Building Complex Computational Models Assignment 2

Name: Nathan Crone

Student Number: 17377463

Email: nathan.crone2@mail.dcu.ie

Code

The code for the simulation created as part of this assignment has been shared with Marija and can be found in the GitLab repository found at:

https://gitlab.computing.dcu.ie/cronen2/ant_agent_based_model_simulation

This GitLab repository contains a 'README.md' file that details more about the code and how you can run it to experiment with various different parameter sets. Despite this, if you have any problems or issues when running this, feel free to contact me and I can help to sort it out.

Model Outline & Purpose

My project focuses on the design and implementation of an Agent Based Model (ABM) to simulate ant behaviour. From a macro perspective, ants may seem like extremely complicated insects and at times it may look as if they are behaving in a chaotic manor, however, in reality, ants behave according to a series of simple rules. By homing in on these simple rules, it is possible for us to re-create ant colony behaviour by means of an agent-based model. By stacking these rules on top of each other, we can simulate the seemingly complex system of behaviour we observe in ant colonies across the globe.

From a simplistic point of view, ants need food to survive, they need mature ants to collect this food, and they need new-born ants to replace these mature ants when they die. In my model, I set out to recreate how these behaves to simulate my own virtual ant colony.

Model Rules & Parameters

In this section, I will explain the rules that I decided to implement in my agent-based model and the reason why these rules were chosen.

1. Initialising the Simulation Environment

Environment Size:

Parameters:

- Environment_width
- Environment_height

The first part of the implemented simulation relates to its environment. The size of the environment can be specified using the 'environment_width' and 'environment_height' parameter. This specifies the number of locations across each axis that the environment contains and not the size of the figure. In this simulation, the anthill always falls in the middle of this specified area.

Food Sources:

Parameters:

- Environment_starting_food_percent
- Environment_food_spawn_prob

This area then has to be populated with food that the ants can eat. This is chosen by the 'environment_starting_food_percent' parameter which specifying how food rich the environment is. This parameters value must lie between 0 and 1 and it specifies what % chance each position in the environment has of containing a piece of food.

Once the simulation starts, the 'environment_food_spawn_prob' parameter specifies the probability that a new piece of food shows up. A new piece of food is given the chance to show up each iteration and can appear at any random place in the environment.

Food Amount:

Parameters:

- Max_food_per_location

We also specify a maximum amount of food that each food initialisation can contain. This is done through the 'max_food_per_location' parameter and helps set a realistic boundary on the amount of food in the environment. Each square, when initialised as a piece of food, has the amount of food contained in it randomly picked from a uniform distribution of values between one and the maximum food amount value. This is similar to real life where the amount of food appearing in one place varies.

Ant Population:

Parameters:

- Starting_population_size

The last step in initialising the simulation is the ant population. The 'starting_population_size' parameter specifies how many mature ants are in the colony from when the simulation starts. All ants start in the anthill upon initialisation of the simulation.

2. Searching for Food

Once the environment has been initialised, the second part of this model is how the ants go about searching for food. Initially, all the mature ants in the anthill will set out in search of a food source themselves as at the start, there is no food source that the ants know about.

Searching for new food:

The simulated ants travel in a random pattern as they scower the environment until they land upon a square that contains food. A random patter was chosen for the ants when searching for food as ants do not have ears and actually have quite poor eyesight. In real life, instead of using their eyes and ears, ants feel along the ground using their antennae to find a suitable food source [1]. As a result, in my simulation, the ants do not know where the food is until they are standing on top of the environment location that contains it. This long and slow process is represented in my simulation by allowing the ants to move one square in any direction at random each iteration. If the ants chosen movement would leave them outside of the environment window, they will remain in the same place until the next iteration where a new direction can be randomly selected.

After food is found:

Parameters:

- Num_food_brought_back_to_nest

Once an ant has found food, they must now bring this food back to the anthill for storage. The 'num_food_brought_back_to_nest' parameter dictates how much food they can bring back to the anthill.

When traveling out from the anthill, ants have a built-in pedometer that tracks the number of steps that each of their legs walks along with the precise order their legs stepped in. This allows them to easily retrace their steps once they have decided that they want to return to the anthill. In the case when the ant has found food, they retrace their steps exactly by repeating the same steps they took to get to the food but in a reverse order. This is done in my simulation by recording the steps that the ant takes at each iteration in a list and then following this list in reverse order to get home.

