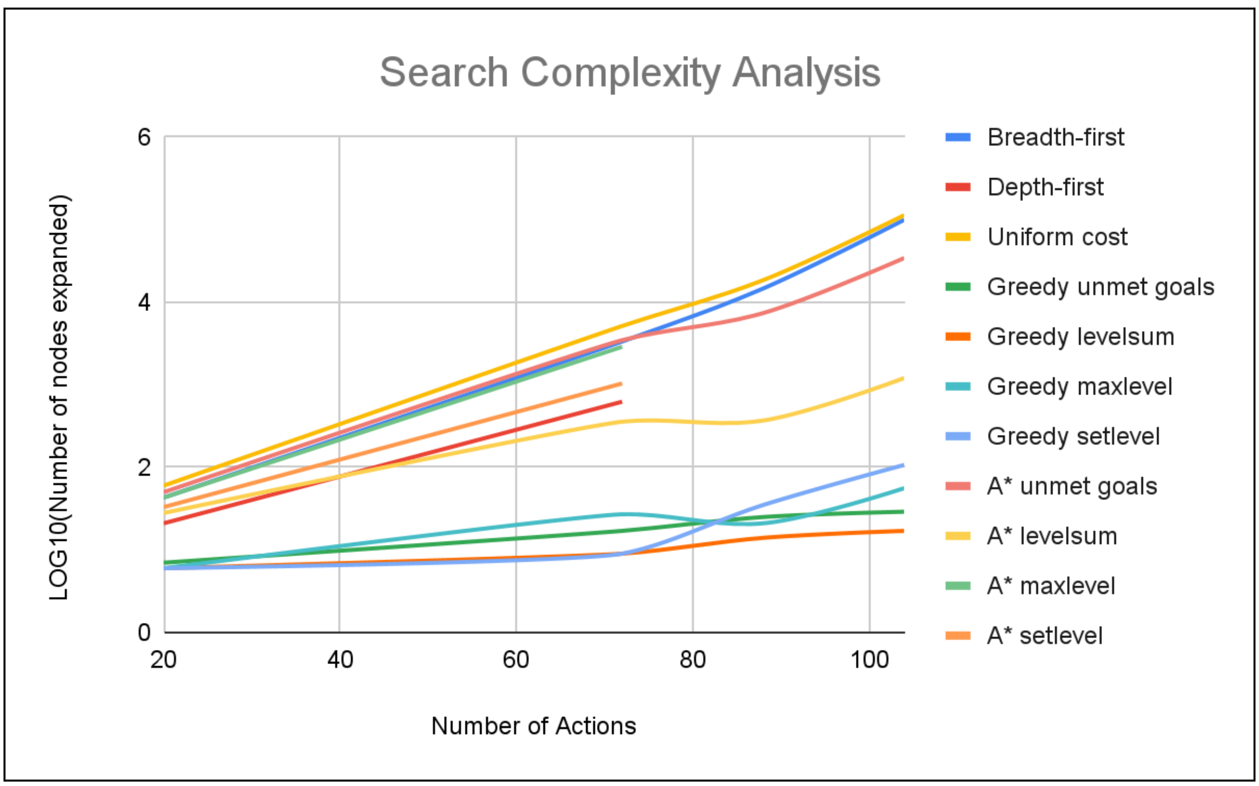
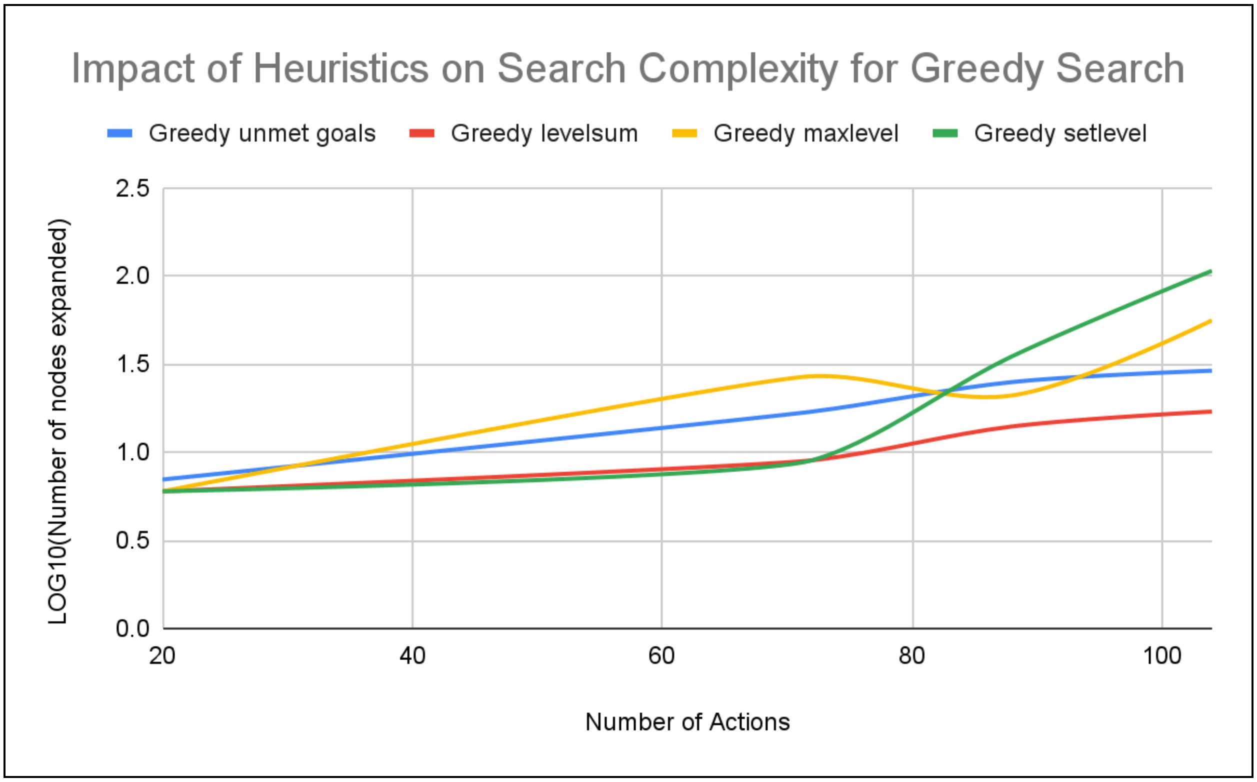
Summary

