TDDC17 Lab 4 Report

Alim Amanzholov (aliam864)

Task 1
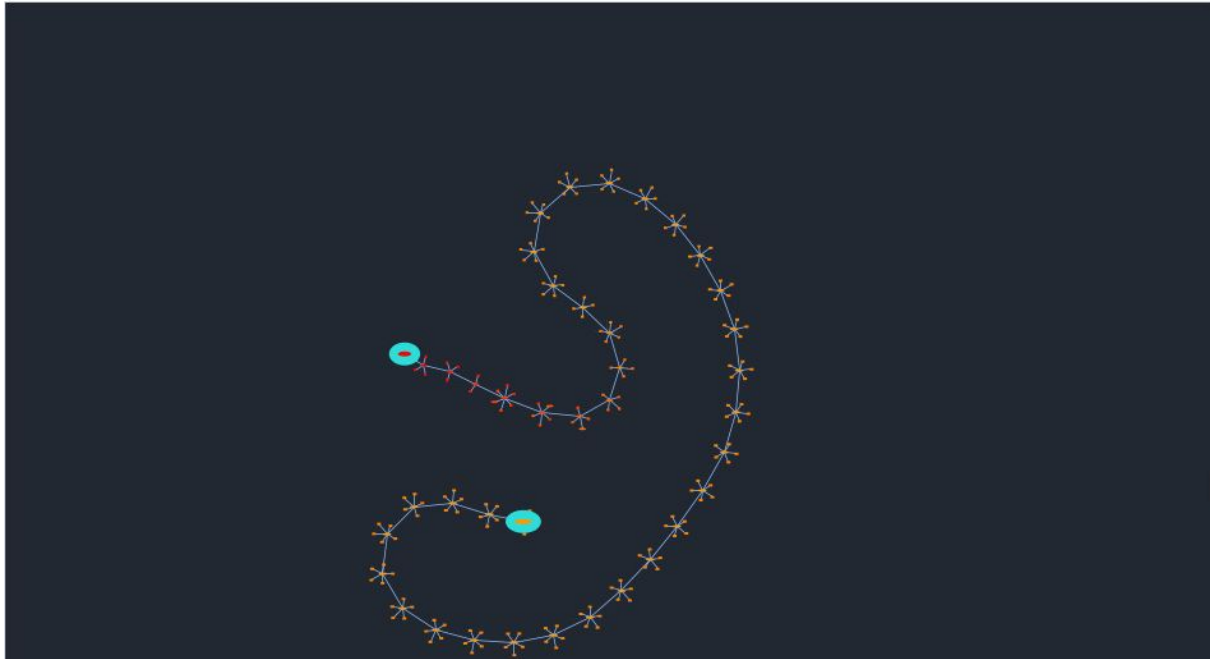
I tested the problem with the domain with FF and LAMA planners and both of them find the solution.
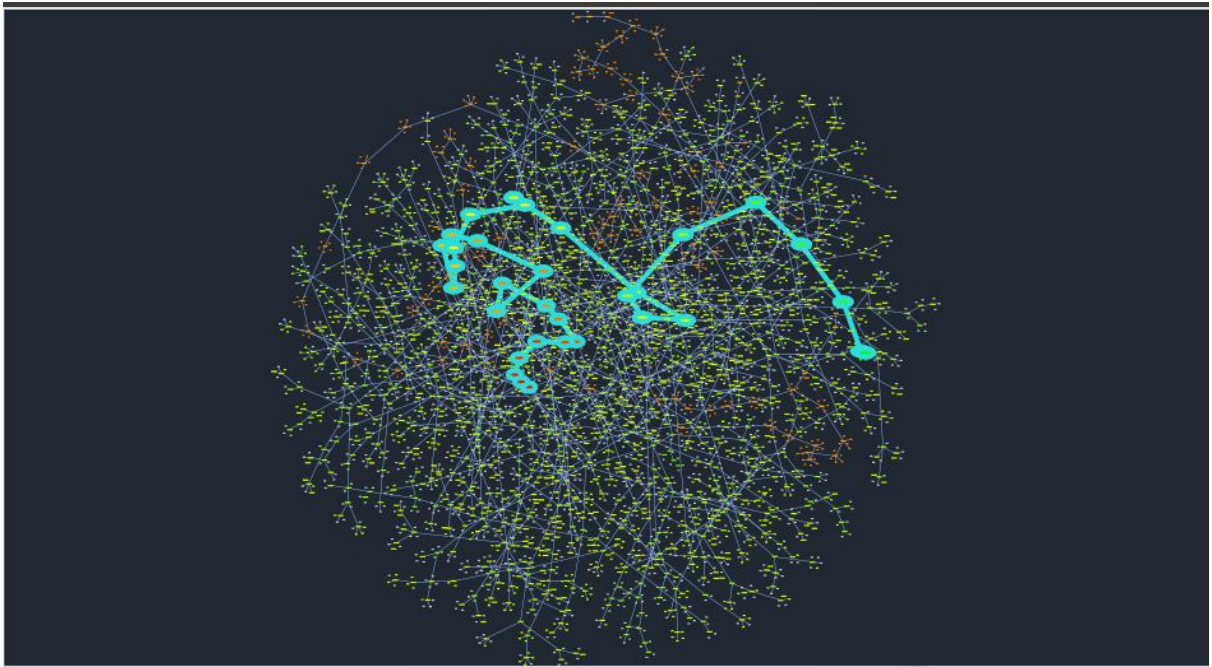
Task 2

