

Lab - 4 - Report

Empirical Investigations

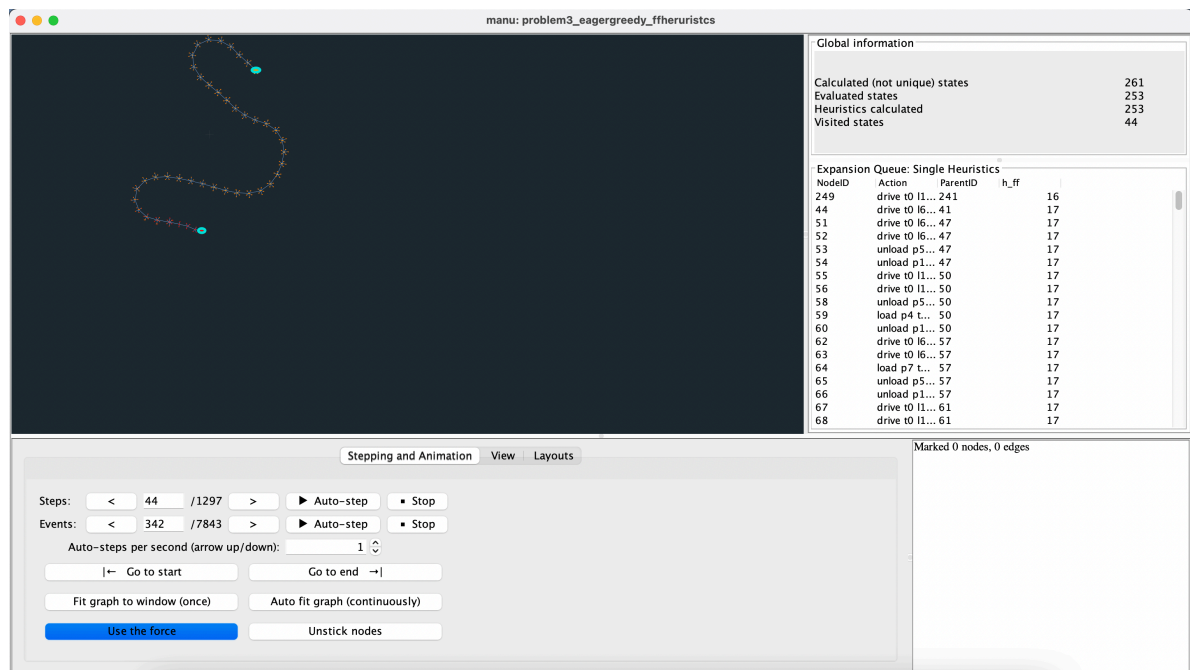
1. Comparing Heuristics, Problem 3

1. Start the Fast Downward visualizer twice, using the log files generated when solving problem 03 with the configuration that uses (a) FF-heuristic and (b) goal count heuristics, respectively.
2. Step both visualisations to time step 40 manually or through auto-stepping. Don't use the "go to" functionality (entering a step number) because you will need to see nodes as they appear!

