Investigating the properties of pheromones on colonies of simulated ants by evolving properties tailored to different habitats.

Submitted in partial fulfillment for the degree of BEng
Department of computer science

PRBX

Vince Da Silva

10/05/2022

Supervisor: Sam Braunstein

Dedication:

Thank you Sam Braunstein for accepting this project and allowing me to work on something which I’m passionate about, providing me with optimistic insight, and enduring my terrible time management

Also thanks mum

Table of Contents:

Table of Figures:

Table of tables:

Executive:

Pheromone communication is an area of interest in a range of fields, spanning from swarm robotics to entomology to optimization problems, in this paper we aim to expand our understanding of pheromones and how certain attributes such as the rate of evaporation and diffusion, affect overall communication. To benchmark the performance of the pheromones, we create a simulation of an ant colony gathering food using trail pheromones. The more food a colony can gather, within a set time window, the better suited the pheromone is to the particular type of food source. By using a genetic algorithm to tailor a set of pheromones used by an artificial colony of ants, we are able to optimize a set of pheromones to specific niches. By comparing these specialized pheromones in all the types of habitats we have been able to conclude that the properties of pheromones have a large impact on the effectiveness of communication and foraging, we conclude that a specific pheromone can be designed to optimize for a specific problem. We consider how varying such properties of pheromones could affect previous literature exploring pheromone communication.

Introduction

The field of computer science has benefited greatly from the observation and study of natural phenomena. Methods have been abstracted and crafted from concepts in all fields of natural sciences, from simulated annealing derived from material science to the revolutionary mimicry of our brains in the form of neural networks.

A recent area of interest in AI has been algorithms based on social insects such as bees, termites, and ants. The interest in social insects is their ability to self-organise, individuals of minimal complexity can generate greater complexity at the population level. This greater complexity can be used to solve tasks as a collaborative population whilst keeping each individual computationally and logically light.

A common task in ant and other insect colonies is *central place food foraging*, which consists of two phases [1].

1) Individually exploring to find a large food source

2) Recruitment of fellow workers to gather and deliver the discovered food back to the nest.

In ant colonies recruitment takes many forms. In its simplest variant, *tandem running,* a second ant is physically led to the food. However, *group recruitment* is more common, this consists of short-term pheromone trails being laid to guide small groups of ants to a food source. In certain species with typically large food sources, *mass recruitment* can take place, this is where hundreds or thousands of ants consecutively reinforce a pheromone path lasting indefinitely [2]. This encoding of useful information via pheromones modifying the environment is termed *stigmergy* [3].

Pheromones are chemicals that an animal produces to alter the behavior of another animal of the same species. Ants can have multiple types of pheromones, with multiple unique purposes [4]. Some ant species such as Lasius niger or the Argentine ant Iridomyrmex humilis use trail pheromones to mark paths on the ground. The paths can be sensed by foragers to help lead them to areas of interest such as food sources [5]. These pheromones are volatile compounds that have properties such as

- Volatility: How quickly the chemical evaporates
- Strength: The attractiveness of a pheromone
- Rate of Diffusion: How quickly the chemical diffuses/spreads

Depending on the purpose of these chemicals their ideal properties may be very different. For example, in the case of the Malaysian ponerine army ant, the pheromones used to lure soldier ants to prey last only 5 minutes, before evaporating to undetectable levels. Whilst the pheromones used for trails leading workers back to the nest can last longer than 25 minutes. [6]

AI-inspired, insect stigmergy literature has focused on developing strategies for agents to optimally place and interpret pheromones, to collectively complete tasks including foraging [7]. However, the effects of the properties of pheromones such as those described above have not been thoroughly investigated.

Therefore, I will explore the effects of pheromone properties on an artificial colony of ants completing foraging tasks. Drawing interpretations of the advantages and disadvantages of certain properties when accomplishing tasks and making overall conclusions about the effect of pheromone properties on the effectiveness of communication. To explore this I will be using an evolutionary algorithm to learn a range of optimal pheromone properties fitted to different foraging tasks. These foraging tasks will range in the size and distribution of food sources being gathered. This combination of factors will be referred to as the *habitat* of a colony hereon. Certain habitats will contain stationary non-exhaustive food sources, whilst others will contain sparsely spread small food sources or moving food sources.

I hypothesize that different pheromones will emerge depending on which habitats the pheromones are adapted for, I believe that correctly chosen pheromone properties can singlehandedly increase the foraging capabilities of a colony of ants. To test this hypothesis I will compare the different pheromones in a range of habitats and attempt to show the advantages and disadvantages of each adaptation. The resulting pheromones should perform notably better in habitats similar to the ones trained upon, but notably worse in other environments. Properties such as having high evaporation rates for example may perform better in environments with smaller, less constant food sources.

Johnson, Hubbell, and Feener defined a model which predicts the optimal degree of cooperation in optimal central place foraging as a function based primarily on the distribution of food in the environment [8]. As patches of food become smaller and sparser the number of foragers decreases, increasing the number of scouts. We, therefore, expect pheromone trails to be followed by a higher quantity of ants when the target food source is large and localized.

