Udacity – AIND – Planning Search and

Heuristic Analysis

By: Cheryl Miller

# Introduction

This project illustrates the use of planning search agents to solve deterministic logistics planning problems, specifically for an Air Cargo transport scheduling problem. It does this first using uninformed/non-heuristic searches such as Breadth First Search, Depth First Search, Uniform-cost, etc. Then we develop Planning Graphs and use domain independent heuristic searches utilizing heuristic functions such as ignore prerequisites and level sum.

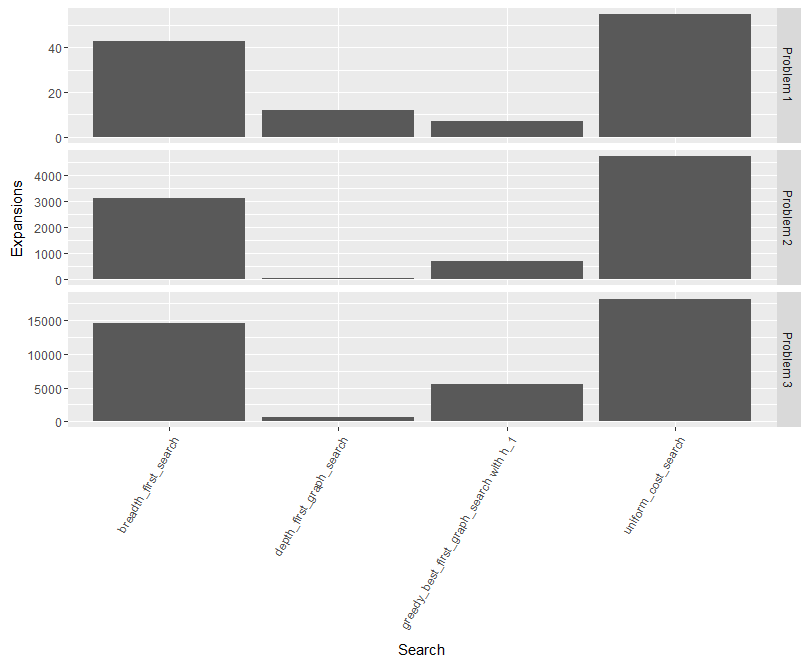
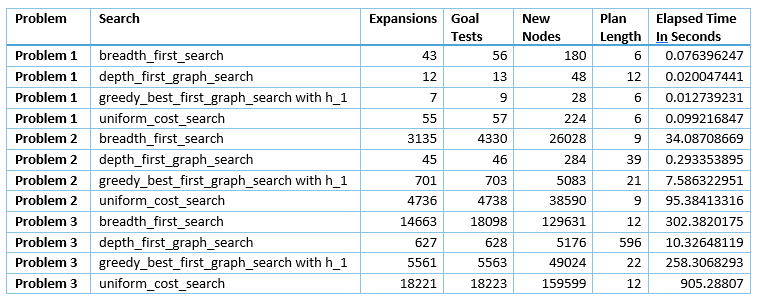
# Uninformed Search

Uninformed search strategies have no additional information about states beyond that known at problem definition, they can only generate successors and distinguish between goal and non-goal states. These searches vary by the order in which nodes are expanded and explored. For this comparison I chose Breadth First, Depth First Graph, Uniform Cost and also included Greedy Best First Graph with h\_1 heuristic (which really is none at all).

## Results

Figure - Uninformed Search Node Expansions

Table - Uninformed Search Metrics



## Analysis

The results highlighted in Figure 1- Uninformed Search Node Expansions and Table 1 show that while Depth First and Greedy Best First Graph searches have very fast elapsed times and low number of node expansions, neither presented an optimal plan. In this case Depth First offered a very bad plan with a plan length of 596 for problem 3. The plan for Greedy Best First Graph search was considerably closer at 22, but still not the optimal length of 12.

Both Breadth First and Uniform Cost searches found the optimal plan every time, but with considerably longer elapsed times and node expansions. Breadth First did better that Uniform Cost, both in elapsed time and nodes expanded. Breadth First stops as soon as it generates a goal state, while Uniform Cost needs to examine all the nodes at a given depth in the event there is a lower cost path.

For the non-informed searches, Breadth First search performed the best both in elapsed time and nodes expanded while still provided the optimal plan.

