Heuristic Analysis

Table of Contents

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

Optimal Plan for Air Cargo Problem 1

Load(C1, P1, SFO)
Fly(P1, SFO, JFK)
Load(C2, P2, JFK)
Fly(P2, JFK, SFO)
Unload(C1, P1, JFK)
Unload(C2, P2, SFO)

Optimal Plan Air Cargo Problem 2

Load(C1, P1, SFO)
Fly(P1, SFO, JFK)
Load(C2, P2, 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)

Optimal Plan Air Cargo Problem 3

Load (C2, P2, JFK)
Load(C1, P1, SFO)
Fly(P2, JFK, ORD)
Load(C4, P2, ORD)
Fly(P1, SFO, ATL)
Load(C3, P1, ATL)
Fly(P1, ATL, JFK)
Unload(C1, P1, JFK)
Unload(C3, P1, JFK)
Fly(P2, ORD, SFO)
Unload(C2, P2, SFO)
Unload(C4, P2, SFO)

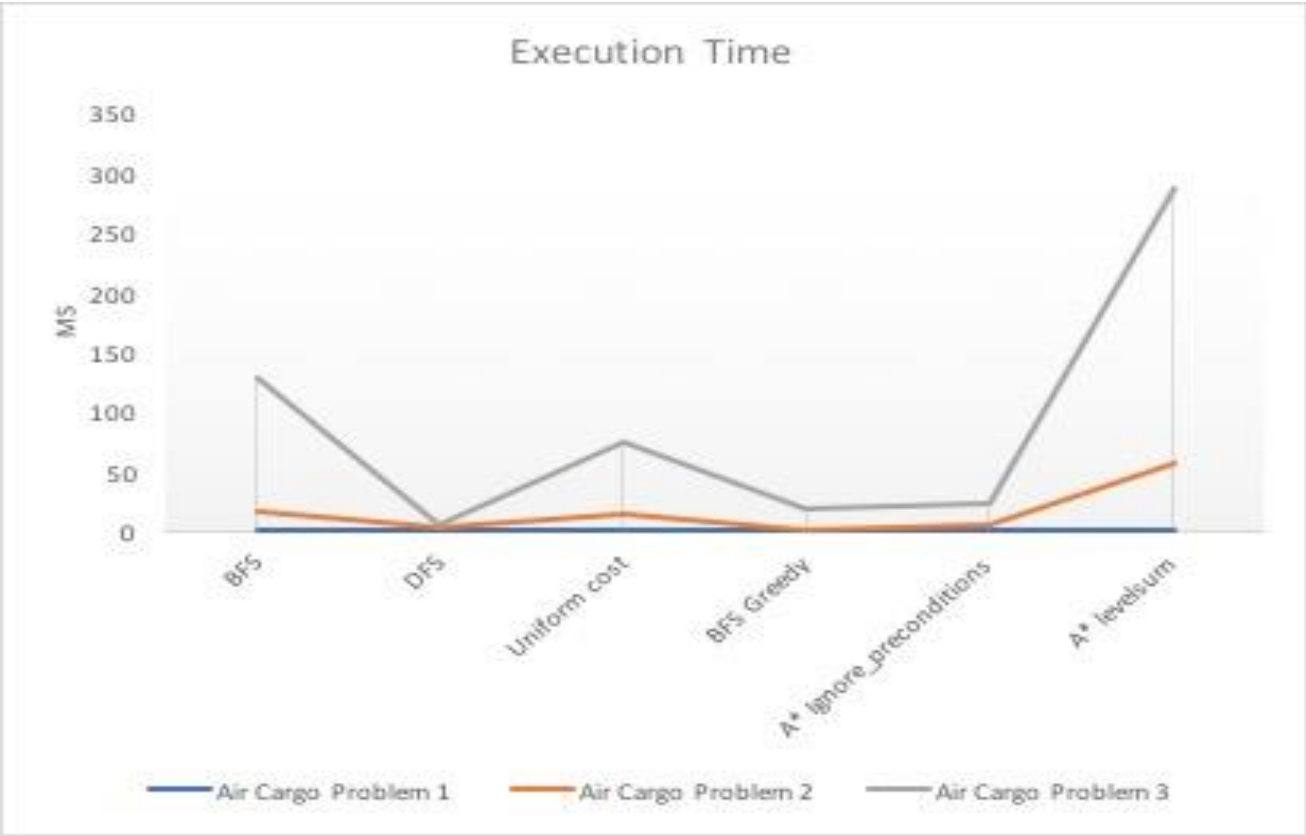
Data Tables and Graphs for Further Discussion