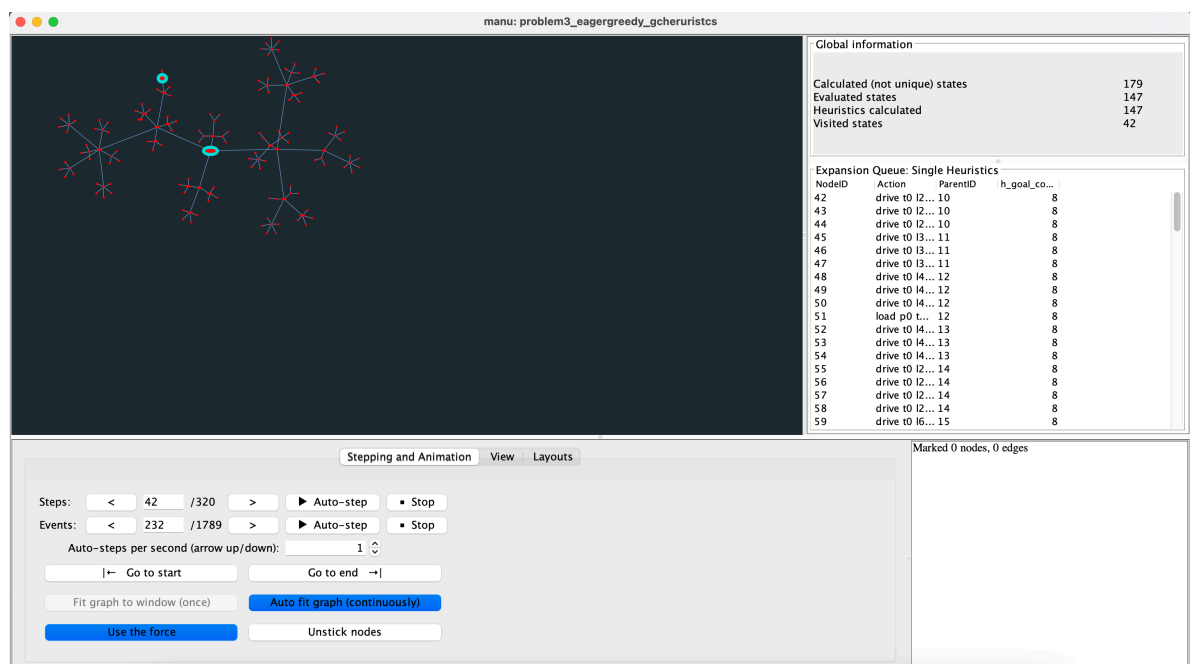
How do the graphs differ, given the same problem and search method but different heuristics? Include screenshots in your report and discuss what you see!

Answer:

- a. FF Heuristics - It causes the graphs to expand in a specific path for a very long time due to the hill climbing algorithm used it starts to expand near by nodes that has better heuristic value until it cannot find any nearby nodes with better heuristic value. This is the behaviour of expanding neighbouring nodes with better heuristic value is observed during the first 40 expansions of problem 3 with FF heuristics



- b. Goal count Heuristics - With goal count heuristics the planner expands nodes that has minimum number of facts left to achieve. This causes the the expanded nodes to be spread out throughout the search space.

