Università degli Studi di Torino
Master degree in
Physics of Complex Systems

Greta Greco

Simulating an Ant Colony

Prof. Pietro Terna

INDEX

1. Introduction

Individual ants are simple insects with limited memory and capable of performing simple actions. However, an ant colony expresses a complex collective behavior providing intelligent solutions to problems such as carrying large items, forming bridges and finding the shortest routes from the nest to a food source. A single ant has no global knowledge about the task it is performing. The ant's actions are based on local decisions and are usually unpredictable. The intelligent behavior naturally emerges as a consequence of the self-organization and indirect communication between the ants. This is what is usually called emergent behavior.

An approximate computation tells us how successful this model is. It is estimated that their total weight is the same order of magnitude as the total weight of human beings; like human beings, they can be found virtually everywhere on the earth. Ants are undoubtedly one of the most successful species on the earth today, and they have been so for the last 100 million years.

An ant colony is the basic family unit around which ants organize their life cycle. The typical colony consists of one, or more, egg laying queens, a large number of sterile females, divided into workers and soldiers. Seasonally, a large number of winged sexual males and females depart the nest in great nuptial flights. The males die shortly thereafter, along with most of the females. A very small percentage of the females survive to initiate new nests.

The elements of a colony are generally divided according to their role: there are workers who take care of the nest and the brood, others who forage for food, and soldiers, with a larger head and above all larger jaws, who defend the territory, patrol the paths that lead to food, and break the larger prey into bits. However, flexibility of behavior is one of the great strengths of ants: the division of labor is never hard and fast so that roles are interchangeable.

Foraging for food is a very expensive activity for a colony, because workers can easily be lost, killed by predators or other causes. Workers exit the colony random walking and identify a source of food and return to the nest leaving a trial marked by an odorous substance secreted by glandular structures. This odor trail quickly leads other workers to food, thus creating the long lines we are used to seeing. This is an extremely effective and secure technique, which - thanks to the evaporation of the odor trail - makes it possible to determine the shortest trajectory between food and nest. This amazing property has fascinated many mathematicians and computer scientist, inspiring the creation of algorithms that, for example, minimize the time it takes data to travel trough communication networks.

Ants have a stomach and a ‘pre-stomach’ (ingluvies), which performs a fundamental role in feeding the colony. The food ingested fills the ingluvies, but only a part of it goes into the ant’s stomach where it is digested. Such ‘social-stomach’ serves the purpose of a deposit, and its contents can be regurgitated for other ants that request it (a form of feeding known as trophallaxis). In this way a food chain is formed and it’s capable of feeding the entire colony in a short time.

A digital representation of ants’ behavior can be created in a computer through an agent-based simulation, specifically with NetLogo. This model includes only the class of workers which possesses a certain number of characteristics. These attributes consist of a series of quantitative variables like energy or pheromones, and some Boolean variables, whose value can only be true or false. One example of the latter variables tells us if the ant is carrying food or if it is near the nest and so on. At each step, if something upset what the ant was doing, all the variables will be updated and thus the behavior opportunely corrected.

Another important role is played by the patches. Since ants communicates trough chemical messages spread here and there by dropping pheromones, patches will keep trace of its concentration, and will let it evaporate as time goes by. The most expensive and maybe difficult task for an ant colony is foraging. Home-pheromones (HP) are released when an ant is leaving the nest, and like Hansel and Gretel fairytale, they are signals that will help the ant and its colleagues to find the way home. Each ant has definite quantity of pheromones and as it walks will leave a descending amount of it, so that higher concentration means closeness to the nest. When the ant finds food, it will take it and leave food-pheromones(FP) returning to nest by following a positive gradient of concentration of HP. Similarly ants that happen to feel FP will follow the positive gradient of the track in order to reach the food source.

Food will be represented by a breed; seeds will sprout randomly from the patches and will die when an ant takes one.

The final goal is to run a realistic simulation and since, by definition, a complex system is composed by a great number of agents, the size of the virtual world must be proper to this scope, and big enough to welcome all the existing and coming ants.

The queen will eventually produce new eggs and the colony will evolve for an indefinite period of time, accordingly to the food available.

2. The Script

The script begins defining the setup. Walls are opened, there are periodicity conditions and the space is thus wrapped in a toroid. At the center of the screen is located a nest that consist in a circle of radius 4.