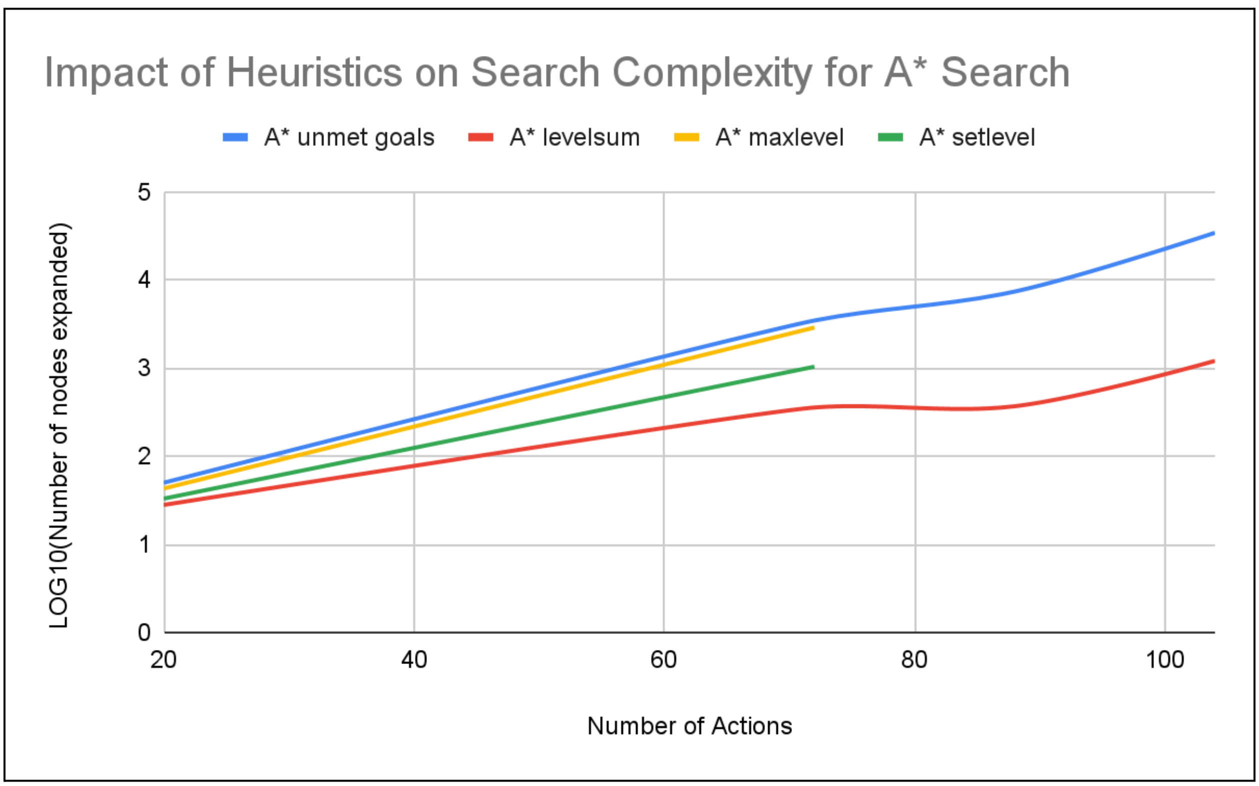
1. Ran uninformed search algorithms (breadth-first, depth-first, uniform cost) and informed search algorithms (greedy best first graph search, A* search) for the air-cargo problem and recorded data to analyze complexity, search completion time and plan length for different search algorithms and heuristics (in case of informed search).
2. Ran all of the search algorithms on the first two problems and observed that among the uninformed search algorithms, breadth-first and uniform cost search algorithms were outperforming depth-first search in terms of length of the search plan generated. Hence, depth-first search was not used for problem 3 and problem 4.
3. For problem 3 and 4, two uninformed search algorithms - breadth-first and uniform cost are used.
4. Both informed search algorithms are used for problem 3 and 4 but all the given heuristics (unmet goals, levelsum, maxlevel, setlevel) are only used for greedy best first graph search.
5. For A* search only two heuristics - unmet goals and levelsum are used as was the requirement in this project but also because those heuristics were observed to be computationally heavy by quite a margin as compared to other heuristics.

