Rizwaan Adil

Udacity AI Nanodegree

Heuristic Analysis

Project 3 - Planning

Table of Contents

[Introduction - Heuristic Analysis for Air Cargo Problems 1](#_Toc511645890)

[Uninformed Search Results 2](#_Toc511645891)

[A\* Search with Heuristics Results 3](#_Toc511645892)

[Optimal Plans 4](#_Toc511645893)

[Problem 1 4](#_Toc511645894)

[Problem 2 4](#_Toc511645895)

[Problem 3 5](#_Toc511645896)

# Introduction - Heuristic Analysis for Air Cargo Problems

This paper discusses the results of uninformed and A\* searches applied to the Air Cargo problem presented in the Project 3 of Udacity AI Nanodegree.

The goal is to find an optimal solution to each problem using a variety of search algorithms. The complexity increases from problem 1 to problem 3.

# Uninformed Search Results

The below table shows the results of the uninformed search algorithms applied to the 3 problems.



Key Points:

* None of the searches exceeded the 10 min threshold.
* As evident from the table, DFGS consistently posted the shortest time across the 3 problems. However it isn’t the most optimal solution since it’s plan length is directly proportional to the problem’s complexity. E.g. the plan length goes from 12 in P1 to 575 and 660 in P2 and P3 respectively. This also impacts the other parameters (Expansion, goal tests etc).
* Across all 3 problems, **GBFS appears to be the optimal solution** when we consider the time and plan length together. E.g in problem 3 while the time taken by GBFS is almost 4 times more than DFS the plan length is the shortest (22).

# A\* Search with Heuristics Results

The below table summarizes the results of A\* searches for the 3 problems



Key points:

* Barring the *astar\_search\_h\_pg\_levelsum* for problem 3, all other algorithm executions finished within 10 minutes. While it is very efficient, the execution time is unacceptable for real world scenarios.
* From a holistic perspective, astar\_search\_h\_ignore\_preconditions consistently proves to be an optimal solution even though the nodes and other parameters are way high. From a user’s perspective, this algorithm will be more responsive and hence must be preferred

# Optimal Plans

Following are the optimal plans for the various problems

## Problem 1

Load (C1, P1, SFO)

Load (C2, P2, JFK)

Fly (P1, SFO, JFK)

Fly (P2, JFK, SFO)

Unload (C1, P1, JFK)

Unload (C2, P2, SFO)

## 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 (C1, P1, JFK)

Unload (C2, P2, SFO)

Unload (C3, P3, SFO)

## Problem 3

Load (C1, P1, SFO)

Load (C2, P2, JFK)

Fly (P1, SFO, ATL)

Fly (P2, JFK, ORD)

Load (C3, P1, ATL)

Load (C4, P2, ORD)

Fly (P1, ATL, JFK)

Fly (P2, ORD, SFO)

Unload (C1, P1, JFK)

Unload (C2, P2, SFO)

Unload (C3, P1, JFK)

Unload (C4, P2, SFO)