Literature review

Ant Colony Optimisation:

The study of ants and their utilisation of stigmergy via pheromones has led to the development of “Ant Colony Optimization” (ACO), A population-based metaheuristic used to find approximate solutions to optimization problems. A classic use case of ACO is the traveling salesman problem (TSP), a problem in which one has to find which is the shortest route that passes through each of a set of points once and only once [9]. In ACO for the TSP, a population of ants traverses stochastically based on a heuristic (such as the inverse of arc length) from node to node leaving pheromones at each edge traveled, the shorter the route made by an ant the more pheromones are left behind on its traversed route. This process is then repeated, however, ants also stochastically favour travelling on edges that have more pheromones present. After each generation, the pheromones are evaporated by a constant multiplier. This allows for the forgetting of inefficient and rarely traveled paths [10]. Over time the ants should converge to an optimal or near-optimal path, at which point the shortest path created by an ant can be used as the best solution produced by the ACO algorithm.

ACO has results comparable to other industry-standard metaheuristics and unlike simulated annealing and genetic algorithms, can also continuously adapt to dynamically changing graphs. This is particularly useful for network routing and urban transportation systems [11].

Implementations of ACO algorithms for different optimization problems vary, the traveling salesman problem is a classic NP-hard problem and one of the most researched implementations of ACO algorithms. It is also the easiest to draw similarities to biological and real-world applications like swarm robotics. Below is an implementation of a simple ACO algorithm called “Ant System ant-cycle” on the TSP [12]. It consists of two main procedures, *tour construction,* and *pheromone update*. The procedure is as follows:

```
Procedure *ACO algorithm for TSPs*
    Set parameters, initialise pheromone trails
    While (termination condition not met) do
        Construct Solutions
        Update Trails
        end
    end  *ACO algorithm for TSPs*
```

Tour Construction:
Initially, each ant begins in some randomly chosen city. At each step in construction, ant k applies a probabilistic action choice rule. Specifically, the probability that ant k, within city i, chooses to move to city j at the t^{th} iteration of the algorithm is:

$$p_{ij}^{k}(t) \; = \; \frac{\left[\tau_{ij}(t)\right]^{\alpha} \bullet \left[\eta_{ij}\right]^{\beta}}{\sum_{l \in N_{i}^{k}} \left[\tau_{il}(t)\right]^{\alpha} \bullet \left[\eta_{il}\right]^{\beta}} \; if \;\; j \in N_{i}^{k} \, , \qquad (1.1)$$

Where $\eta_{ij} = 1/d_{ij}$ is a heuristic value, α and β are two parameters which determine the relative influence of the pheromone trail and the heuristic information on the decision process, and N_{i}^{k} is the feasible neighbourhood of ant k, this is the set of cities which and k has not yet visited. If $\alpha = 0$, then selection would be entirely based on the distance of cities within the feasible neighbourhood (this resembles a stochastic greedy algorithm). If $\beta = 0$, pheromones would be the sole factor of the ants' decisions, this quickly leads to a *stagnation* of search where all ants follow the same path and generally produce extremely suboptimal solutions [13]

Pheromone Update:
After every ant has constructed their tours, the pheromone trails are updated. First, the pheromone strength of all arcs are lowered by a constant factor, secondly, each ant adds pheromones to the arcs it has visited.

$$\tau_{ij}(t + 1) = (1 - \rho) \bullet \tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t), \qquad (1.2)$$

Where τ is the pheromone matrix, $0 < \rho \leq 1$ is the rate of evaporation, m is the number of ants and $\Delta\tau_{ij}^{k}(t)$ is the amount of pheromone ant k places on the arcs it has visited, it is defined as the following:

$$\Delta\tau_{ij}^{k}(t) = \begin{cases} 1/L^{k}(t) & if\ arc\ (i, j)\ is\ used\ by\ ant\ k \\ 0 & otherwise, \end{cases} \qquad (1.3)$$

Where $1/L^{k}(t)$ is the length of the kth ant's tour. Equation 1.3 means that ants with shorter paths place more pheromones, this makes all its traversed arcs more desirable to be traversed in future generations.

As expected many similarities can be drawn between ACO and foraging simulations, such as pheromone properties. Of the three parameters being investigated in this report we can see the rate of evaporation appears in equation 1.2 and the strength of a pheromone can be perceived as the ratio of α and β in the arc selection procedure (equation 1.1)

The rate of diffusion has also made appearances in ACO literature, a novel version of ACO with the application of pheromone diffusion showed promising results with the "overall solution and convergence speed of the algorithm in this thesis have been greatly improved." when comparing the novel approach to the classical ACO algorithm described above [14]. However, the implementation of diffusion in this approach is complex and not easily convertible to a coordinate system.

Choosing appropriate values for parameters is key to a successful and quickly converging solution. The importance of such features in

ACO algorithms leads me to believe that there is value in investigating if the same applies to biological simulations.

Ant foraging simulations:

Literature developing different pheromone foraging techniques such as AntFarm [15] and Liviu A. Panait and Sean Luke [16] have established a grid framework consisting of a nest and food, used to benchmark the performance of colonies at collectively collecting food. Ants are initialised in the nest and can move to any tile within a 3x3 grid centred on themselves once per time step, if they travel to a food tile they may pick up 1 unit of food and must then deliver it back to the nest. The performance of a colony is judged on how many units of food are gathered and returned to the nest within a given time period.

In AntFarm, ants may place “to food” pheromones to help direct other ants to a food source. To find their way back to the nest they use a compass to navigate, contrastingly, in Liviua et al. ants also have the ability to drop “to home” pheromones to help guide them back to the nest.

