MAS – Ex2
Erfan Moosavi Monazzah - 401722199
April 23, 2023

Introduction:

The vacuum agent problem involves designing an agent that can navigate a rectangular room and clean all the dirt present in it. The agent is equipped with sensors that allow it to detect the presence of dirt in the tile it hovered. The agent can perform four actions - move forward, rotate 90 degrees, check for dirt, and suck dirt. This problem can be generalized as a multi-agent problem by increasing the number of vacuum agents. In this sense, agents are competing to each other. One course of competition can be the number of dirty tiles that got cleaned by each agent.

Problem Statement:

The task of the multi vacuum agent is to clean more tiles than the other agents until the entire room get cleaned., while minimizing the number of movements and actions taken. The agents should be able to move efficiently and avoid unnecessary movements or revisits to already cleaned areas.

Proposed Solution:

Our solution to the vacuum agent problem is based on a simple yet effective strategy. The agent will start by rotating until it faces the nearest wall. Once facing the wall, the agent will move toward the wall for one tile. It again checks for the nearest wall and rotate toward it. It sucks up all the dirt along the way. This process will continue until the entire room is cleaned. One problem may arise when two agents want to go to the same tile, to address this issue we proposed a notion of randomness of agent selection to the problem. So in each step, the order of agent action performance is random. If an agent wants to go to a tile which is selected by one of the previous agents, then this agent is punished and do not move or rotate at this step.

Algorithm:

1. Start positioning agents facing a random direction in a random tile
2. Generate a random permutation of agent orders.
3. For each agent do step 4 to 7 until all tiles traversed once by each agent.
4. Check for dirt
5. If dirt is present, suck it and go to step 3, else go to next step
6. if the agent is not facing the nearest wall, rotate it once and go to step 3, else go to next step.
7. Move toward the wall for one tile and go to step 3

Evaluation:

We evaluated our proposed solution by running simulations of the agents in various room configurations. The results show that our solution is effective and efficient in cleaning the entire room.

You can visit the attached GIF files to this report to better understand the algorithm:
Brown squares: Dirt
White squares: Clean
Bot at left: (Alice)
Bot at right: (Bob)
In these pictures, the agents are facing north.

Analysis of Vacuum Cleaner Simulation Results

We ran the simulation of the vacuum cleaner problem three times, each time with different inputs. The cost of sucking a dirty tile is 2 and the cost of moving to a new tile is 1. No cost is defined for rotating.

The results of the simulations are as follows:

Simulation 1:
Alice sucked 3 tiles for 16 costs (punished)
Bob sucked 5 tiles for 20 costs

Bob wins. (Sucked more)

Simulation 2:
Alice sucked 2 tiles for 14 costs (punished)
Bob sucked 2 tiles for 13 costs

Bob wins. (Sucked the same amount of tiles as Alice with less cost)

Simulation 3:
Alice sucked 4 tiles for 18 costs
Bob sucked 2 tiles for 14 costs (punished)

Alice wins. (Sucked more)

We can see that the agents that get punished, cleaned less tiles because they had to wait for at least one step. This waiting process may sounds small but it may create opportunities for other agent to clean more tiles or get at a path which ends up in more tile to be cleaned by that agent. For this exercise since both agents use the same algorithm and there are no tools to detect dirt on other tiles, we decided to introduce the notion of punishment to create a competitive environment for the agents.

Since the lines of codes for this exercise is large, the code explanation is done through commenting in the simulation.py file.