Project 2: Planning

1 Analysis of Problem 1 and Problem 2

All of the search algorithms are run to solve problem 1 and problem 2.

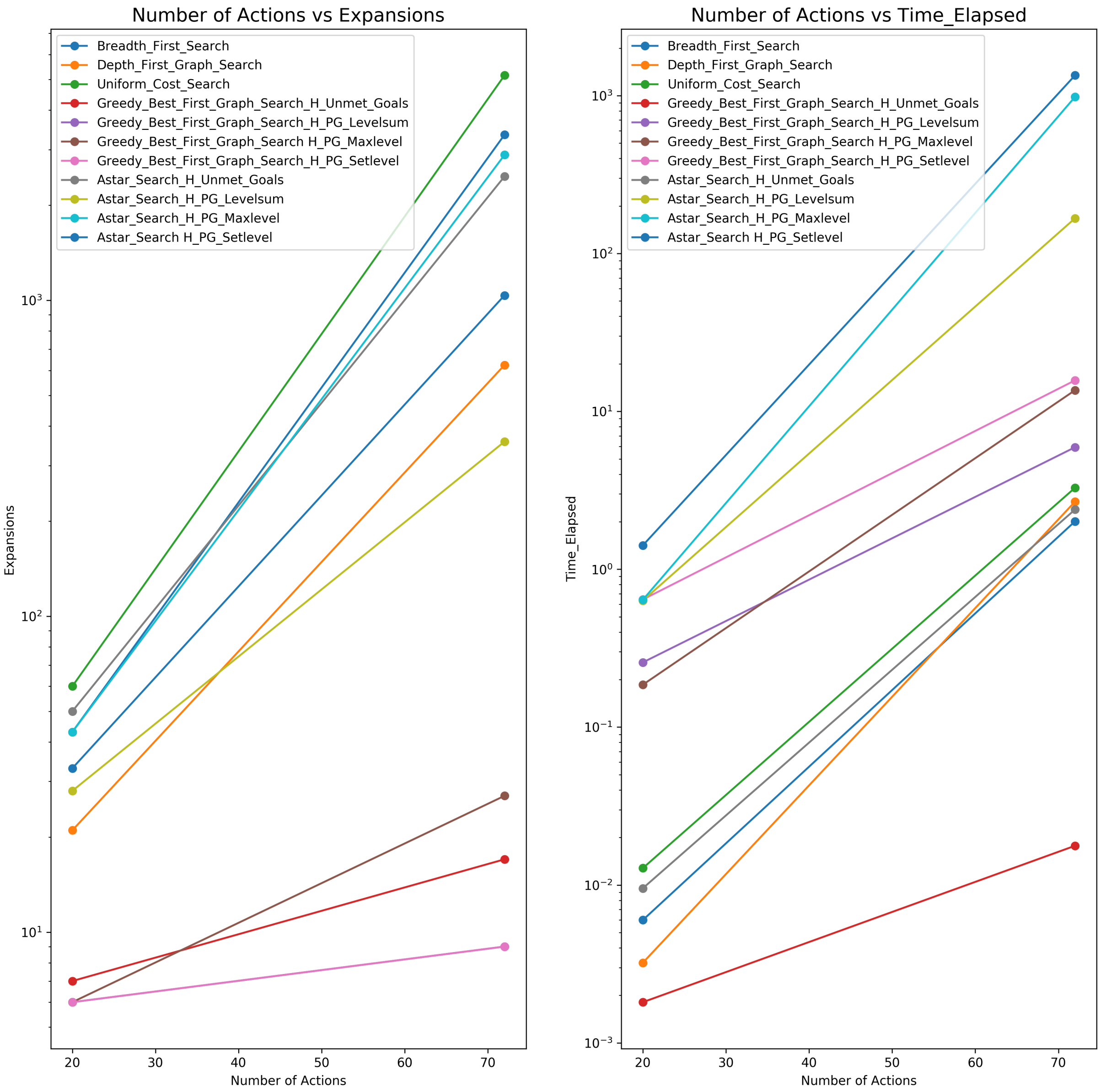

Figure 1 Number of actions vs expansions and number of actions vs search time comparisons for all algorithms (Problem 1 and Problem 2)