Both AntFarm and Liviu et al. are designed to develop strategies for pheromone-based foraging using neural networks and strongly typed GAs respectively, in combination with a genetic algorithm.

However, both papers do not allow learning of different pheromone properties, AntFarm does not even implement any evaporation or diffusion whilst Liviu uses constant rates of 0.1% for diffusion and evaporation. Below are some useful comparisons of the implementation of the generic algorithms and simulations used by these methods

Paper	AI type being trained	Grid size	Ants per colony	Population (number of colonies)	Multiple food locations	Amount of food available
AntFarm	Neural Network	16x16	128	16,384	Yes	196 Units
Liviu et al.	Strongly typed GA	10x10 33x33 100x100	50 50 500	100	No	Infinite

Table 1: *The parameters used in the implementations of the genetic algorithms and ant colony simulations from AntFarm and Liviu et al.*

Swarm robotics:

In swarm robotics, pheromone communication is an area of interest as it provides agents with “extremely simple sensory-motor and memory capabilities, to achieve fully decentralised control” [17]. This has been implemented in a range of mediums such as chemicals, RFID, and virtual environments.

COS-Ω is a light-based artificial pheromone system that allows for a high level of flexibility and fidelity [18]. This work investigated the effects of factors such as evaporation, on the effectiveness of a swarm of *Colias* micro-robots attempting to follow a pheromone trail left behind by a leading agent. This paper found that increasing the pheromones’ half-life was beneficial to the agent's ability to follow a pheromone depositing leader.

A new model of COS-Ω expanded on the methods by investigating the effects of diffusion and advection (advection is the transfer of substances or any quantity by bulk motion of a fluid, like wind) on a swarm of micro-robots attempting to reach a certain area containing a pheromone source. It found that a higher rate of diffusion increased the swarm's ability to get close to the pheromone source [19].

COS-Ω’s implementation consists of an LCD screen that displays light to represent the intensity of pheromones at a given pixel, the brighter the light, the stronger the pheromones. The *Colias* micro-robots contain two light intensity sensors used to detect light emitted from the LCD screen on which they are deployed.

With computational simplicity in mind, diffusion was implemented using a Gaussian filter. Gaussian filters contain analogous features to how real-world diffusion occurs, namely:

- The total amount of pheromones remains the same
- Pheromones are spread to neighbouring positions

The pheromone diffusion update rule for COS-Ω is the following:

$$\Phi^{k+1}(x, y) = (\omega \star \Phi^{k})(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} \omega(s, t) \bullet \Phi^{k}(x - s, y - t), \quad (2.1)$$

Where ω is a kernel matrix of size $(2a + 1) \times (2b + 1)$ which is convolved with the matrix of the pheromone strength at the k^{th} iteration and w is defined by the equation below:

$$\omega(x, y) = \frac{1}{2\pi\sigma^{2}} e^{-\frac{x^{2}+y^{2}}{2\sigma^{2}}}, \ \omega \in \mathbb{R}^{(2a+1)\times(2b+1)}, \qquad (2.2)$$

where the element at ((a + 1), (b + 1)) is assigned as (0,0) of ω and σ is the standard deviation of elements of ω. The elements of the kernel are determined based on the Gaussian distribution. The further an element from the center of the matrix is, the smaller value this element has.

Method

Evolving pheromones:

To learn parameters a genetic algorithm is implemented. Genetic algorithms are used to generate approximate solutions to complex optimization problems [20], often case, this solution is a string of parameters (called chromosomes) that approximately optimises a particular function.

A genetic algorithm evolves a population of strings of parameres by allocating a fitness score to each chromosome based on the quality of a solution. The parameters being learned in this genetic algorithm are the rate of diffusion, volatility, and strength of a colony's pheromones, resulting in a genome composed of 3 real numbers. The rate of evaporation is bounded within the range [0,1], the rate of diffusion can be any non-negative real number, and the strength can be any real number. A negative strength represents a pheromone that is repellent, causing ants to move away from high concentrations sensed.

Genetic algorithms can be broken down into four fundamental parts:

1) Allocating fitness scores
2) Selection
3) Mating/Crossover
4) Mutation

In this paper, the function attempting to be optimised is a simulation of a colony's foraging for food collectively, the more food gathered the better. Since the simulation is subject to randomness, each chromosome will be tested multiple times and averaged to produce a fitness that more accurately represents its fitness.

Selection is an integral part of genetic algorithms since it significantly affects convergence and is largely responsible for maintaining genetic diversity in the population [21]. Tournament selection is an efficient selection method of time complexity $O(N)$.

Tournament selection consists of drawing a random group of size G from the population and making them compete in a tournament in which the agent with the highest fitness score is selected.

The advantage of this method, compared to others such as roulette wheel selection, is that selection is not directly proportional to fitness score. This means an extremely high-scoring outlier in an early generation cannot immediately swamp the next generation with its children, massively reducing the diversity of the population [22].

Since chromosomes with higher fitness scores are more likely to be passed on to the next generation, each generation should roughly improve on the latter and after many generations converge to near-optimal solutions.

