

# Heuristic Analysis

## 1. Provide an optimal plan for Problems 1, 2, and 3.

Problem 1:

Plan length: 6

Load(C1, P1, SFO)

Fly(P1, SFO, JFK)

Unload(C1, P1, JFK)

Load(C2, P2, JFK)

Fly(P2, JFK, SFO)

Unload(C2, P2, SFO)

Problem 2:

Plan length: 9

Load(C1, P1, SFO)

Fly(P1, SFO, JFK)

Unload(C1, P1, JFK)

Load(C2, P2, JFK)

Fly(P2, JFK, SFO)

Unload(C2, P2, SFO)

Load(C3, P3, ATL)

Fly(P3, ATL, SFO)

Unload(C3, P3, SFO)

Problem 3:

Plan length: 12

Load(C2, P2, JFK)

Fly(P2, JFK, ORD)

Load(C4, P2, ORD)

Fly(P2, ORD, SFO)

Unload(C4, P2, SFO)

Load(C1, P1, SFO)

Fly(P1, SFO, ATL)

Load(C3, P1, ATL)

Fly(P1, ATL, JFK)

Unload(C3, P1, JFK)

Unload(C1, P1, JFK)

Unload(C2, P2, SFO)

## 2. Compare and contrast non-heuristic search result metrics (optimality, time elapsed, number of node expansions) for Problems 1,2, and 3. Include breadth-first, depth-first, and at least one other uninformed non-heuristic search in your comparison; Your third choice of non-heuristic search may be skipped for Problem 3 if it takes longer than 10 minutes to run, but a note in this case should be included.

problem	Algorithm	Expansions	Goal Tests	New Nodes	Plan length	Time elapsed in seconds
	breadth_first_search	43	56	180	6	0.02435915
	depth_first_graph_search	21	22	84	20	0.0129857
	astar_search with h_1	55	57	224	6	0.030764691
	breadth_first_search	3346	4612	30534	9	12.0330284
	depth_first_graph_search	107	108	959	105	0.279368343
	astar_search with h_1	4634	4636	42095	9	9.54517459
	breadth_first_search	14120	17673	124926	12	75.09224548
	depth_first_graph_search	292	293	2388	288	0.841519947
	astar_search with h_1	17065	17067	149856	12	34.22608549

We can see that for all three problems, BFS gets the optimal plan, DFS does not, however DFS has less expansion, takes much less time than BFS. A\* with no heuristics also gets the optimal plan, while takes more expansions but less time than BFS when the scale is large, and still much slower than DFS. The difference in performance between BFS and DFS by their nature, as stated in the AIND Lesson 11 videos. The BFS expand all the nodes at the same level before it goes to the next level, so it can surely find the optimal solution. The DFS will search to the end of a trace of expansion before it goes back to the fork, so it might take a little time to finish the search because it will not experience the exponential expansion, but it might not end up the optimal solution.

### 3. Compare and contrast heuristic search result metrics using A\* with the "ignore preconditions" and "level-sum" heuristics for Problems 1, 2, and 3.

problem	Algorithm	Expansions	Goal Tests	New Nodes	Plan length	Time elapsed in seconds
	astar_search with h_ignore_preconditions	41	43	170	6	0.025051694
	astar_search with h_pg_levelsum	11	13	50	6	0.672447482
	astar_search with h_ignore_preconditions	1371	1373	12554	9	2.72521201
	astar_search with h_pg_levelsum	79	81	769	9	59.9751652
	astar_search with h_ignore_preconditions	4676	4678	41409	12	11.04835642
	astar_search with h_pg_levelsum	279	281	2560	12	289.2524772

Both the A\* heuristics algorithms gets the optimal plans for all the three problems. And for all three problems, the ignore preconditions heuristics has larger expansions but less time compare to level-sum.

**4. What was the best heuristic used in these problems? Was it better than non-heuristic search planning methods for all problems? Why or why not?**

The best heuristic is the ignore preconditions because it outperforms levelsum. Although it takes longer time than DFS, but it always finds the optimal solution like BFS and non-heuristic A\*, but takes much less time. So it is better than non-heuristic search planning.