Simulating Ant Colonies To Investigate How Minor Worker vs Major Worker Ant Distributions Affect Colony Energy

Harry Howarth, Frazer Bennett-Wilford and Blayze Millward

May 14, 2018

Contents

List of Figures

List of Tables

Abstract

Ant colonies have the unique ability to gather food from their surrounding environments using dynamic, emergent behaviours that have been well explored by a variety of past studies [1, 2, 3, 4, 5, 6], often focusing on examining the paths formed to and from food sources [6, 7, 5]. Many species of ants have a number of different castes within their ranks [8]. This study examines what affect different distributions of two castes of ants in particular – the *minor* and *major* – worker ants within a colony have its overall ability gather food.

Ant colonies with descritised terrain and differing ratios of worker-caste ants are simulated and food is spawned randomly throughout. Their success – the total energy collected – is plotted against these ratios in order to identify the most sucessfull distribution of minor and major workers.

Our model seems to emulate the behaviour of ant colonies in the real world, with the model indicating that a ratio of around 32% minor to 68% major workers is the most successful, reflecting the average distribution of workers seen in mature nests in the real-world as measured in a previously published work [9].

1 Introduction and Background

Ants are *eusocial* creatures, meaning that they possess the highest level of organisation that any social species of animal can possess [10, 11]. Collectively their behaviour allows them to perform incredible feats as a whole, even when individually each ant has very little actuation over it's environment. One such example of this is an ant colony's ability to forage for food from its surrounding area, while avoiding obstacles and successfully navigating changes in its environment. This ability in itself has been examined a significant amount in past research [5, 6], and has even lead to algorithmic methods inspired by the behaviour being developed to solve problems faced in areas of computer science, such as those similar to the *travelling salesman problem* [12, 13].

Another characteristic of eusocial creatures – and the focus of our own investigation – is the separation of groups within the colony into differing *castes* that exhibit specialized behaviour, making them more fit for certain types of jobs over others in their colony [11]. Ants in particular are often divided into up to four differing casts: Queens, Minor Workers (Workers), Major Workers (Soldiers) and Drones [14, 15]. According to work performed by W. R. Tschinkel in his journal article on the distribution of worker ant populations within colonies, *Colony growth and the ontogeny of worker polymorphism in the fire ant, Solenopsis invicta** [9], minor workers typically make up 65% of the worker population within a mature colony with major workers making up just 35%. Tschinkel measured these stable distributions only in more mature colonies (colonies over six years of age), in particular, in *fire ant* colonies.

Our experiment is interested in the area of finding the ratio of minor to major worker ants that has the most positive impact on an ant colony's ability to thrive within a changing environment. It will revolve around an agent based simulation of ant foraging behaviours around a nest in an environment where food is randomly spawned to be consumed by the colony, and the colony can produce two differing types of ants representing the major and minor worker ants.

Agent based models (ABMs) are models, typically of natural systems, that simulate the system they are modelling through the use of individual *agents*. As opposed to mathematical models that aim to simulate a system by generalizing the behaviour of potentially many components into a few mathematically described expressions (such as population over time, or growth rate with food consumed), agent based models simulate each component individually and the interactions between them [16], allowing them to simulate potentially unexpected emergent behaviour that is hard to describe using mathematical approaches – this makes them particularly useful when evaluating complex systems that exhibit strong emergent behaviour such as the flocking of birds or, as in our case, the retrieval of food exhibited by ant colonies.

A number of past studies have examined the way that ant colonies search their surrounding areas to locate and retrieve food [5, 6]. Often the these studies focus on simulating ant colonies to apply the emergent behaviour that they exhibit to solve abstracted technical problems [12, 13], with a seemingly smaller number of past works focusing on simulating ant colony behaviour as it is seen in nature [5].

The 2004 work by Vittori et al.[5] examined the way that ants navigate their environment, using trails of deposited pheromones which attract other ants from the colony toward food or back to the colony. This work in particular was of interest to us, as it evaluated the results of the model by comparing them to real ants in a number of experiments. Vittori et al implemented a very reduced model for their simulation that allowed each ant to only move

through a relatively small graph of various positions, with each node having a reference to a real-world counterpart and ant behaviour when deciding which path to take not only being based on pheromone depositions, but also partially on the angle at which the ant would have to move through to get to the next node. The model also introduced restrictions to certain actions that the ants could take, such as the a maximum number of U-turns an ant could perform on a long branch, allowing the ants a way to backtrack in a similar way to observed behaviour in real ants, while not allowing them to get caught in a repeated loop of U-turns.

