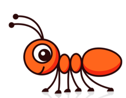
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Agent-Based Modeling and Social System Simulation

Desert Ant Behaviour

Tobia Ochsner, Robin Schmidiger, Pascal Strebel



Contents

1 Introduction

Especially in deserts, ants need to travel big distances when looking for food. How does a creature with a brain that small finds its way back home? As often in biology, there is no exact answer but there are several theories based on experiments which have shown interesting results.

Since an ant's search behaviour is pseudo-random, the outward path (home $\rightarrow$ food) is more likely to be indirect (curvy) as we would imagine. It however occurs that the way back home (food $\rightarrow$ home) is often straight. It seems that an ant abbreviates the inverse direction of a path it has taken before rather than retracing it.

Experts explain this phenomenon due to something called **path-integration**:
"...[the ant] uses path-integration as an egocentric guideline to acquire continually updated spatial information about places and routes."[1]
In simpler words, path integration is the ability to take the (more or less) direct way back home. An ants brain remembers virtual **landmarks** on the outgoing path which simplify path integration. From each landmark, the ant kind of knows the direction and distance to the previous landmark (reversed vector). On its way back home, it is able to reduce those directions and distances within some **vision radius** and therefore to find a way that is more direct.

People working in the corresponding research fields have been coming up with formulas in order to approach this path integration process by evaluating experiments. We want to simulate and visualize the animal's foraging process based on these formulas and understand the influence of certain physical properties that an ant might have towards its ability to correctly perform the path-integrating. We do this by introducing parameters such as vision radius or movement speed before we predict and measure the inaccuracies that may occur.

What we won't do is making the ants learn. Our ants will have their skill set (parameters) which don't change. The ants won't influence each other and their success is mostly based on their ability to perform path-integrating. This may be very simplified. However, the way path integration has been explored so far was influenced by so many experiment-specific factors that would be exaggerated to introduce if we want to simulate pure nature.

2 Basic Setup

Our playground has finite size and consists of the following things:

- Home
- Food
- Ant
- Landmarks
- No obstacles since we're in the desert

During the simulation, we release a number of ants with biased-random search behaviour which is influenced by (static) parameters. During the search, landmarks can be memorized. After the food is found, each ant tries to find its way back home by performing path-integration. This is also affected by its parameters, namely they may cause deviation of distance and angle between two landmarks. We track the ants during their whole journey. Ants don't interact with each other.