1.Analyze Search Complexity

		Problem 1	Problem 2	Problem 3	Problem 4
	Number of actions	20	72	88	104
Number of new node expansions	**Breadth-first**	43	3343	14663	99736
	Depth-first	21	624	-	-
	Uniform cost	60	5154	18510	113339
	Greedy unmet goals	7	17	25	29
	Greedy levelsum	6	9	14	17
	Greedy maxlevel	6	27	21	56
	Greedy setlevel	6	9	35	107
	A* unmet goals	50	3467	7388	34330
	A* levelsum	28	357	369	1208
	A* maxlevel	43	2887	-	-
	A* setlevel	33	1037	-	-




Impact of Heuristics on Search Complexity for Greedy Search
Greedy unmet goals
Greedy levelsum
Greedy maxlevel
Greedy setlevel
LOG10(Number of nodes expanded)
2.5
2.0
1.5
1.0
0.5
0.0
20
40
60
80
100
Number of Actions

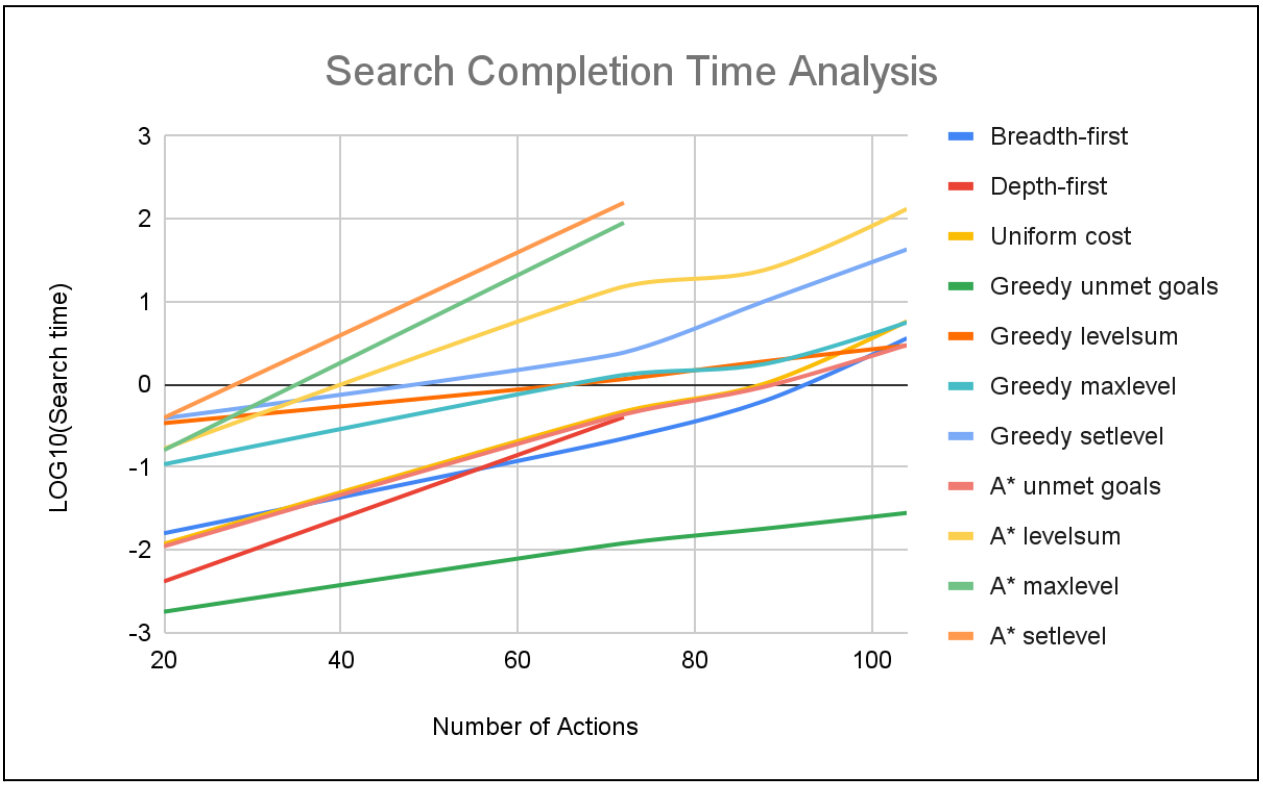

Impact of Heuristics on Search Complexity for A* Search
A* unmet goals
A* levelsum
A* maxlevel
A* setlevel
LOG10(Number of nodes expanded)
5
4
3
2
1
0
20
40
60
80
100
Number of Actions

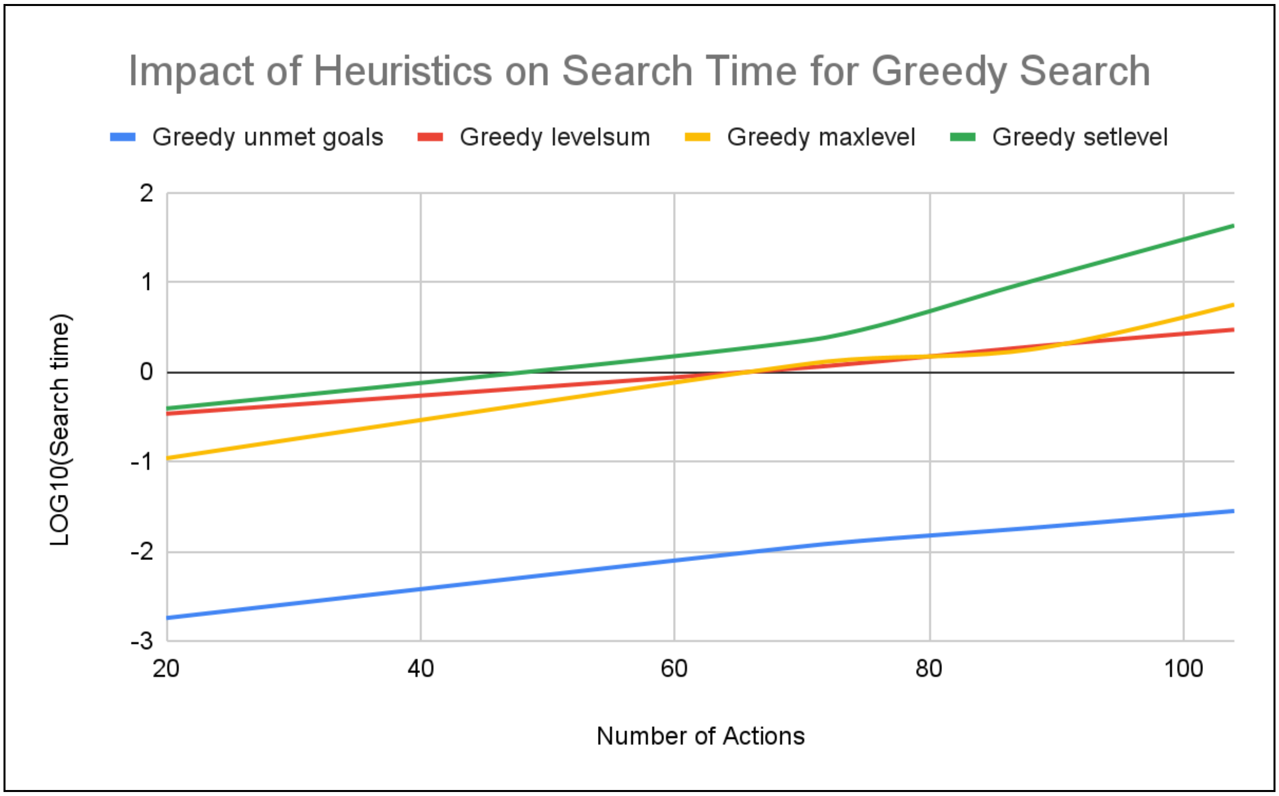
Observations:

1. In general, the above search complexity analysis graph trend shows that as the domain size increases i.e. as the number of actions in the domain increase, the search complexity increases for any search algorithm, in most cases multiple orders of magnitude.
2. Purely, from a search complexity point of view, the ***greedy best first graph search algorithm*** performs the best i.e. expands the least amount of new nodes to find a solution, whereas, for uniform cost search and breadth-first search, the complexity grows at a high rate as the problem size increases.
3. Different heuristics affect the search complexity as well. It can be seen in the above plots for greedy best first graph search and A* search
4. Search complexity is comparatively lower when ***levelsum*** heuristics are used for any informed search algorithms.
5. ***maxlevel*** and ***setlevel*** heuristics increase the search complexities by multiple order of magnitude

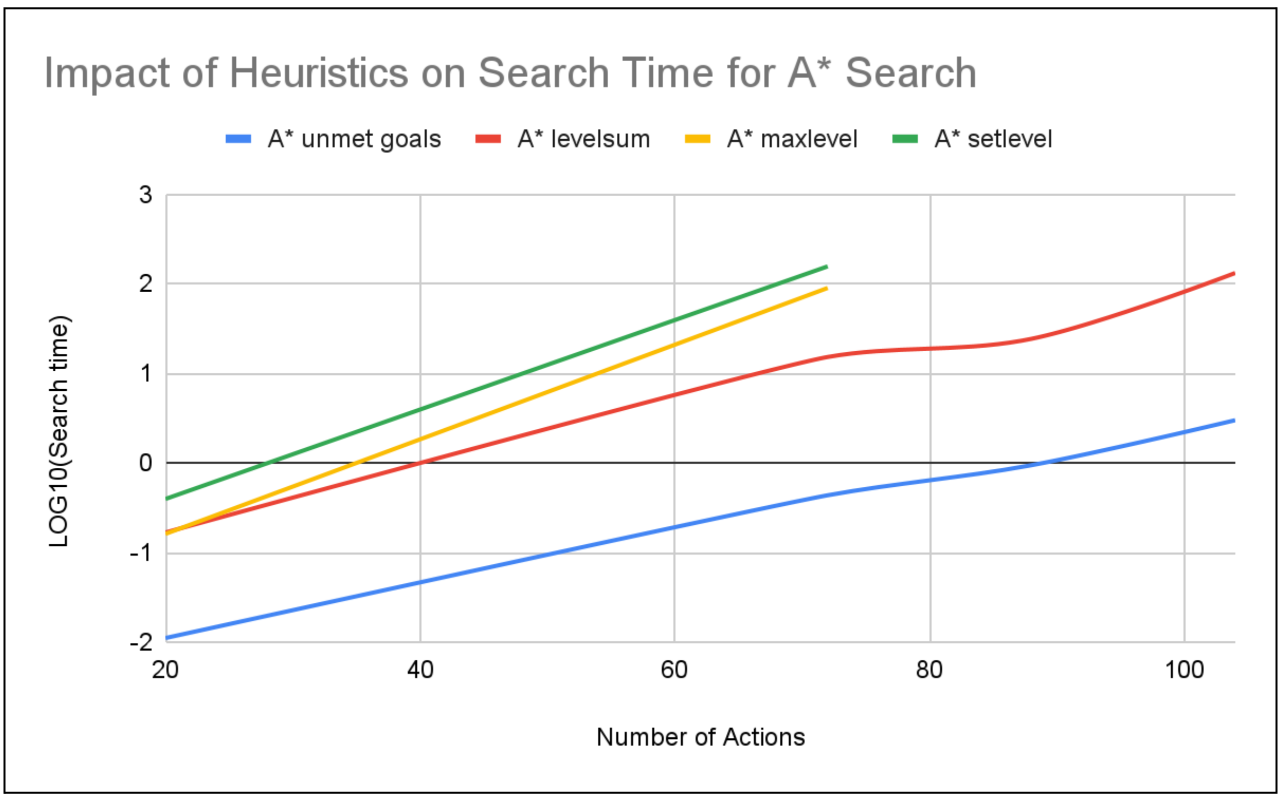
2. Analyze Search Time

		Problem 1	Problem 2	Problem 3	Problem 4
	Number of actions	20	72	88	104
Search time	**Breadth-first**	0.016	0.223	0.640	3.622
	Depth-first	0.004	0.405	-	-
	Uniform cost	0.012	0.473	1.032	5.795
	Greedy unmet goals	0.002	0.012	0.018	0.028
	Greedy levelsum	0.341	1.164	1.908	2.957
	Greedy maxlevel	0.109	1.306	1.777	5.631
	Greedy setlevel	0.391	2.432	10.219	42.842
	A* unmet goals	0.011	0.436	0.945	3.004
	A* levelsum	0.168	15.262	24.344	132.343
	A* maxlevel	0.161	90.043	-	-
	A* setlevel	0.396	156.884	-	-