Panait et al. introduces the concept of a dual-pheromone system in "Ant Foraging Revisited" [6] demonstrating the use of a two-pheromone system allowing for learning of paths to both food sources and the colony's nest. This is an alternative to many previous systems that used ingrained knowledge within each ant or the environment itself to direct returning ants toward the nest. It appears to have the benefit of better mirroring real-world ants, but does also increase the total complexity of the system. Comparing the dual-pheromone method explored by Panait et al. [6] with the more direct methods of behaviour coding described by Vittori et al. [5], it can be seen that there benefits to both systems. Vittori et al. often choose to simplify ant behaviour, resulting in faster models that still appear to reflect behaviour exhibited in the real world, whereas Panait et al. opt for more realistic real-world behaviors based on the assumption that the ants have no intrinsic knowledge of their environment.

However, there are examples of ant species such as desert ants that can find their way back to their nest seemingly using 'natural pedometers' [17], lending support to the idea of modelling a system in which the agents know intrinsically information about themselves and the position of their nest.

2 Methodology

2.1 System Description

At the highest level our ABM consists of one agent, Ants, and an environment that contains their colony, food and then the pheromones that the ants leave behind. This high level behaviour is described by the state diagram Figure 5 in Appendix A. We have modelled two types of ants, major workers and minor workers. These are called workers, for major workers, and scouts, minor workers, in this project's code and some of the diagrams.

The system aims to simulate colonies of ants as they forage for food in a 2-D environment. Each colony of ants exists as a single spawn point out of which ants leave to look for food. The state diagram in Figure 6 found in Appendix A details this behaviour. Each tile contains a value for the strength of food pheromone for each colony. Ants can move from tile to tile to follow pheromone trails towards food. If an ant cannot see a food pheromone trail then they will explore away from the nest for new food sources or food pheromone trails to follow. As can be seen in Appendix A Figure 8 Ants will reinforce a pheromone trail when they return to the nest with food. Over time these pheromone trails will decay. This is described by Robinson's journal article [3] which details the decay rates of varying types of pheromones in ants' foraging networks. Ants will only reinforce those food pheromone trails that they find food at the end of by leaving additional pheromones when returning with food.

2.2 Assumptions

An assumption made was to not model factors that affect ant speed. One example is ant encounters, this being where two ants move over each other and cause a slowdown. The reasoning for this was that according to one study 'direct or interaction effects, has a much smaller effect on walking velocity than does body size' [2], therefore the speed of an ant is mostly determined by their body size, which is given as the most important factor. Another example is ant size, we are assuming that ants of all sizes move at the same rate.

Additionally, although ants were modelled to have continuous locations, the environment is grid based and so pheromone trails were located by their tile position. The purpose of this model is that it finds statistical patterns based on the proportion of scout/worker ant populations in each colony, therefore, it is not important that the entire system is modelled in a way that is physically accurate on a ground level, rather that the simulation provides results which are statistically representative of ant colony behaviour.

The properties of food were assumed. The two main assumptions that were made here were the generation rate of the food and the decay rate of the food. The generation of the food was done by starting with n tiles of food and then generating n tiles of food in random places in the environment every t iterations. The decay rate of the food was not implemented so food was assumed to exist from creation until it is used up by the Ants. Another food based, assumption that was made was that food remained in the same place as it was generated.

2.3 Experimental Setup

This sub-section outlines the various parameters used in our experimentation. It starts by detailing the values used for this models agents, two types of ant. Where values have been taken from literature, they are clearly cited. Any values that have been assumed have been described in the sub-section above.

The two types of ant are major workers and minor workers. We are modelling them using four parameters: speed, strength max energy level and energy use per iteration. The values for each of these are in table 1. The speed value comes from the journal article [2] and the fact that they are the same has previously been explained. The values for the minor worker's strength, maximum energy level and energy use per iteration were arbitrarily chosen. However, we used literature to scale the difference between the minor worker and major worker. Tschinkel's journal article [9] states that the average major worker's size is 4 times that of a minor worker. We then took this and applied it to the characteristics as seen in table 1.

Table 1: This table details the parameters used for each ant in our experimentation

Ant Type	Speed ($\mathrm{mm\,s^{-1}}$)	Max Energy Level	Energy Use Iteration^{-1}	Strength
Major Worker	51.9	600	4	120
Minor Worker	51.9	150	1	30

As detailed in the introduction with reference to W. R. Tschinkel's journal [9], minor workers typically make up 65% of the worker population and major workers make up just 35%. Our project is investigating how minor worker vs major worker distributions affect

colony energy. Hence our experimentation is going to be focusing on changing the proprtion of major workers in the colony to find the optimum for our colony. The values, percentage of major worker ant, used in experimentation were 35% and 0% to 100% with a step size of 10.