When an ant in the real world has found food, as they travel back to the anthill, they also lay down a pheromone trail. This pheromone trail then tells other ants that this path leads to food and enables the other ants to follow this trail as well. In my simulation, once an ant has returned to the anthill after retrieving the food, the path they took back to the anthill from the food (the pheromone trail) is added to a list of all the pheromone trails that have been created. This list is then used as a central hub of all trails that other ants can follow.

Following trails:

Parameters:

- Follow_prob

Once food has been found and a pheromone trail has been established and stored, ants may now decide to follow this trail instead of setting about on their own randomised search for food. While in reality, if an ant comes across a pheromone trail while they are out on their search for food, they may decide to change their course and follow this trail instead, in my simulation, ants can only decide to follow a trail when they located are in the anthill. This is due to the added difficulty this would take to implement, as the ants would not be following the trail from the start, and also the time complexity this would add to the model. As a result, in my simulation, once an ant is in the anthill, it has a choice: to follow a pheromone trail to an already found food source, or to set out on its own search for a new food source. The probability of the ant following a trail can be set through the parameter 'follow_prob' with this value being between 0 and 1. In real ant colonies, this probability is not 100% as there are 'scout ants' whose job it is to find new food sources. Despite this, the majority of ants are worker ants and do follow trails, so a high value for this probability of about 90% seems reasonable. If it is selected that the ant will follow a trail, they will have a look inside the list of all pheromone trails and select one that they wish to follow. This selection is probabilistically weighted by the pheromone strength of that given trail.

When following a trail, the ant will incrementally retrace the steps that the original ant took in retrieving the food from the source. If the ant gets to the end of the trail and there is no food left, this ant will then change course and will no longer be following a trail but instead will set out randomly searching for food. This ant will still have recorded the steps he took so can easily return to the anthill once they find food.

Trail Strength:

Parameters:

- Trail_depreciation_time

The way these trails work among real ants is that the pheromones get stronger as more food is retrieved from a specified trail and gets weaker if no food is retrieved from a trail. This means that if food runs out at a trails end, no food will be retrieved from that location and hence the pheromones associated with that trail will dwindle. After a specified length of time of no food being retrieved from a trail, the scent of this trail will no longer exist. I have accounted for this in my simulation where you can specify the 'trail_depreciation_time' parameter which helps set the length of time between food being retrieved by an ant from a trail and the trails scent disappearing. This length of time is deduced by multiplying this parameters value by the time it would take to follow this trail. This ensures that the length of the trail does not affect its disappearance rate.

In cases when lots of ants are going to the same trail, the pheromone strength of this trail will increase rapidly and as a result, this will be an extremely popular trail for ants to follow.

3. The Ant Lifecycle

The lifecycle of the ants must also be mentioned as a key player in this simulation. Naturally, over time, in the real world, we will see new ants being born and old ants dying, so this must be factored into our model.

Colony Growth:

Parameters:

- Num_ants_laid_daily
- Time_till_egg_hatch
- Time_till_larvae_become_pupa
- Time_till_pupa_become_mature_ants

The initial state of the simulation is that there are a number of fully mature ants that can go out, search for food, and bring it back to the anthill. However, alongside this, there are also immature ants who are in the process of developing into fully mature ants. These ants remain in the anthill until they are fully developed, whereupon they can venture out to collect food. In real ant colonies, there is a queen ant who stays inside the anthill and produces ant eggs as her only job. This queen is aided by a number of worker ants who look after the produced eggs and feed the ants until they mature. While this simulation does not involve a queen or these helper ants, I have based the model parameters used in this simulation for the ant development process, and the time associated with this, on real ant development schedules.

The first part of this is creating ant eggs. When looking online, I could not find a value for the number of ants laid per day by the queen. As a result, the 'num_ants_laid_daily' parameter is worth experimenting with in this simulation.