The interface allows the user to choose the initial population of ants which only consist of workers. They are given a series of variables with the respective initial values

food and home pheromone	10
energy	600
social-stomach	200
Boolean hasSeed?	False

```
to populate
  set-default-shape workers "ant"
  crt population [
    set size 3
    set color black]
  ask turtles [set breed workers]
  ask workers [
    set hasSeed? false
    set ant-home-pheromon 10
    set ant-food-pheromon 10
    set energy 600
    set social-stomach 200 ]
end
```

After the setup of the population, all the patches are painted in white, except for the nest which is yellow. The user can also choose the number of food piles that will be present at the beginning. To create food I added a new breed – the seeds – and I asked to number of patches (equal to the number of initial food piles) to sprout an amount of seeds randomly chosen between 0 and 5, the same I did for the neighbors in radius 2.

```
to setup-food
  set-default-shape seeds "egg"
  ask n-of initial_food_piles patches [
    sprout-seeds random(5)
    ask patches in-radius(2) [ sprout-seeds random(5)]
    ask seeds [set color green]]
end
```

Now the ants are ready to emerge from the nest with specific tasks: if they are not carrying a seed they must look for it, and when they pick one, they are asked to bring it back to the nest. Since the variable hasSeed? is setup on False everyone will go in search.

```
to go
  if count workers > 0 and totfood1 > 0 [
    ask workers [
      consume-energy
      ifelse energy < 50 [eat2] [
        ifelse hasSeed?
        [return-to-nest]
        [go-foraging]] ]
. . .
. . .
  tick]
end
```

It is important to see that if their energy goes below 50, ants will try to survive instead of dying and will try to return to nest to take food.

During this phase ants will random walk: at each step there is a 20% probability of changing their direction, equally bipartite among turning left and right of an angle between 0 and 30 degrees.

```
to move
  fd 1
  let turn-angle random(10)
  if turn-angle < 1 [
    lt random(30) ]
  if turn-angle > 8 [
    rt random(30) ]
end
```

They will also release a quantity of home-pheromone increasingly smaller as they walk away from nest (10% less at each step). This procedure will be fundamental in the next phase in which they try to find the way home.

```
to release-home-pheromon
  if (ant-home-pheromon > 0 and not hasSeed?) [
    set ant-home-pheromon (ant-home-pheromon * 0.9)
    ask patch-here [
      set home-pheromon home-pheromon + (ant-hp * 0.9) ]]
end
```

At each step they will consume a unit of energy and insects will die when they run out of forces. However there are many strategies to survive. First of all if they find a seed and will manage to bring it back to the nest they can take 2 pieces of food if their energy happen to be below 100.

```
to eat1
  if energy < 100 and nest? [
    if [storage] of patch-at 0 0 > 1 [
        set energy 600
        set social-stomach 200
        ask patch-at 0 0 [
          set storage storage - 2]]]
end
```

When they are performing their tasks and the energy goes below 100, they will try to return to nest, but it is not always possible. For example they can get lost or can find themselves too far from the nest; in this situation if the energy is decreasing reaching the value of 50, they will ask for help. Trophallaxis allows hungry ants to get some food from a colleague that has stored it in a social stomach. This way they will get a little bit of energy (100) to return to nest and eat an abundant amount of food and totally recharge the battery.

```
to eat2
    return-to-nest
    let help one-of workers in-radius 2
    if help != nobody and [social-stomach] of help > 0 [
      set energy energy + 100
      ask help [
        set social-stomach social-stomach - 100 ]]
end
```

They random walk until they will smell a seed far at most 2 patches, they will approach and pick it. To take the food, the turtle “seed” will die, the variable hasSeed? turns True and the ant changes shape, appearing with food in the mouth. Now it’s time to get home.

```
to pick-seed
  let food one-of seeds in-radius 2
    if food  != nobody [
  face food
  fd 1]
  let food1 one-of seeds-here
  if food1  != nobody [
```

```
      set hasSeed? true
      ask food [die]
      set shape "ant-seed"
      rt 180]
end
```

The most difficult part of the whole project has been teaching ants how to follow tracks. The idea behind is very trivial: go where there is a greater concentration of chemical. Many attempts were tried without obtaining sufficient results.
I first tried to ask the ant to look ahead, left and right of 45 degree and see where the pheromone was stronger, then proceed to that patch. A first look gave the hope everything was going right, some ant reached the nest but unexpectedly the ants started to change direction even though there was a clear path ahead. This happened because many paths crossed each other and produced a great concentration in those points destabilizing the insects.