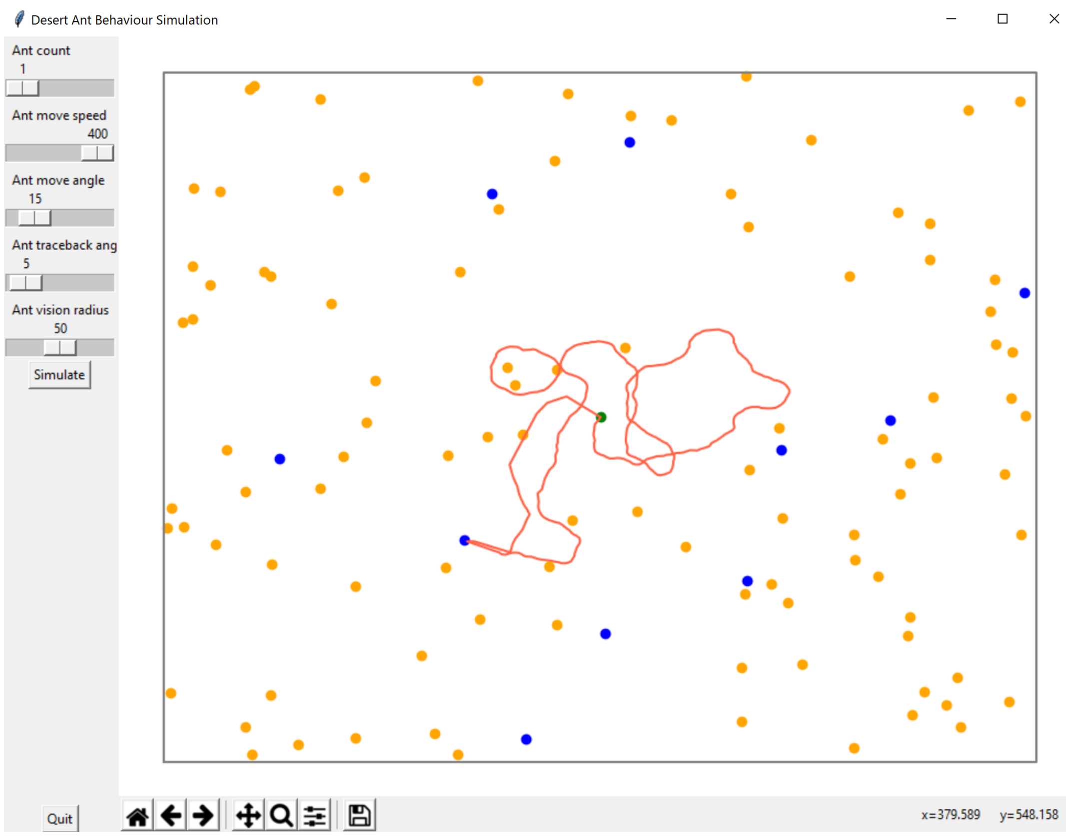


Figure 1: Program Environment

3 Behind the Code

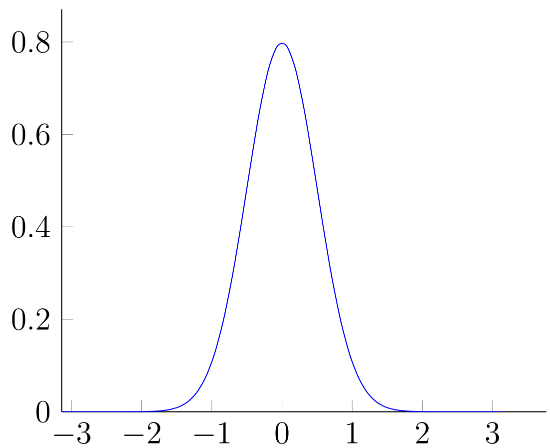
As already mentioned, we are using parameters to specify an ants abilities:

- **Vision radius** r
- **Movement speed** v
- **Move angle** α which is the standard deviation of its current direction during search behaviour
- **Traceback angle** β which is the standard deviation of the direction to the next landmark's when heading to the nest

These parameters completely determine an ants behaviour during search phase and path integration.

3.1 Random Walk

We make the search behaviour biased-random by drawing random samples from a **normal (Gaussian) distribution** to simulate changes in direction.



That way, the ant is more likely to keep moving into a similar direction at every time step we reevaluate its direction. If we made the walk truly random, it would keep changing its direction all the time and probably stay on the same spot forever. As papers have shown [3], this actually reflects nature quite well.To prevent the ant from walking too far away and never finding any food, we introduce boundaries.

When the food appears in its vision radius, an ant directly walks towards it and then proceeds with the path integration to find home.

3.2 Path Integration

Müller and Wehner[2] suggested the following equations to approximate path-integration in a mathematical sense:

$\varphi_{n+1} = \frac{l_n \cdot \varphi_n + \varphi_n + \delta}{l_n + 1}$
$l_{n+1} = l_n + 1 - \frac{2 \cdot \delta}{\pi}$

φ_n is the direction in which the ant has covered the distance l_n after the n_{th} step. δ is the angle about which the ant has turned after that step and therefore $\varphi_n + \delta$ is the direction in which it proceeds for another unit length.

They also mentioned that when ants had to make bigger turns (angle-wise), this model wouldn't fit as good as for smaller turns. However, since an ant that keeps changing its direction completely (without a reason) while searching probably wouldn't find the food in time and therefore die rather sooner than later (thanks to Darwin), we don't expect ants to behave in such a way (the ants in Müller's and Wehner's experiments were made to do so by tunnels). Therefore we can work with this model as a basis for our own implementation.

The path integration in our model can be described in the following visualized way:

- After food is found, the ant heads (more or less directly according to its parameters) towards the last landmark it has seen.
- Every time another known landmark pops up in its vision radius, it considers the corresponding vector in addition to its current direction. This means that all known landmarks within the vision radius are reduced to one vector pointing to the next prior landmark.
- An ant vaguely knows in how many steps it should reach the next landmark. For this purpose, we introduce a **certainity** that starts to drop as the walked steps exceed the predicted ones to prevent an infinite walk in the same direction if a landmark was missed. When this certainity becomes too low, the ant goes back to search behaviour until it finds a known landmark or its home.