Once the eggs are produced, there are three development phases that an ant goes through before it becomes a mature ant, phase one = Ant Egg, phase two = Larvae, and phase three = Pupa. While the length of time the ant takes to develop into each of these phases varies depending on species, one website pointed out that eggs usually take a week or two to hatch, larvae usually develop into pupa after about a month and pupa become fully mature adults after a period of six to ten weeks [2]. While these values can be changed, I proceeded to use the values stated on this websites as default in my model. The development parameters are 'time_till_egg_hatch', 'time_till_larvae_become_pupa', and 'time_till_pup_become_mature_ants', respectively.

While eggs do not eat food, once the ants become larvae, they start needing to be fed. The food to feed them is taken from the storage which is collected by the mature ants. The model is set up that the more developed the ant gets, the more food they eat. The growth in consumption for an individual ant is uniform up until the day they become a fully mature ant, where their meal then consists of one food piece.

Colony Decline:

On the other side of the lifecycle, we have the death of ants. This is governed by two things, the first being the maximum lifespan an ant can have before they die of old age, and the second being the maximum time an ant can live without eating before they die of starvation. Again, with these parameters I have sought a value that closely aligns with that of real ants.

Maximum Lifespan

Parameters:

- Max_lifespan

For the maximum lifespan, there is no fixed amount as it varies with species. I saw some online content that stated that ants live for a few months while others said they live up to fifteen years. Depending on the species you are modelling, the 'max_lifespan' parameter can be experimented with, however, the default value in my simulation is a year and a half as this lay as a common middle ground between all of the quoted lifespans.

Time Before Starvation

Parameters:

- Max_without_food

The value for the maximum time for an ant to live before it dies of starvation was less variant with most sources stating that ants can survive without food for up to two weeks [3]. As a result, this is the default value for the 'max_without_food' parameter. While ants can only survive for up to 5 days without water, as water does not exist in my model, I did not include it as a parameter for the ants.

Time Until Hungry

Parameters:

- Time_till_hungry

Following on from this rule governing how long ants can go without eating, I also implemented a minimum time that ants can go without eating. This parameter sets the time it takes for an ant to become hungry after they eat. I did not want the ants eating every iteration if there was a food source right beside the anthill, so I implemented this rule to prevent that. [4] states that a common time for ants to eat is every 3 days so this is the default value for the 'time_till_hungry' parameter. This rule also helped to conserve food in the simulation and to ensure there was enough left for other ants to stay alive.

When Ants Eat

Ants will only eat if the time since they have last eaten falls between the time till they are hungry and the time till they starve. Ants only get the opportunity to eat if they are at the anthill, where there are food supplies, or if they are at a food source. When an ant arrives at a food source, if they are hungry, they will eat and if there is any food left, they will then bring this back to the anthill for storage.

The time till hungry and time till starvation also affects the immature ants. A baby ant will not get fed unless it is hungry, but if it is hungry and there is no food in storage, it will remain hungry until either food arrives, or it dies of starvation. Once food arrives to the anthill, in my simulation, the ants are fed in the order of oldest to youngest. This is to ensure that larvae, who would otherwise not survive, do not eat all the food and in turn cause in the starvation of an older ant who is more near maturity. This ensures to prioritise the development of mature ants to give an extra helping hand in collecting the food to feed the young.

Results Tracking

What results are being tracked?

Over the course of each simulation, I plot a number of graphs alongside the simulations current state. These graphs update every iteration and help to track the relationships between the different results in the simulation. The first of these graphs tracks the adult ant population, the second tracks the immature ant population and the development stage of these ants, and the third graph shows the total amount of food stored in the anthill at each iteration.

Why are these results being tracked?

These graphs are plotted in order to give the viewer an understanding as to the results of the model at its current state and to summaries what is going on in the environment. The adult ant population is the key statistic to record as this essentially shows the current strength of the colony. Alongside this, the collected food supply and the number of immature ants are also key values to track as the colony depends on these when it comes to its future success. I broke the immature ant plot down further into the development stages of these ants as well, this added an additional layer of understanding of the current state and where the results of the model may be heading.