Mating consists of combining the genetic information of two parents into a single offspring. To do so, only parts of each parent's genetic information can be passed to an offspring. Classic approaches of recombination are one-point and two-point crossover, these consist of lining up the parent chromosomes, and picking one or two points respectively to crossover the chromosomes. Resulting in two unique chromosomes made up of genetic information from both parents. Uniform crossover can be seen as L-point cross over where L is the length of chromosomes, this results in offspring where each gene randomly belongs to either parent. Uniform crossover has been shown to be superior to one-point and two-point crossover in most cases [23]

In this implementation, as there are only 3 loci to be passed on to the offspring, two-way crossover, and uniform crossover can be interpreted as the same operation. Therefore either method can be used to yield the same results. Uniform crossover will be used as it clearly signifies that each gene randomly belongs to either parent. From the two resulting chromosomes, one is randomly discarded, and another is made subject to possible mutation before entering the next generation.

During mutation, each allele has a random chance p, of being mutated. If an allele is selected for mutation, a random number from a Gaussian distribution centred at 0, with a standard deviation of σ, will be added to the allele. If the resulting allele is not within its valid range of values it will be cast back into the closest value that is in the valid range.

Ant simulation:

In order to assess the fitness of differing pheromones, a microanalytic simulation similar to AntFarm and Liviu et al. will be used. The simulation consists of ants that traverse an environment consisting of a single nest tile, empty tiles, and food tiles. At the beginning of a simulation, all the ants are in the nest and facing in a random direction.

The quality of a pheromone is defined as the average amount of food a colony can gather and return to the nest, using the pheromone as its sole tool for communication within an amount of time T.

An ant can be in one of two states, either searching for food or returning food to the nest. These states are broken down below:

“To Food” -Searching for food:

> Ants begin each simulation in this state. Each unit of time each ant senses the pheromone levels of tiles directly and diagonally in front of them (Figure 1). A stochastic procedure based on the pheromone levels present is used to determine which tile the ant moves to.

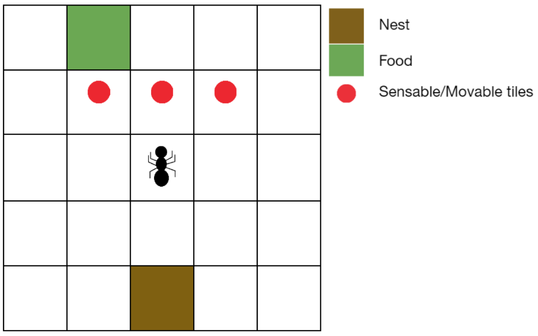


Figure 1: *An illustration of a scaled-down environment with an ant's sensable/movable tiles depicted with red circles. Pheromones are not shown but may be present on all tiles. The actual environment is 32 x 32.*

The decision procedure for movement is based on an altered version of the decision procedure used in ACO (equation 1.1). The heuristic is a hand-coded logic consisting of favouring walking straight instead of turning. The pheromone levels are normalised to sum to one, and then multiplied by the pheromone strength γ, this allows us to determining relative influence of the pheromones to the heuristic, replacing the exponents used in ACO equation 1.1. The heuristic is then added to the strength scalled pheromone values and the equation is normalised once again to produce probabilities which sum to one. The probability of an ant k, at time step t, currently positioned at (r, t) moving to (x, y) is calculated by the following:

$$p^{k}_{(r,t)(x,y)}(t) = \frac{\gamma \bullet \frac{\tau_{(x,y)}(t)}{\sum_{(x,y) \in N^{k}} \tau_{(x,y)}(t)} + \eta^{k}_{(x,y)}(t)}{\gamma + 1} \qquad (3.1)$$

Where $\tau_{(x,y)}(t)$ is the level of pheromones present on tile (x, y) , N^{k} is the group of sensible/movable tiles of ant k, γ is the pheromone strength, and $\eta^{k}_{(x,y)}(t)$ is a heuristic used to defined a simple produce where walking forwards its favoured to turning defined by:

$$\eta^{k}_{(x,y)}(t) = \begin{cases} 0.7 & if\ (x,y)\ is\ directly\ in\ front\ of\ ant\ K \\ 0.15 & otherwise, \end{cases}$$

(3.2)

experimentally values of 0.7 for continuing straight and 0.15 for turinng in each direction performed well.

If all of the sensible tiles are out of bounds, the ant is flipped 180 degrees and must wait until the next time step to move again.

If an ant is able to sense food in any of its three sensible tiles it will move to the tile, pick up a unit of food, flip 180°, and change states to the “To Nest” state.

“To Nest” -Returning food to the nest:

When returning to the nest, AntFarm’s solution of a compass directing ants straight back to the nest leads to the shortest path from the food to the nest always being immediately discovered. This nullifies a colony's ability to learn the shortest path over time [1] which is a major factor in the interest of research on ants.

Adding a second pheromone as Liviu et al. depicted would require an overly complex hand-coded algorithm. Therefore, an implementation where each ant simply backtracks its steps to the nest based on an internal memory has been implemented. This is not dissimilar to how some real ants navigate as they use elaborate techniques including memorising landmarks to orient themselves [24].

Once all the ants have finished their actions for a time step, each ant in the “To Nest” phase deposits pheromones on the tile they are positioned on, this is defined in the pheromone update stage described below. An ant completing its journey to the nest thereby creates a trail of pheromones which can

> be used by other ants to steer themselves from the nest to the food source. The amount of pheromones deposited per ant is a constant ε of arbitrary value.
>
> Once an ant has returned to the nest, it will deposit the food it is carrying, flip 180°, and swap states to the “To Food” once again.