4 Program Structure

For simplicity, we split our program into the following classes:

Simulation	Contains list of all ants Has an **Environment** Simulates ants
Ant	Search behaviour Go to nest behaviour Go to food behaviour Ant state (pos/rot/behaviour)
Parameters	Movement speed Angle parameters Vision radius Dist/angle error Ant count
Environment	Nest Food Landmarks
Data	Contains list of all paths taken
GUI	Widgets for setting simulation **Parameters** Displays **Ant Paths** Displays **Environment**

We will now look at some of these classes a little more in detail.

4.1 Simulation

Ants basically may take unlimited time searching for either food or their nest. To prevent the program from crashing in such a case, we run the simulation on multiple threads which could each be killed in such a case.

4.2 Ant

Describes a single ant (state and behaviour). An ant can either be searching, going back to the nest or be on its way to the food (if it is found). Initially the ant will be pseudo-randomly searching for food. Once food is found, it remembers which landmarks it has seen in the past and how to get from each previous landmark to the next. That way the ant will be able to trace back the path it has taken from its home to the food. This means that the ant associates a single **directional vector** with each landmark which denotes the relative position of the next landmark on the path from the nest to the food. If it sees multiple landmarks, it will take the sum of the directional vectors.

4.3 Environment

Describes an environment containing landmarks and food on different places. The home is assumed to be a landmark. The **visual range** describes the distance to landmarks which are seen, the **action range** the minimal distance of the food to eat it and to the nest to enter it. Food only gets noticed in the action range, whereas landmarks and the nest get noticed in the visual range.

5 Hypothesis

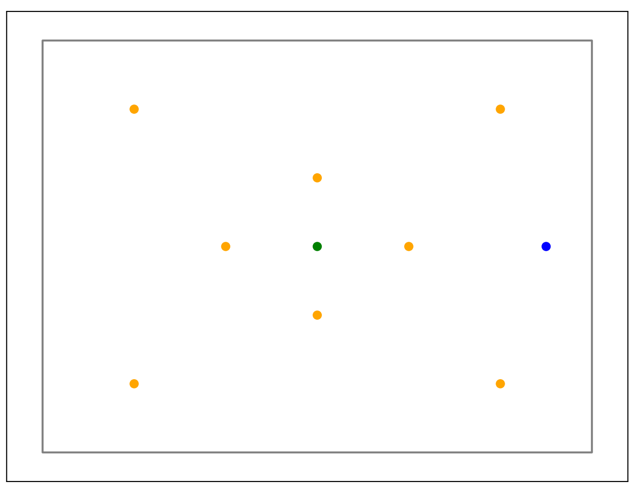
Before actually running our simulation, we will make some predictions considering special parameter settings.

- It takes faster to go home than to find some food (i.e. path integration works).
- If the **vision radius is too small**, an ant will get stuck.
- The **bigger** the **vision radius** gets, the more direct the way back home should be.
- If there are **too few landmarks**, many ants will get lost multiple times and therefore return into search behaviour.
- If there are **too many landmarks**, it might happen that the ant doesn't find home because it keeps walking in a cycle between multiple narrow landmarks that are all in the vision radius and close to the nest.

We will come back to the evaluation of these at a later point.

6 Simulation

To end up with general results, we simulate 100 ants per step and use the same fixed $600x600$ environment for all our tests.