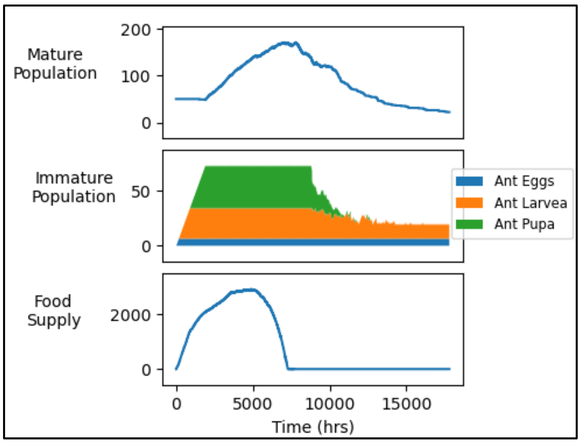
As the simulation is run, these graphs show the relationships between these variables and enable us to see how the number of mature ants affect the amount of food that is collected and stored in the anthill, how the amount of food stored in the anthill affects the immature ant populations, and how the number of immature ants affect the mature ant population.

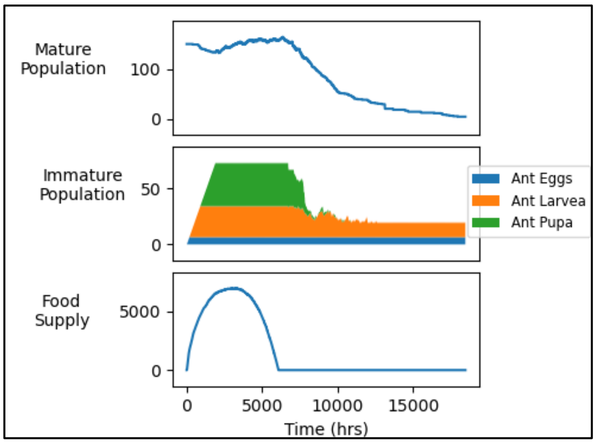
Parameter Scenarios

In this section, I will attempt to experiment with different parameter sets to see the effect that they have on the simulation results. As seen from the above detailed description of the model, there are a total of sixteen parameters used in this model. While some of the values for these parameters do not need to be experimented with as they have been chosen using evidence on real ant biology or are only used to specify the width of the environment, this still leaves eight parameters to choose for the simulation. Due to the constraint of the 8-page report for this assignment, it would not be possible to detail the experiments on all of these eight parameters, so in this section I will pick three different parameters that represent different things in the model and discuss them. The results of each parameter scenario will be judged over a period of two years or until the death of the colony, whichever comes first, and the metrics used to judge how the simulation faired will be the plotted results described above.

Starting Population Size

The first scenario I wanted to investigate was the affect that having a higher starting ant population has on the simulation. Using the same parameters outside of this population size, I will run the simulation with a starting mature ant population of 50 and then again with a starting mature ant population of 150. Below we can see the results of these simulations, respectively.