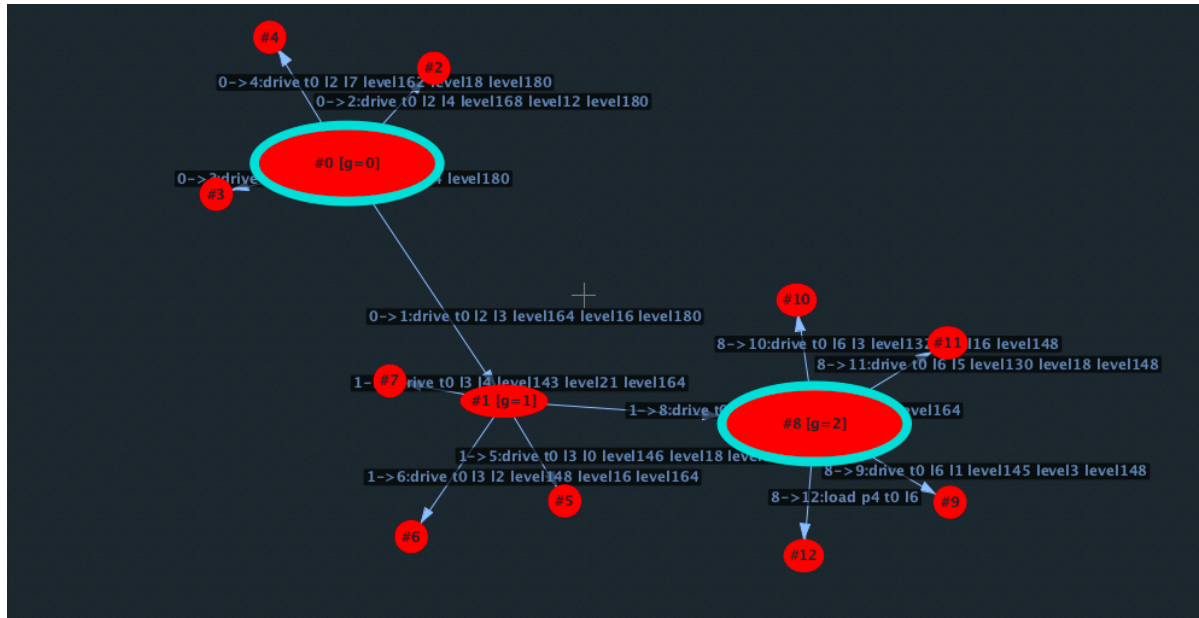


3. Do the different configurations use different actions? Zoom in on the edges to see which actions are used and discuss the differences.

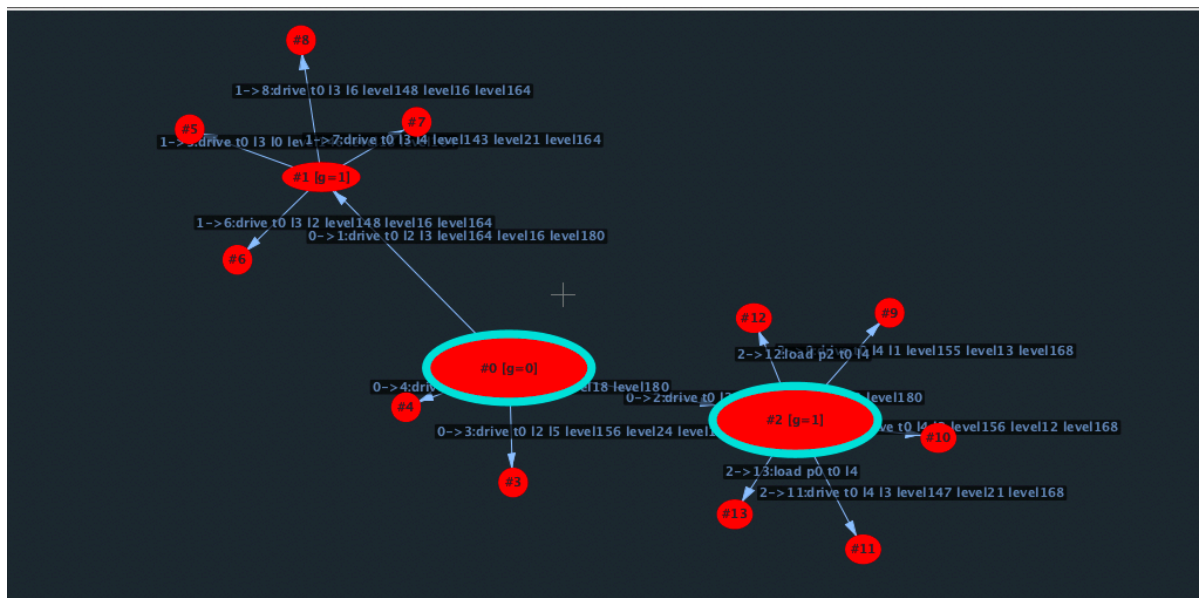
Answer:

No, different configurations does not use different actions this can be seen from the screenshots attached below. The actions to go from state 0 to 1 as well as between other state stay the same.

