Planning Search Analysis

- ## Problem Definition

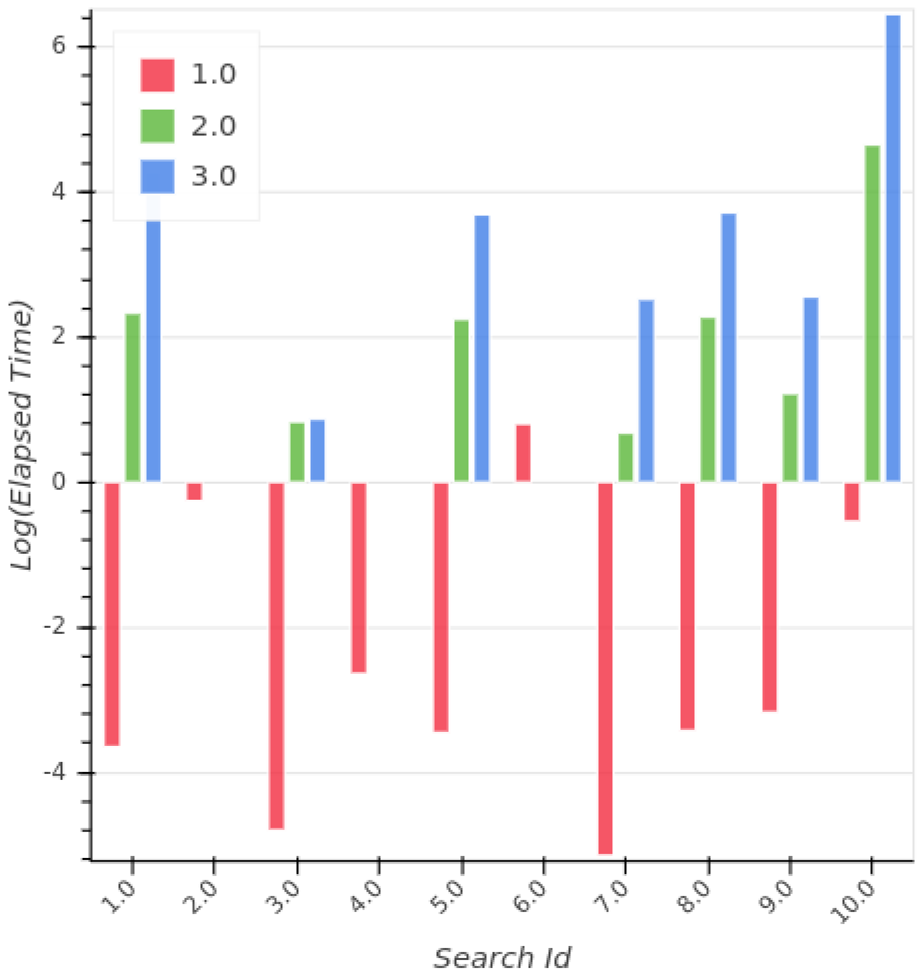
  Advancing from the last project on isolation game playing agent we here are trying to solve a planning problem by defining them as a search problem i.e. search from the initial states through the intermediate states to ultimately find the goal state. The problem being addressed here is the air cargo transport problem which involves loading the cargo, flying it from place to place to ultimately unloading it at the specified destination. The problem has been defined with 3 actions Load, Unload and Fly. The actions here affect two predicates that are ‘In(c, p)’ i.e. when cargo ‘c’ is in aeroplane ‘p’ and At(c/p, a) i.e. when cargo ‘c’ or aeroplane ‘p’ is at airport a. To solve this problem uninformed and informed search algorithms were used. The results of these methods have been analysed below.

- ## Comparison of search results for all three problem

  Three unique problems have been defined and their uninformed and informed search results have been analysed based on metrics of Elapsed Time, expansions and optimal path lengths. The values have been compared on logarithmic scale. The information on problem 2 and 3 has not been analysed due to long running time for 2, 4, and 6th search strategies. The values 1 to 10 on x axis refer to the 10 search strategies provided in the project.
  The results have been illustrated below.