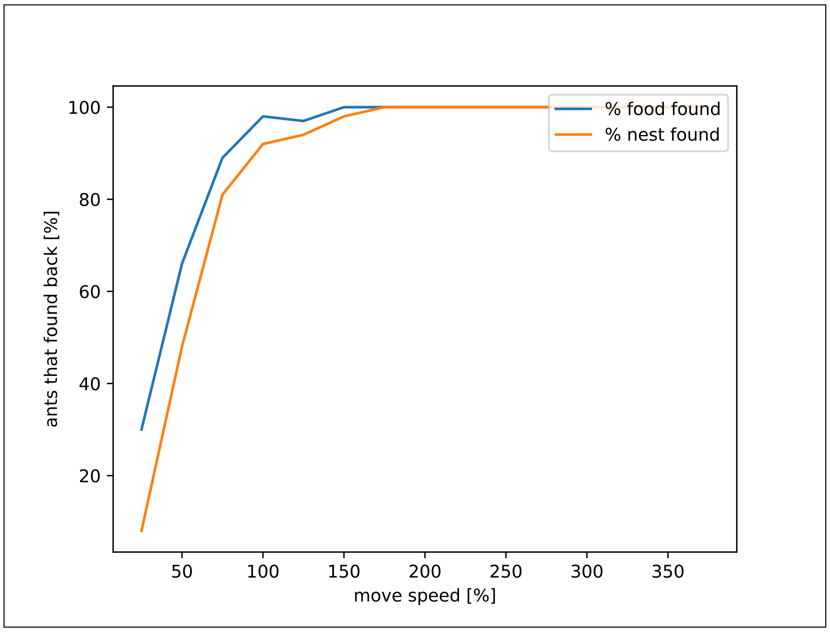


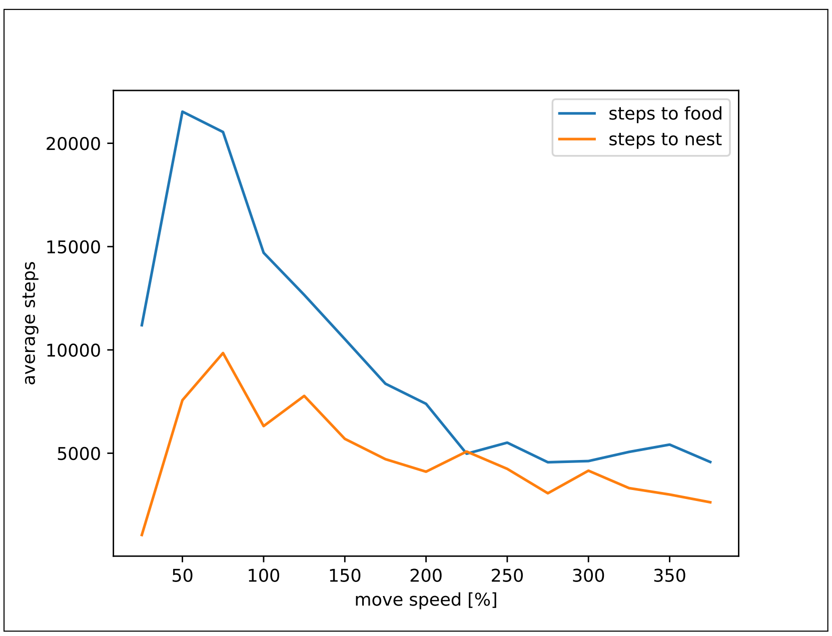
Furthermore, we set the following standard parameters:

Movement speed:	100%
Search Angle:	15°
Traceback Angle:	5°
Vision Range:	25

6.1 Movement Speed Test

We see that with increased movement speed, it is more likely for ants to find their way back home within the time limit. At a movement speed of around 160% basically all of them manage to find the food and make it back (Graphic 1). It is interesting to observe that at a certain point of about 225% movement speed, the search for food stops to become faster while there remains a gap between the *number of steps to food* and the *number of steps to nest* (Graphic 2). This makes sense however since the former is *randomized* while the latter is *calculated*.

