion

All non-heuristic search strategies and find heuristic search strategies, find a solution to all air cargo problems. Depth first graph search is never optimal because it only examines single nodes and does not determine which node is the best. Breadth first search is always optimal and find the shortest path. These solutions work very well in problems 1 and 2. For problem 3, the best solution is heuristic search strategy. It is better in search that the complexity increased. The most optimal and the fastest search strategy was the A* search with ignore preconditions heuristic. The most complex and slower was the A* with level sum heuristic search, and it . We did not get an optimal result for this strategy in problem 3. The complexity scale is (3 > 2 > 1).

References:

[1] Stuart J. Russell, Peter Norvig (2010), *Artificial Intelligence: A Modern Approach (3rd Edition)* Available at: https://dcs.abu.edu.ng/staff/abdulrahim-abdulrazaq/courses/cosc208/Artificial%20Intelligence%20A%20Modern%20Approach%20(3rd%20Edition).pdf