1.1 Comparing the number of nodes expanded against number of actions in the domain

By comparing the number of nodes expanded against number of actions in the domain for all the algorithms, we can see from Figure 1 that the following algorithms have comparable growth trend as the problem complexity increases:

- *Uniform_Cost_Search*
- *Breadth_First_Search*
- *Astar_Search_H_PG_Maxlevel*
- *Astar_Search_H_Unmet_Goals*

- *Astar_Search H_PG_Setlevel*
- *Depth_First_Graph_Search*
- *Astar_Search_H_PG_Levelsum*

Meanwhile, the following algorithms have relatively lower growth trend in the number of nodes expanded as the problem complexity increases than the aforementioned algorithms:

- *Greedy_Best_First_Graph_Search_H_PG_Maxlevel*
- *Greedy_Best_First_Graph_Search_H_Unmet_Goals*
- *Greedy_Best_First_Graph_Search_H_PG_Setlevel*
- *Greedy_Best_First_Graph_Search_H_PG_Levelsum*

1.2 Comparing the search time against number of actions in the domain

By comparing the search time against number of actions in the domain for all the algorithms, we can see from Figure 1 that the following algorithms have comparable growth trend as the problem complexity increases:

- *Breadth_First_Search*
- *Astar_Search_H_PG_Maxlevel*
- *Astar_Search_H_PG_Levelsum*
- *Uniform_Cost_Search*
- *Depth_First_Graph_Search*
- *Astar_Search_H_Unmet_Goals*
- *Astar_Search H_PG_Setlevel*

Meanwhile, the following algorithms have substantially lower growth trend in the search time as the problem complexity increases than the aforementioned algorithms:

- *Greedy_Best_First_Graph_Search_H_PG_Maxlevel*
- *Greedy_Best_First_Graph_Search_H_Unmet_Goals*
- *Greedy_Best_First_Graph_Search_H_PG_Setlevel*
- *Greedy_Best_First_Graph_Search_H_PG_Levelsum*

1.3 Comparing the length of the plans against number of actions in the domain

By comparing the search time against number of actions in the domain for all the algorithms, we can see from Table 1 that all algorithms generate the same length of plans for both problem 1 and problem 2, except the *Depth_First_Graph_Search*, which has substantially greater plan length compared to the other algorithms.

Table 1 The length of the plans returned by each algorithm on all search problems

Algorithm	Problem 1	Problem 2
Breadth_First_Search	6	9
Depth_First_Graph_Search	20	619
Uniform_Cost_Search	6	9
Greedy_Best_First_Graph_Search_H_Unmet_Goals	6	9
Greedy_Best_First_Graph_Search_H_PG_Levelsum	6	9
Greedy_Best_First_Graph_Search_H_PG_Maxlevel	6	9
Greedy_Best_First_Graph_Search_H_PG_Setlevel	6	9
Astar_Search_H_Unmet_Goals	6	9
Astar_Search_H_PG_Levelsum	6	9
Astar_Search_H_PG_Maxlevel	6	9
Astar_Search_H_PG_Setlevel	6	9

From the comparisons, it can be inferred that the *Depth_First_Graph_Search* should be excluded for problem 3 and 4, mainly because of its inability to find the optimal plan.

2 Analysis of Problem 3 and Problem 4

For problem 3 and 4, *Depth_First_Graph_Search* is excluded. For problem 4, *Astar_Search_H_PG_Maxlevel* and *Astar_Search_H_PG_Setlevel* are excluded because of considerable running time (even with pypy3). Figure 2 is presented to compare the results of the selected algorithms for all problems.