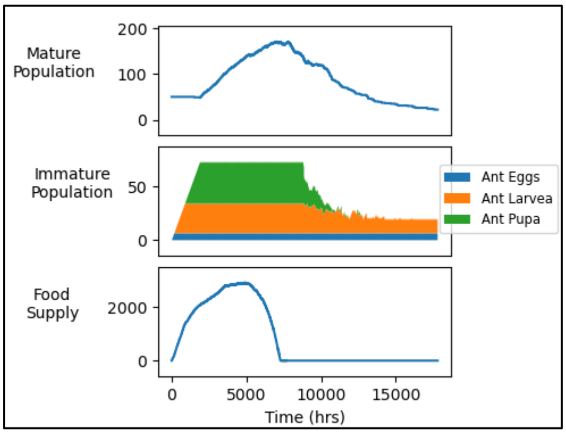


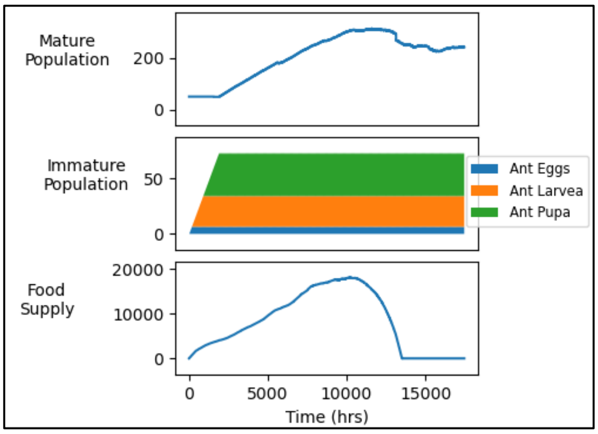
As you can see, the end result after the two years is quite similar. We see that in both simulations, the ants start off strong collecting lots of food, but then once the immature ants get older and start consuming more food, the built up food supplies start to deplete. Another reason for this decline which can't be seen here is that at the start, the food is dispersed evenly across the environment, but once the ants start collecting, they collect all of the food supplies that are close to the anthill meaning the supplies that are left are further away. The further the ants have to travel to collect food, the more time it takes to replenish the depleting food supplies. Once these supplies run out, the immature ants start to starve which results in no new mature ants being developed.

We also see that having the higher starting population size didn't actually mean more ants were around. While at the very start this was the case, what we saw when there was 50 ants was that the ant population started to increase as new mature ants were developed, however, with the higher population start size, we saw more ants dieing of starvation and the new mature ants merely mainting the population. While it's not quite clear from these grahs, at the end of the two year period, the number of ants remaining was higher in the simulation that started with only 50 ants. While the higher starting ant population collected more food initially, they also depleated their supplies faster in sustaining this population.

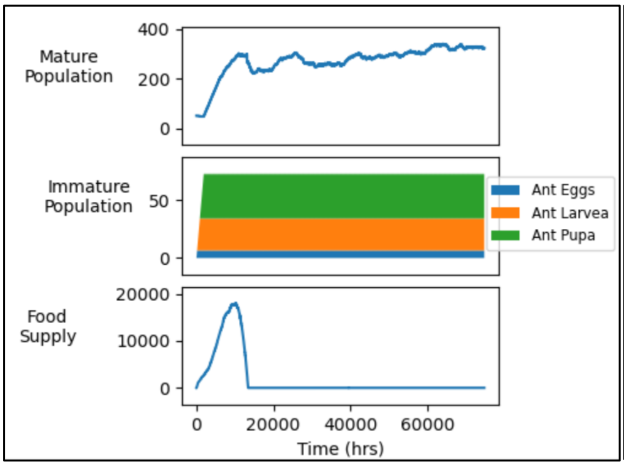
New Food Spawn Rate

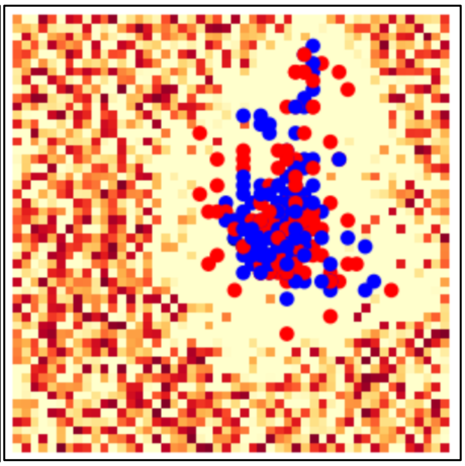
A population start size of 50 seemed best but the next thing to look at was how to help the ants maintain a high food supply and not collect lots at the start and then drain this over time. While I have various food related parameters, such as the number of food items per food location or the number of food items each ant brings back to the anthill, in this experiment I will look at increasing the food spawn rate. In the previous simulation, each hour there was a 1% probability of new food spawning, but I will compare this to a 5% probability and see how the simulation reacts.





As you can see from this graph, the 5% new food spawn rate gave more longevity to the simulated colony. We see by the end of the two years; the mature ant population was flourishing, and the immature ant population was still strong at maximum levels. While the food supply got up to really elevated levels and lasted much longer than previously, once again, in the end, it was depleted. Unusually, however, this did not have a massively negative effect on the ants. I suspect that this was due to the ants collecting just enough food each iteration to maintain the immature ant population, but then having no more food left over in storage. To check the sustainability of the populations under these conditions, I ran the simulation for 8.5 years and here are the results and final environment state.