The length of each simulation is 1200 iterations. In real world terms an iteration is 1 minute, this means our system is simulating 20 hours of real-world time. We allow the ants to move 1 step each iteration at a pace of $51.9\mathrm{mm\,s^{-1}}$. Hence the size of a tile is 3.1m by 3.1m. The environment is set to a fixed 50x50 tiles therefore the area that our experimentation is covering is $24025\mathrm{m}^2$ (155m by 155m).

To produce our results 5 runs, with 5 separate seeds (1:5), for each percentage of major worker ant were done. Each run takes c.63s to complete, so doing this many runs that this level of iterations is not computationally prohibitive.

3 Results

3.1 Control Results

To test our system had the expected behaviour we ran a set of simulations with no food being generated in the map. In these simulations it was expected that the ants would forage for food until they had lost enough energy to need to return to the colony. At this point they would not be able to regain energy so they would again leave the nest and lose half the amount of energy required before returning to the colony and forage again. This process would repeat until the ants 'died' at which point there would be no ants left. This occurred as expected and a video of this can be found at `https://www.youtube.com/watch?v=mHyaiDkegGc`

3.2 Main Results

As already described, the length of each simulation is 1200 iterations and the environment is set to a fixed 50x50 tiles for each run. The independent variable of this experiment is the percentage of major and minor workers. Given the stochasticity in our system, we took the average of 5 runs, with 5 separate seeds (1:5), for value in the experimentation. The average runtime of each run in an experiment was 63s. However all of the results were generated using Matlab's parallelisation ability where multi-core CPUs can be leveraged (more detail on this in Appendix D).

The results of our experimentation can be seen in the line graph in figure 1. There are 12 lines on this plot, 35% represents the 'natural system' value taken from the literature [9] and the rest 0-100% in steps of 10 represent our experimental values. As you can see in the figure 1 the natural system produces the 'best' result in terms of overall colony energy. This can also be seen in the bar chart of the same data found in Appendix C figure 11.

This positive result around the value that the internationally renowned Professor Tschinkel's journal article [9] motivated further experimentation around the values between 30% and 40% of the colony being major workers. The values for this experiment were {30%, 32%, 35%, 38%, 40%} The results for this are in the bar chart in figure 2. We were expecting a distribution that looked Gaussian around 35% but this shows that the peak for our model is at 32%.

A visualisation of our system can be viewed at `https://www.youtube.com/watch?v=tgsPYGt2TbI`. This is at the optimal ratio of major:minor workers for our system of 32:68

4

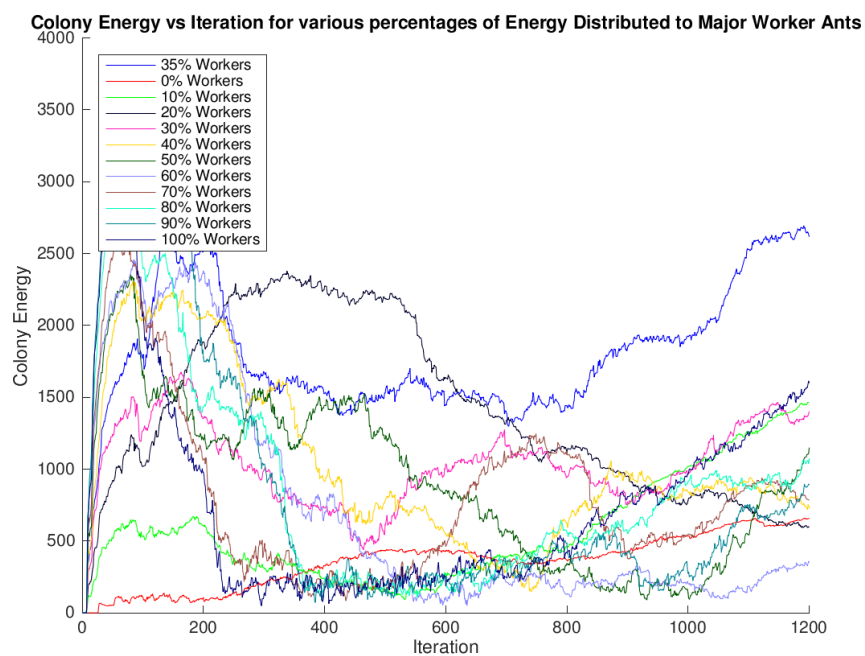and runs for 1200 iterations which equates to 20 hours.