The second idea was allowing the ants to look all around them for 360 degree. The result was even worse. Carefully watching dozens of ants doing their job I found that an uncomfortable situation took place too often: if there was a couple of adjacent patches with a greater concentration than their neighbors, the insect were stuck in loops where they changed continuously direction without going anywhere. They were not moving but still releasing chemical: other ants were attracted and fell into this black hole.
This experience suggested that in no case the ant should turn itself for more than 90 degrees.

Then I thought about diffusion. How about asking the patches to diffuse some of the chemical to facilitate the ants reaching the destination? The problem is that the diffusion is exponential and very soon all the space was pervaded with a smell of chemical (too easy, exactly what I didn’t want to happen) and once again diffusion altered the odor trails making them impossible to be followed.

Actually, the tries have been much more and after hundreds of simulations suddenly the bright idea came to me. My ants lacked of a wider vision: they were looking only at their neighbors limiting the knowledge of the environment, what seemed more useful was looking also at the patches that were 1, 2 o 3 units far and moving toward them.

```
to follow-fp

  let fs (patch-set patch-ahead 1 patch-right-and-ahead 45 1 patch-right-and-ahead 90 1 patch-left-and-ahead 45 1 patch-left-and-ahead 90 1 patch-ahead 2 patch-right-and-ahead 45 2 patch-right-and-ahead 90 2 patch-left-and-ahead 45 2
```

```
patch-left-and-ahead 90 2 patch-ahead 3 patch-right-and-ahead 45 3 patch-right-
and-ahead 90 3 patch-left-and-ahead 45 3 patch-left-and-ahead 90 3)

   let f max-one-of fs [food-pheromon]
  ifelse [food-pheromon] of f > food-pheromon [face f fd 1] [move]

end
```

Once the ant gets a seed will follow track until the nest, will deposit the seed and will turn 180 degrees going back to the food source and so on. Every time an ant brings a seed to the nest, the amount of food stored increases by one.

```
to leave-food
  if nest? [ if hasSeed? [ set hasSeed? False
    ask patch-at 0 0 [ set storage storage + 1
      set totfood1 totfood1 - 1]
    set shape "ant"]
    set ant-home-pheromon 10
    set ant-food-pheromon 10
    rt 180
    ]
end
```

With this procedure of following pheromones it is possible not only to avoid stochastic fluctuations finding the right direction even in proximity of crosses, but also to slightly reduce the covered distance.

With a great surprise the insects stopped behaving abnormally and performed quite well, and formed the typical lines. One problem left to be fixed: when a food in a source finishes, the odor of pheromone persists and ants that happen to smell it, will continue to walk along the path. One possible solution is a highest evaporation rate for very high concentration: once the concentration reaches the value of 20 on one single patch, the evaporation will increase by reaching 5%. The rate will return to 1% after a reduction of the concentration below 20 units.

```
to evaporate
   if food-pheromon > 0 [set food-pheromon food-pheromon * 0.99 ]
   if home-pheromon > 0 [set home-pheromon home-pheromon * 0.99 ]
   if food-pheromon > 20 [set food-pheromon food-pheromon * 0.95 ]
   if home-pheromon > 20 [set home-pheromon home-pheromon * 0.95 ]
end
```

The concentration can be seen on the screen by enabling the proper switch. When the setting is ON, hp are showed in red and fp in blue. The shade of the color depends on the value of concentration: the higher the concentration, the stronger the color. If the switch is set on OFF during the simulation, it's not sufficient to stop showing the pheromone because the screen will display the last frame before changing the setting. A more drastic operation is required: the script restore the color as it was at the beginning: all patches in white except for the yellow nest. This procedure is included in 'To go'

```
ask patches [
      ifelse showTrack?[print-chemical] [ifelse nest? [set pcolor yellow][set pcolor white]]
      evaporate]
```

To print-chemical I used a particular tool of Netlogo that prints different shades of a given color according to a scale of values given. This scale sets the boundaries: the lower is the value, the lighter is the color and viceversa.