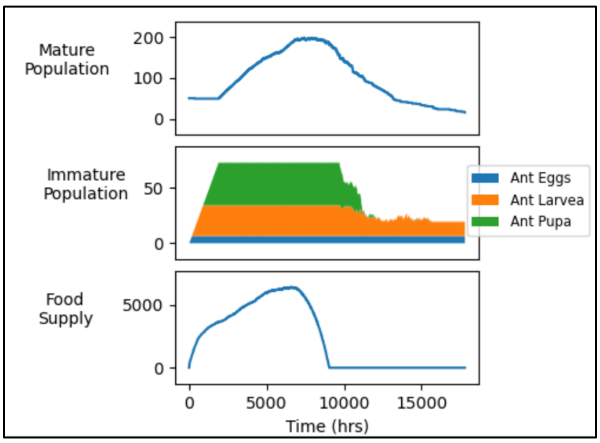


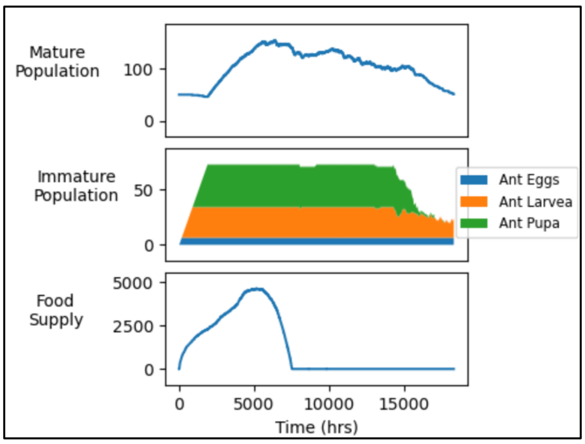


We can see that the populations have remained reasonably stable around this parameter selection. We can also see that the ants are only collecting food that is close to the nest and are not venturing outside a certain distance. This makes sense as it is more time efficient to focus on food sources closer to the nest.

Trail Pheromone Deprecation Rate

The final thing I will experiment with in this report is the trail depreciation rate. This parameter dictates the decrease rate of the trails pheromone strength. If this number is high, it takes a longer time of no food being retrieved from that food source before the trails scent disappears. If this number is lower, previously found trails are forgotten faster. Once the trails strength gets to 0, no ants follow that trail anymore. To test the difference this parameter makes, I will run the simulation with this parameter set to 1 and then again with it set to 100. When run alongside the simulation having a 5% chance of new food spawning, the results were not clear over the two-year period, so the below graphs show the results with the new food spawn chance set to 1%.





The results noticed here show that when the trail deprecation rate is set to 1 and the storage of collected food runs out, the immature ant's population immediately dies off and in turn this causes a decrease in the mature ant population. However, when this parameter is set to 100 and the storage of collected food runs out, the immature ant population lasts much longer, and the mature ant population holds up better. As suspected in the last experiment, I think this is due to the mature ants bringing in enough food each day to feed the immature ants, but not bringing in enough to create food storage reserves. This proves unsustainable in this simulation over the course of the two years but shows how the longer time before deprecation gave the ants a much better chance of survival.

I suspect that the new spawned food sources are the cause of the mediocre performance when the time before the trail's strength decreases is low. When a new food source spawns close to the anthill, the ants immediately turn their attention to depleting this as the distance required to travel to this source is low, which in turn allows more ants can collect food from it in shorter time, increasing its strength and the probability that more ants travel there. However, once this food source is gone, the ants must turn their attention back to collecting food from other sources. When the trail strength decrease time is low, the trails to the old food sources disappear as all the ants turn their attention away from these sources towards this new source. However, when this time is higher, once the recently spawned food source has been exhausted, the trails to the old food sources are still active and the ants can return to collecting from these.

References

[1] - https://www.westernexterminator.com/ants/everything-you-should-know-about-ants/#:~:text=known%20as%20parthenogenesis.-,How%20do%20ants%20eat%3F,the%20rest%20of%20the%20colony.

[2] - https://animals.mom.com/life-cycle-ants-6537.html

[3] - https://pestsamurai.com/how-long-do-ants-live/#:~:text=Most%20sources%20say%20that%20ants,only%205%20days%20without%20it.

[4] - https://www.antsalive.com/ant-care-tips.htm