# Udacity Artificial Intelligence Nanodegree – Build a Forward Planning Agent

**By Cristian Alberch, submitted on May 6, 2020.**

This report evaluates different search algorithms used for air cargo problems. Refer to the github repository for programming code, readme with problem description. https://github.com/cristiandatum/AI_projects.git

There are 4 air cargo problems have different number of actions and these are evaluated against 11 different search algorithms and heuristics for forward planning.

| Problems | Search Algorithms |
|---|---|
| 1. Air Cargo Problem 1 (20 actions)<br>2. Air Cargo Problem 2 (72 actions)<br>3. Air Cargo Problem 3 (88 actions)<br>4. Air Cargo Problem 4 (107 actions) | 1. Breadth First Search<br>2. Depth First Graph Search<br>3. Uniform Cost Search<br>4. Greedy Best First Graph Search with heuristic: unmet goals<br>5. Greedy Best First Graph Search with heuristic: planning graph level sum<br>6. Greedy Best First Graph Search with heuristic: planning graph maxlevel<br>7. Greedy Best First Graph Search with heuristic: planning graph setlevel<br>8. A* Search with heuristic: unmet goals<br>9. A* Search with heuristic: planning graph level sum<br>10. A* Search with heuristic: planning graph maxlevel<br>11. A* Search with heuristic: planning graph setlevel |

**Analysis:** Number of nodes expanded against number of actions in the domain.

| | Number of Expansions | | | |
|---|---|---|---|---|
| | **P1: 20** | **P2: 72** | **P3: 88** | **P4: 104** |
| 1. Breadth First Search | 43 | 3,343 | 14,663 | 99,736 |
| 2. Depth First Graph Search | 21 | 624 | 408 | 25,174 |
| 3. Uniform Cost Search | 60 | 5,154 | 18,510 | 113,339 |
| 4. Greedy Best First Graph Search with heuristic: unmet goals | 7 | 17 | 25 | 29 |
| 5. Greedy Best First Graph Search with heuristic: planning graph level sum | 6 | 9 | 14 | 17 |
| 6. Greedy Best First Graph Search with heuristic: planning graph maxlevel | 6 | 27 | 21 | 56 |
| 7. Greedy Best First Graph Search with heuristic: planning graph setlevel | 6 | 9 | 35 | 107 |
| 8. A* Search with heuristic: unmet goals | 50 | 2,467 | 7,388 | 34,330 |
| 9. A* Search with heuristic: planning graph level sum | 28 | 357 | 369 | 1,208 |
| 10. A* Search with heuristic: planning graph maxlevel | 43 | 2,887 | 9,580 | 62,077 |
| 11. A* Search with heuristic: planning graph setlevel | 33 | 1,037 | 3,423 | 22,606 |