Foragers leave the nest together in a direction influenced by the patrollers, a small group of ants that scout the foraging area early in the morning, and a forager's memory of where it last collected food.

Experimentally it was found that releasing ants from the nest in two waves can increase the performance and predictability of a colony. This is because the second wave of ants has the opportunity to immediately follow a pheromone trail to a food source laid by an ant from wave 1. However, if all ants are released at the same time, by the time a pheromone trail is laid, a majority of the ants will be too dispersed to find and follow it. It is then up to random chance for the lost ants to find the paths laid out by the rest of the colony, this increased the variance of results when assessing the fitness of pheromones and therefore impeded the performance of the genetic algorithm. Some species of real ants use similar systems, where a small group of patrol ants finds safe directions containing food sources before releasing thousands of forager ants in those directions [25]

For a map size of 32 x 32, it was found experimentally that initially releasing 25% of the population of the colony until the 20th time steps when the remainder of the colony is released leads to improved consistency of results.

Pheromone update:

The pheromones placed by ants in an environment are stored within a matrix of equal dimension to the board. Once per time step, after each ant has completed its actions, a pheromone update takes place. The pheromone update consists of adding all the newly deposited pheromones as well as simulating the effects of evaporation and diffusion. Evaporation and diffusion can be simulated by a single kernel being convolved over the pheromone matrix. The kernel ω is calculated by combining the Gaussian filter equation for diffusion (2.2) with a multiplier for the rate of evaporation as seen in ACO equation (1.2) to produce:

$$\omega(x, y) = (1 - \rho) \bullet \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}, \ \omega \in \mathbb{R}^{(2a+1)\times(2b+1)}, \quad (3.3)$$

where the element at ((a + 1), (b + 1)) is assigned as (0,0) of ω, σ is the standard deviation of the gaussian matrix and also interpreted as the rate of diffusion and ρ is the rate of evaporation. α and β are used to define the size of the diffusion kernel.

The new pheromone matrix Φ can be calculated by convolving ω over the previous pheromone matrix, as well as adding the pheromone deposits of each ant in the "To Nest" phase.

$$\Phi^{k+1}(x, y) = (\sum_{k=1}^{m} \Delta\tau_{xy}^{k}(t) + \omega \star \Phi^{k})(x, y) =$$

$$\sum_{k=1}^{m} \Delta\tau_{xy}^{k}(t) + \sum_{s=-a}^{a} \sum_{t=-b}^{b} \omega(s, t) \bullet \Phi^{k}(x - s, y - t), \quad (3.4)$$

Where ω is a kernel matrix of size $(2a + 1) \times (2b + 1)$ which is convolved with the pheromone matrix Φ at the k^{th} time step, and $\Delta\tau_{xy}^{k}(t)$ is defined by:

$$\Delta\tau_{xy}^{k}(t) = \begin{cases} \varepsilon & if\ ant\ k\ is\ at\ (x, y),\ and\ in\ state\ "To\ Nest" \\ 0 & otherwise, \end{cases} \quad (3.5)$$

Environment Selection and Map Design:

An ecological study found that “The principal determinants of ant foraging strategy are the distribution of food resources in size, time, space, and quality”[26]. Since pheromones are at the heart of ant foraging, varying the factors described above allows for a range of different pheromones to emerge, each suited to their environmental niche.

Out of the mentioned determinants, the size, time, and space of food sources will change in the environments our pheromones adapt to. The pheromones most suited to a particular environment should emerge as the result of the genetic algorithm. In this paper, three different environments with fundamentally different food sources will be investigated. The environments are described below with biological examples of similar food sources, alongside their designed map represented in the 32x32 grid system:

- **Unlimited stationary food sources**, akin to leaf-cutter ants collecting leaves from a large tree to feed a fungus that they cultivate in their nets [20], this is an extremely large source of food and is unlikely to quickly deplete.

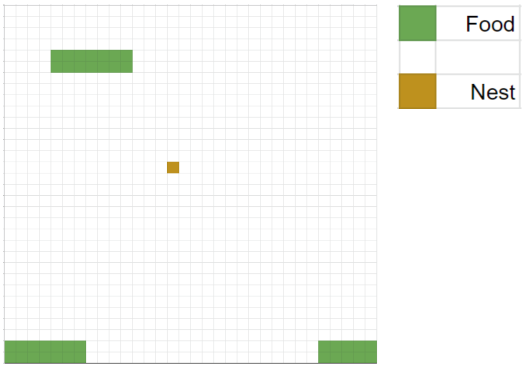


Figure 2: *The unlimited stationary food sources map. Food tiles are not depleted when an ant collects a unit of food from them. The time to gather T is set to 100.*

- **Limited stationary food sources**, akin to the foraging of seeds, other insects' eggs, fruits, and many more small food sources [21]