Impact of Heuristics on Search Time for Greedy Search
Greedy unmet goals
Greedy levelsum
Greedy maxlevel
Greedy setlevel
LOG10(Search time)
2
1
0
-1
-2
-3
20
40
60
80
100
Number of Actions



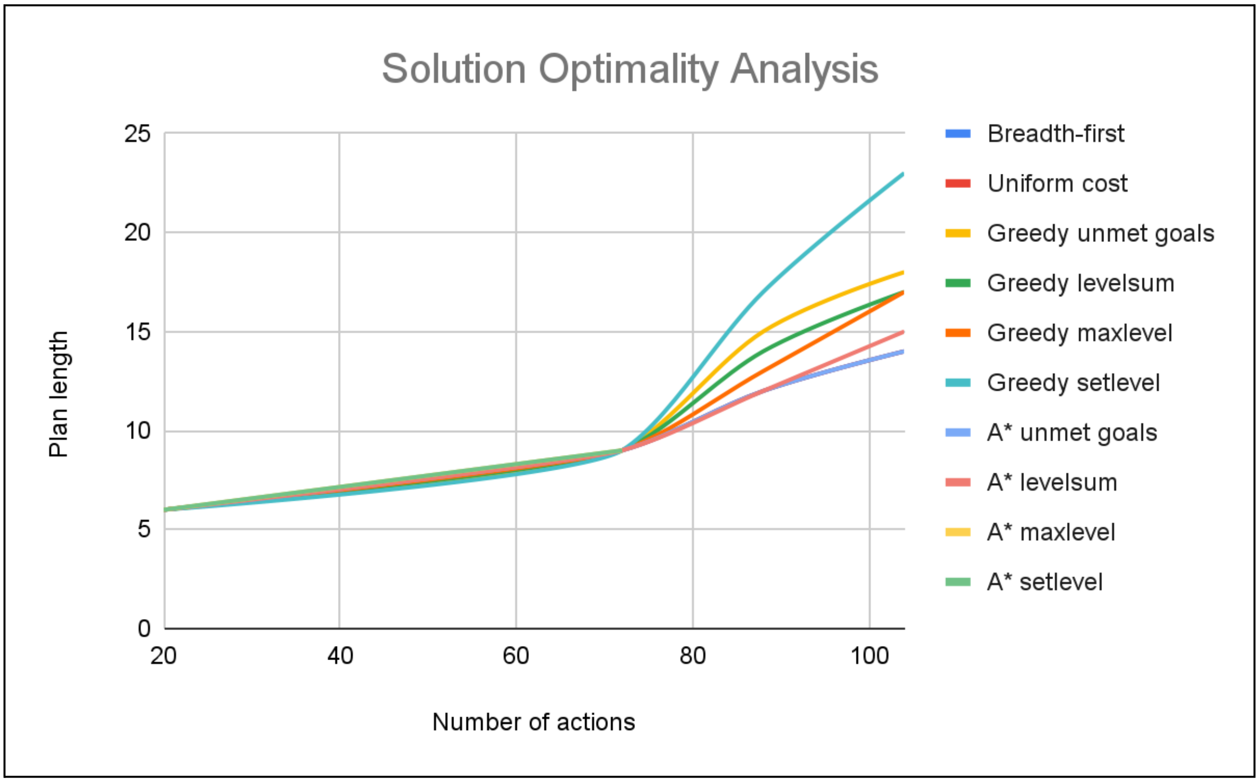
Impact of Heuristics on Search Time for A* Search
A* unmet goals
A* levelsum
A* maxlevel
A* setlevel
LOG10(Search time)
3
2
1
0
-1
-2
20
40
60
80
100
Number of Actions