Figure 1: Line graph depicting colony energy against iterations (time) for various percentages of Major Worker Ant.
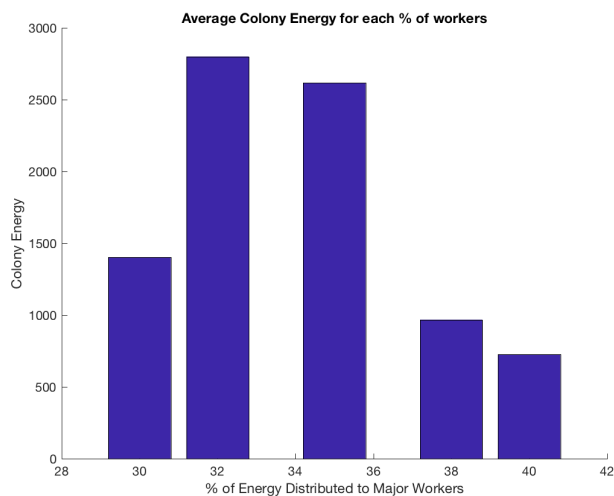

Figure 2: Bar Chart of more detailed experimentation based on positive results between 30% and 40%

3.3 Computational Efficiency

When running the simulation with graphics enabled the largest portion of computation is spent on drawing the simulations results. In order to find bottlenecks in the simulation itself the profiler was used with the graphics disabled, to give a time plot table of what functions are the most expensive during the running of an entire simulation. Appendix B, figure 9 shows the summary of the system and figure 10 of the same appendix shows that the environment is the place where most computation happens.

Children (called functions)

Function Name	Function Type	Calls	Total Time	% Time	Time Plot
TerrainTile>TerrainTile.step	class method	2500000	61.226 s	71.0%	
Colony>Colony.step	class method	1000	11.306 s	13.1%	
Environment>Environment.generateFood	class method	20	0.004 s	0.0%	
Self time (built-ins, overhead, etc.)			13.649 s	15.8%	
Totals			86.185 s	100%	

Figure 3: Profiling for the child functions of Environment - Environment.step

Children (called functions)

Function Name	Function Type	Calls	Total Time	% Time	Time Plot
Pheromone>Pheromone.step	class method	5000000	30.581 s	49.9%	
Self time (built-ins, overhead, etc.)			30.644 s	50.1%	
Totals			61.226 s	100%	

Figure 4: Profiling for the child functions of TerrainTile - TerrainTile.step

When overviewing the computational costs of the environment step, it's highlighted in figure 3 that the terrain tile is where the large majority of time is spent. As shown in figure 4, a large amount of the computation that is done during the simulation of the terrain tile is a result of the pheromone step function. The pheromone step is taking a large amount of computational power to run as each tile of the map contains a pheromone which is then updated each timestep of the simulation. Running it this way means that tiles without active pheromones will still be receiving updates each timestep. However, a way of improving this to greatly reduce computation would be to instead keep a list of active pheromones and then only update those inside that list. When a pheromone decays to zero strength, the pheromone can be removed from the list and won't be unnecessarily updated each timestep.

3.4 Repeatability

Our results are fully repeatable. See Appendix D for a more in-depth view of this and how to run the system.

4 Discussion

4.1 Implications, Reliability and Realism

Our model indicates that the optimum ratio for minor to major worker ants is 32:68. This reflects very well the findings that Tschinkel et al. found in their 1988 paper [9] that we originally examined when searching for a baseline reference of real-world data. In order to ensure that our results provided a true representation of the average behaviour of our model at the various ratios, each ratio simulation was run 5 separate times and an average of the energy collection was used, limiting the chance of abnormal results being reported. This number of repeats, however, could be higher in order to gain a better sample of possible simulation runs, further increasing the representative power of these experiments with respect to our model's overall behaviour.

Another limitation of the parameters we used when experimenting with the various ratios was the number of ants in each simulation. Each simulation that we ran only contained 30 ants. This limited the resolution of possible results in our distribution experiments, as the minimum unit of change we could make was one ant, which equates to a change of 3.3% in worker ratio. This limited our ability to observe fine-grained results when narrowing our experimental range in order to observe the ratio-efficiency distribution around the 35% mark. This also indicates an area where our model begins to lack realism, as real-world ant colonies can have a significantly larger population (Tschinkel et al. measure 220000 workers in colonies 4 to 6 years old [9]).

It is possible for our model to run larger colony sizes and an increased number of times, however doing so requires significantly more computation time and would be outside of the scope of this study.

4.2 Future Improvements

By improving the computational efficiency of the simulation, tests can be run for longer amounts of time to provide longer term trends and repeated to give averaged results. Implementing the improvement suggested in the computation efficiency section of this report would greatly improve the run-time of the simulation and such improvements to computational efficiency will allow for larger amounts of experimentation.

In order to further improve the model itself, there are several properties of real ant colonies which could be modelled to improve biological accuracy. One example of this is reproduction, by giving ant colonies the ability to reproduce would allow the simulation to run over longer periods of time and provide long term trends of ant behaviour. The current system does not include reproduction and so is only suitable for shorter simulations, where reproduction has very little impact. If a colony were to have a large amount of ants die, then as the environment becomes dense with food, the population will slowly increase again until an equilibrium point is reached. Modelling such behaviour would provide results which are more closely aligned to nature, where colonies do have the ability to produce new ants.

