Air Cargo Planning Analysis Report

Air Cargo Problems

Three Problems are listed for testing with the following initial state and goal:

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))

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))

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))

Uninformed Planning Search Results

After implementing the codes, the 7 methods of search results/efficiency for each of the 3 problems are represented below:

Search #	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Method	**Breadth First**	**Breadth First Tree**	**Depth First Graph**	**Depth Limited**	**Uniform Cost**	**Recursive Best First**	**Greedy Best First Graph**
Air Cargo Problem 1							
Expansions	43	1458	12	101	55	4229	7

Goal Tests	56	1459	13	271	57	4230	9
New Nodes	180	5960	48	414	224	17029	28
Plan length	6	6	12	50	6	6	6
Sec elapsed	0.030	0.956	0.008	0.089	0.038	2.957	0.007
Air Cargo Problem 2							
Expansions	3343	-	582	-	4853	-	998
Goal Tests	4609	-	583	-	4855	-	1000
New Nodes	30509	-	5211	-	44041	-	8982
Plan length	9	-	575	-	9	-	21
Sec elapsed	15.012	>5 min	3.535	>5 min	11.249	>5 min	2.543
Air Cargo Problem 3							
Expansions	14663	-	627	-	18151	-	5398
Goal Tests	18098	-	628	-	18153	-	5400
New Nodes	129631	-	5176	-	159038	-	47665
Plan length	12	-	596	-	12	-	26
Sec elapsed	114.979	-	3.504	-	55.056	-	16.755

Note: Optimal algorithm Search Methods were marked in RED.
Search algorithms take longer than 5 minutes were killed and marked as '-'.

Here we attempted all 7 methods for each problem and the methods cost more than 5 minutes without yielding a result were killed in terminal. We focused on comparing the Breadth first, Depth first graph and Uniform cost methods across the problems.

Breadth first approach searches the nodes at each level before moving to the next level and find the shortest route for each level. The method has relatively more expansions/nodes visited compared to depth first graph approach but yields to the optimal result in a reasonable time for all three problems.

While depth first graph approach exhausts each branch for solutions before moving back to the upper level to explore other branches. Therefore, it computes planning result in a short time but provides less inefficient results. This approach serves better for less complex problems.

Uniform cost search is similar to breadth first search but evaluate the cost of the path rather than shortest path. Therefore, it also yield to optimal results in a reasonable time span for all three algorithms. Because the uniform cost evaluate the cost rather than breadth, it will take a bit more memory to visit/expand more nodes in order to save a bit less time compared to breadth first search. The method works well when there is less limit with hardware/memory for a relatively more complex problem.

Therefore, breadth first and Uniform cost approaches are good for the first attempt to problems in order to get an optimal result in reasonable time. Depth first graph search works better and faster for less complex problem.

Heuristic Search Results

Three methods of search results/efficiency for each of the 3 problems are represented below:

Search #	**8**	**9**	**10**
Heuristics	**A* Search**	**A* Ignore Preconditions**	**A* Pg level Sum**
Air Cargo Problem 1			
Expansions	55	41	11
Goal Tests	57	43	13
New Nodes	224	170	50
Plan length	6	6	6
Sec elapsed	0.036	0.042	0.493
Air Cargo Problem 2			
Expansions	4853	1450	86
Goal Tests	4855	1452	88
New Nodes	44041	13303	841
Plan length	9	9	9
Sec elapsed	12.12	4.380	39.458
Air Cargo Problem 3			
Expansions	18151	5038	314
Goal Tests	18153	5040	316

New Nodes	159038	44926	2894
Plan length	12	12	12
Sec elapsed	50.971	16.069	192.077

As the result shows, all 3 A* heuristic searches yield to the optimal result for the problems. Thus heuristic methods could be used to refine/replace the uninformed search methods. We will walk through each method for comparison.

A* ignore-precondition method is the optimal among the 3 heuristics that it calculates the optimal result time-efficiently and with reasonable number of expansion/nodes. The A8 pg-level-sum method will visit/calculate through each goal and level which could be very time-consuming for complex problems. Therefore, it is the least time-efficient methods but visits the least number of nodes. The A* search heuristic performs all well for the three problems and takes a bit more time and space compared to ignore-precondition heuristic.

Optimal Search Methods and Algorithms

The following table shows the optimal plan algorithms for the three air cargo problems respectively.

Uniform Cost Search	**Air Cargo Problem 1**	**Air Cargo Problem 2**	**Air Cargo Problem 3**
Plan length	6	9	12

Plan algorithm	Load(C1, P1, SFO)
	Load(C2, P2, JFK)
	Fly(P1, SFO, JFK)
	Fly(P2, JFK, SFO)
	Unload(C1, P1, JFK)
	Unload(C2, P2, SFO)
		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)
			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(C1, P1, JFK)
			Unload(C2, P2, SFO)
			Unload(C3, P1, JFK)
			Unload(C4, P2, SFO)

Problem 1 is the least complex problem among the 3. As we walk across the above 10 search methods, multiple search approach could provide the optimal planning algorithm in a short time (< 1sec), including breadth first, breadth first tree, uniform cost, greedy best first graph and all 3 A* heuristic search methods. Recursive best first search also yields the optimal planning algorithm but is inefficient, which turns out to not work for the problem 2 and 3.

Problem 2 is more complex compared to problem 1 and it turns out to be solved with an optimal planning algorithm by breadth first, uniform cost and all 3 A* heuristic search methods in 1 minute. With a wider range of time elapsed among these search methods, A* ignore-precondition heuristic search yields the optimal planning result fastest; A* Pg level-sum heuristic search uses the minimum space/memory for searching.

Problem 3 is the most complex planning issue among the 3 problems and inefficient search methods will not function to solve the problem. We found breadth first, uniform cost and 3 A* heuristic search methods could provide optimal planning for this problem. Among these methods, A* ignore-preconditions performs most efficiently to yield the optimal result. Even though, the A* Pg level-sum heuristic search has the minimum new nodes/expansions, it took weigh longer than other methods to find the optimal result. Thus, this method is preferred in case when there is very limited memory/hardware resource and processing time is not a big issue.