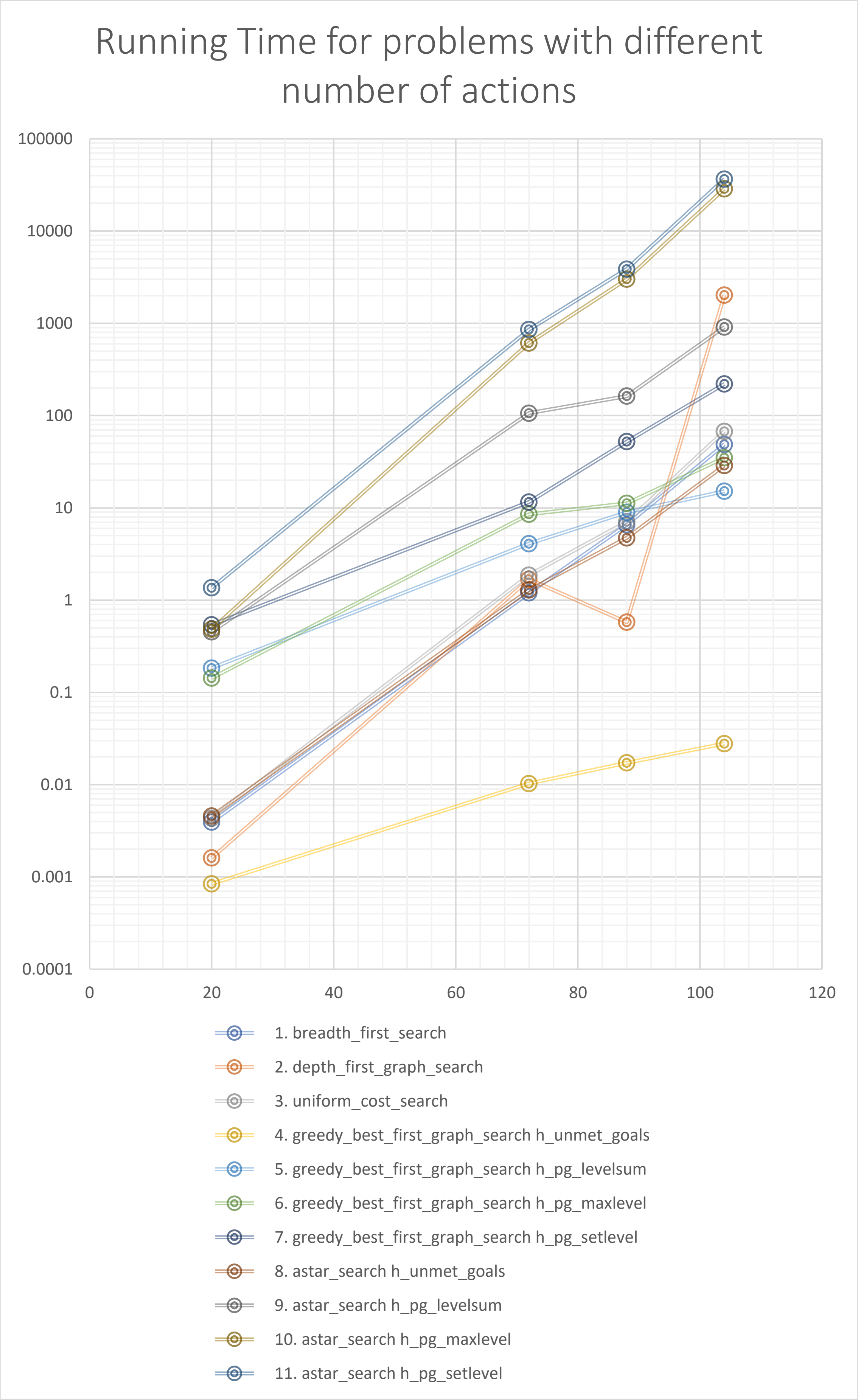
The number of expansions resulting in creation of new action nodes varies greatly according to the algorithm used.

As expected, greedy best-first search results in solutions with very limited number of node expansions as this algorithm tries to expand only the node that is closest to the goal, on the grounds that this is likely to lead to a solution quickly. While this may be efficient in terms of number of operations, it is likely to not lead to an optimum search.

The opposite is true of the A* algorithms, that while guaranteeing an optimum solution, this results in a very large number of operations stemming from a very limited number of actions.