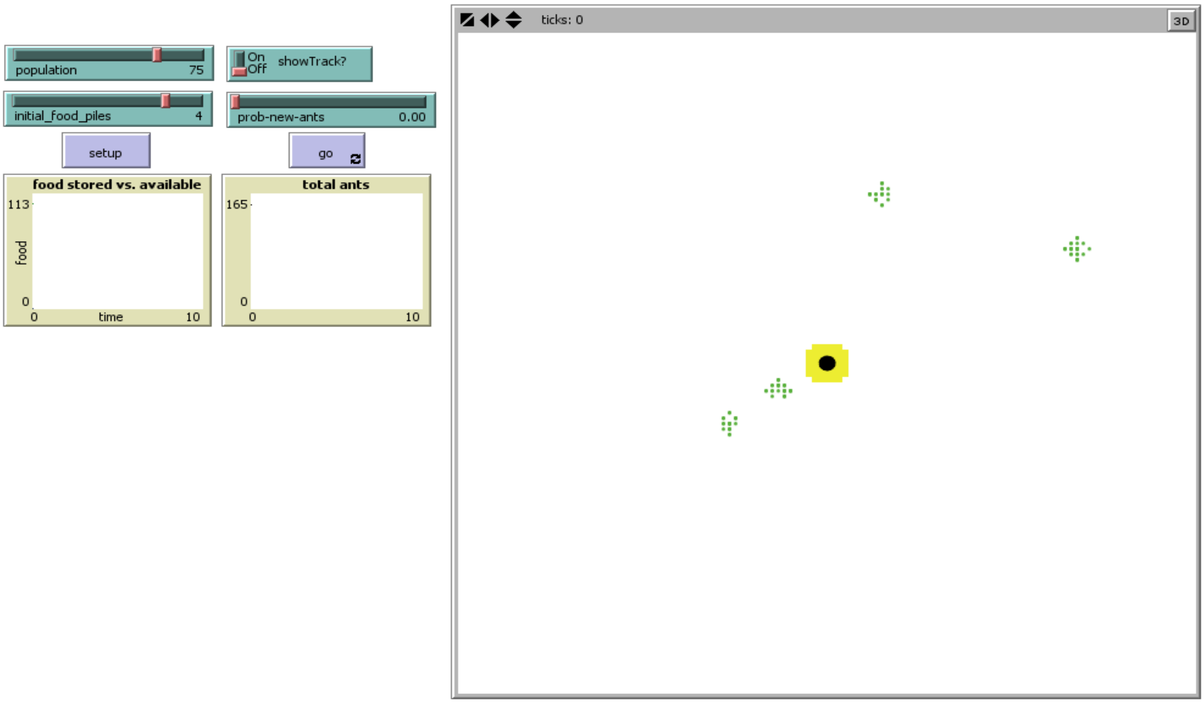
```
to print-chemical
  if home-pheromon > 0 [ set pcolor scale-color red  home-pheromon 20 0 ]
  if food-pheromon > 0 [ set pcolor scale-color blue food-pheromon 20 0 ]
end
```

The simulation will stop when all ants die or when all the food is collected. The second condition requires an explanation in terms of coding: it is not possible to stop the game when the breed 'seeds' has no more items. In fact this condition will stop the game even if there are ants carrying food home (since the seeds die as soon as they are collected from the ground). I created a new variable called 'totfood1' that is set equal to the food available at the beginning of the simulation; when a seed arrives to the nest totfood1 decreases by one and the run can stop when it reaches zero. Differently from the food stored, this value is not affected by ants eating provisions.