Comparing heuristics, problem 3:

How do the graphs differ, given the same problem and search method but different heuristics? Include screenshots in your report and discuss what you see!
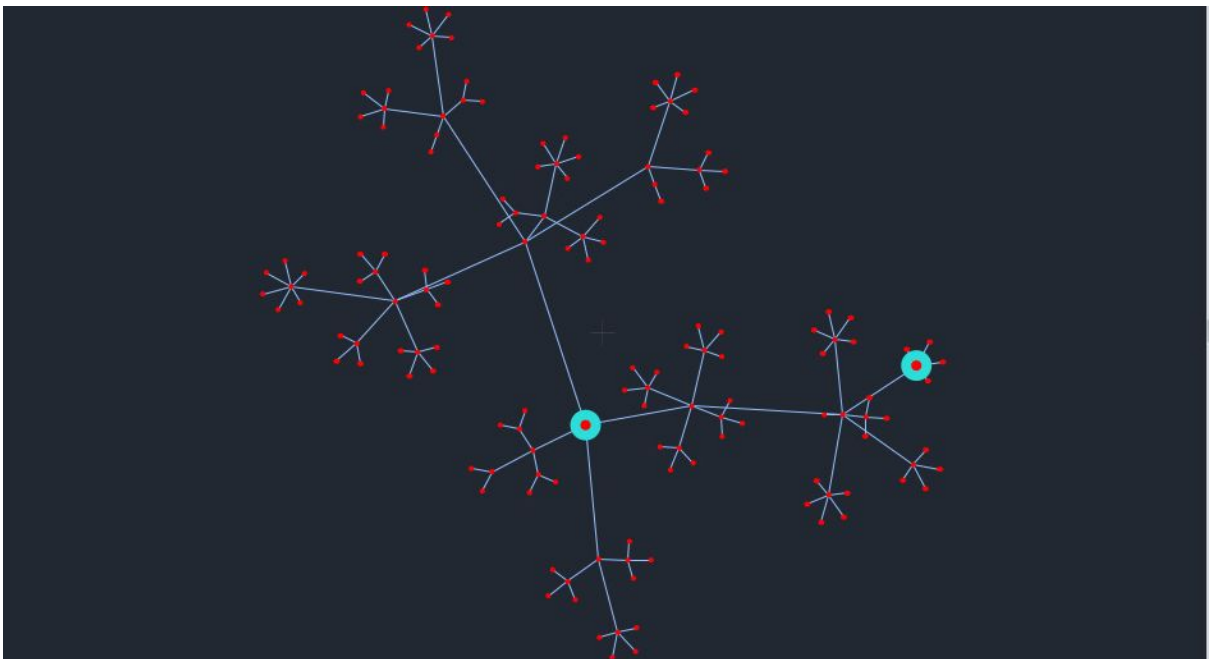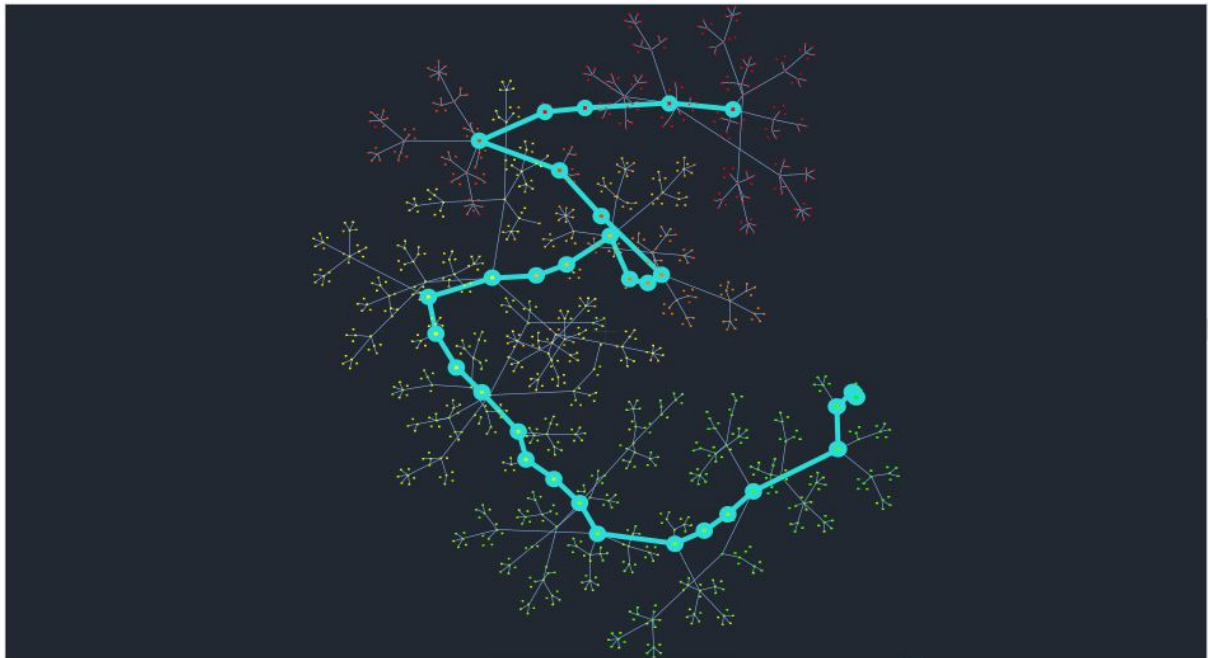
FF-heuristic (at step 40)