**Analysis:** Search time against the number of actions in the domain.

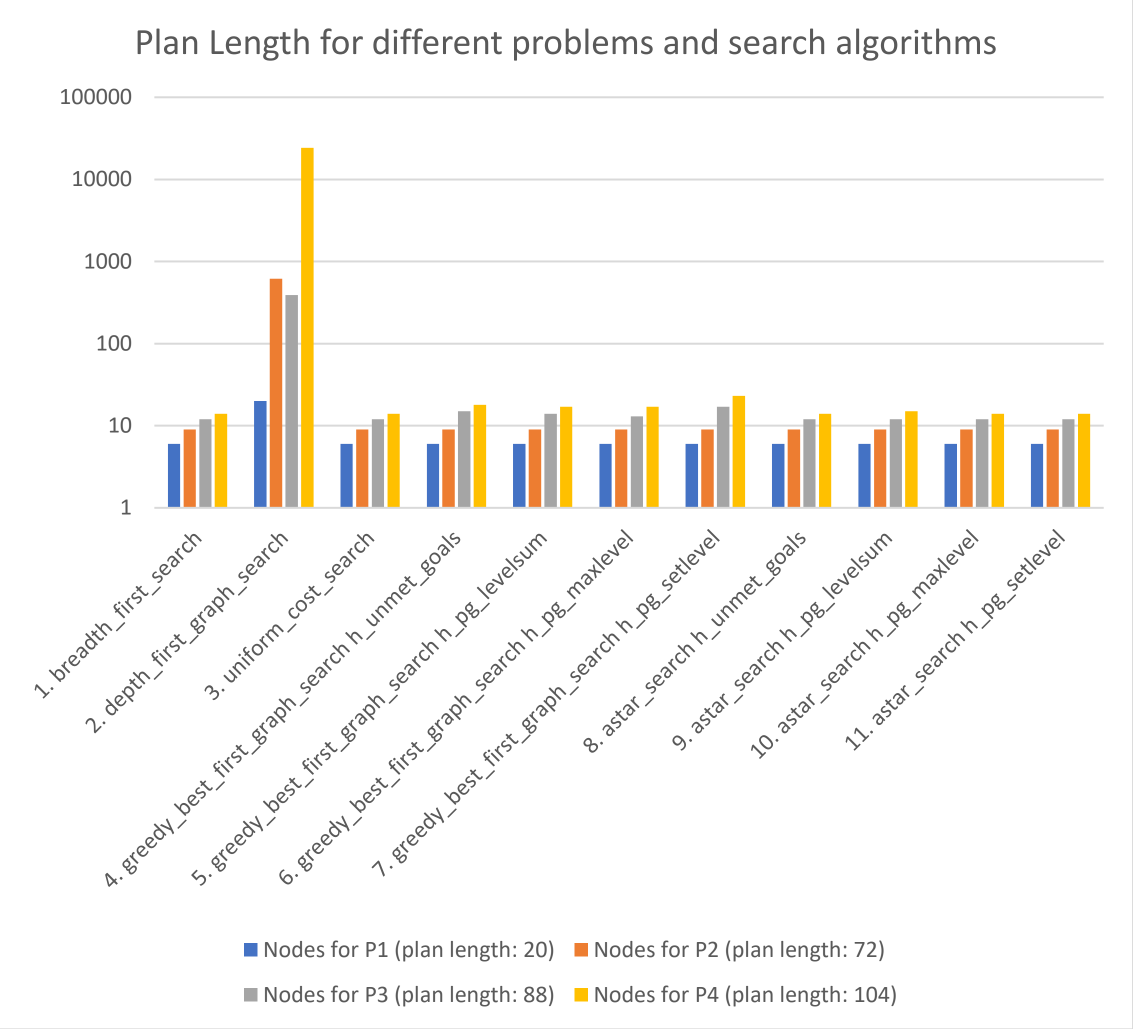
As shown in the graph, the search time increases exponentially. A* algorithms are resulting in dramatic increases in running time. However, A* with heuristics level sum is 10 x faster than the max level and set level heuristics.