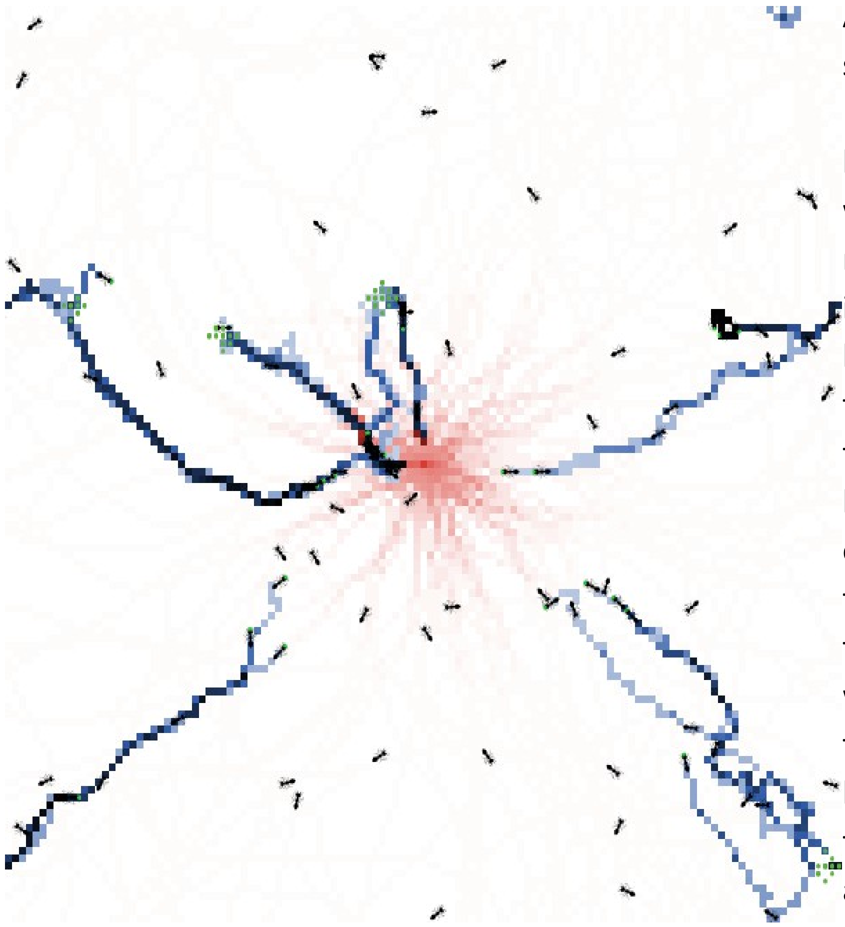
```
to go

  if count workers > 0 and totfood1 > 0 [
    ask workers [ … ]
end
```

3. Experiments

Once the script is completed it's time to test it. At first sight the user faces the following screen



There are various possibility of customization: initial number of ants and food piles, either to show the presence of pheromones or not, and finally the probability new ants will leave the nest at each step.
Once the simulation is started, and the showTrack? Slide is on, the next figure shows how the world looks like when it contains 90 ants.



At the center, the nest is surrounded by the presence of HP (in red) and there are some blue lines connecting the nest with the food piles and they reveal the presence of FP. It's important to notice that these lines are supposed to optimize the distance between nest and food and thus the paths are likely to be straight. The lack of complete straightness derives from the fact that ants leaving the nest perform a random walk, even though the chance to turn left and right are very low. Of course the further the food is, the weaker the odors are, and it becomes more difficult to optimize the path and make it as straight as possible. Nevertheless, ants seem to find the way home with not so many difficulties.

The chance to move the slides gives the possibility to perform different tries to study their behavior in different conditions. For this purpose I created an experiments using the 'Behavior Space' tool available on NetLogo. I varied the number of ants starting from 10 and ending at 250, taking a step of 10. In addiction I decided to record 20 repetitions, this seemed a proper number that captured the necessary information. The data plotted hereafter are the result of a mean of 20 values.

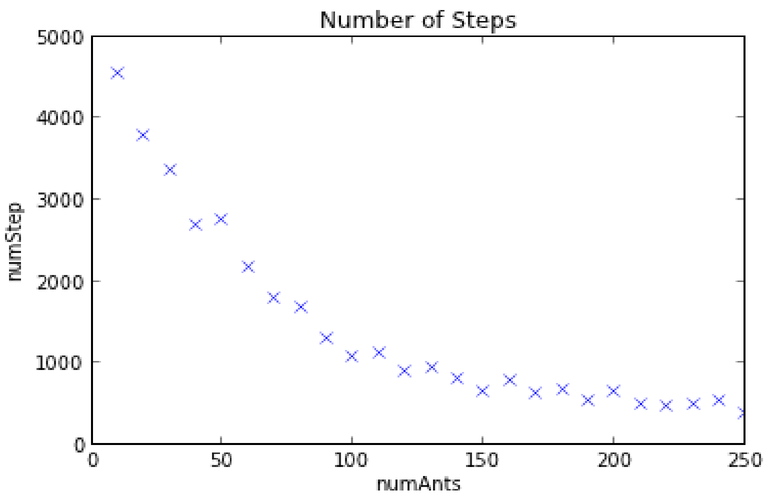
The quantities of interest were 7 in total:
- Initial and final number of ants,
- Initial seeds presents and stored food,
- Number of steps necessary to collect all the seed (or, eventually, necessary for all ants to die).

