Juan Kikushima

Report

Planning Agent

The following charts come from the data acquired from the `run_search.py` script. All the search algorithms were run on the four different Air Cargo problems:

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

Problem	Number of Actions
Air Cargo Problem 1	20
Air Cargo Problem 2	72
Air Cargo Problem 3	88
Air Cargo Problem 4	104

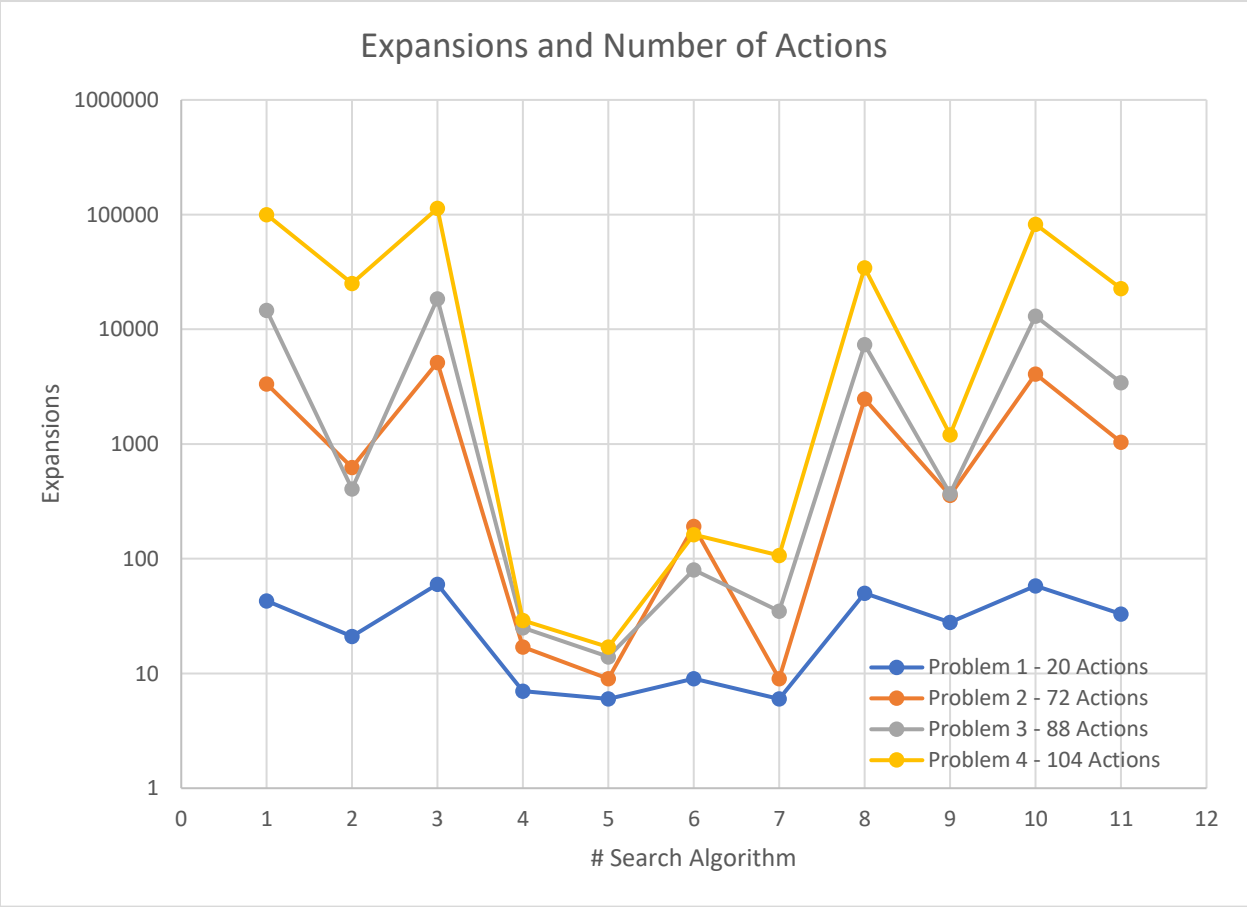

Figure 1

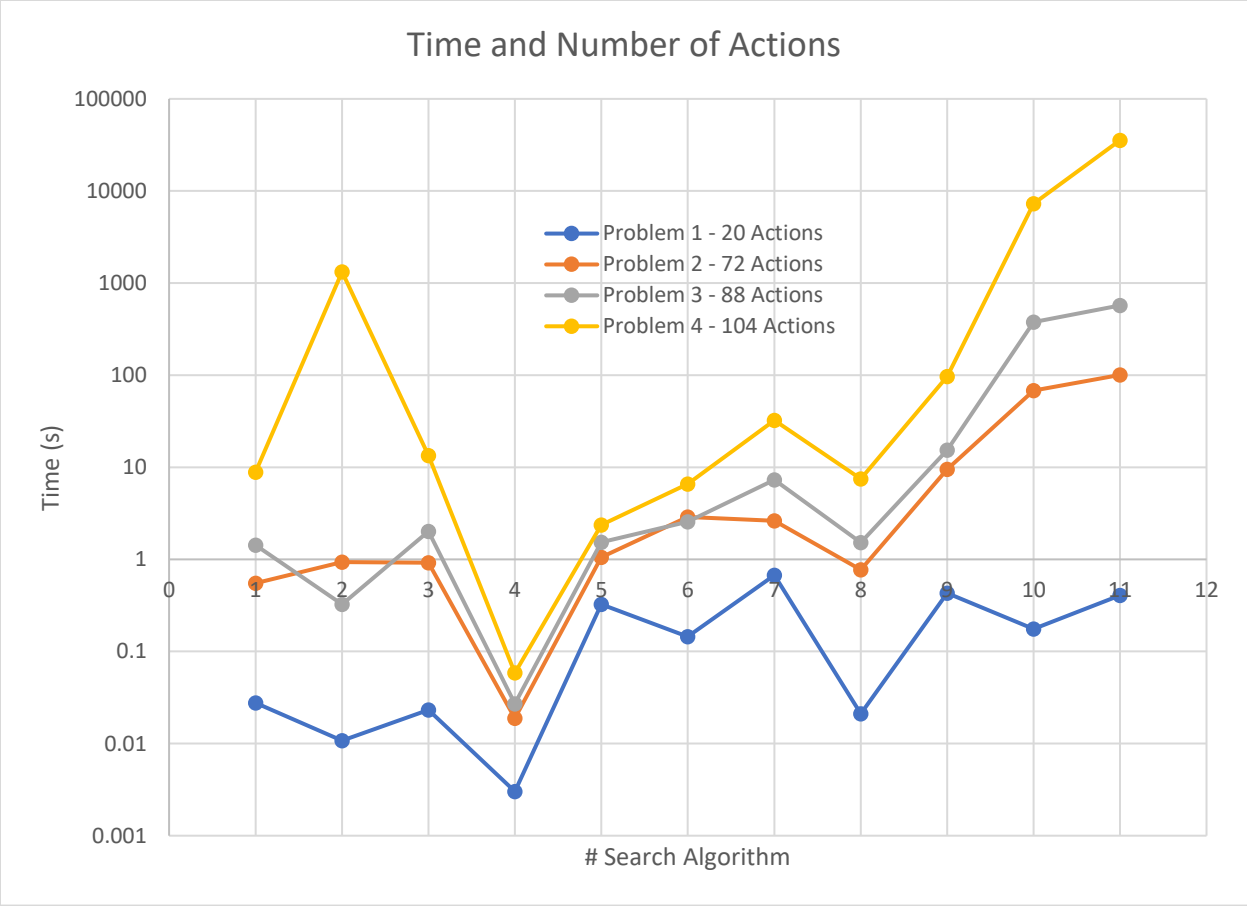
Time and Number of Actions
100000
10000
1000
100
10
1
0.1
0.01
0.001
Time (s)
0
1
2
3
4
5
6
7
8
9
10
11
12
Search Algorithm
Problem 1 - 20 Actions
Problem 2 - 72 Actions
Problem 3 - 88 Actions
Problem 4 - 104 Actions