FF-heuristic (at the end):

GC=heuristic (at step 40)

GC-heuristic (at the end):

From what I observed when FF heuristic is used the search resembles a DFS. On the other hand, when GC heuristic is used the search resembles a BFS.

Do different configurations use different actions? Zoom in on the edges to see which actions are used and discuss the differences.

The differences between actions in two different configurations are which node gets expanded with each heuristic. Both heuristics result in expansions of different nodes at different times.

In the beginning, the FF visualization expanded nodes along a long path before it started expanding at other places. Are the nodes in this path used in the final plan?
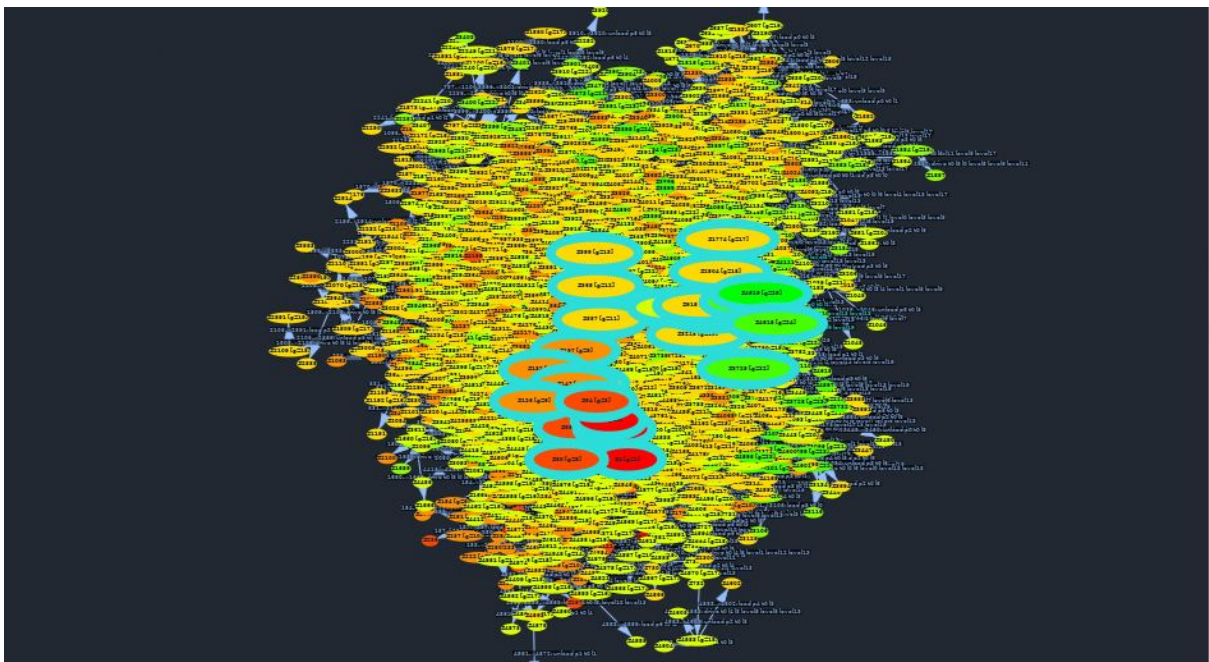
The nodes expanded along a long path before it started expanding at other places are not used in the final plan except for the first 3 nodes.

Comparing heuristics, problem 2:

FF-heuristic:
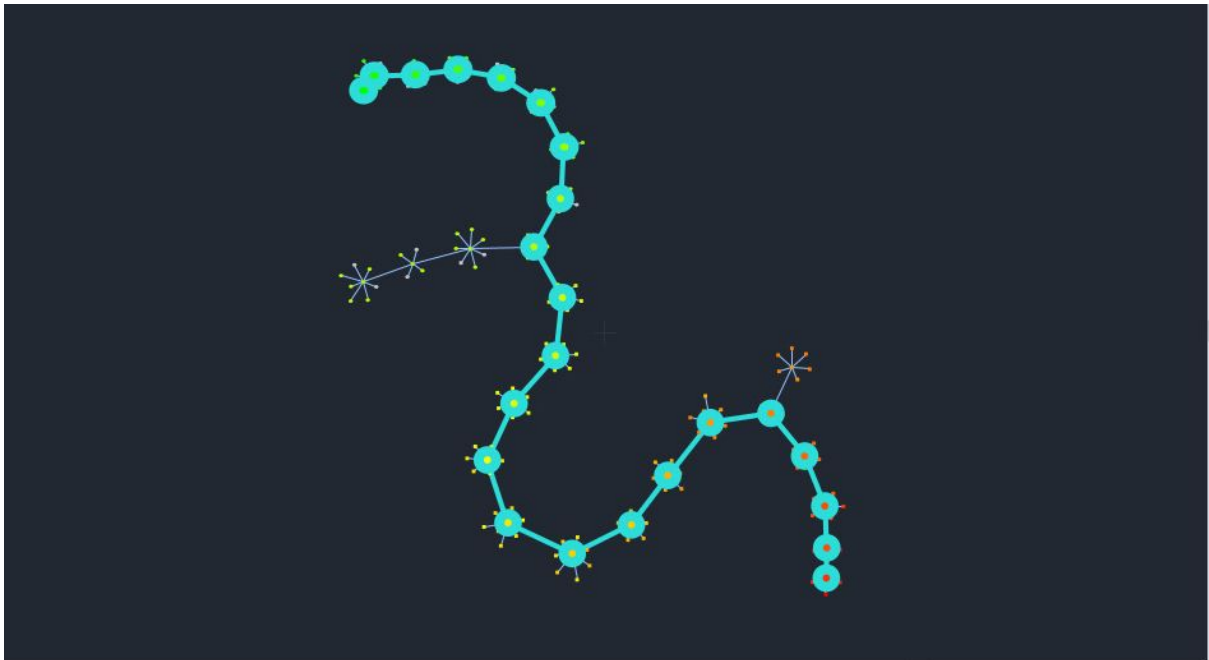
GC-heuristic:

When the planner adds a new action to a branch in the search tree, this doesn't necessarily cause the heuristic function to decrease. In which time step does the planner first find a new lower value for the main heuristic function in each example (FF and GC, respectively)?