Another area of improvement could be to simulate colony interactions. In the current system several colonies can be simulated simultaneously, however ants from two colonies do not interact. Giving ants the ability to fight each other would allow for colonies to compete for food sources. If the simulation was extended to include ant interactions and reproduction,

the simulation would likely depict a more natural representation of ant colony behaviour.

4.3   Modelling Ant Colonies with an Agent Based Model

In order to simulate the behaviour of an ant colony, each individual ant was modelled as an agent and functioned following a list of behaviours. This agent based approach allowed for emergent behaviour to result from a collection of relatively simple rules. This being the main advantage of an agent based model, behaviour becomes easy to modify, understand and the simulation of ants can be extended to include new behaviours with ease. However, the primary disadvantage of using an agent based model in this instance was that by modelling each individual ant, the computational costs become very large. When compared with a mathematical model, when considering the type of system developed, it can be considered appropriate that an agent based model was used because the mathematical equation of such a complex system could not easily be understood, tested and modified.

5   Conclusions

In conclusion, the main take-away from this report and the results of our simulated ant colony is that, like in the real world, a mix of major and minor workers is optimal for the overall 'energy' of the colony. For our system a ratio of 32:68 major to minor worker ants was optimal based on the seeds that we used. In a 1988 journal article [9] Walter Tschinkel, a renowned myrmecologist, gave the ratios in a mature fire ant colony as 35:65 of major to minor workers. Given that the optimal value for our colony is close to this, it can be concluded that our model gives a reasonable representation of optimal worker ant ratios for foraging, although in order to add additional weight to our results further simulations and evaluations could still be performed with larger colony sizes and a larger sample size of initial seed conditions.

Overall, the use of an agent based model proved to be an effective way of simulating ant colony behaviour. However, is it important to note that given that our colony is unrealistically small and other assumptions have been made, we should not make too much of this positive result, instead encouraging further work.

References

[1] S. Camazine, J.-L. Deneubourg, N. R. Franks, J. Sneyd, and G. Theraulaz, *Self-Organization in Biological Systems*. Princeton studies in complexity, Princeton University Press, 2003.

[2] M. Burd and N. Aranwela, "Head-on encounter rates and walking speed of foragers in leaf-cutting ant traffic," *Insectes Sociaux*, vol. 50, pp. 3–8, Feb 2003.

[3] E. J. H. Robinson, K. E. Green, E. A. Jenner, M. Holcombe, and F. L. W. Ratnieks, "Decay rates of attractive and repellent pheromones in an ant foraging trail network," *Insectes Sociaux*, vol. 55, pp. 246–251, Sep 2008.

[4] J. M. HERBERS and E. CHOINIERE, "Foraging behaviour and colony structure in ants," *Animal Behaviour*, vol. 51, no. 1, pp. 141 – 153, 1996.

[5] K. Vittori, J. Gautrais, A. F. R. Arajo, V. Fourcassi, and G. Theraulaz, "Modeling Ant Behavior Under a Variable Environment," in *Ant Colony Optimization and Swarm Intelligence*, Lecture Notes in Computer Science, pp. 190–201, Springer, Berlin, Heidelberg, Sept. 2004.

[6] L. A Panait and S. Luke, "Ant foraging revisited," tech. rep., George Mason University, Jan. 2004.

[7] L. A Panait and S. Luke, "Learning ant foraging behaviors," tech. rep., George Mason University, May 2018.

[8] "Caste Terminology - AntWiki." `http://www.antwiki.org/wiki/Caste_Terminology`. [Online; accessed 13-May-2018].

[9] W. R. Tschinkel, "Colony growth and the ontogeny of worker polymorphism in the fire ant, solenopsis invicta," *Behavioral Ecology and Sociobiology*, vol. 22, pp. 103–115, Feb 1988.

[10] E. Wilson, *The Insect Societies*. Harvard University Press, Jan. 1971.

[11] D. Hadley, "What Are Social Insects?." `https://www.thoughtco.com/what-are-social-insects-1968157`. [Online; accessed 13-May-2018].

[12] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Computational Intelligence Magazine*, vol. 1, pp. 28–39, Nov. 2006.

[13] Y. Zhang, Z. l. Pei, J. h. Yang, and Y. c. Liang, "An Improved Ant Colony Optimization Algorithm Based on Route Optimization and Its Applications in Travelling Salesman Problem," in *2007 IEEE 7th International Symposium on BioInformatics and BioEngineering*, pp. 693–698, Oct. 2007.

