Alex Mitro

Assignment 1 Report

Overview:

The goal for this experiment is to determine which algorithm that we create would be the most effective in gathering crops/materials on a plot. These algorithms will be used on a robot object that will be traversing the plot to gather the objects. The first algorithm that we will be creating will pick a random crop location and move the robot to that location, where it will then remain for the remainder of the time. The second algorithm we will create will pick a random direction for the robot to go in until it hits a wall, where it will then pick another random direction and do the same thing. This continues until the time is complete. The robot will attempt to pick up crops at every unit of time passed. Algorithm three will involve the robot choosing a random direction after every unit of time/step. Algorithm four will choose a random crop location, go to that location, and then choose another random crop location to go to. In similar fashion, the robot will pick crops up as it moves. The fifth algorithm will calculate which plot location has a greater chance at creating the most crops in the allotted time, and will then move around that pivot area for the rest of the time, while collecting the crops.

I would have to guess that the most effective algorithm of the original four algorithms would be the fourth, because the robot will automatically be going in the direction of materials. As it moves towards them, it can collect others on its way there. Algorithm one will not be as efficient, because although it is going to the plot with the highest growth probability, the respawn time of the crop will be too high and result in less crop growth overall. The other two algorithms are probably going to be unreliable because of the randomness feature that they have.

Experimental Design:

In order to answer the question of which algorithm would be most efficient and gather the most crops, I must make sure to I have to create a design for how I will make these algorithms. The first and most important step is to create an object to represent the robot and the plot of land he will be traversing in the language Python. The robot will hold its coordinates, number of steps taken, and number of crops picked up. The plot object will have a method to update the plot, whether it be determining the chance of a crop growing or decrementing its respawn timer. The plot itself will be stored in an dictionary(hashmap) for quick lookup capabilities. While reading in the csv file, we will also store all of the wall coordinates in their own separate arrays. This way, we can use them when traversing through the plot.

For the first algorithm, we must find the coordinate with the highest probability of finding a crop, and stay there for the remainder of the time. First, we will find the coordinate with the highest value. Then, the robot will make create the path as it goes. It will increase its x and y coordinates as needed per unit of time/step. If at any point the robot hits a wall, we will use the lists that contained the wall coordinates to maneuver the robot around the wall. Once the robot reaches its destination, we will halt the robot for the remainder of the trial, and it will only check the current location for the remainder of the time.

For the second algorithm, all we will need to do is determine which of the eight directions that the robot could go. Since this is random, we will use a random number generator. Once the direction is chosen, it will continue in that direction, picking up materials as it goes, until it hits a wall. There, it will choose another random direction. Repeat this until the time runs out.

For algorithm three, it will be set up similar to algorithm two, except it will choose a random direction at every step. If the random direction leads to a wall, it will choose another direction. The robot stops at the end of the time limit.

For algorithm four, we will look at a list of crop coordinates and choose a random point, where the robot will then go towards. The robot will traverse in the same manner as the robot did in algorithm one. The only difference, is that the robot will receive a new coordinate to go to when it has completed the current one. As the robot moves, it will pick up materials if it finds any.

For algorithm five, which is my custom design, I will create a formula to determine which coordinate would have the highest probability of creating the most crops in the allotted time. The robot would then go to that coordinate and traverse the immediate area for the remainder of the time.

In order to properly determine which algorithm is the most efficient, it will be important to run the algorithms at multiple different time specifications to see if they will deliver a higher average at the end of the experiment. At the end of each experiment, if run multiple times, the average of the experiments will be calculated for further analysis.

Based on what I’ve written and my intuition, I still believe algorithm four will perform the best out of the original four based on its lack of randomness and its constant moving towards probable crop locations. However, if done correctly, algorithm five will probably have the highest average outcome of materials gathered. Regardless, this is what I plan on creating, and it should give us some results that I can use to determine which algorithm is the best.

Experimental Results/Analysis:

Results are averaged

Number of test runs per experiment: 500

Algorithm One:

Time: 195 (NxMx25%) Result: 2 Materials

Time: 100 Result: 1 Material

Time: 300 Result: 3 Materials

Time: 500 Result: 5 Materials

Algorithm 2:

Time: 195 Result: 7.726 Materials

Time: 100 Result: 2.952 Material

Time: 300 Result: 13.824 Materials

Time: 500 Result: 25.14 Materials

Algorithm 3:

Time: 195 Result: 5.54 Materials

Time: 100 Result: 2.61 Material

Time: 300 Result: 8.002 Materials

Time: 500 Result: 13.48 Materials

Algorithm 4:

Time: 195 Result: 26.016 Materials

Time: 100 Result: 11.188 Material

Time: 300 Result: 42.178 Materials

Time: 500 Result: 74.324 Materials

Algorithm 5:

Time: 195 Result: 48.03 Materials

Time: 100 Result: 23.462 Material

Time: 300 Result: 75.404 Materials

Time: 500 Result: 127.668 Materials

Obviously, these numbers are subject to change if they were to be run multiple times over and over again. These numbers, because they are subject to change, must be taken with a grain of salt, but we can see which algorithms clearly have a higher number of materials found. Also, we can see that the more time we give the algorithms, the more materials they find. Some algorithms get more material to a higher magnitude, specifically algorithms four and five. Algorithm four and five are obviously the most efficient, which is what I figured would be the case.

Final Thoughts:

As we can see, when we use the hw1map in its original form, we can see that algorithm five is the best. However, when just considering the original 4, the fourth algorithm is the best. This trend makes sense to me based on what was mentioned previously. The randomness makes some of the algorithms just too unreliable. It’s possible that there are bugs in my program and it is possible that it may not work on drastically different plots, but for the most part, these algorithms are viable, and the tests prove their effectiveness on the plot.