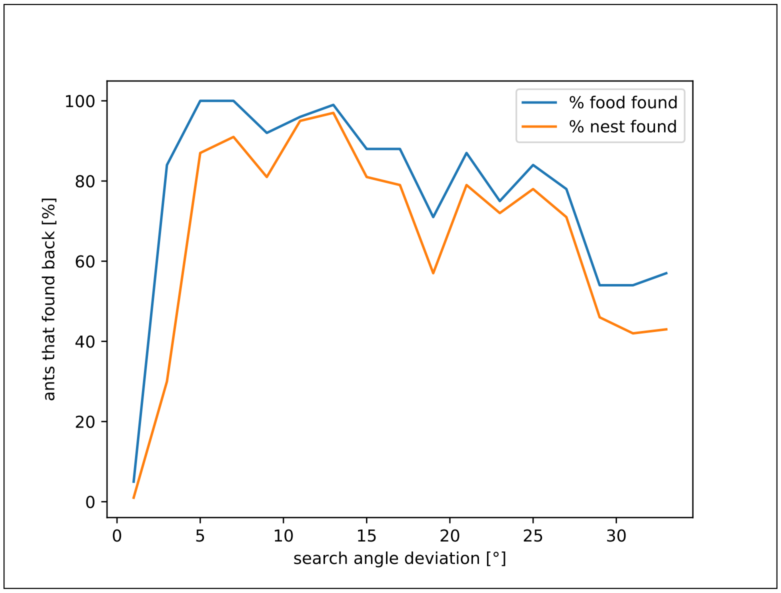


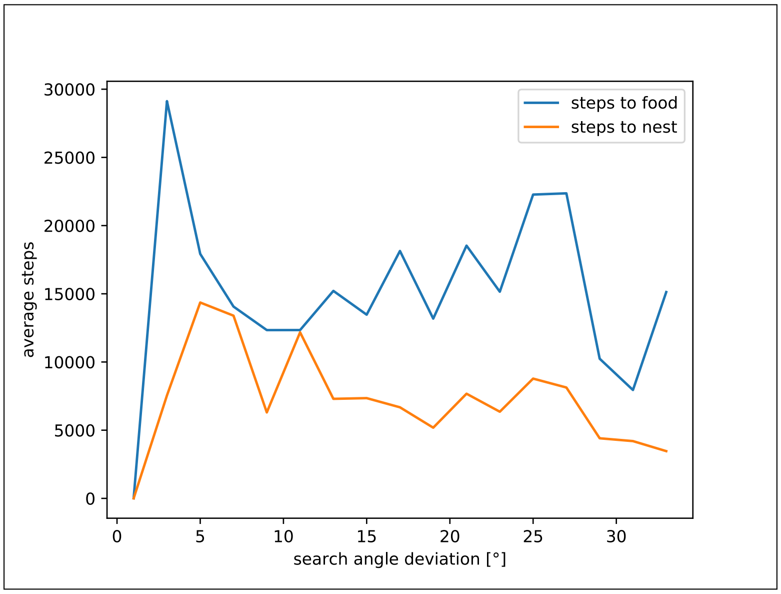


6.2 Search Angle Test

When the search angle deviation is too small, the ants are too conservative. This might be fatal for both, the random walk and the path integration, since the ants never even get to look for their way back since they will just walk in the same direction during the random walk with a low possibility to ever find some food (Graphic 1).

In fact, there is no *best* value for this since it is really situation-dependent whether an ants search behaviour should be more or less conservative. This reflects in the curve that doesn't converge too much (Graphic 2) after the compulsory fails described above. It is also interesting to see that the two curves behave similarly due to going back is dependent on finding food.

