CS/SDP 262 Final Project Report

Badvi, B.; Hamza, A.; Usaid, M.

December 8, 2020

Contents

1 Introduction

1.1 Abstract

Anthropocentrism and the capitalistic commodification of resources has led humans down a spiral of self-destruction, and equally increased a general disregard for nature. For most nature is an external entity, unbeknown to them that humanity is not exclusive from it. Following from this stems the negligence of intelligence shown in nature however, as we shall be exploring in our project, signs of intelligence can be found everywhere in nature. Our project deals with modelling the semi-intelligent behavior shown by an acellular organism known as Physarum Polycephalum or more commonly as Yellow slime mold. This organism has baffled biologist and researcher over its exhibition of intelligence besides the fact that it does not have a central nervous system. Surely without a central processing unit how can an organism function let alone solve difficult problems? Intriguingly enough P.Polycephalum cannot only perform basic life function but is able to solve many complex problems such as path optimization (Nakagaki et al., 2000). Research suggest that a pulsating flow of biochemicals within its tube-like structure handles P.Polycephalum's intricate mobility and that the control of this fluid is the cause of its coordinated growth (Alim et al., 2013). Another research suggests that P.Polycephalum makes use of an external 'spatial' memory by using pheromones to mark visited regions (Reid et al., 2012).

1.2 Purpose

We are studying *Physarum polycephalum* in the context of an agent-based model because it will allow us to better understand its situational behavior. By implementing an agent-based model we can think about the behavior of P. polycephalum in algorithmic terms. It will also allow us to introduce different parameters and observe the changes in the behavior based on those parameters.

1.3 Approach

To try and obtain a holistic understanding and to model the behavior of P. polychephalum with a reasonable degree of accuracy, we will be using two different agent based models - each with a separate goal. The first ABM attempts to model the behavior of the slime mold as it coalesces around food sources in a maze. The maze is generated randomly and slime mold nuclei are randomly distributed across the maze and then are linked. The ABM intends to model the primitive behavior that the mold exhibits - it moves closer and closer to the food source, while the nuclei away from the food source tend to die off. However in doing so, it effectively solves the maze.

2 Details of Simulation Model

2.1 Agent Based Model (Minimum Food Distance)

1. **Motivation**

2. **Agents**
   Two different types of agents are used, both serve a different purpose. One of these are used to generate a random maze when setup-maze is triggered. The main agents used in this simulation are an abstract of P.Polycephalum's nuclei, these agents would own an attribute which would store the euclidean distance to the closest food source.

3. **Environment**
   The environment is a Geo-spacial environment where we can setup mazes/walls to show how P.Polycephalum is able to overcome such scenarios.

4. **Learning**
   The agent's movement is such that in each step they either move randomly, remain static, or they move towards a neighbor which is closer to the food node. The movement of an agent towards their neighbor is motivated from the fact that the nuclei prioritizes movement towards the direction it's obtaining the greatest nutrition hence an assumption is made that the flow of nutrients from the neighbor closest to the food source would be greater.

5. **Interactions**
   Each agent is connected with another agent via a link these links are thence contribute to the movement the agents make throughout the simulation as shall be discussed further on.

6. **Time**
   In reality Slime mold grows at about a centimeter an hour so it's ideal to have the time scale of one tick being equal to an hour. Anything less would be redundant. In our environment we've made it as such so that a step of 0.3 is equivalent to a step of $1cm$ in a real-world scenario.

7. **Variables and Constants**

   - **Global Variables**:
     (a) `prob_random` - Represents the probability that a nucleus would move randomly, lies within the range 0 and 1 and the default value is 0.6
     (b) `prob_static` - Represents the probability that a nucleus would remain static, lies within the range 0 and 1 and the default value is 0.6
     (c) `prob-reproduce` - Represents the probability that a nucleus divides creating another,lies within the range 0 and 1 and the default value is 0.15

   (d) `num-agents` - Represents the number of initial nuclei in the model, lies in range 0 to 6000 default value is 1000
   (e) `avgfooddist` - Represents the average distance of a nucleus from a food source.

- **Local Variables**
   (a) `distfood` - The Euclidean distance from the nucleus to the food source.
   (b) `midptx`, `midptx1`, `midptx2` - The x-coordinates of three midpoints of a link.
   (c) `midpty`, `midpty1`, `midpty2` - The y-coordinates of three midpoints of a link.

