4/16/2024

Robot Vacuum Project

To begin, I made small adjustments to vacuum.py in order to pass the world and agent position from vacuum.py to the agent functions in vacuum_agents.py. This change was made to the agent_function call in take_action(). It is now:

action = agent_function(world[x][y] == 'dirt', world, agent)

Throughout the project, I never use memory and I use the same model for both action and dirt loss functions. Therefore, I have a total of 3 agent functions. All loss calculations I mention are going to be calculated on a 20 by 20 grid, with 100 runs and 50,000 maximum steps. For the set of agents with full access to the world coordinate, the model, informed_search1(), runs through pre-determined subdivisions of the board, navigating to each spot consecutively. It calls an A* function to calculate the most efficient path from the current position to the current goal position and takes the first step of the found path. With each call the process repeats, approaching the next dirty square or cleaning a dirty square until the board is clear.

Within the A* function, each action has a cost of 1. The frontier is sorted by the total path cost and heuristic function. In my implementation, my heuristic is always 1, because it has been working and no other attempts at a heuristic have gone well. Both models I thought of for this function take forever to run 100 times.

The losses for informed_search1() are 161,080.48 for dirt and 851.44 for actions. I think sometimes with 100 runs it gets skewed by outliers quite a bit. It might go from taking 30 seconds to 30 min. On the run I'm using for these loss values, it took 10 minutes. The previous run it took 5. I'm not sure what makes it get stuck occasionally, maybe if the closest dirt isn't accessible, it repeatedly tries to reach an impossible state. I hadn't found time to try and debug that.

For agents with access to neighboring tiles, I simply had the agent choose randomly between dirty options. If there are no dirty squares available, it picks a random available action. I named it agent_neih1, as I have a difficult time spelling neighbor. I attempted to find the corner and use a grid search algorithm from my known position, but had much trouble guaranteeing a corner. Fear of outliers that might tank the average stopped me from pursuing the grid search idea further. The losses are 231,685.25 for dirt and 4,419.62 for actions. 100 runs take about .15 minutes to run.

My agent with no spatial awareness or memory cleans or picks a random direction. That is the entire function. The losses are 619,609.52 for dirt and 10,774.56 for actions. 100 runs take about .3 minutes to run.

The sum of all dirt and action losses from this 100 run is 1,028,420.87.