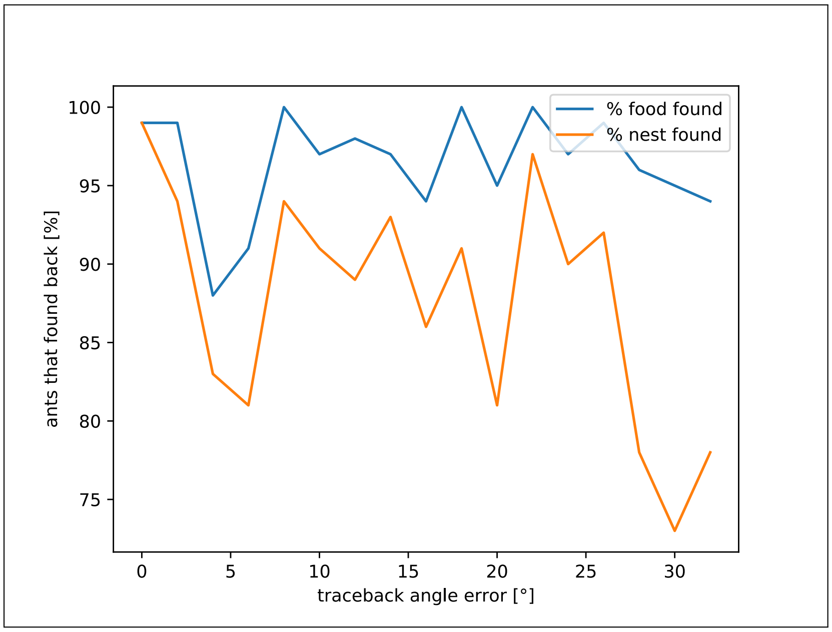


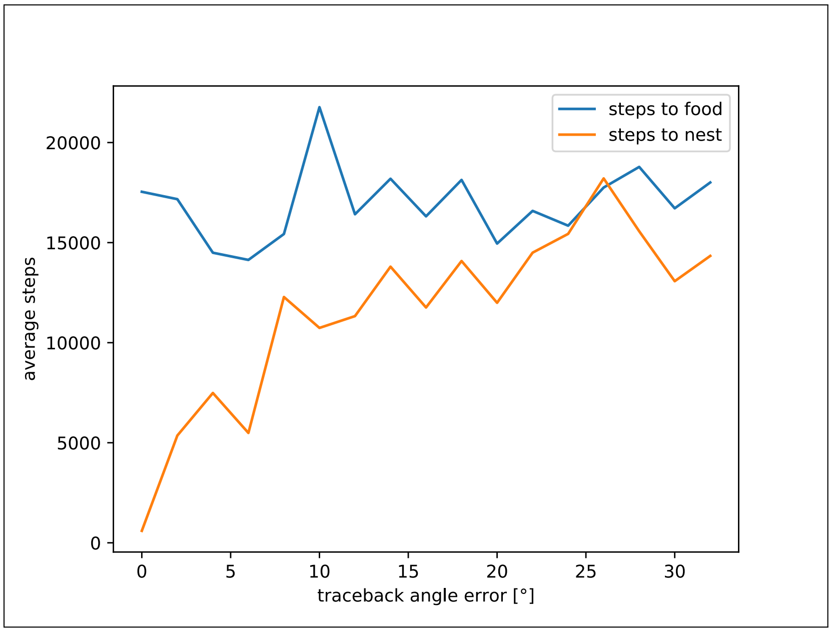


6.3 Traceback Angle Error Test

The error in the traceback angle only influences path integration which means that the peaks for *food found* and their corresponding influence on the *nest found* graph are of random nature (Graphic 1).

While this parameter is increasing, the ants obviously become less successful. Despite most of them managing to find home, the amount of steps needed for therefore growing merely linearly (Graphic 2).

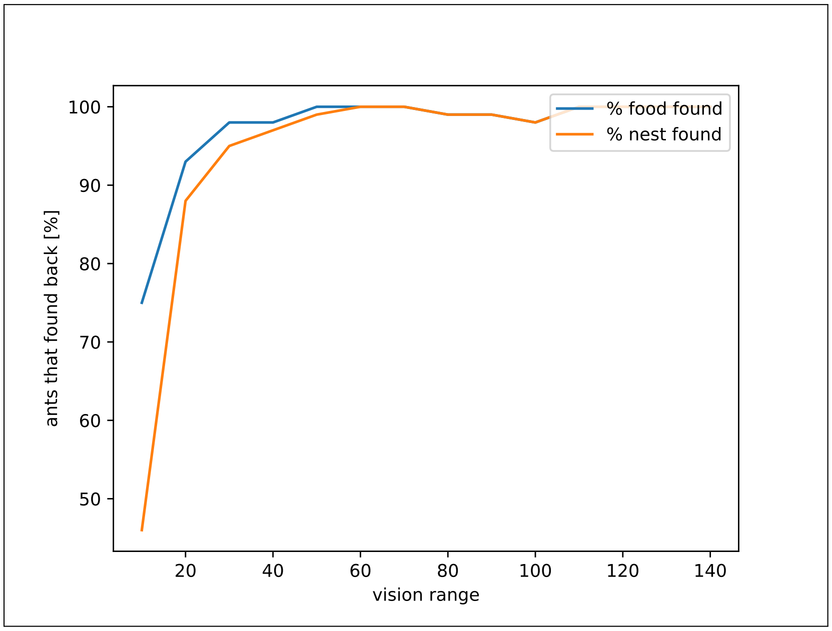


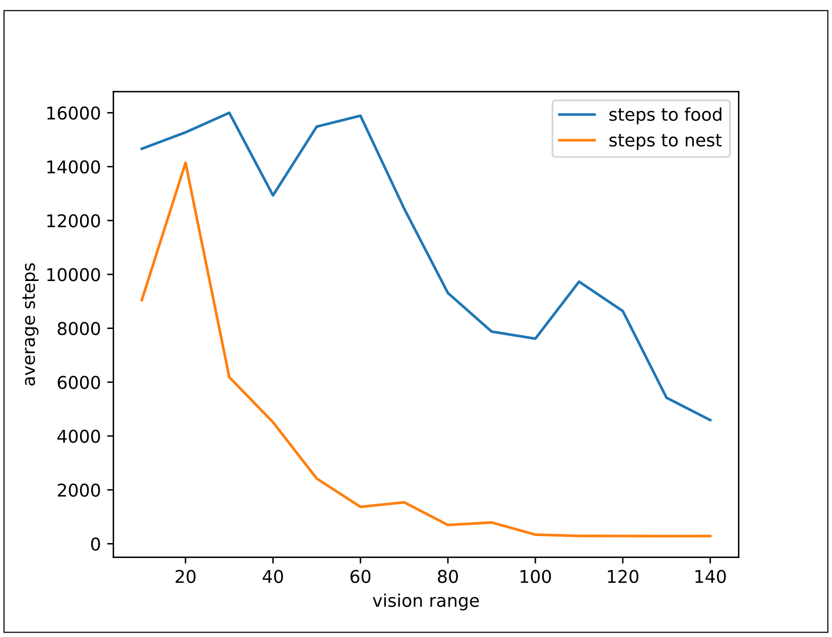


6.4 Vision Range Test

It is clear that the vision range has the most beneficial influence onto ant performance. When too low, there is the possibility to keep missing important landmarks. Increasing vision range only has positive effects on path integration but obviously is a bottleneck in real nature.