# Domain independent heuristic Search

Informed or Independent Heuristic searches use problem specific knowledge to find solutions more efficiently. These searches are similar to the Uniform Cost search, except they use a heuristic function to evaluate the cost rather than a set cost value. In this analysis, we look at the AStar searches using the h\_1 heuristic as a baseline for AStar in general, then also look at the h\_ignore\_preconditions and the h\_levelsum heuristics in combination with using a Planning Graph.

The h\_ignore\_precondition heuristic is a ‘relaxed problem’ in which all preconditions on an action are ignored so that all actions become applicable in every state and any single goal state can be achieved in a single step. This turns out to be an instance of the set-cover problem that can be solved using a simple greedy algorithm that returns a set covering size within a factor of log n of the true minimum covering.

The i\_levelsum heuristic returns a sum of level costs of all the goals based on the **subgoal independence assumption** in which the cost of solving a conjunction of subgoals is approximated by the sum of the costs of solving each subgoal independently.

## Results

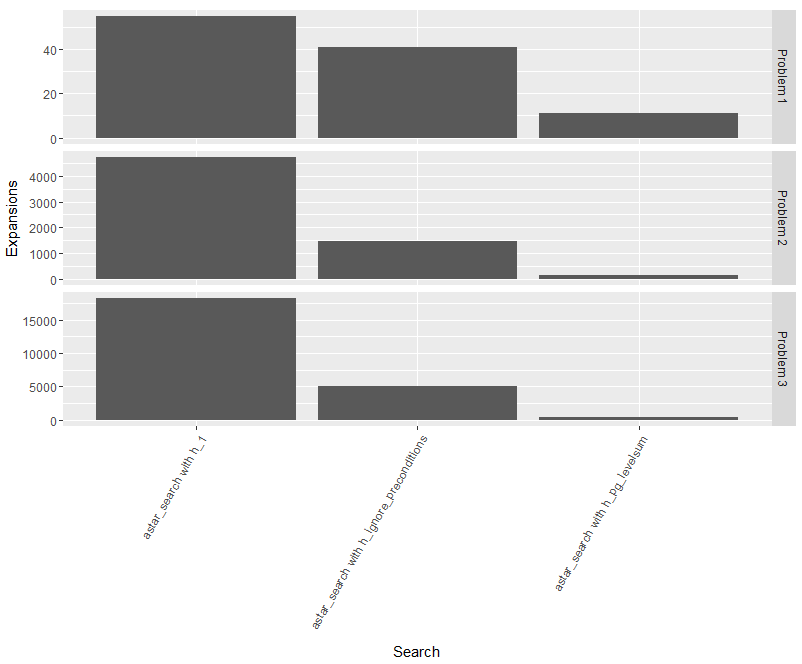
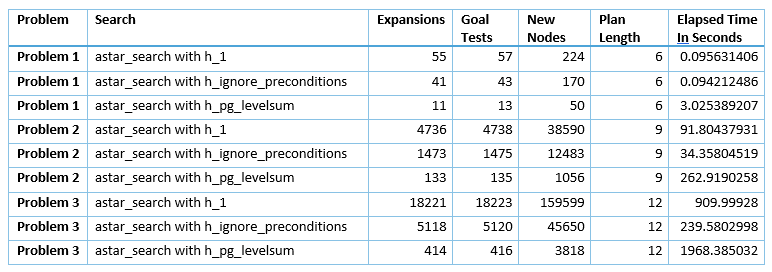
Figure - Heuristic Search Node Expansion

Table - Heuristic Search Metrics



## Analysis

As the results in Table 2 and Figure 2 show, in general for the larger problem 2 and 3, both the AStar/h\_ignore\_preconditions and AStar/h\_levelsum perform much better than the non-informed searches. The exception to this was for problem 1 in which the Greedy Best First Graph search did exceptionally well due to the simple and small graph generated. The Uniform Cost search was identical to the AStar/h\_1 search, since they are ultimately equivalent due to the constant cost of 1 returned by the heuristic function.

# Recommendation

For very small and simple planning graphs, the Breadth First search and Greedy Best First Graph searches prove to work well without a lot of complex overhead from the generation of a graph plan. For the larger and more complex problems, generating the graph plan and using the heuristics proved to be fast and complete. The h\_levelsum heuristic proved to be the fastest and with the least number of node expansions and can be very useful for searching large problem graph plans.