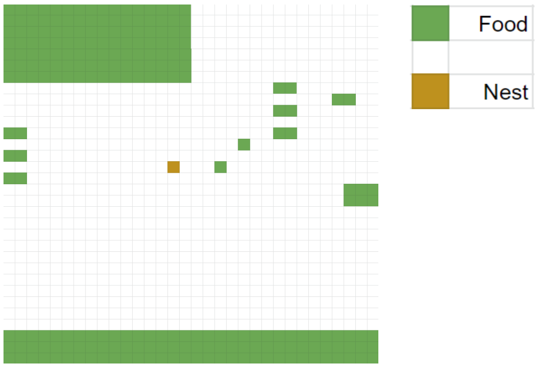


Figure 3: *The limited stationary food sources map. Food tiles contain only a single unit of food and are removed when gathered. The time to gather T is set to 100*

- **Unlimited moving food sources**, akin to scavenging dead large animals such as birds or reptiles [2] which may be quickly and unexpectedly consumed by a larger scavenging species

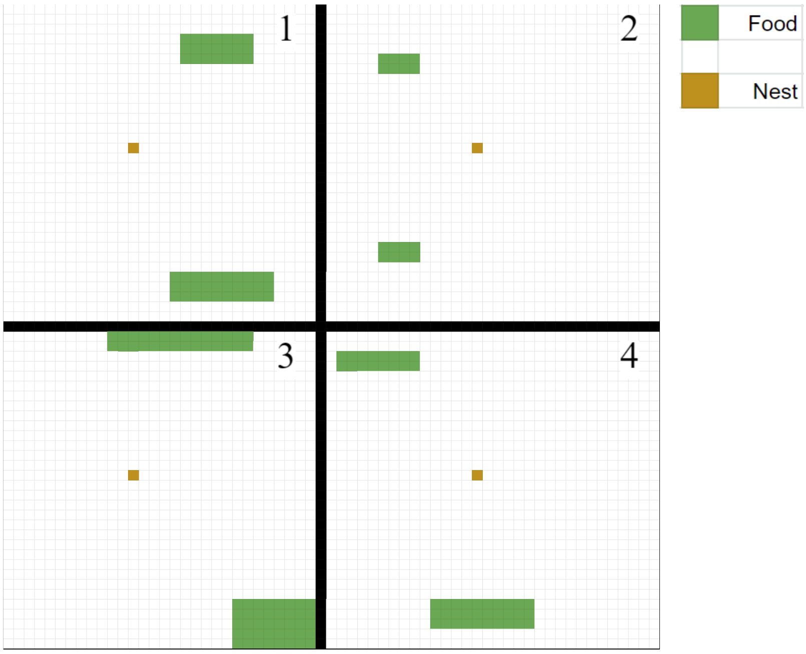


Figure 4: *The unlimited moving food sources maps, this environment consists of 4 maps, labeled 1 to 4, which take turns being the active map in numerical order. Each map is active only*

> *once, and all four maps are active for an equal amount of time. Food tiles are not depleted when an ant collects a unit of food from them. The time to gather T is set to 400 to account for extra exploration time.*

Computational Issues and parallelisation:

Whilst each ant's decision process is very computationally low, genetic algorithms are population-based, and individuals are assessed multiple times to retrieve more accurate fitnesses, this leads to a very large number of computations per generation. The breakdown of the total number of simulations and ants is the following:

A population size of 50 agents, each of which is assessed 5 times to obtain an accurate fitness, requires 250 simulations per generation (Each consisting of 128 ants, for a total of 32,000 ants per generation). Since we are learning 3 separate pheromones, each of which uses 75 generations within the genetic algorithm until the algorithm is halted, we require a total of 225 generations. This results in a total of 56,250 simulations consisting of 7.2 million total ants.

This computational load is so large that implementing parallelization is necessary to generate results on a normal work machine within hours instead of days.

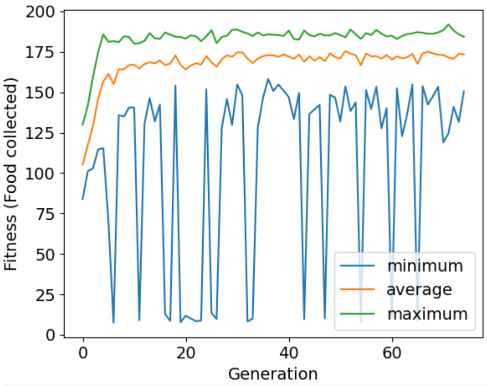
When assessing the fitnesses of a given generation, a collection of subprocesses are created, allowing multiple threads within a single multi-core CPU to work together to complete large tasks much faster. Each subprocess is allocated the task of calculating the fitness of an individual. Once the subprocess has computed the fitness, it returns this result to the main process and fetches another unassessed individual if there are any remaining.

As the evaluation of an individual is an independent process, asynchronous non-blocking parallelization is implemented. This allows each subprocess to work at any speed without slowing down other processes since no communication is required. The only blocking implemented is the main process, which waits for all the individuals to be assessed before proceeding to the selection phase of the genetic algorithm.

Results, Analysis, and Evaluation

The pheromone's small genotype of three real numbers results in a fitness landscape that cannot be overly complex, therefore the genetic algorithm is able to learn near-optimal values relatively quickly. To learn the optimal pheromone properties of each environment, 75 generations of the genetic algorithm with a population size of 50 provided results that have reached near-optimal performance. Parameters for selection and mutation are group size $G = 5$, mutation probability $p = 0.3$, and mutation standard deviation $\sigma = 0.2$.

Below we observe the fitness graphs of the training of each pheromone.