Later, we will discuss some bugs introduced in context with vision range.



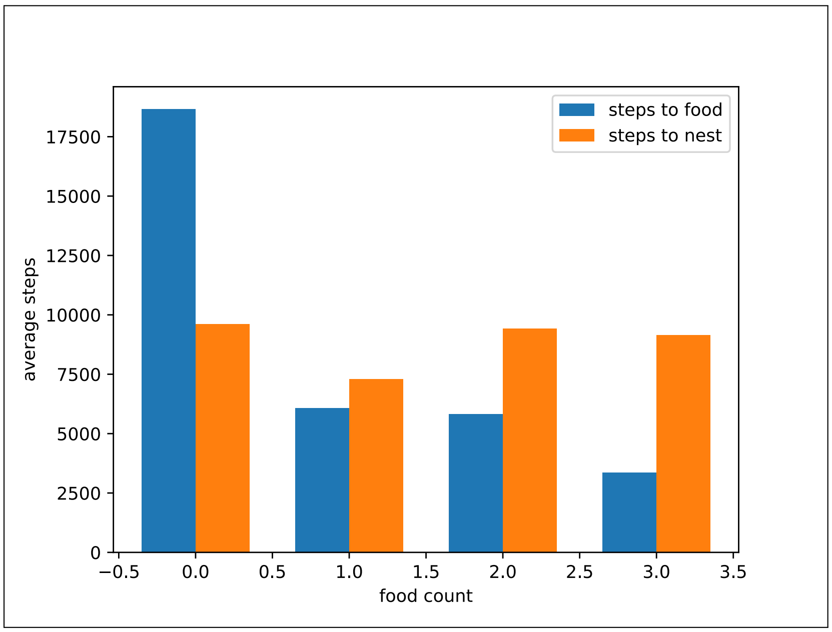


6.5 Food Count Test

All the previous tests have, as mentioned, taken place on the same playground which contains only one piece of food. Last but not least, we will look at the performance changes if we increase the amount of food on the environment.

When doubling the food, the number of *steps to food* massively decreases since it is logical to find things with random search behaviour if there are more things. Further increases to this have a way lower impact but still are beneficial.

Also, this has no effects on *steps to nest*.



6.6 Observations

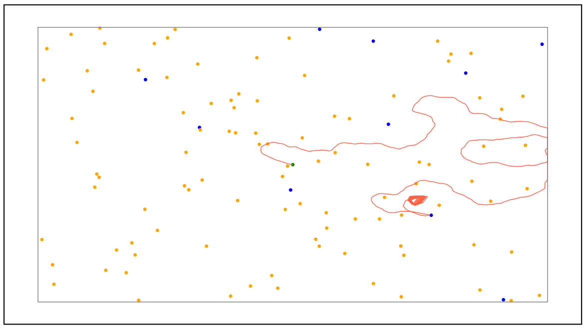
We observe that when simulating really many ants, most of them achieve similar performance. However, there are some "stray bullets" that take **much** longer since there is no bound in time. These sometimes strong deviations between single data points probably occur due to the strongly random nature of the simulation.

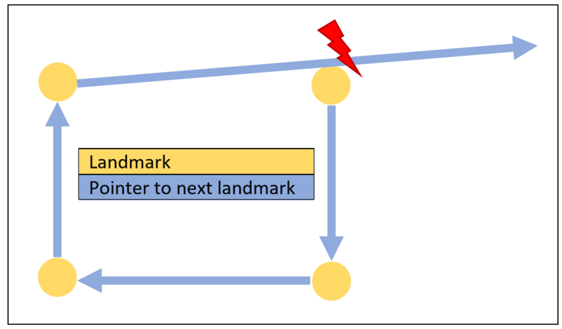
Furthermore, thermal throttling might influence the percentage of ants that find back to the nest. This is because the simulation has a timer based cutoff which favours faster CPUs. But if an ant really gets lost it usually is easy to tell, since the lost ant takes orders of magnitude longer to find the nest than an ant that isn't lost.