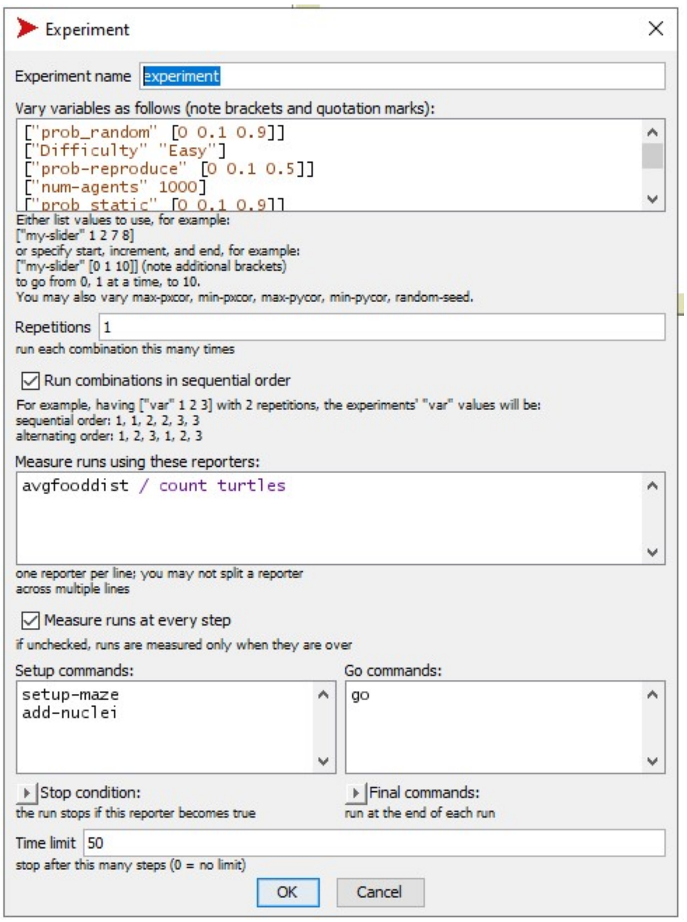
8. **Processes**
   Following processes are involved whilst setting up the environment:
   - `setup-maze, setup-start-finish, make-maze` (Details of these processes can be referred from BAM's maze generation model inspired from `http://ccl.northwestern.edu/netlogo/models/community/BAM20Maze20Generator2)`
   - `kill` This process is used to kill the maze-making agents once the maze is constructed
   -
9. **Stochastic Measures**
10. **Actions**
11. **Inputs**
    This ABM does not take any input.
12. **Outputs**

2.2 Agent Based Model (Minimum Spanning Network)

1. **Motivation**
2. **Agents**
    - Nuclei
    - Food
3. **Environment**
4. **Learning**
5. **Interactions**
6. **Time**
7. **Variables and Constants**
8. **Processes**
9. **Stochastic Measures**
10. **Actions**
11. **Inputs**
    - Number of Food Nodes
    - Angle of Vision
    - Number of Nuclei
    - Nuclei Secretion
    - Food Secretion
    - Evaporation Rate
    - Step
12. **Outputs**

3 Analysis of Data

Since, the primary goal of this project was an inquiry into the inner workings of the slime mold, we chose to not limit our study to just using ABM's. Through the use of the ABM we built that is referenced in section 2.2 we were able to acquire 3 independent variables that could be used to model the slime mold, and predict its decisions. We did this by using the BehaviorSpace tool present in Netlogo 6.1. We set out a range of changes in the independent variables that we were interested in to see how these changes impact the final average euclidean distance from the two food sources. The experiment was setup as follows:



Experiment

Experiment name experiment

Vary variables as follows (note brackets and quotation marks):

```
["prob_random" [0 0.1 0.9]]
["Difficulty" "Easy"]
["prob-reproduce" [0 0.1 0.5]]
["num-agents" 1000]
["prob_static" [0 0.1 0.9]]
```

Either list values to use, for example:
["my-slider" 1 2 7 8]
or specify start, increment, and end, for example:
["my-slider" [0 1 10]] (note additional brackets)
to go from 0, 1 at a time, to 10.
You may also vary max-pxcor, min-pxcor, max-pycor, min-pycor, random-seed.

