AIND Project 3 – Implement a Planning Search

1. Optimal plans for solving air cargo problems 1, 2, 3

Problem 1:

$$\begin{aligned}&\text{Init}(\text{At}(C1, SFO) \land \text{At}(C2, JFK)\\&\quad \land \text{At}(P1, SFO) \land \text{At}(P2, JFK)\\&\quad \land \text{Cargo}(C1) \land \text{Cargo}(C2)\\&\quad \land \text{Plane}(P1) \land \text{Plane}(P2)\\&\quad \land \text{Airport}(JFK) \land \text{Airport}(SFO))\\&\text{Goal}(\text{At}(C1, JFK) \land \text{At}(C2, SFO))\end{aligned}$$

Optimal Solution:

Actions	Solution Cost (in length)
Load C1 and C2 on P1 and P2 respectively	2
Fly P1 and P2 to JFK and SFO respectively	2
Unload C1 and C2 from P1 and P2 respectively	2
Total Solution Length: 6	

Problem 2:

$$\begin{aligned}&\text{Init}(\text{At}(C1, SFO) \land \text{At}(C2, JFK) \land \text{At}(C3, ATL)\\&\quad \land \text{At}(P1, SFO) \land \text{At}(P2, JFK) \land \text{At}(P3, ATL)\\&\quad \land \text{Cargo}(C1) \land \text{Cargo}(C2) \land \text{Cargo}(C3)\\&\quad \land \text{Plane}(P1) \land \text{Plane}(P2) \land \text{Plane}(P3)\\&\quad \land \text{Airport}(JFK) \land \text{Airport}(SFO) \land \text{Airport}(ATL))\\&\text{Goal}(\text{At}(C1, JFK) \land \text{At}(C2, SFO) \land \text{At}(C3, SFO))\end{aligned}$$

Optimal Solution:

Actions	Solution Cost (in length)
Load C1, C2 and C3 on P1, P2, P3 respectively	3
Fly P1, P2 and P3 to JFK, SFO and SFO respectively	3
Unload C1, C2 and C3 from P1, P2 and P3 respectively	3
Total Solution Length: 9	

Problem 2:

$$\begin{aligned}&\text{Init}(\text{At}(C1, SFO) \land \text{At}(C2, JFK) \land \text{At}(C3, ATL) \land \text{At}(C4, ORD)\\&\quad \land \text{At}(P1, SFO) \land \text{At}(P2, JFK)\\&\quad \land \text{Cargo}(C1) \land \text{Cargo}(C2) \land \text{Cargo}(C3) \land \text{Cargo}(C4)\\&\quad \land \text{Plane}(P1) \land \text{Plane}(P2)\\&\quad \land \text{Airport}(JFK) \land \text{Airport}(SFO) \land \text{Airport}(ATL) \land \text{Airport}(ORD))\\&\text{Goal}(\text{At}(C1, JFK) \land \text{At}(C3, JFK) \land \text{At}(C2, SFO) \land \text{At}(C4, SFO))\end{aligned}$$

Optimal Solution:

Actions	Solution Cost (in length)
Load C1 and C2 on P1 and P2 respectively	2
Fly P1 and P2 to ATL and ORD respectively	2
Load C3 and C4 on P1 and P2 respectively	2
Fly P1 and P2 to JFK and SFO respectively	2
Unload C1 and C3 from P1	2
Unload C2 and C4 from P2	2
Total Solution Length: 12	

2. Metrics from experiments of non-heuristic planning solution searches

Air Cargo Problem 1					
Search Algorithm	**Node Expansions Required**	**No. Goal Tests**	**New Nodes**	**Time Elapsed (s)**	**Optimality of Solution**
breadth_first_search	43	56	180	0.038	Yes (Length 6)
depth_first_graph_search	21	22	84	0.018	No (Length 20)
uniform_cost_search	55	57	224	0.047	Yes (Length 6)

Air Cargo Problem 2					
Search Algorithm	**Node Expansions Required**	**No. Goal Tests**	**New Nodes**	**Time Elapsed (s)**	**Optimality of Solution**
breadth_first_search	3343	4609	30509	14.23	Yes (Length 9)
depth_first_graph_search	624	625	5602	3.62	No (Length 619)
uniform_cost_search	4853	4855	44041	45.90	Yes (Length 9)

Air Cargo Problem 3					
Search Algorithm	**Node Expansions Required**	**No. Goal Tests**	**New Nodes**	**Time Elapsed (s)**	**Optimality of Solution**
breadth_first_search	14663	18098	129631	104.67	Yes (Length 12)
depth_first_graph_search	408	409	3364	1.84	No (Length 392)
uniform_cost_search	18223	18225	159618	376.13	Yes (Length 12)

From the above metrics, both breadth first search (BFS) and uniform cost search (UCS) results in an optimal solution at the cost of more node expansions and computational resources, as compared to depth first graph search (DFGS) which does not result in an optimal solution. Between BFS and UCS, BFS is more computationally efficient, producing the optimal solution with lesser number of node expansions compared to UCS.

3. Metrics from experiments of A* searches with domain-independent heuristics

Air Cargo Problem 1					
Search Algorithm	**Node Expansions Required**	**No. Goal Tests**	**New Nodes**	**Time Elapsed (s)**	**Optimality of Solution**
astar_search, h1	55	57	224	0.042	Yes (Length 6)
astar_search, h_ignore_preconditions	41	43	170	0.052	Yes (Length 6)
astar_search, h_pg_levelsum	8	10	35	1.05	Yes (Length 6)

Air Cargo Problem 2					
Search Algorithm	**Node Expansions Required**	**No. Goal Tests**	**New Nodes**	**Time Elapsed (s)**	**Optimality of Solution**
astar_search, h1	4853	4855	44041	44.1	Yes (Length 9)
astar_search, h_ignore_preconditions	1506	1508	13820	13.89	Yes (Length 9)
astar_search, h_pg_levelsum	39	41	368	70.08	Yes (Length 9)

Air Cargo Problem 3					
Search Algorithm	**Node Expansions Required**	**No. Goal Tests**	**New Nodes**	**Time Elapsed (s)**	**Optimality of Solution**
astar_search, h1	18223	18225	159618	373.77	Yes (Length 12)
astar_search, h_ignore_preconditions	5118	5120	45650	85.50	Yes (Length 12)
astar_search, h_pg_levelsum	18223	18225	159618	376.13	Yes (Length 12)

From the above metrics, all three A* search algorithms achieved an optimal solution within finite amount of time for all three problems. A* Search with the heuristic of ignoring preconditions consistently outperforms the other two A* heuristics. It is worth noting that A* with ignoring preconditions outperforms A* with planning graph and level sum by approximately a factor of 4 in Problem 2 and 3, even while expanding almost 40x number of node expansions. One possible reason for the disappointing performance of A* with planning graph and level sum is that h_levelsum as implemented in the code, is very slow to compute.

The best A* algorithm, astar_search with h_ignore_preconditions, outperforms the best non-heuristic search algorithm in Part 2, BFS in all three Air Cargo Problem in terms of compute time and node expansions. Hence the conclusion of this report is that astar_search with h_ignore_preconditions is the best heuristic used in solving these problems.