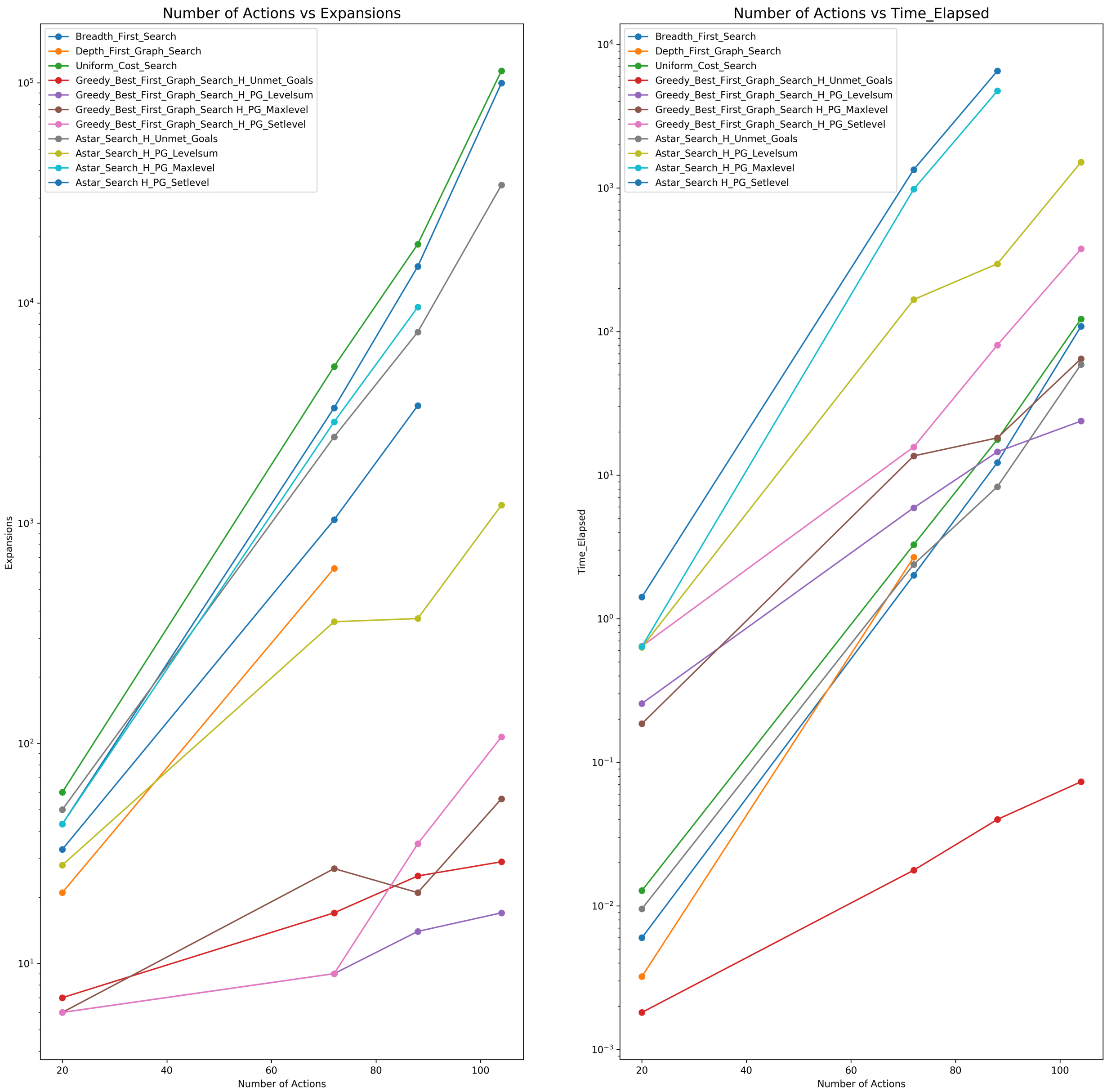

Figure 2 Number of actions vs expansions and number of actions vs search time comparisons for all algorithms (Problem 1, Problem 2, Problem 3, and Problem 4)

