Introduction to Artificial Intelligence

Lab 4 (Week 4) -Intelligent Agents: Part 2 - Goal and Utility-based Agents

2023 - 2024

26 February 2024

Objectives

- Implement different architectures of intelligent agents(Simple Reflex agent, goal-based agent, Utility-based agent).
- Compare and analyze the behavior of different types of agents within a Vacuum world environment.

Overview:

In this comprehensive Lab, the objective is to create a variant version of a vacuum cleaner agent operating in a multi-room environment. The environment consists of sequentially numbered rooms from 1 to 10 (Figure 1), each having a designated level of dirtiness. The agent receives two key percepts: the room's location and its cleanliness level, with the ultimate goal of cleaning all the rooms.

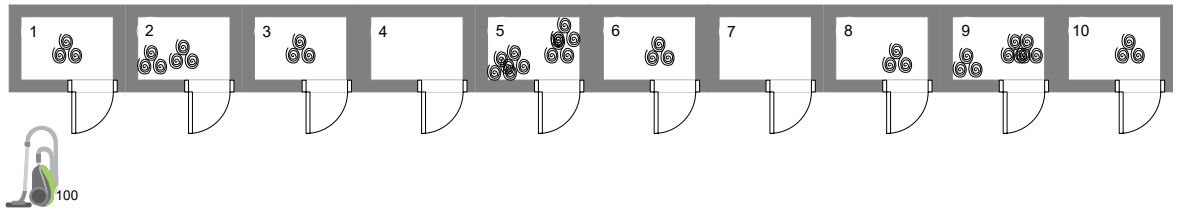

Figure 1: Vacuum Cleaner world

You are asked to design and implement variant agents, specifically reflex-based, goal-based, and utility-based agents. These agents will utilize the percepts received from the environment, as well as any additional relevant information, to make a relevant decision that improves the performance measure. The decisions include actions such as moving to `RIGHT` or `LEFT` and performing cleaning actions at the current location.

Your mission involves the development of three distinct agent structures, each of which must be meticulously delineated according to the following specifications:

- ***Simple Reflex Agent:***
  - This agent perceives its present location and engages in room-cleaning activities if it detects any dirt.
  - It employs two distinct movement strategies: random displacement following an action and sequential navigation, adjusting its direction when reaching boundary rooms.
- ***Goal-Based Agent***:
  - This agent employs a goal function to decide when to stop its movements.
  - It incorporates two distinct move strategies: random displacement and the selection of the room with the highest level of dirtiness.
- ***Utility-Based Agent:***
  - Characterized by its reliance on a utility function that takes into account both `dirtiness level` and `energy consumption`.
  - Embraces a move strategy of relocating that maximizes the output of the utility function.

Performance measure

It's crucial to note that the agents consume energy during their operations. Specifically, they use 1 unit of energy when moving from the current room to the next room (moving from Room 4 to Room 5 incurs 1 unit of energy; from 1 to Room 6 consumes 5 units). Moreover, there is an additional cost of 1 unit on the vacuum cleaner's activity for each cleaning action. The cleaning action results in a reduction of 0.25 from the dirtiness level.

Getting started:

To get started, you are provided with the Vacuum environment class code and a sample of a reflex-based agent class.

```python
import random

class VacuumWorldEnvironment:
    def __init__(self, num_rooms):
        self.num_rooms = num_rooms #size limit
        self.rooms = [random.uniform(0.2, 1) for _ in range(num_rooms)]  # Random
    dirtiness levels

    def __str__(self):
        return f"Dirtiness levels of rooms: {self.rooms}"

    def clean(self, room):
        #clean the room

    def is_dirty(self, room):
        # a boolean function that checks whether a room is dirty or not. A room is dirty
    if the dirtiness level is greater than 0.2
```

```python
class SimpleReflexAgent:
    def __init__(self, environment):
        self.environment = environment
        self.direction = 0 # 0 for forward 1 for backward
        self.energy = 100
        self.agent_location = 0

    def __str__(self):# Adjust this method to print the Dirtiness levels
        return f"Agent is at location: {self.agent_location}, Energy level :{self.energy}
    "
```

Using the aforementioned code snippets, proceed to perform the following tasks:

Task 01: Simple Reflex Agent

- Develop a vacuum cleaner (agent) that takes into account the precepts from the environment, including the status of its current room and the level of dirtiness.
- As the agent engages in random movement across various rooms, implement the requisite actions that enable the Vacuum cleaner to transition between rooms and initiate room cleaning if the room is identified as dirty.
- Create a display method that outputs the current Environment **Configuration**, including the dirtiness level of each room, along with the energy level of the vacuum cleaner agent.
- Test your agent in the specified environment and display the final configuration, What do you notice?
- Adjust your agent to move continuously (sequentially) in one direction until reaching a boundary room, then reverse its direction (go back) and continue moving. Display the final configuration, do you get the same performance?
- Considering the agent's behavior in both random and sequential movement scenarios, do you think we can consider this agent as intelligent? Why or why not?

Task 02: Goal-Based Agent

The second variant of our agent is distinguished by the incorporation of a `"goal function"`, which assesses whether all rooms have been cleaned (indicated by a dirtiness level is less than 0.2). Furthermore, the agent ceases its activity under two conditions:

1. if all rooms are cleaned, as determined by a dirtiness level below 0.2; and
2. if the agent's energy level becomes depleted, reaching zero.

Considering the new definition of the vacuum cleaner, your mission is to implement the following functionalities:

- Using the same Vacuum Cleaner Environment initial **configuration**, Implement the corresponding Agent.
- Display the initial Environment configuration before launching the agent.

- Initiate the agent, adhering to the specified constraints. Given its sequential movement, the agent transitions to the subsequent room upon completing the cleaning of the current one.
- Display the final configuration after the agent is stopped (dirtiness level and energy level).
- Re-launch the agent after resetting the environment. In this iteration, the agent opt for random room selection, including the current room.
- What notable observations can be made regarding the agent's behavior in the random movement scenario?

Task 03: Utility-based Agent

For this particular assignment, our newly designed agent will incorporate a utility function that takes into account the consumed energy level, focusing on two key aspects:

1. Maximizing the cleanliness level.
2. Minimizing the energy consumption.

The agent is programmed to select the action (move to the room) that maximizes this utility function.

Considering the new definition of the vacuum cleaner, your mission is to implement the following functionalities:

- Using the same Vacuum Cleaner Environment initial **configuration**, Launch the agent to clean the rooms, adhering to the specified constraints.
- Display the final configuration after the agent is stopped.
- What do you notice?

Discussion

- Which of the above three agents gives the best results? Explain your viewpoint.