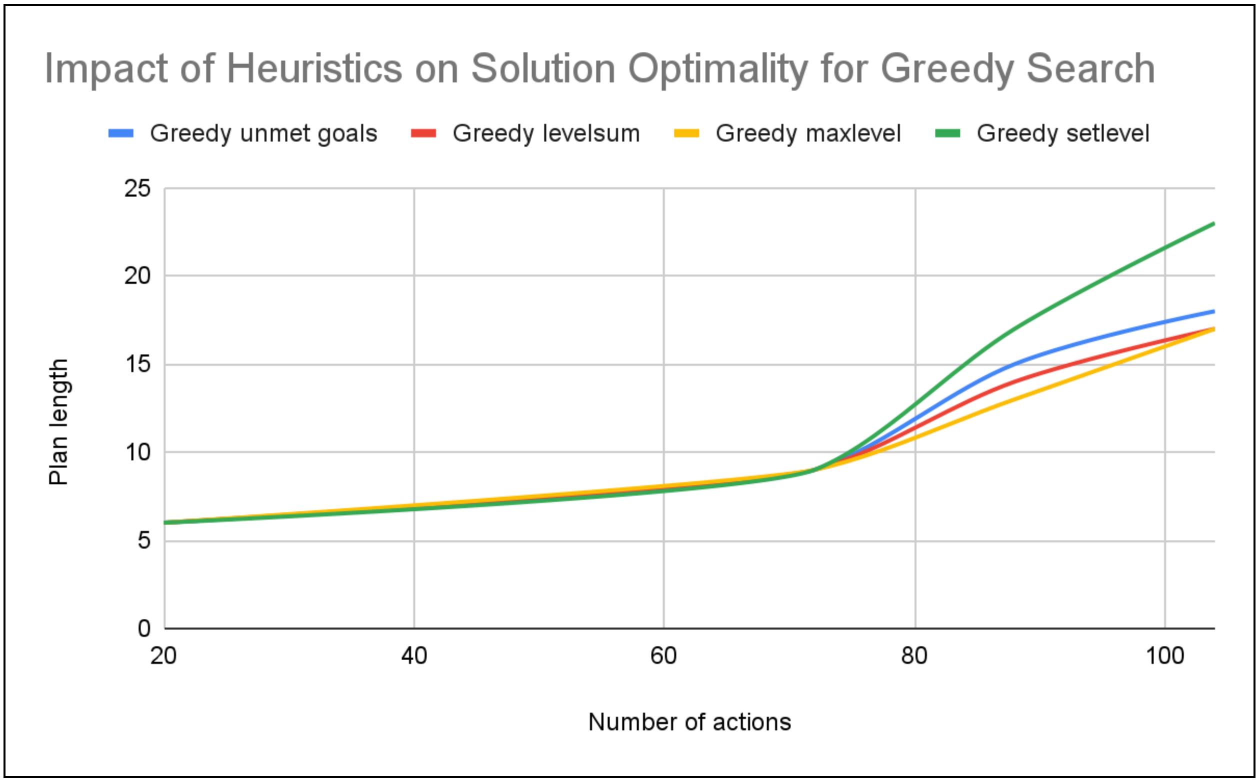
Observations:

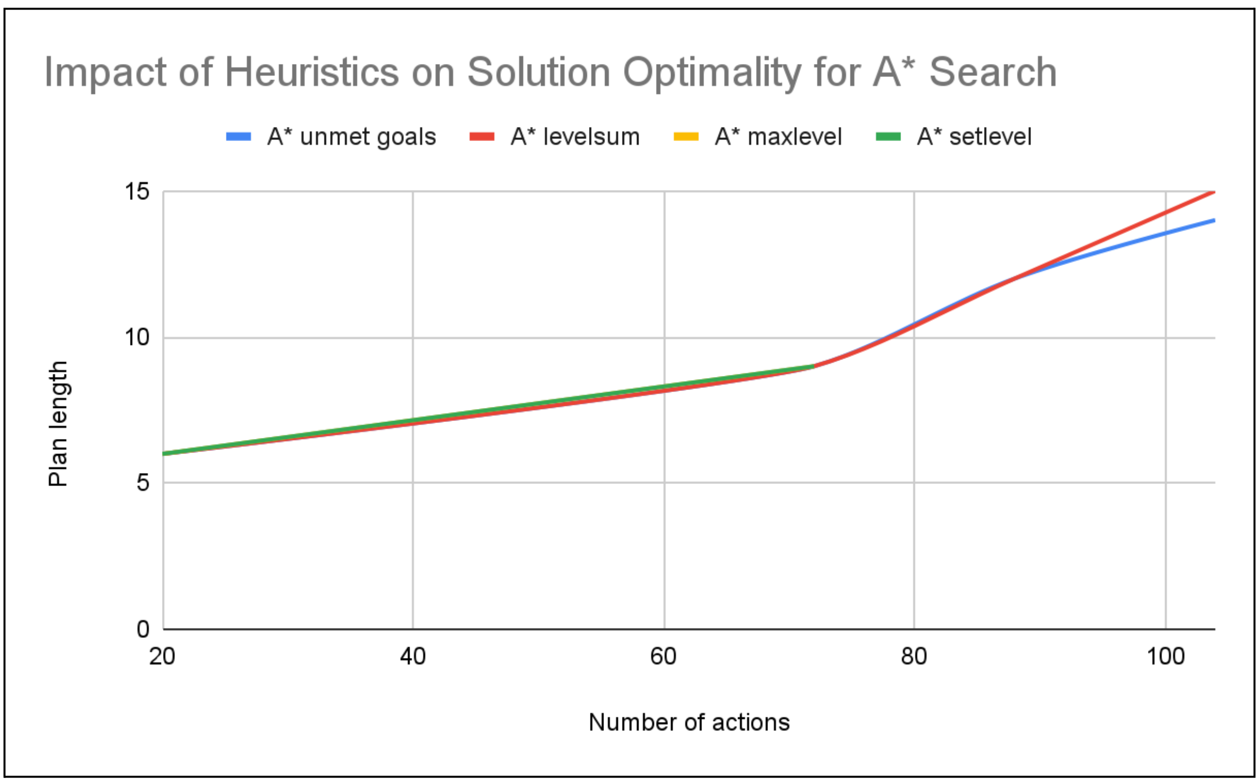
1. The search time analysis graph trend shows that as the domain size increases i.e. as the number of actions in the domain increase, the search time increases for any search algorithm, in most cases multiple orders of magnitude.
2. Search time for A* search with most heuristics (all except for unmet goals) show a very high search time indicating it to be the most computationally heavy algorithm.
3. In general, the informed search algorithms (for most of the heuristics) have higher search times as compared to the uninformed search algorithms for a given problem.
4. Different heuristics affect the search time as well. It can be seen in the above plots for greedy best first graph search and A* search
5. Search time is considerably lower when ***unmet goals*** heuristics are used for any informed search algorithms.
6. A significant impact on search time is observed when greedy first best graph search algorithm is used with unmet goals heuristics. Data shows that among the available search algorithms, ***greedy first best graph search algorithm*** when used with ***unmet goals*** heuristics gives the solution in the least time.
7. ***maxlevel*** and ***setlevel*** heuristics increase the search time by multiple order of magnitude

3. Analyze Optimality of Solution

		Problem 1	Problem 2	Problem 3	Problem 4
	Number of actions	20	72	88	104
Search time	**Breadth-first**	6	9	12	14
	Depth-first	20	619	-	-
	Uniform cost	6	9	12	14
	Greedy unmet goals	6	9	15	18
	Greedy levelsum	6	9	14	17
	Greedy maxlevel	6	9	13	17
	Greedy setlevel	6	9	17	23
	A* unmet goals	6	9	12	14
	A* levelsum	6	9	12	15
	A* maxlevel	6	9	-	-
	A* setlevel	6	9	-	-




Impact of Heuristics on Solution Optimality for Greedy Search
Greedy unmet goals
Greedy levelsum
Greedy maxlevel
Greedy setlevel
Plan length
0
5
10
15
20
25
20
40
60
80
100
Number of actions


Impact of Heuristics on Solution Optimality for A* Search
A* unmet goals
A* levelsum
A* maxlevel
A* setlevel
Plan length
0
5
10
15
20
40
60
80
100
Number of actions

4. 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 search algorithm with unmet goals heuristic would be the appropriate planning algorithm to use in a restricted domain, the one that needs to operate in real time for the following reasons:

  a. It comes up with a plan in least amount of time amongst the studied search algorithms, thus making it the most suitable for real time application.
  b. Since, it is given that the domain is restricted, the optimality of solution for any search algorithm is similar so the plan generated by greedy best first search algorithm with unmet goals heuristic would be optimal or near optimal.

- *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)*

  A* search with unmet goals heuristic would be one of the appropriate planning algorithms that can be used for planning in large domains as from the experimentation performed in this project it can be inferred that although the search can get complex as the number of nodes expanded would be very high, it comes up with an optimal solution (which might be needed for company like UPS to efficiently manage its resources for planning deliveries) with not so high search times. Similar inferences also apply to ***breadth-first search*** or ***uniform cost search*** as they are able to find optimal plans as well, the trade-off being high complexity.

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

  A*, ***breadth-first*** and ***uniform cost*** search guarantee finding optimal plans for any planning problem. Hence, they would be the most appropriate for the problems that require only optimal paths.