This significant amount of data has been used to get the trend of three quantities as functions of the number of ants:
- Number of steps,
- Food Efficiency,
- Ants Efficiency.

This analysis was performed using python optioned with a couple of libraries (Numpy and Matplotlib).

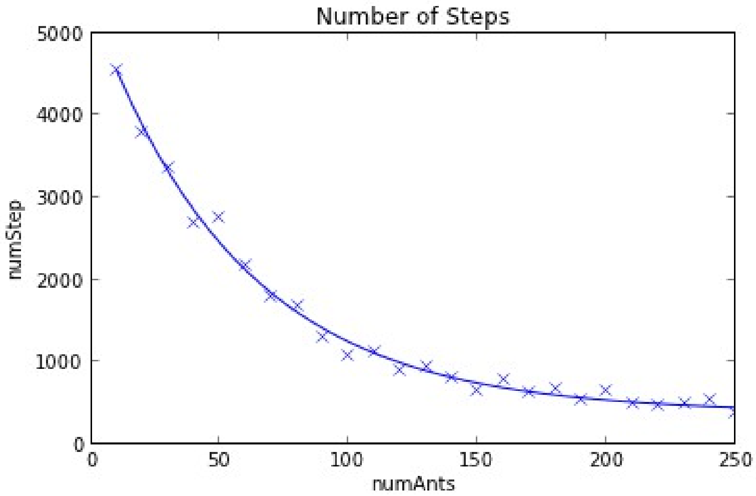
3.1 Number of steps vs. Number of ants

When it comes about forecasting the trend of the number of step in function on the ants it's immediate to think that, since an ant colony is a complex system, the best configuration happens with an increasing number of components. Moreover the initial distribution of ants in all the direction and progressively the homogeneous spreading of the insects makes it easier to detect food sources. What is not immediate is to guess how the steps decrease and here is when collected data helps. The following figure was obtained simply plotting the two quantities just mentioned.



The effective trend confirms the prediction but I considered worthwhile trying to approximate the points with a curve. I tried both with a parabolic and with an exponential fitting, but the second won since it scored the smallest Chi Squared.

This picture shows the result.

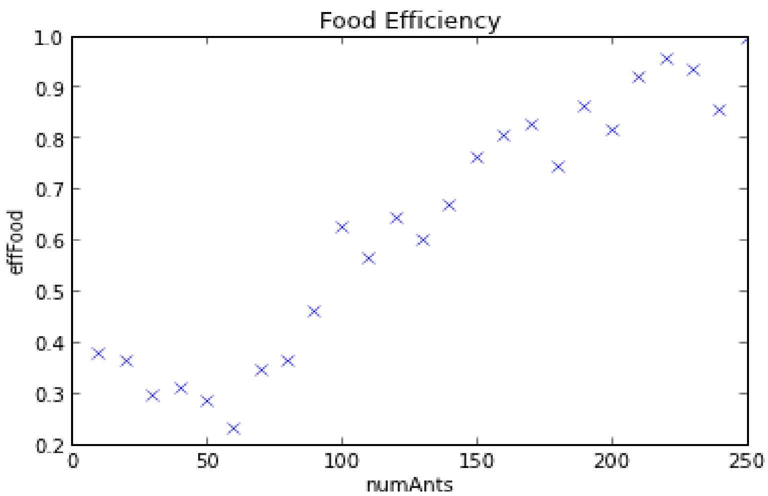


3.1 Food efficiency vs. Number of ants

The goal of an ant colony is to collect the largest amount of food possible deploying the least resources. I called thus 'Food efficiency' a quantity that takes into consideration the food collected in the expedition and the food eaten by ants to survive and complete their tasks:

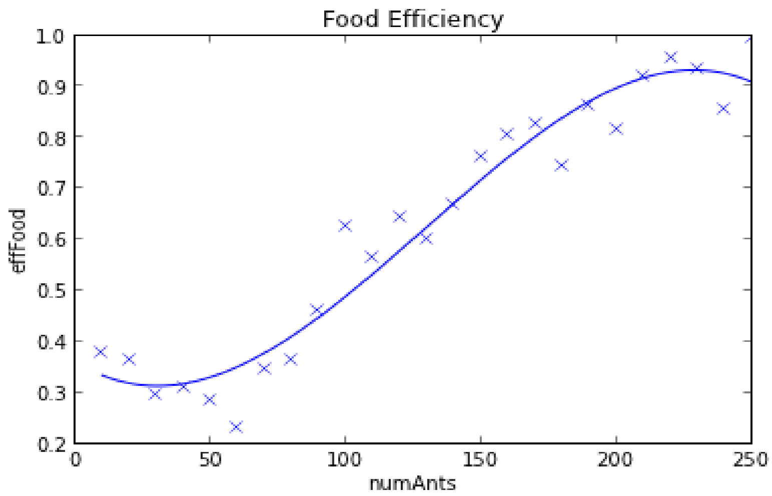
$$E_{Food} = \frac{FoodCollected - FoodEaten}{TotalFood}$$