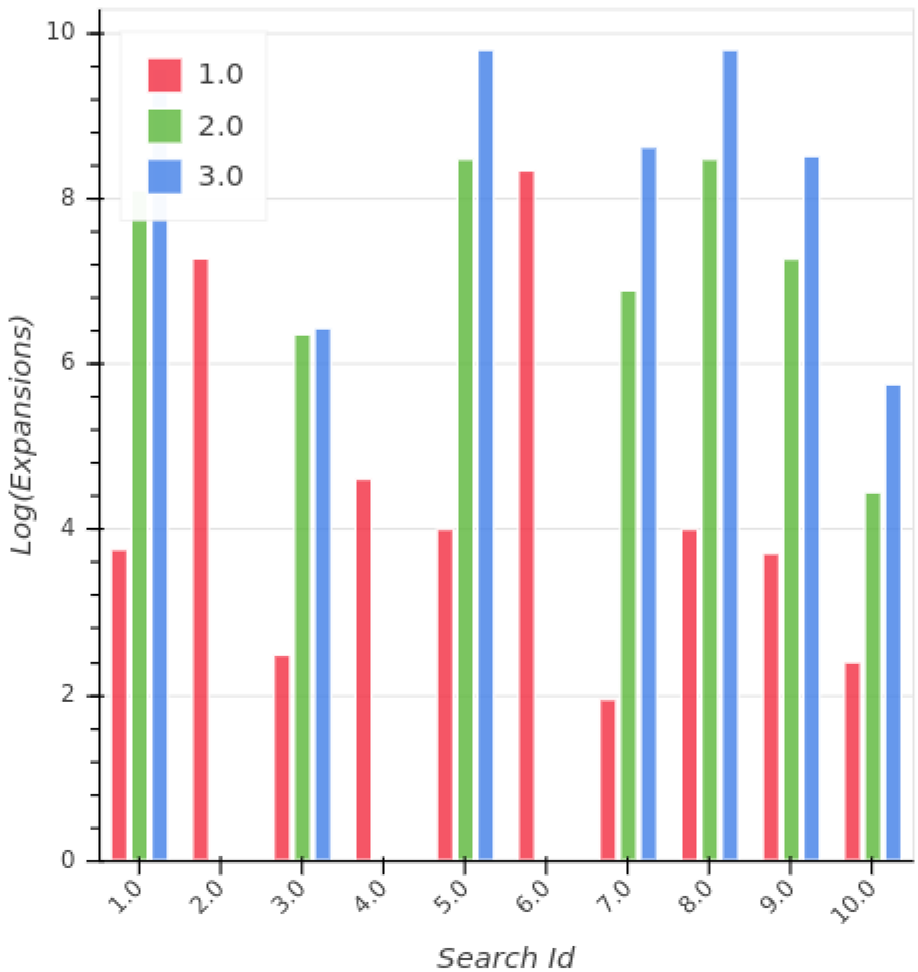
➢ Based on **Elapsed Time** :

Elapsed time basically refers to the time required for the search strategy to find the optimal path for the goal state. Going by these results, it can be said that for all problems 1,2,3 ; **depth first graph search** had the least time in case of uninformed path search and **greedy best first search** had the best time in case of heuristic based search. Although these strategies gave results in optimal time they are **not guaranteed** to give a solution in least amount of time for a more complex problem than the ones defined. The graph also shows us how the time increases as the complexity of the problem increases.