- a. FF Heuristics -



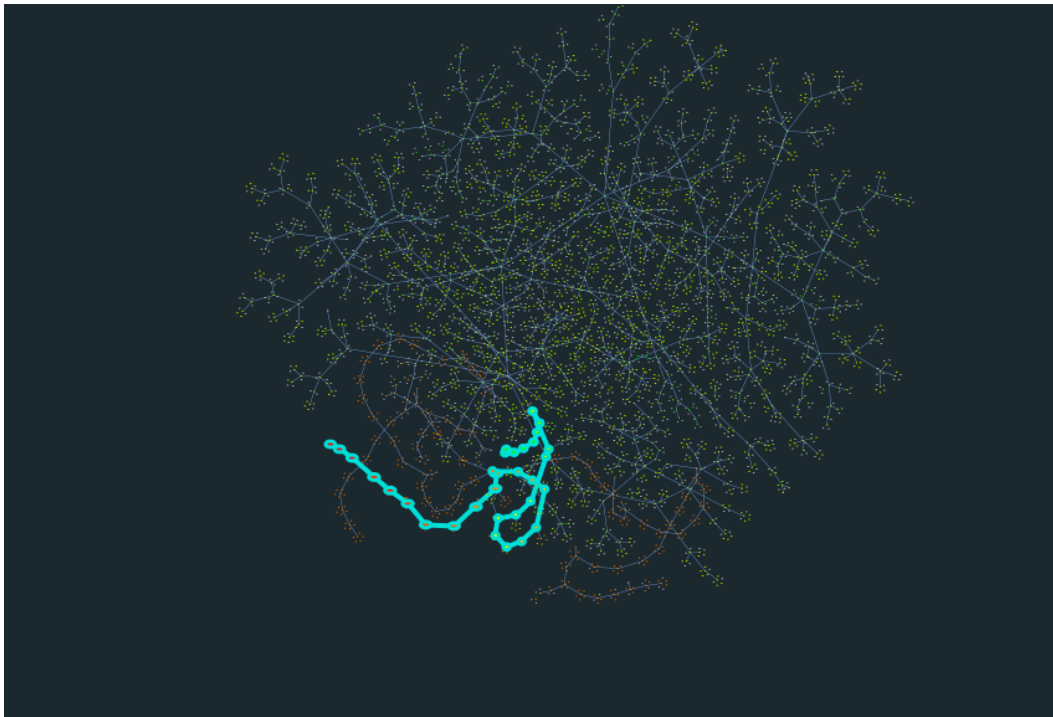
- b. Goal Count Heuristics -



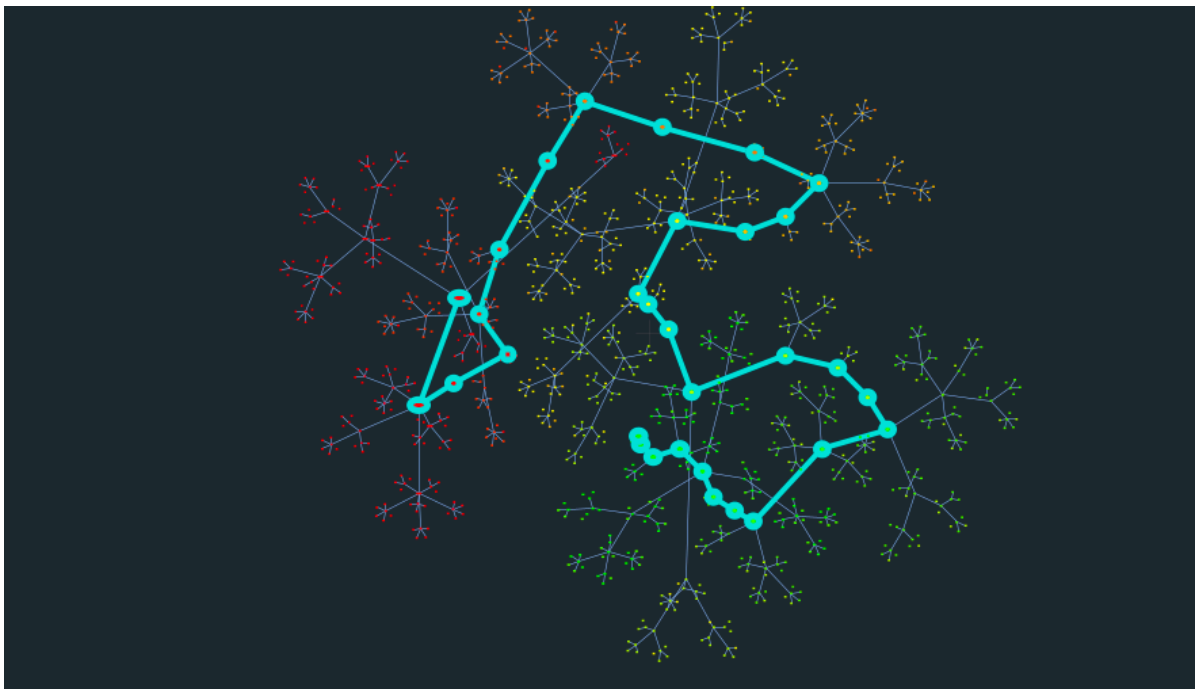
4. Step the FF visualisation to the end through auto-stepping (hint: set the "Auto-steps per second" to a value greater than 1 but small enough so that you can follow the search process). Don't use the go to functionality. At the beginning, the FF visualisation expanded nodes along a long path before it started expanding at other places. Are the nodes in this path used in the final plan? Hint: the nodes are numbered incrementally as they are expanded so this can be used to find the answer.