Note – Depth Frist Limited Search has not been included in the following statistics to preserve scale

Execution Time Table

Execution Time - millisecond						
Problem	**BFS**	**DFS**	**Uniform cost**	**BFS Greedy**	**A* Ignore_preconditions**	**A* levelsum**
Air Cargo Problem 1	0.030112452	0.008266761	0.040799025	0.004679458	0.036977633	0.594330518
Air Cargo Problem 2	15.7850817	3.787136518	13.48370305	1.628596639	4.525138589	56.42213105
Air Cargo Problem 3	127.4747322	4.374486951	73.8525242	18.67165855	22.58467513	286.8719626

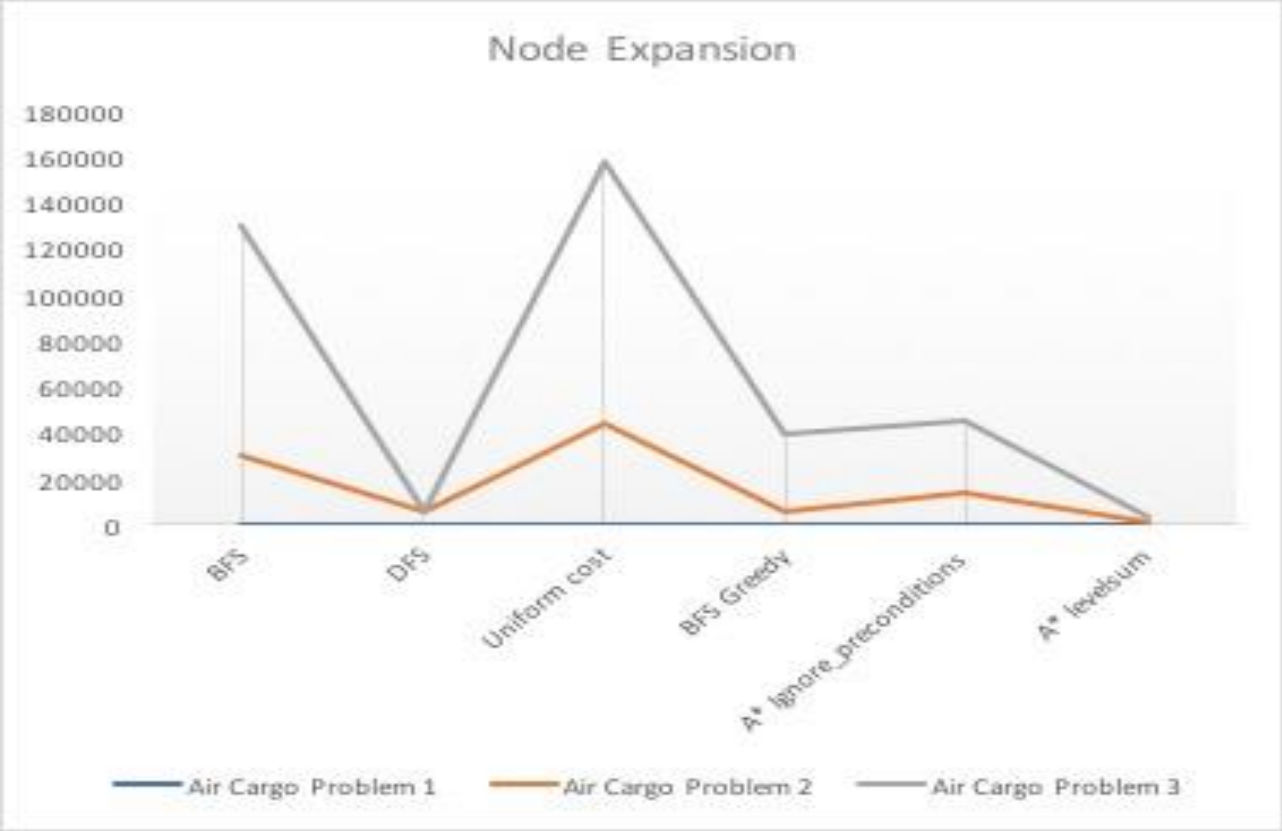
Execution Time Graph



Node Expansion Table

Node Expansion						
Problem	**BFS**	**DFS**	**Uniform cost**	**BFS Greedy**	**A* Ignore_preconditions**	**A* levelsum**
Air Cargo Problem 1	180	48	224	28	170	50
Air Cargo Problem 2	30509	5211	43370	5310	13303	841
Air Cargo Problem 3	129631	5176	157055	39595	44728	2965

Node Expansion Graph



Compare Non-Heuristic Search for Problem 1, Problem 2 and Problem 3

The Non-Heuristic Search considered are

Bread First Search

a. Looks for the Goal near the root of state space
b. Optimal solution is provided
c. Exponential increase of Execution time and nodes expanded

Depth First Graph Search

a. Optimal solution is NOT provided
    a. Reason being DFS keeps on expanding the search tree in one particular direction and stop as soon as a goal is found. The path might not always be optimal.
b. linear increase of nodes and execution time is a positive

Depth First Limited Search

Problem	Execution Time	Nodes Expanded
Air Cargo Problem 1	0.091936213	414
Air Cargo Problem 2	999.389323	2054119

a. It is DFS with a limited downward movement, after that it will expand the next node – Essentially it is a cross between DFS and BFS exploring each branch till a predefined depth.
b. Optimal solution is not provided

c. Exponential increase of Execution Time and Nodes Expanded – Makes the search least desirable search

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

A* Search with level Sum

1- The Heuristic used is to evaluate the level sum of all goal states.
2- The execution time increases exponentially
3- The number of nodes expanded increases linearly

A* Search with Ignore Preconditions

1- The heuristic used here is that we ignore all preconditions while finding the result.
   a. ***The heuristic is faster to calculate hence it takes lesser time to find the goal than A* Search with level Sum***
2- The Execution time increases linearly
3- The number of nodes expanded increases linearly