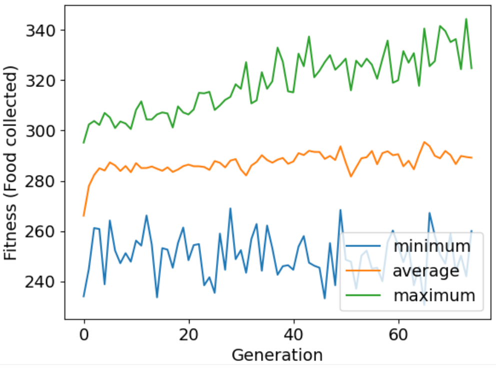


Fig 5: *Limited stationary food source Pheromone fitness graph.*

Fig 6: *Unlimited moving food source pheromone fitness graph.*

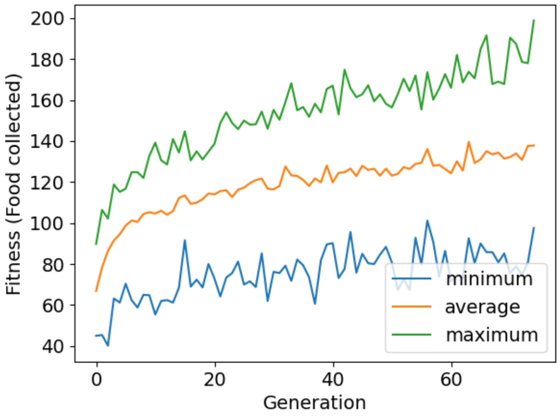


Fig 7: *Unlimited stationary food source pheromone fitness graph.*

Figures 5 and 6 share many features, both populations quickly learn within the first 10 generations but then stagnate for the remainder of training, this further suggests that the fitness landscape may not be overly complex leading to quick convergence. Learning properties within the unlimited stationary food source environment had the same initial fast improvement but, continued to gradually improve afterward (Figure 11). Table 1 below, shows the best-learned properties in training, the strength of this pheromone is a very high value of approximately 5.6, a possible reason for the constant small improvements in fitness is that increasing the value for strength was extremely beneficial but mutations were too infrequent and small to quickly reach an optimal value. Increasing mutation frequency and mutation standard deviation or changing the implementation of strength to have an exponential effect on tile decision instead of a scalar one could increase the learning rate.

Environment food type	Diffusion standard deviation (3 d.p)	Rate of evaporation (3 d.p)	Strength (3 d.p)
Unlimited stationary	0.283	0	5.6323
Limited stationary	2.407	0.458	-0.571
Unlimited moving	0.127	0.123	3.865

Table 2: *The properties of the best individuals for each habitat after 75 generations of the genetic algorithm.*

The pheromone from unlimited stationary food sources does not evaporate, this is most likely due to only gathering from a single location, and therefore the ability to forget old paths is not required. The strength of this pheromone is also very high, this is most likely as exploration does not benefit colonies as once they have found a food source it is abundant and permanent. Diffusion is marginally present, which could slightly help increase the chances of shortening the path to the food source.

The pheromone for limited stationary food sources has learned very interesting properties, most not-worthy is the learning of a negative pheromone strength, this means that pheromones are repellent and lower the chances of an ant moving to a tile with a high concentration of pheromones. The pheromones are used to signify where to not explore, since food in this environment is limited, when an ant finds food it's likely searching this same area is not beneficial as the resource could be depleted. However, the rate of evaporation is extremely high, removing 50% of pheromones at every time step, this is combined with a very high rate of diffusion making what can be pictured as "bursts" of repellent pheromones, which serve to spread ants apart increasing total exploration.

The pheromone developed inside of the unlimited moving environment has to evaporate approximately 12% per time step, therefore, for a trail to evaporate 90% of its concentration it requires 19-time steps. This allows for forgetting paths that lead to food sources that are no longer available. This pheromone's strength is lower than the strength of the pheromones developed for a stationary food source, this is likely to increase the number of ants which are exploring instead of gathering, increasing the speed of finding a new food source when the current source disappears. The diffusion standard deviation of 0.28 (2 d.p) results in under 1% of the pheromones spreading to neighboring tiles per time step. A possible reason for this pheromone having such slow diffusion is to stop the spacial influence of the trails so that ants that have "broken free" from the pheromone trail are not immediately lured back in from the diffused tiles.

To compare the pheromones, we test how they perform in all the environments discussed in this paper. To ensure the adaptations learned are generalised, and not specific to the maps used in training, a new set of maps are implemented which represent the same food sources. These can be used to compare the pheromones' impact on a colony's gathering ability. The new maps are depicted below:

- Unlimited stationary food sources

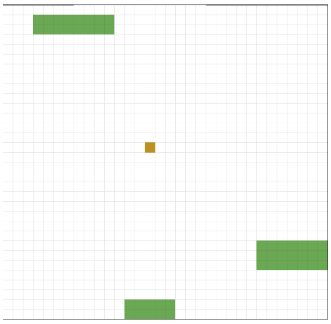


Figure 8: *Test map for unlimited stationary food sources, food tiles are not depleted when an ant collects a unit of food from them. The time to gather T is set to 80 to further differ from the training case.*

