Project 2 “Classical Planning”

# Experiment Results

I tried to run locally all combinations of problems and search algorithm just to have the full picture. The only one I was not able to run locally was the problem 4 with Astar\_h\_pg\_setlevel due to its really long time to complete (I did not installed pypy because I saw from the official web page that there are some libraries that are not fully supported or not supported at all by pypy and that I usually use, so I preferred to keep clean my local python installation and do not do this exercise combination)

## Actions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Problem 1 | Problem 2 | Problem 3 | Problem 4 |
| BFS | 20 | 72 | 88 | 104 |
| DFS | 20 | 72 | 88 | 104 |
| UCS | 20 | 72 | 88 | 104 |
| Greedy\_BFS\_h\_unmet\_goals | 20 | 72 | 88 | 104 |
| Greedy\_BFS\_h\_pg\_levelsum | 20 | 72 | 88 | 104 |
| Greedy\_BFS\_h\_pg\_maxlevel | 20 | 72 | 88 | 104 |
| Greedy\_BFS\_h\_pg\_setlevel | 20 | 72 | 88 | 104 |
| Astar\_h\_unmet\_goals | 20 | 72 | 88 | 104 |
| Astar\_h\_pg\_levelsum | 20 | 72 | 88 | 104 |
| Astar\_h\_pg\_maxlevel | 20 | 72 | 88 | 104 |
| Astar\_h\_pg\_setlevel | 20 | 72 | 88 | 104 |

## Node Expansions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Problem 1 | Problem 2 | Problem 3 | Problem 4 |
| BFS | 43 | 3343 | 14663 | 99736 |
| DFS | 21 | 624 | 408 | 25174 |
| UCS | 60 | 5154 | 18510 | 113339 |
| Greedy\_BFS\_h\_unmet\_goals | 7 | 17 | 25 | 29 |
| Greedy\_BFS\_h\_pg\_levelsum | 6 | 9 | 14 | 17 |
| Greedy\_BFS\_h\_pg\_maxlevel | 6 | 27 | 21 | 56 |
| Greedy\_BFS\_h\_pg\_setlevel | 6 | 9 | 35 | 107 |
| Astar\_h\_unmet\_goals | 50 | 2467 | 7388 | 34330 |
| Astar\_h\_pg\_levelsum | 28 | 357 | 369 | 1208 |
| Astar\_h\_pg\_maxlevel | 43 | 2887 | 9580 | 62077 |
| Astar\_h\_pg\_setlevel | 33 | 1037 | 3423 |  |

## Time to Complete (seconds)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Problem 1 | Problem 2 | Problem 3 | Problem 4 |
| BFS | 0.0039497 | 1.7106573 | 9.1515743 | 79.4936872 |
| DFS | 0.0028455 | 2.1253408 | 0.9016738 | 3792.9974517 |
| UCS | 0.0094644 | 2.6933768 | 15.4364058 | 91.5444883 |
| Greedy\_BFS\_h\_unmet\_goals | 0.0013931 | 0.0197747 | 0.0331642 | 0.0435025 |
| Greedy\_BFS\_h\_pg\_levelsum | 0.2024466 | 4.0753295 | 9.6263346 | 16.6113856 |
| Greedy\_BFS\_h\_pg\_maxlevel | 0.1285541 | 8.5296306 | 10.6825531 | 36.1015673 |
| Greedy\_BFS\_h\_pg\_setlevel | 0.5880906 | 21.8557797 | 150.489032 | 644.5688521 |
| Astar\_h\_unmet\_goals | 0.0079307 | 1.9783416 | 7.4542304 | 42.2038915 |
| Astar\_h\_pg\_levelsum | 0.4715595 | 133.3397137 | 223.7447806 | 1048.9292734 |
| Astar\_h\_pg\_maxlevel | 0.5465259 | 751.7285616 | 3901.7926626 | 49435.7638472 |
| Astar\_h\_pg\_setlevel | 1.3718278 | 1943.9594379 | 11521.649492 |  |

## Plan Length

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Problem 1 | Problem 2 | Problem 3 | Problem 4 |
| BFS | 6 | 9 | 12 | 14 |
| DFS | 20 | 619 | 392 | 24132 |
| UCS | 6 | 9 | 12 | 14 |
| Greedy\_BFS\_h\_unmet\_goals | 6 | 9 | 15 | 18 |
| Greedy\_BFS\_h\_pg\_levelsum | 6 | 9 | 14 | 17 |
| Greedy\_BFS\_h\_pg\_maxlevel | 6 | 9 | 13 | 17 |
| Greedy\_BFS\_h\_pg\_setlevel | 6 | 9 | 17 | 23 |
| Astar\_h\_unmet\_goals | 6 | 9 | 12 | 14 |
| Astar\_h\_pg\_levelsum | 6 | 9 | 12 | 15 |
| Astar\_h\_pg\_maxlevel | 6 | 9 | 12 | 14 |
| Astar\_h\_pg\_setlevel | 6 | 9 | 12 |  |

# Questions

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

# Answers

1. I would prefer the Greedy\_BFS\_h\_pg\_levelsum because it runs really fast (not faster for small problem, little bit slower but comparable) also wrt the fastest A\* search, i.e. Astar\_h\_unmet\_goals. Especially because on larger and more complex problem like the 4th the Astar\_h\_unmet\_goals is definitively slower than the Greedy\_BFS\_h\_pg\_levelsum and we can accept the not optimal (but anyhow really good) solution of the Greedy\_BFS\_h\_pg\_levelsum for the sake of speed gaining.
2. Here we have to look for the optimal solution if possible, in a less than 1 day of computation. For that reason the Astar\_h\_pg\_levelsum is the best choice because it reaches near optimal solutions with a computational time increasing as a function of the size of the problem that is the best possible (the other search strategies, optimal or near optimal ones, have a computational time resolution that grows faster with the problem size … for example the Astar\_h\_unmet\_goals seems to be better of the Astar\_h\_pg\_levelsum in both exacteness of the solution and computational timimng BUT the more the size of the problem increase the slower the search becomes)
3. Given that we do not care of time but we want only optimal solution we can chose among: BFS, UCS, and A\* with admissible heuristic, i.e. Astar\_h\_pg\_setlevel.