Repetitions 1
run each combination this many times

☑ Run combinations in sequential order

For example, having ["var" 1 2 3] with 2 repetitions, the experiments' "var" values will be:
sequential order: 1, 1, 2, 2, 3, 3
alternating order: 1, 2, 3, 1, 2, 3

Measure runs using these reporters:

```
avgfooddist / count turtles
```

one reporter per line; you may not split a reporter
across multiple lines

☑ Measure runs at every step
if unchecked, runs are measured only when they are over

Setup commands:

```
setup-maze
add-nuclei
```

Go commands:

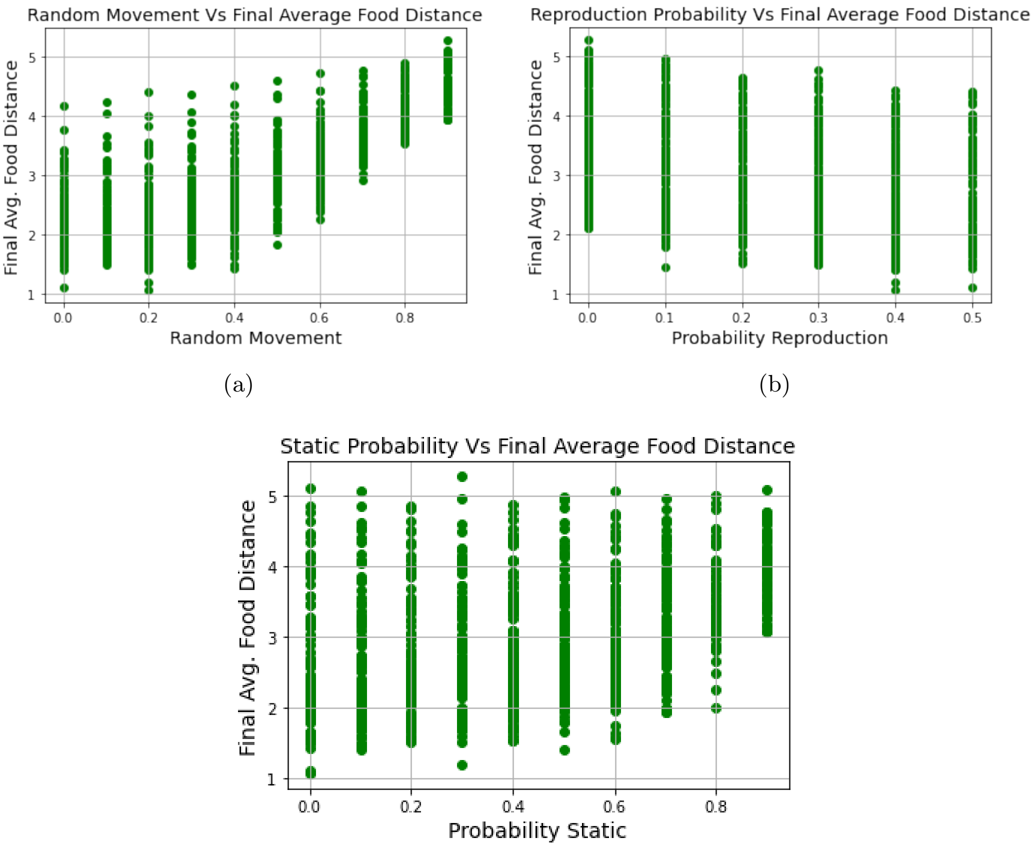
```
go
```

Stop condition:
the run stops if this reporter becomes true

Final commands:
run at the end of each run

Time limit 50
stop after this many steps (0 = no limit)

We were able the run the experiment twice and obtained 920 datapoints, which we then analysed using Python, and some libraries. Through this we were able to obtain visualization between the 3 independent variables and the dependent variable.



(a)

(b)

(c)

Figure 1: Scatter Plots of 3 Independent Variables against the Dependent Variable

4 Limitations of the Model

4.1 Computational Limitations

Since, our project was focused on an organism that has trillions of of nuclei, we knew computational complexity would be a problem for us from the start. We tried to explore avenues of incorporating multi-threading or perhaps use the CUDA cores of the discrete GPUs in our machines but that required the usage of another ABM modelling software called, Flame GPU. This was, obviously, outside the defined scope of the project so we were unable to rectify this issue.

4.2 Linking the Nuclei

Another major challenge was to ensure that links were being formed properly. We had to make sure that the links between two nuclei do not cross over a wall in the maze (a white patch). We simulated the slime's inability to create linkages through walls by simulating only links that were not crosses the wall. We calculated the 3 midpoints across each link and if any of those were intercepting the x and y coordinates of a white patch, we would eliminate the links. This added another layer layer of computational complexity within the model which furthered the limitations of the computational aspects of our ABM.

4.3 Preventing Clustering

Another challenge was the fact that clusters form in areas in the maze. Theoretically, these clusters should die because they get stuck and cannot access food source, and are disconnected from the rest of the P. polycephalum. Unfortunately, eliminating these clusters would be very computationally expensive - which means that its more effective to let them be for the sake of practicality of the model, since the model itself is also computationally expensive due to the large number of nuclei and links that need to be iterated very often.

5 Conclusions

6 Bibliography