For FF new lower value for the main heuristic function is found in step 3. For GC, new lower value for the main heuristic function is found in step 19.

How many goal facts are left to achieve according to the goal count heuristic (see the description of Eager greedy search, Goal count heuristic) in the state that has progressed the longest?

At step 27, 6 goal facts are left to achieve according to the goal count heuristic in the state that has progressed the longest.

Follow the marked path from the start node (the initial state) to the last node (the first found goal state) in the highlighted path in the graph (the plan). Does the solution ever increase the value of the goal count heuristic between one state and the next?

The solution never increases the value of the goal count heuristic between one state and the next.

Running one of the configurations on your own domain and problem:

I run the Fast downward with an FF-heuristic and obtained the following search graph:
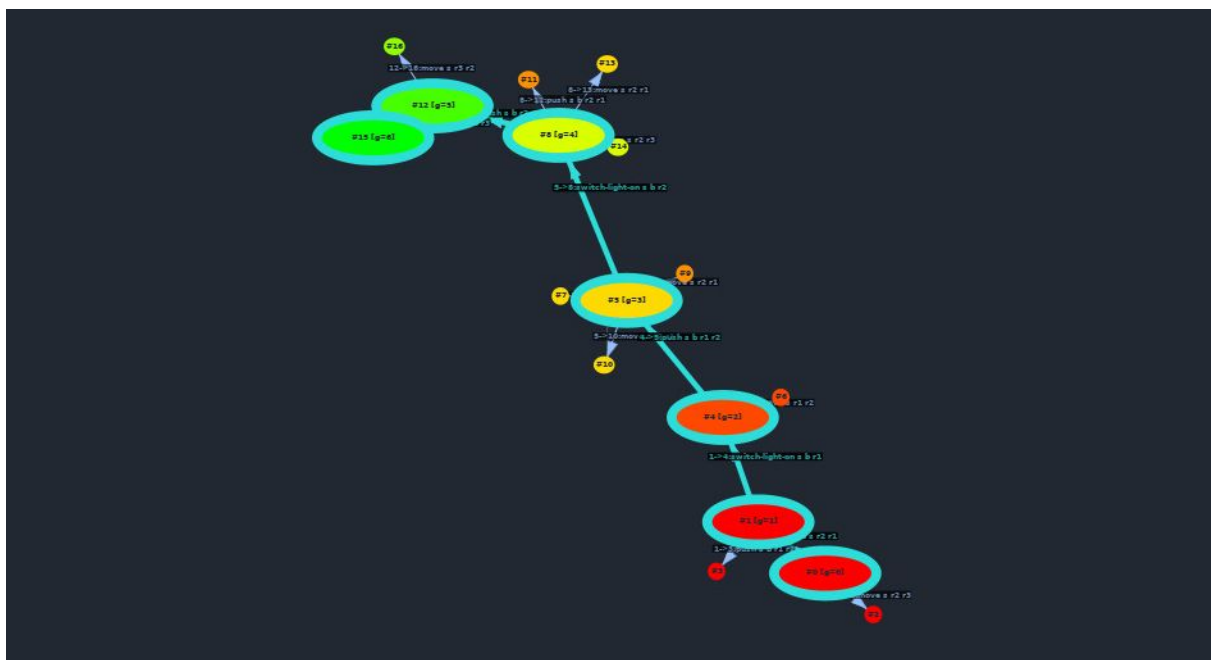
From what I observed the graph for my domain and problem is similar to the graph that I obtained from running Fast downward with FF-heuristic in problem 2, however it is not similar to the graph obtained from running Fast downward with FF-heuristic in problem 3. The reason is both in my domain and problem and in problem 2 the final plan lies on the first expanded long path, but in case of problem 3, the graph had to expand along several long paths until it found the correct path.

Conclusion

Was any configuration better than the other? Was it better on everything or just on some problems?

From running the problems above with different heuristics we can see that FF-heuristic will perform better on certain problems than GC, but on other problems GC can perform much better than FF. So the conclusion is that which heuristic is the best will vary for every problem.