[14] "Ant Family Castes." `http://antark.net/ant-life/the-family/`. [Online; accessed 13-May-2018].

[15] Y. Dobrev, "Castes of ants." `http://www.antday.com/?lang=en&pageid=castes`. [Online; accessed 13-May-2018].

[16] "Agent, based, modelling." `http://www.agent-based-models.com/blog/2010/03/30/agent-based-modeling/`. [Online; accessed 14-May-2018].

[17] R. Khamsi, "Ants use pedometers to find home." `https://www.newscientist.com/article/dn9436-ants-use-pedometers-to-find-home/`, Jun 2006. [Online; accessed 14-May-2018].

A   Appendix State Machines

This section of the Appendix contains the state diagrams that are required for the Methodology section but are too large to include in the actual body of the report.
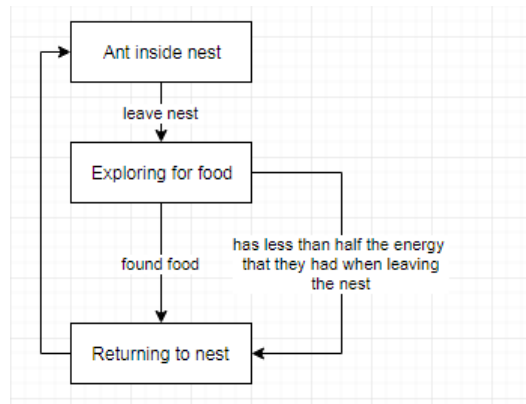

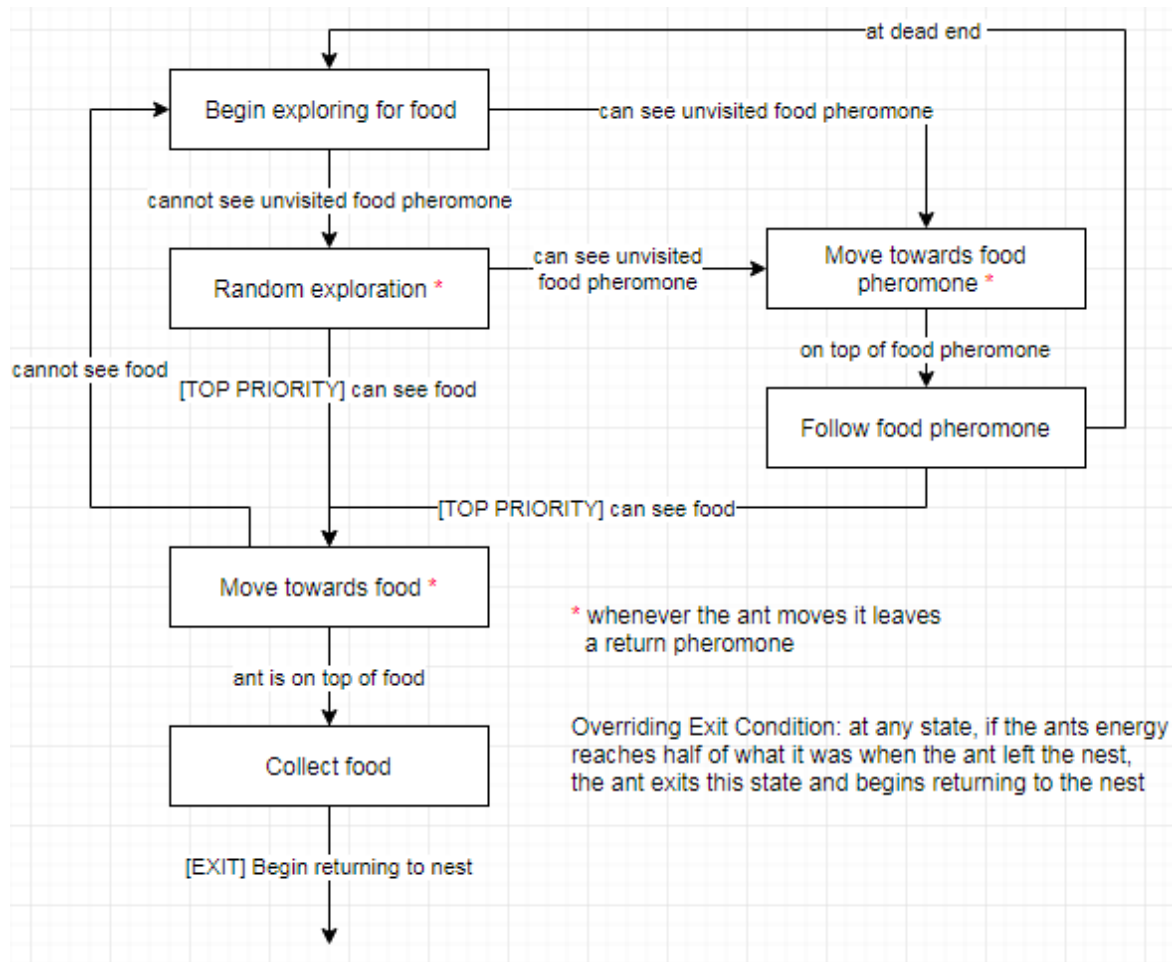Figure 5: High Level State Diagram for the Ant

