Planning Search Heuristic Analysis
by Jordan Carson

Introduction
For this project, we implemented a planning search agent to solve deterministic logistics planning problems for an Air Cargo transport system.

Planning Problems
We were given three Air Cargo planning problems that follow the same below action schema:

Action(Load(c, p, a),
PRECOND: At(c, a) ∧ At(p, a) ∧ Cargo(c) ∧ Plane(p) ∧ Airport(a)
EFFECT: ¬ At(c, a) ∧ In(c, p))

Action(Unload(c, p, a),
PRECOND: In(c, p) ∧ At(p, a) ∧ Cargo(c) ∧ Plane(p) ∧ Airport(a)
EFFECT: At(c, a) ∧ ¬ In(c, p))

Action(Fly(p, from, to),
PRECOND: At(p, from) ∧ Plane(p) ∧ Airport(from) ∧ Airport(to)
EFFECT: At(p, from) ∧ At(p, to))

From this, we had to solve for three air_cargo_problems() that have the following initial states and goals. This exercise was to solve the air_cargo_problems() and build A* search heuristic of informed search and compare these to the AIMA uninformed search heuristics.

Air Cargo problem 1:
Initial State and Goal:

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)

Air Cargo problem 2:
Initial State and Goal:

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))

Air Cargo Problem 3:
Initial State and Goal:

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))

The goals above can be obtained through a different set of plans and lengths of plans. The optimal plan lengths for problems 1, 2, and 3 are 6, 9, and 12 actions. Below are the optimal sequence of actions:

Problem 1:
Load(C1, P1, SFO)
Load(C2, P2, JFK)
Fly(P2, JFK, SFO)
Fly(P1, SFO, JFK)
Unload(C2, P2, SFO)
Unload(C1, P1, JFK)

Problem 2:
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(C2, P2, SFO)
Unload(C3, P3, SFO)
Unload(C1, P1, JFK)

Problem 3:
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)

	Air Cargo Problem	Expansion Nodes	Goal Tests	New Nodes	Length	Time Elapses in Seconds	Optimality
Breath First Search	1	43	56	180	6	0.0287	Optimal
Breath First Search	2	2899	3845	25527	9	10.1856	Optimal
Breath First Search	3	14663	18098	129631	12	89.5827	Optimal
Breath First Tree Search	1	1458	1459	5960	6	0.7823	Optimal
Depth First Graph Search	1	21	22	84	20	0.0108	Bad
Depth First Graph Search	2	1524	1525	12704	557	4.7730	Bad
Depth First Graph Search	3	408	409	3364	392	1.5544	Bad
Depth Limited Search	1	101	271	414	50	0.0692	Bad
Greedy Best First Graph Search with h1	1	7	9	28	6	0.0066	Optimal
Greedy Best First Graph Search with h1	2	726	728	6334	30	1.8481	Bad
Greedy Best First Graph Search with h1	3	5579	5581	49159	22	15.9469	Good
Recursive Best First Search with h1	1	4229	4230	17023	6	2.5431	Optimal
Uniform Cost Search	1	55	57	224	6	0.0305	Optimal
Uniform Cost Search	2	4000	4002	35439	9	10.7097	Optimal
Uniform Cost Search	3	18223	18225	159618	12	47.2723	Optimal

Comparison of Uninformed Planning Searches

The results for the uninformed deterministic searches can be found in the pdf titled "uninformed search results.pdf" or below:

- Breath First Search (BFS): Also known as shortest first search, will always find the shortest way in terms of node searching to the goal, however it will take more time than other searches.
- Depth First Search (DFS): This method is faster than BFS, but the path to the goal takes longer as it generates a longer search path. This is not an optimal case.
- Uniform Cost Search (UCS): Also known as Cheapest first search picks the path with the lowest total cost. Thus this would be the optimal method compared to BFS or DFS.

Comparison of Informed Planning Searches

The A* search finds the shortest length path while it is expanding the minimum. It depends on the heuristic, that keeps the algorithm focused to reach the goal.

Informed Search Results

	Air Cargo Problem	Expansions	Goal Tests	New Nodes	Plan Length	Time Elapses in Seconds	Optimal?
A* with H1	1	55	57	224	6	0.0346	Optimal
A* with H1	2	4000	4002	35439	9	9.4668	Optimal
A* with H1	3	18223	18225	159618	12	45.5419	Optimal
A* with Ignore Preconditions	1	41	42	170	6	0.0323	Best
A* with Ignore Preconditions	2	1317	1319	11820	9	3.6357	Best
A* with Ignore Preconditions	3	5040	5042	44944	12	15.2359	Best
A* with level-sum	1	56	58	228	6	1.4970	Good
A* with level-sum	2	4581	4583	40439	9	1595.2919	Good
A* with level-sum	3	20049	20051	174481	12	11439.4219	Good

Looking at the above results, there are far less expansions required for the A* with ignore preconditions heuristic, however this is far faster then H1 or level-sum. The level-sum then requires less goal tests but if you compare the amount of time taken the ignore preconditions heuristic is the most optimal.

Informed vs Uninformed Search

The best search strategies, as the strategies that generate optimal plans, are BFS, UCS, and A* search with all heuristics.

The results of the A* search (informed) strategies with custom heuristics over uninformed search techniques when searching for an optimal plan. This means that there are true benefits of building custom-made heuristics for a particular problem, and these benefits can be seen in terms of speed, memory usage, and the plan's length.