Simulating "*only*" 100 ants per iteration might seem like a small number. But since each ant can take up to 5 seconds, each core handles 25 ants and there are about 30 iterations per simulation, a single simulation already takes around 45-70 minutes.

6.7 Hypothesis Evaluation

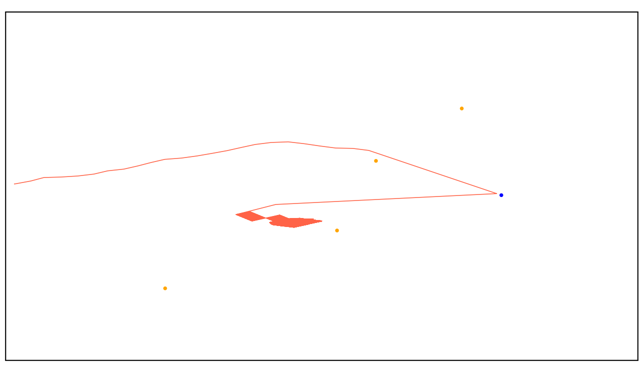
The predictions we made were mostly fulfilled. While the effect of the size of the vision radius was rather trivial to foresee, the landmark frequency in fact played an important role in the performance of the path integration. Especially the occurrence of **too many landmarks** lead to some interesting results which we will explain in detail.





As we can see, the ant keeps walking in a repeating pattern between several landmarks. This happens due to a special constellation where there are multiple known landmarks in a **similar direction**. If the ant needs to walk less steps than expected to find a next landmark, it will use the found one (which is in it's vision radius) rather than going further to the one it was actually looking for (which is not yet in it's vision radius). However, the landmark it found "by accident" might point back to a landmark it has already passed on it's path integration which leads to such cycles.

Therefore it may also occur that the ant keeps going back and forth (zigzagging) between two landmarks with vectors facing in nearly opposite directions as seen in the next graphic.



This could be fixed by either introducing a bigger vision-radius or less landmarks. However, having too many landmarks makes it unlikely to ever miss a landmark and return into search behaviour.

7 Reflection and Optimizations

We successfully created a simple simulation of real-life ant behaviour which we then simulated with regard to different factors. With this, we could evaluate the effect of settable parameters on the performance of path integration. Questions that remain unanswered but still are of interest (and thus may be tackled by the responsive reader using our simulation as a base) are:

- Does path integration improve if we add **multiple entrances** to the nest?
- How is the result affected when we **change the landmark vectors** from each pointing *to the previous seen landmark* to all pointing *to the nest*? (The way theoretically becomes shorter but this comes with cost in accuracy since errors in the angle have more influence)
- What happens if we **introduce obstacles** that ants have to walk around?

Also, there are many optimizations that could be implemented to possibly gain even further insight in desert ant behaviour, some of which may be:

- Simulate with multiple **different environments**, which takes a lot of time
- Simulate with **more ants**, which takes a lot of computation power
- Use a number of steps **cutoff** rather than cutting off at after a certain amount of time (since processors behave differently)
- **Improve code** to make gains in efficiency
- Make ants **interact** with each other (e.g. by leaving pheromones behind)
- Introduce a **learning effect** by simulating the "*same*" ant again and again
- **Expand the GUI** visually but also content wise with useful information, numbers and statistics that are directly visible during the simulation

During this course in *Agent-Based Modeling and Social System Simulation*, we learned a lot about mathematical and computational ways to model natural behaviour. We faced problem that arise in development and implementation of such models and experienced the possible applications of simulations in practice.

Appendix A Declaration of Originality

ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

Desert Ant Behaviour

Authored by (in block letters):
For papers written by groups the names of all authors are required.

Name(s):	First name(s):
Ochsner	Tobia
Schmidiger	Robin
Strebel	Pascal

With my signature I confirm that
- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date
05.12.2019

Signature(s)

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.

Appendix B References

[1] R. Wehner:
Desert ant navigation: how miniature brains solve complex tasks.
Department of Zoology, University of Zurich, 2003.

[2] M. Müller, R. Wehner:
Path integration in desert ants, Cataglyphis fortis.
Department of Zoology, University of Zurich, 1988.

[3] M. Collett, T.S. Collett, S. Bisch, R.Wehner:
Local and global vectors in desert ant navigation.
Department of Zoology, University of Oxford 1998