Conclusion - Since A* Search with Level Sum increases execution time exponentially A* Search with Ignore Preconditions is preferred.

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?

Best Heuristic Search

The A* Search with heuristic Search with ignore preconditions is the best method among the heuristic based search.

Comparison of Heuristic Search with non-heuristic search

After analysis of non-heuristic search, we realize that

1- **Non-Heuristic Search**
   a. **Depth First Search**
      i. ***Optimal Solution is not provided*** – This is a major drawback
         1. Reason being DFS keeps on expanding the search tree in one particular direction and stop as soon as a goal is found. The path might not always be optimal.
   b. **Breath First Search** –
      i. The time execution and nodes expanded is really large
      ii. As the nodes increase the execution time and node expand exponentially
      iii. Since the Time increases exponentially BFS is not suited for problem with large number of states

c. **Depth First Limited Search** – This search was very inefficient because of the number of nodes expanded and the time taken for the overall search to execute.

2- **Heuristic Search** - Heuristic search is bound to be faster as we are trying to solve a relaxed problem which is easier to solve.
   a. **A* Search with Level Sum**
      i. The Heuristic used is to evaluate the level sum of all goal states.
      ii. The execution time increases exponentially – This is not suited for problems with large state space.
      iii. The number of nodes expanded increases linearly
   b. **A* Ignore Preconditions**
      i. The Execution time increases linearly
      ii. The number of nodes expanded increases linearly
   c. The heuristic used here is that we ignore all preconditions while finding the result.
      i. ***The heuristic is faster to calculate hence it takes lesser time to find the goal than A* Search with level Sum***

Explanation for choosing A* Search with Ignore Preconditions

So, the A* Search with Ignore Preconditions is the preferred method because-

1- **Optimality** - Optimal solution is always provided.
2- **Performance** - The time taken is the least for all search that return optimal solution.
3- **Scalability** - The increase in execution time and nodes expanded is linear – hence it is better suited for problems with large number of states

Reference

- Russell, Stuart J; Norvig, Peter (2010). Artificial Intelligence: A Modern Approach (3rd ed.). Upper Saddle River, New Jersey: Prentice Hall. ISBN 0-13-604259-7.