Answer:

- a. FF-Heuristics - It can be seen that a larger number of initially explore nodes are used in the final solution in the case of FF-Heuristics the image of the full search space and solution is given below, the bottom left edges has the set of initially explored it can be seen that a number of nodes from this space has been used.



- b. Goal count Heuristics - It can be seen that a lesser number of initially explore nodes are used in the final solution in case of goal count heuristics. Nodes are still used from the initially explored path but not as much when compared from FF Heuristics. This can be seen in the image attached below left most edges are the nodes that have expanded initially.



2. Comparing Heuristics, Problem 2

1. Start two instances of the Fast Downward visualization, using the log files generated when solving problem 02 with (a) the configuration that uses FF-heuristic and (b) the configuration using GC heuristics with preferred operators from FF.

2. 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)?

Answer:

- FF-Heuristics - It find the node in the first time function time step itself. The heuristics value of the initial state 0 is 18 at time step 1 it find the state 4 with the heuristics function value 17
- Goal count Heuristics preferred operator from FF - The heuristics function value for the initial state is 7. It find the first state with lower heuristic function value at time step 19 which is the Node 64 with the heuristics value 6.
- Goal count Heuristics - Here also, the observation is same as GC heuristics with preferred operator from FF - the heuristics function value for the initial state is 7. It find the first state with lower heuristic function value at time step 19 which is the Node 64 with the heuristics value 6.

3. Step both visualisations to time step 27 (which is the last time step for the configuration using FF as heuristic, but not for the goal count configuration).

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?

Answer:

- Goal count Heuristics preferred operator from FF - At time step 27 the node expanded is Node 72, its heuristic function value is 6. So 6 goals are left to be achieved.
- Goal count Heuristics - Here also, the observation is same as GC heuristics with preferred operator from FF - At time step 27 the node expanded is Node 72, its heuristic function value is 6. So 6 goals are left to be achieved.

4. Auto-step the visualisation for the goal count heuristic to the end.

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?