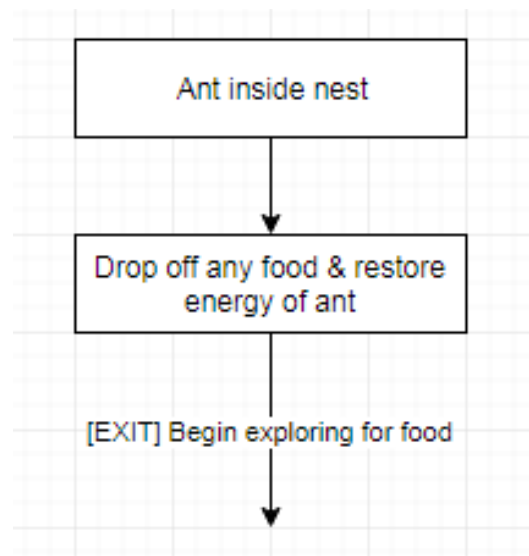Figure 6: State Diagram for the Ant Exploring

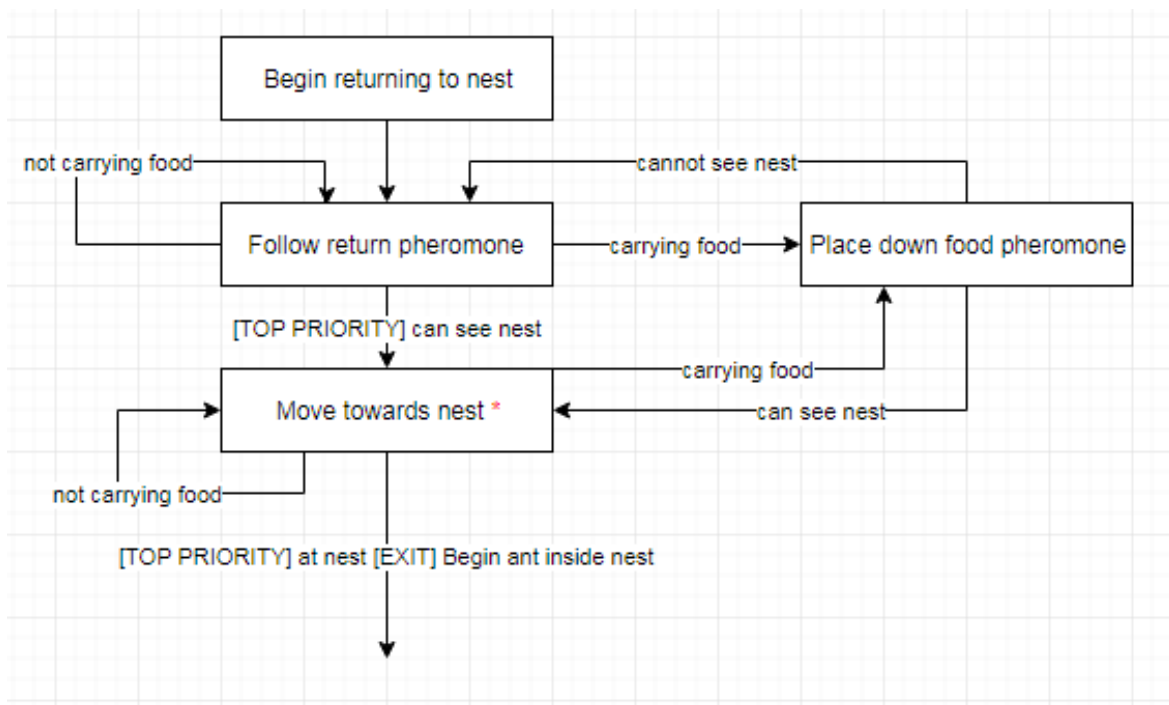Figure 7: State Diagram for the Ant interacting with the nest

Figure 8: State Diagram for the Ant returning to the nest

B Appendix Computational Efficiency Profiling

This section of the appendix contains some of the relevant computational efficiency profiling screenshots.

Profile Summary
Generated 13-May-2018 22:35:06 using performance time.

Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
ant_simulation	1	87.474 s	0.242 s	
Environment>Environment.step	1000	86.185 s	13.649 s	
TerrainTile>TerrainTile.step	2500000	61.226 s	30.644 s	
Pheromone>Pheromone.step	5000000	30.581 s	30.581 s	
Colony>Colony.step	1000	11.306 s	0.318 s	
Ant>Ant.step	29000	10.985 s	0.408 s	
Ant>Ant.moveStep	22973	6.641 s	0.179 s	

Figure 9: Profile summary of our Matlab application

Children (called functions)

Function Name	Function Type	Calls	Total Time	% Time	Time Plot
Environment>Environment.step	class method	1000	86.185 s	98.5%	
legend	function	1	0.337 s	0.4%	
Environment>Environment.Environment	class method	1	0.247 s	0.3%	

Figure 10: Child functions of the ant-simulation, which contains the majority of the computational cost

C Appendix for Figures for Results Section

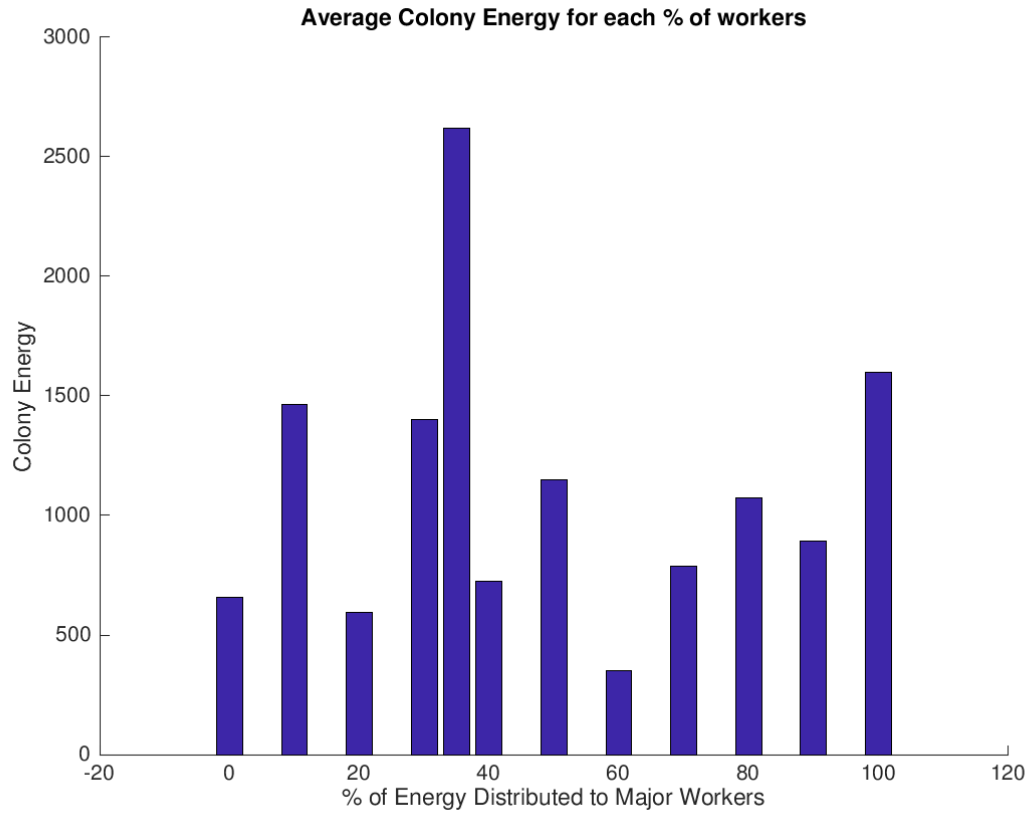This section of the appendix contains some of the relevant result figures.

Figure 11: Bar Chart depicting total colony energy after 1200 iterations for each percentage of ants that was experimented with.

D   Repeatability

Given the nature of this simulation it is important that our results are repeatable. This is, obviously, the case for this project. We used 'rng(seed)' with a seed value to set the seed used for generating random numbers. For an example of the repeatability, run 'simple_example' exactly as it has been handed in. Although if you have changed parameters, use the values in the table **??**. This will give you a value of '6530' displayed in the console and saved to 'final_energy'. Hence, as you can see we have repeatable results.

Table 2: This table gives the variable values for the file simple_example.m. They should be this for the hand-in but they are here for repeatability.

Variable	Value
seed	2
environment_size	50
colony_count	1
worker_percentage	0.5
colony_ants_total	30
simulation_length	1200

To get the graphs that we have produced, we ran the experiments in parallel. If you run 'ant_parallel' you will be able to generate exactly the same graphs. However, if you don't have a quad core processor you may have to change 'cpu = 4;' to something that is a better value for you. If you don't want to run in parallel then set this to 'cpu = 1'. However, this will will increase the time it takes for the simulations to finish.