This quantity is a percentage (ranging from 0 to 1) and the expectation, as above, is somehow an increase of the efficiency. Here the points obtained are reported.



Unexpectedly there is a minimum not for very few ants but around 60. A possible explanation is that 60 ants perform similarly to a lower number of ants but, since they are more, they consume more food. In fact there is a short plateau staring from 10 ants and decreasing around 60. This effect is unlike to be a consequence of causality: 20 repetitions are made for each step and the chance this point doesn't respect reality is very low. I would like to precise that the data plotted above must be referred to this specific world (with its specific dimension, number of piles, etc..) and the position of the minimum is with all probabilities a peculiarity. As the insects increases, the system in its total becomes more efficient and the typical ant lines are fundamental for raising the efficiency. Since the quantity plotted is a percentage, the improvements made by ants cannot overcome the value of 1, and the curve saturates.

In order to capture the minimum a possible function that can interpolate the points is a third degree polynomial (or higher). In order not to loose generality I decided to use the third grade since it seemed to follow the effective trend.



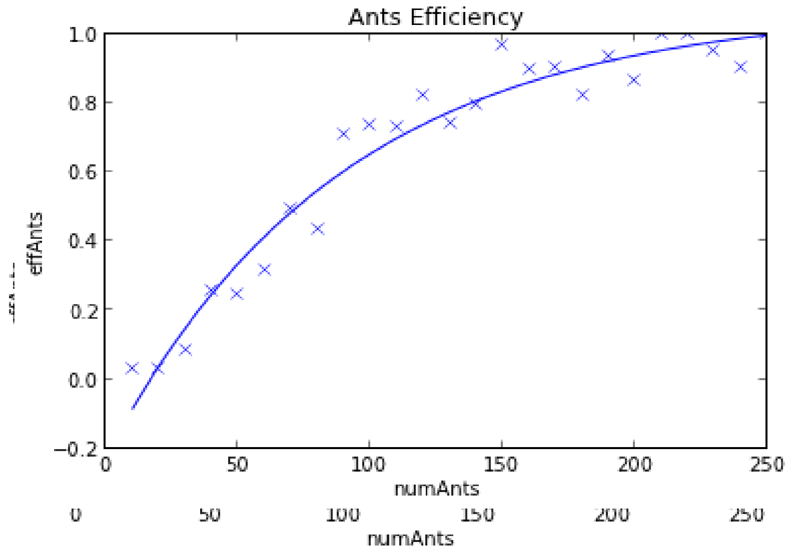
3.1 Ants Efficiency vs. Number of ants

The last analysis regards how many ants succeed in staying alive for the whole duration of the simulation. Together with the food stored over the total, this value can give another information about efficiency of the colony.

Like the previous case, it's important to define a quantity that describes this efficiency:

$$E_{Ants} = \frac{AliveAnts}{InitialAnts}$$

It reaches its maximum when all the ants stay alive. Since they have a definite amount of energy, and they consume 1 item at each step, most probably the death rate will decrease following the decreasing of the step of the simulation as we have seen above. The expectation is thus an increase of the efficiency that follows an exponential growth, and that saturate at the value of 1. The next figures show the data collected in the experiment and the points interpolated by an exponential.



In conclusion, I can say that the improvements made by an increasing number of ants follow an exponential function, except for the food efficiency that has a minimum whose position depends on the specific world the ants are acting in.

4. References

- S. Cacace, E. Cristiani, D. D'Eustacchio, Mymedrome: Simulatinifg the life in an ant colony
M. Emmer, Imaging Math. 2. Between Culture and Mathematics Springer-Verlag
Italia, 2013.

- Deborah Gordon, The emergent genius of ant colonies
TED Conference, 2008.

- David Shindleman, The behavior of ants
Department of Biochemistry and Molecular Biophysics
The University of Ariziona, 2002.