Figure 2

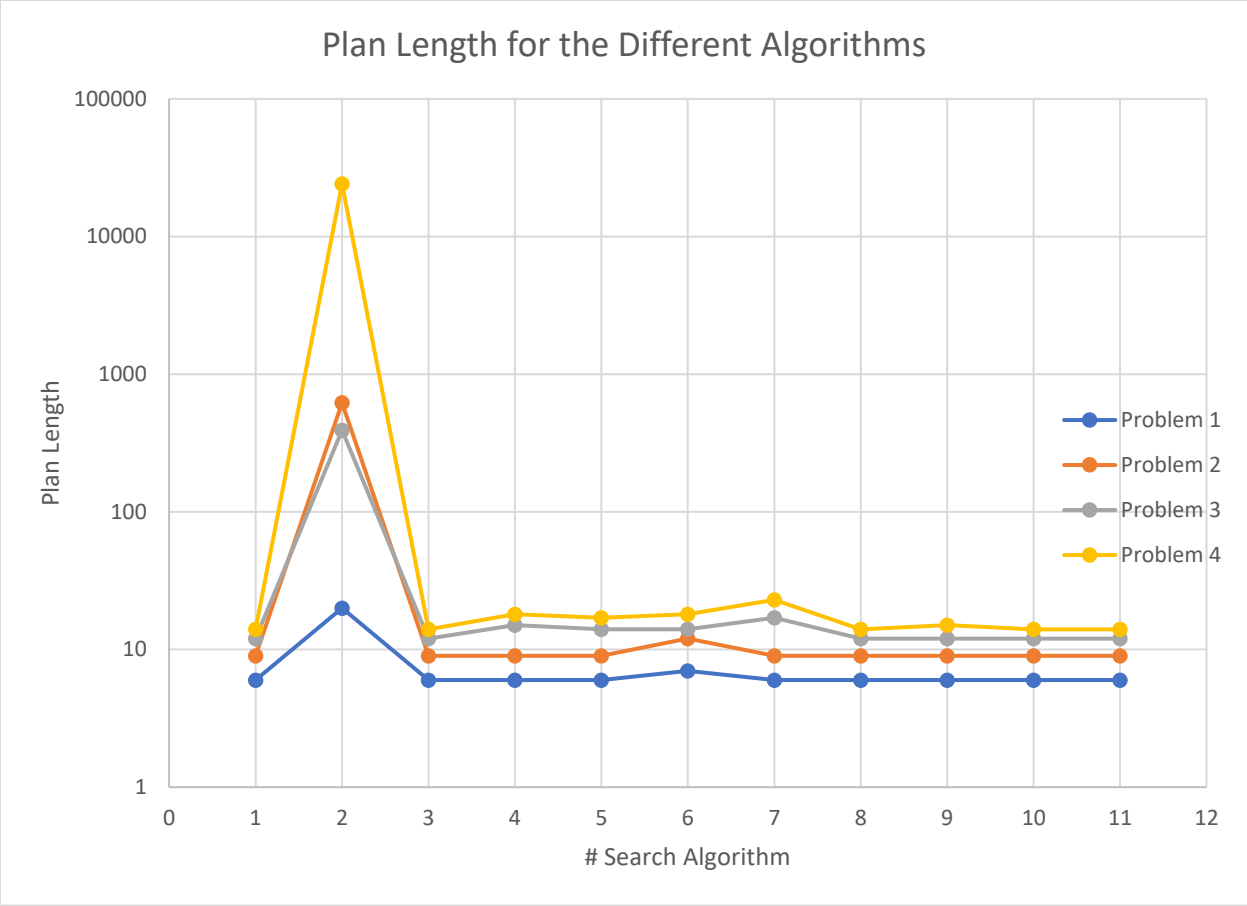

Figure 3

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?

As we can see in Figure 2, the Greedy Best-First Graph Search using the unmet goals heuristic is the fastest, even for the problem with the larger number of actions. The uninformed search strategies might be used if you have a problem with few actions, but as can be seen on the same Figure 1, for Problem 2 which consists of 72 actions there is significant increase in time from the Greedy Best-First Search algorithm and the uninformed search methods.

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)

Greedy Best-First Graph Search using the unmet goals heuristic is fast even for Problem 4 which is the one with the largest number of actions, although Greedy Best-First Graph Search using the levelsum heuristic results in less expansions as can be seen in Figure 1. Depth-First Search is definitively not recommended as both the time and number of expansions are significantly higher than the other algorithms, especially for the Problem 4, the one with the largest number of actions.

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

Any of the informed search algorithms will return an optimal plan, however we can see from Figure 1 that Greedy Best-First Graph Search using the unmet goals heuristic and A* Search using the unmet goals heuristic are the fastest, probably because the other heuristic takes longer to calculate as they become more complicated. As we can see DFS is nonoptimal for all experiments and the time as well as the number of expansions and the path length grow significantly for large domains when using uninformed search algorithm.