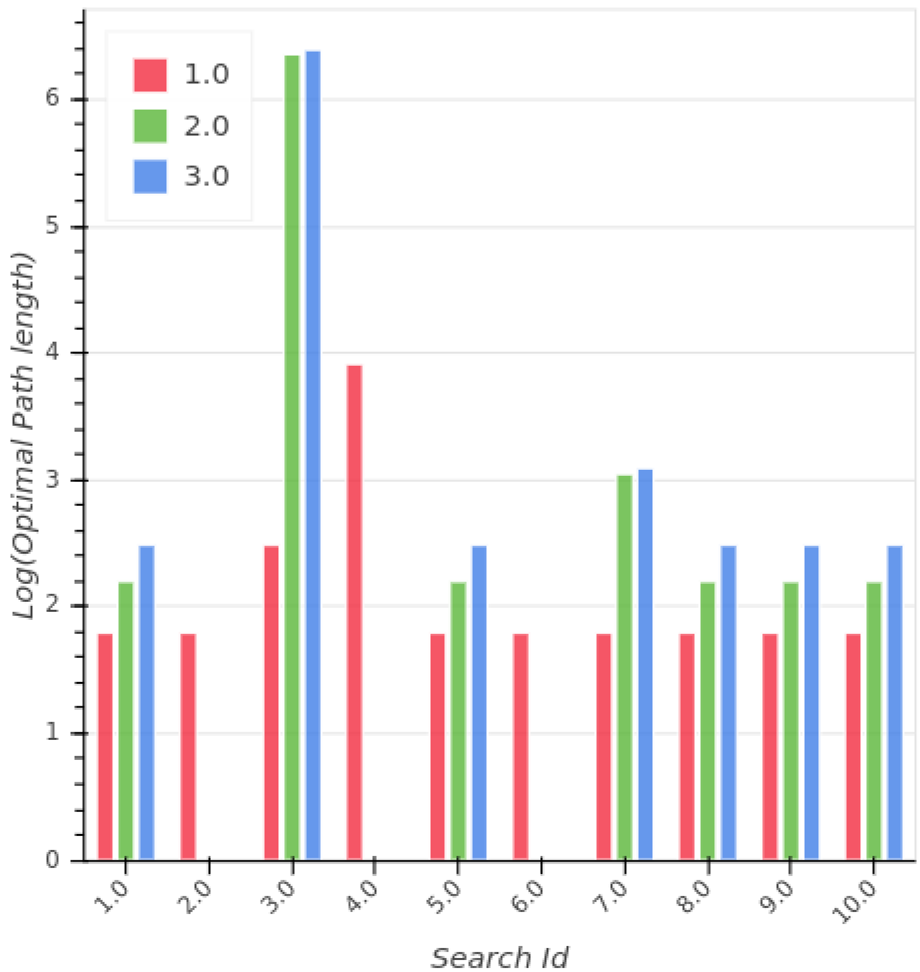
➢ Based on **number of nodes expanded**:

On observing the graph of the number of nodes expanded to reach the optimal solution, we get an understanding of how deep the search algorithm had to search from which we get an indirect implication of space complexity. As uninformed search strategies don't take domain knowledge in consideration and the problems defined here are not that complex, hence depth first graph search can find found solution in least expansion of nodes. But now for informed search strategies we see that for problem 1 greedy best first search with heuristic function h_1 obtains solution in least expansion of nodes and for more complex problems 2 and 3, **A* search with 'h_pg_levelsum'** heuristic function search finds solution in least expansion of nodes.



➢ Based on **Optimal Path length :**

Ultimately when finding the best solution path with least number of nodes to be traversed then the graph clearly shows that dfs fails in it where other all strategies give same results for problem 1. For problem 2 and 3 in case of informed search strategies all the algorithms an optimal path length is same i.e. all the heuristic function help A* in finding the same optimal path length. Overall the shortest path length for problem 1, 2, 3 found is 6, 9 ,12 respectively. But when analysing based all three metrics parallel we understand which solution is the best depending on our time and computation constraints. It also helps in evaluating the best heuristic function and optimising and fine tuning that function.



- Best Optimal plans :

  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)

Fly(P1, SFO, JFK)

Fly(P2, JFK, SFO)

Load(C3, P3, ATL)

Fly(P3, ATL, SFO)

Unload(C3, P3, SFO)

Unload(C1, P1, JFK)

Unload(C2, P2, SFO)

Problem 3 :

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(C4, P2, SFO)

Unload(C3, P1, JFK)

Unload(C1, P1, JFK)

Unload(C2, P2, SFO)

- 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?

  In case of informed search strategies i.e. search algorithms conditioned on the heuristic function the A* search with h_ignore_predictions heuristic function and greedy best first search with heuristic function h_1 perform best based on prioritizing time the highest for nearly all the mentioned problems and optimal path computed by A* search with h_pg_level function performs best based on lowest number of expansions for all mentioned problems.

  Heuristic Function based search strategies perform better than most of the non-heuristic search planning methods. This is primarily because the informed search planning methods take the domain dependent information into account and as the problems in real world start getting more and more complex where we have to plan large store inventories and flight travel schedules for 1000s of airports uninformed search strategies will not provide optimal path schedules and even if they would they would take an enormous amount of time for evaluating it.

  This question has been addressed in Artificial Intelligence Artificial Intelligence A Modern Approach Edition 3 on Pg 374 in section 10.2.1 where he has said that planning problems often have large state spaces and even for a relatively small problem determining a solution without a heuristic is hopeless.