2.1 Comparing the number of nodes expanded against number of actions in the domain

By comparing the number of nodes expanded against number of actions in the domain for all the algorithms, we can see from Figure 2 that the following algorithms have comparable growth trend as the problem complexity increases:

- *Uniform_Cost_Search*
- *Breadth_First_Search*
- *Astar_Search_H_PG_Maxlevel*
- *Astar_Search_H_Unmet_Goals*
- *Astar_Search H_PG_Setlevel*
- *Depth_First_Graph_Search*
- *Astar_Search_H_PG_Levelsum*

Meanwhile, the following algorithms have relatively lower growth trend in the number of nodes expanded as the problem complexity increases than the aforementioned algorithms:

- *Greedy_Best_First_Graph_Search_H_PG_Maxlevel*
- *Greedy_Best_First_Graph_Search_H_Unmet_Goals*

- *Greedy_Best_First_Graph_Search_H_PG_Levelsum*

Greedy_Best_First_Graph_Search_H_PG_Setlevel seems to have lower growth trend when the number of actions is lower, but the growth trend increases when the number of actions is higher.

2.2 Comparing the search time against number of actions in the domain

By comparing the search time against number of actions in the domain for all the algorithms, we can see from Figure 2 that the following algorithms have comparable growth trend as the problem complexity increases:

- *Breadth_First_Search*
- *Astar_Search_H_PG_Maxlevel*
- *Astar_Search_H_PG_Levelsum*
- *Uniform_Cost_Search*
- *Depth_First_Graph_Search*
- *Astar_Search_H_Unmet_Goals*
- *Astar_Search H_PG_Setlevel*

Meanwhile, the following algorithms have substantially lower growth trend in the search time as the problem complexity increases than the aforementioned algorithms:

- *Greedy_Best_First_Graph_Search_H_PG_Maxlevel*
- *Greedy_Best_First_Graph_Search_H_Unmet_Goals*
- *Greedy_Best_First_Graph_Search_H_PG_Setlevel*
- *Greedy_Best_First_Graph_Search_H_PG_Levelsum*

Greedy_Best_First_Graph_Search_H_PG_Setlevel seems to have lower growth trend when the number of actions is lower, but the growth trend increases when the number of actions is higher.

2.3 Comparing the length of the plans against number of actions in the domain

From Table 2, it can be seen that *Breadth_First_Search, Uniform_Cost_Search*, and *Astar_Search_H_Unmet_Goals* are able to find the most optimum plan compared to the other algorithms in all problems. *Astar_Search_H_PG_Levelsum* is also an algorithm that is able to find the most optimum plan.

Table 2 The length of the plans returned by each algorithm on all search problems

Algorithm	Problem 1	Problem 2	Problem 3	Problem 4
Breadth_First_Search	6	9	12	14
Depth_First_Graph_Search	20	619	None	None
Uniform_Cost_Search	6	9	12	14
Greedy_Best_First_Graph_Search_H_Unmet_Goals	6	9	15	18
Greedy_Best_First_Graph_Search_H_PG_Levelsum	6	9	14	17
Greedy_Best_First_Graph_Search_H_PG_Maxlevel	6	9	13	17
Greedy_Best_First_Graph_Search_H_PG_Setlevel	6	9	17	23
Astar_Search_H_Unmet_Goals	6	9	12	14
Astar_Search_H_PG_Levelsum	6	9	12	15
Astar_Search_H_PG_Maxlevel	6	9	12	None
Astar_Search_H_PG_Setlevel	6	9	12	None

3 Answers to Questions

3.1 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_H_Unmet_Goals is the best algorithm to perform in a very restricted domain because of its fast performance and the ability to find the most optimum plan when the number of actions is small.

3.2 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)

Astar_Search_H_Unmet_Goals because of its ability to find the most optimum plan in short amount of time (in large domains).

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

Breadth_First_Search, Uniform_Cost_Search, and *Astar_Search_H_Unmet_Goals*