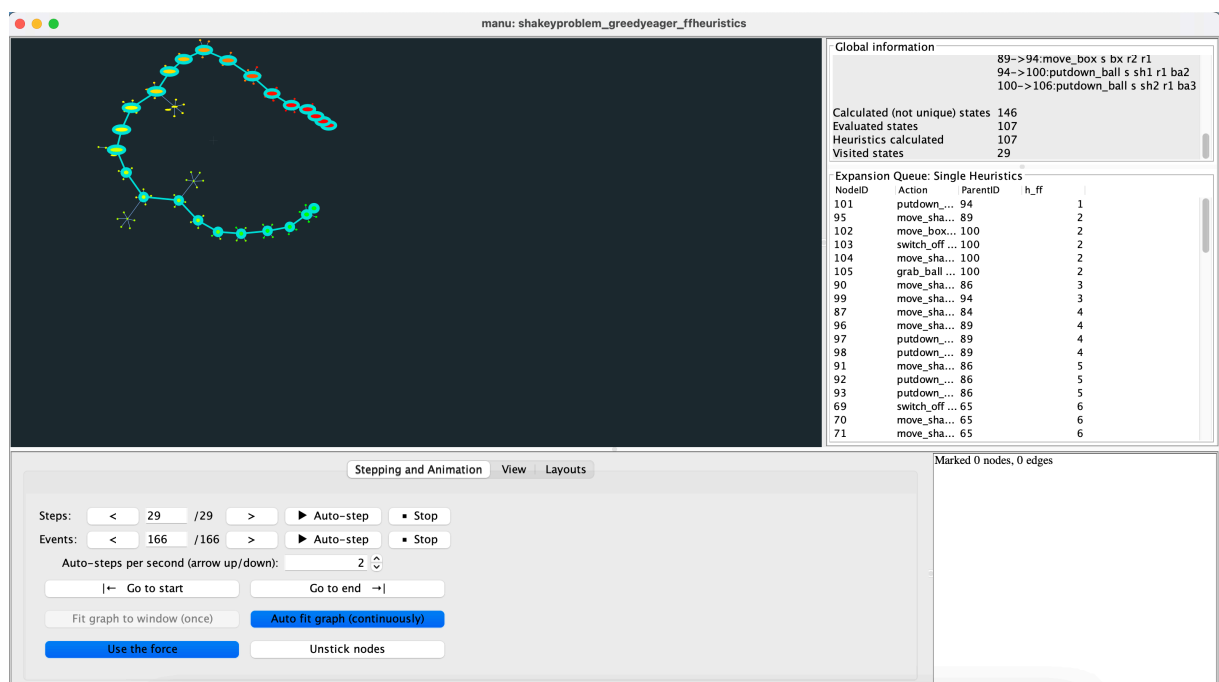
Answer:

- Goal count Heuristics preferred operator from FF - The heuristic function value was never increased from one state to the next in the path to the goal.
- Goal Count Heuristics - The heuristic function value was never increased from one state to the next in the path to the goal.

3. Run one of the configurations on your own domain and problem. Is the graph similar to any of those that you have examined above? Can you explain why or why not?

Answer:

We tried running the shaky domain and problem that was defined in task 1 with the configuration - *Eager Greedy search, FF heuristics*. The graph obtained was similar to the graph obtained in other problem we got a long chain in which the search was proceeding to the nearest sated with least heuristic value lucky in this problem we were able to find the solution with hill climbing part of the search itself which lead to other nodes not being expanded - the screenshot of the graph is attached below



Questions

1. The statistics from using the different configurations to solve the given problems. Was any configuration better than the other? Was it better on everything or just on some problems?

Answer:

Different configurations worked differently for different configurations.

FF Heuristics worked better in Problem 2, it took only 27 time steps for it to reach a solution at the same time Goal count heuristics took 12222 time steps to reach a solution.

Goal Count heuristics worked better in Problem 3 in which, it solved the problem in 320 time steps while FF heuristics took 1297 time steps.

I think the performance of the heuristics depended on the landscape of the search space generated by the search problem. So there might not be one specific heuristics that works for all problems. So I don't think one heuristic is better than the other. Its performance depends on the search space generated by the problem

2. Are your findings applicable to all the infinite many possible domains and problems? If so why? Else, why not?

Answer:

Yes these finding should be applicable to infinitely many domains and problems as the heuristics used are generated from states and actions provided so its performance should be independent of the domain and these finding that we had should hold in them also. The landscape generated by the domain and the problem specified will affect the performance of the heuristic for the specific problem statement.