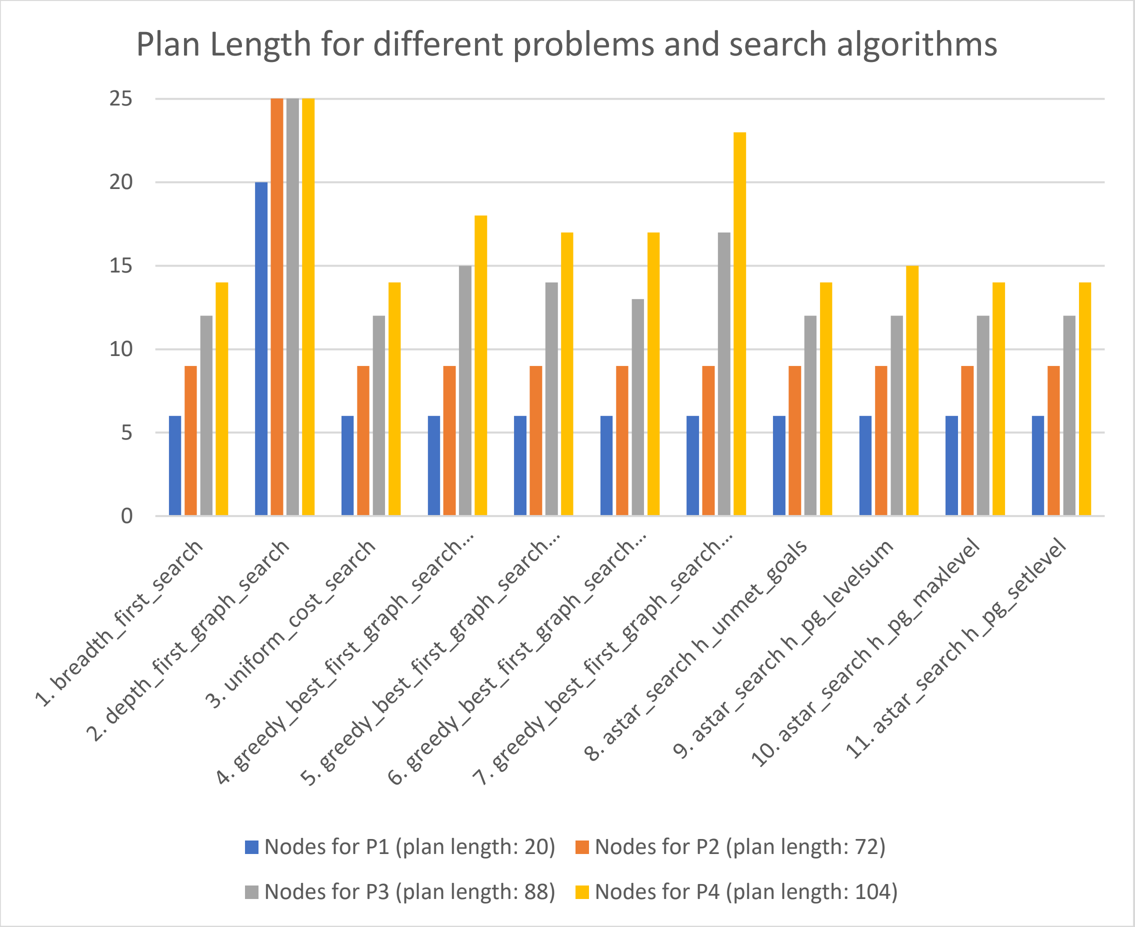

Running Time for problems with different number of actions
100000
10000
1000
100
10
1
0.1
0.01
0.001
0.0001
0
20
40
60
80
100
120
1. breadth_first_search
2. depth_first_graph_search
3. uniform_cost_search
4. greedy_best_first_graph_search h_unmet_goals
5. greedy_best_first_graph_search h_pg_levelsum
6. greedy_best_first_graph_search h_pg_maxlevel
7. greedy_best_first_graph_search h_pg_setlevel
8. astar_search h_unmet_goals
9. astar_search h_pg_levelsum
10. astar_search h_pg_maxlevel
11. astar_search h_pg_setlevel

**Analysis:** The impact of selection of algorithms against length of plans

The plan length is an important measure to determine which algorithm provides the optimum (or closest to) solution.

As expected, the A* algorithms provide the smallest length of plans, providing the ideal solutions. Depth first algorithm is completely unsuitable due to the long length of plan, and uniform cost search performs well.





Questions:

- Which algorithm or algorithms would be most appropriate for planning in a very restricted domain (i.e., one that has only a few actions) and needs to operate in real time?

  Greedy Best First Graph Search with heuristic unmet goals is very fast and provides a good results for the plan length as long as the number of actions remains small.

- Which algorithm or algorithms would be most appropriate for planning in very large domains (e.g., planning delivery routes for all UPS drivers in the U.S. on a given day)

  Uniform cost search and all A* algorithms using heuristics 'unmet goals' and planning graph 'level sum' provide good solutions as it is assumed that the cost of additional plan length is high (extra number of trips for UPS drivers) and high amount of computing power is available.

- Which algorithm or algorithms would be most appropriate for planning problems where it is important to find only optimal plans?

  A* algorithms provide optimal plans even though these can be computationally expensive. The A* algorithm 'max level' provides an optimal plan and appears to be more computationally efficient than 'set level' heuristic.