Simulation of ants gathering food using pheromone trails

David Åsberg - dasberg@kth.se

February 2022

1 Background

Ant colony optimisation algorithm is a technique typically used for finding a short path through a graph. The algorithm models the behaviour of real ants by having a large number of agents (ants) move through the graph whilst leaving pheromone trails. These trails allow the ants to communicate in a sense. An typical scenario is where an ant wanders randomly until it discovers food, from where it will release pheromone trails until it finds its way back to the nest. Now other ants can follow this trail instead of wandering randomly to find the food. The pheromones also evaporate over time, which causes the algorithm to favour shorter paths. This is because on a longer path specific parts will be marched over less frequently and therefore have a weaker trail than on a shorter path.

2 Problem

The problem is to simulate how ants search for food using pheromone trails that evaporate over time. Two types of pheromones will be used, one for when the ants are searching for food and one for when they are returning to the nest. A similar simulation has been done by other people before.[1]

3 Implementation

The implementation is divided into two different main parts:

1. **Food gathering**
   The ants will wander randomly within the map boundary and leave a trail of pheromones that will lead back to the nest. The ant will at all times sample the concentration of pheromones in front of it and move towards the direction of highest concentration. This means that "highways" will

form on which a large number of ants will move back and forth between clusters of food and the nest.

2. **Crowd avoidance**
   To simulate how the ants would move in real life, a simple crowd avoidance algorithm will be used to keep the ants from walking into each other. This means that a simple separation force will be applied to the ants to keep them from getting too close to each other.

4 Extensions

The following features will also be considered if there is time for implementation:

1. **Quad-tree implementation for improved performance**
   Using a Quad-tree data structure to
2. **Use Unity DOTS for better parallelisation and scalability**

References

[1] Sebastian Lague. Coding Adventure: Ant and Slime Simulations, Mar 2021. [Online; accessed 24. Feb. 2022].