- Limited stationary food sources

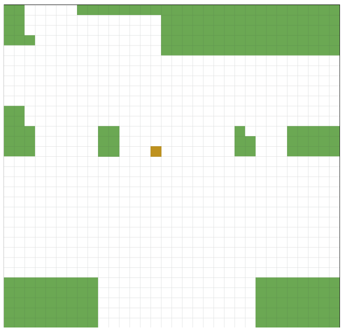


Figure 9: *Test map for limited stationary food sources. Food tiles contain two units of food. The time to gather T is set to 120 to further differ from the training case.*

- Unlimited moving food sources

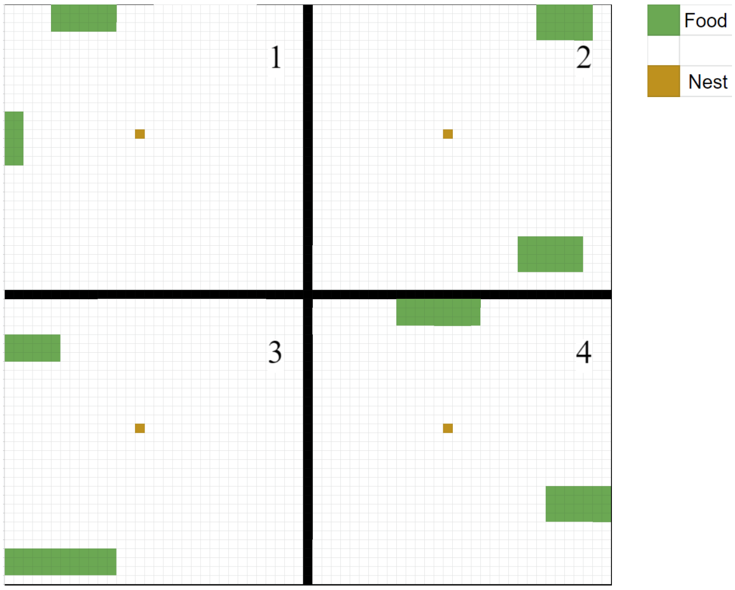


Figure 10: *Test maps for the unlimited moving food sources, this environment consists of 4 maps, labeled 1 to 4, which take turns being the active map in numerical order. Each map is active only once, and all four*

maps are active for an equal amount of time. Food tiles are not depleted when an ant collects a unit of food from them. The time to gather T is set to 440 to further differ from the training case.

Each pheromone is applied to every test case 50 times to account for some randomness of the ant's movements, below is a table showing the result averages and standard deviations in each environment:

	Test Habitat		
Pheromone habitat	Unlimited stationary	Limited stationary	Unlimited moving
Unlimited stationary	**49.82** (13.51)	89.88 (14.14)	305.08 (28.56)
Limited stationary	20.3 (8.09)	**192.84** (9.50)	264.48 (19.76)
Unlimited moving	44.72 (9.19)	96.86, (12.78)	**335.06** (36.51)

Table 3: *The of performance tests of evolved pheromone properties across three habitats. Numbers (Mean performance followed by the standard deviation in parenthesis) summarise food units collected and returned to the nest. (Numbers in bold performed significantly better down their column)*

Every pheromone statistically performed significantly better in the environment they were adapted for (at a 0.05 significance level). This indicates that a pheromone's properties can heavily influence the effectiveness of a pheromone at a given task. However, it is not known to be true that a high level of generalisation was learned by the pheromones since only a single set of maps was utilized for testing. The utilisation of a larger set of maps to both test and train the pheromones would increase the confidence of the findings in this paper. Automatic map generation could allow for the creation of a large test and training map set, ideal for ensuring the generalisation of the learned parameters.

AntFarm's goal was to determine when pheromone communication would evolve in a simulated environment. Answering this question might be aided by adding learnable pheromone properties to their simulation. This paper suggests, that even if evolving pheromone communication is an optimal evolutionary step, using suboptimal pheromones as implemented in AntFarm could reduce the effectiveness of communication such that the adoption of pheromone communication is evolutionarily unfavorable. The incorporation of learnable pheromone properties could therefore potentially improve understanding of the emergence of pheromone communication in social insects.

Combing AntFarm with learnable pheromone properties could also lead to the development of unique unresearched implementations of pheromone communications. The bursting repellent pheromones learned by the pheromone adapted to the limited stationary food sources is an example of an unexpected use of pheromone communication, combining complex depositing and interpretation strategies with modular pheromones may result in intricate and insightful communication strategies being discovered which could help us further understand biological ants or implement new techniques into fields such as swarm robotics.

Conclusion:

In conclusion, our understanding of pheromone strategies implemented by real ants is far from perfect, when researching foraging strategies the properties of pheromones can not be ignored as they can have a very large impact on performance and overall strategy of foraging. Our results show that specific tasks favor specific pheromones, and that there isn't any perfect pheromone that thrives in all environments.

The model suggested by Johnson, Hubbell, and Feener of the optimal degree of cooperation being a function of food distribution holds true with the results of this paper. The model claims the largest, long-lasting food source should result in the highest number of foragers and the lowest number of scouts. In my implementation, a low evaporation rate, and a high pheromone strength is most likely to lead to high quantities of ants engaging in foraging rather than exploring/scouting. In our results, the environment with unlimited stationary food sources was the only pheromone to adapt to not evaporate at all, and also had by far the highest pheromone strength.

Bibliography

[1] Koch, F. (1989), Sudd, J. H. und Franks, N. R.: The behavioural ecology of ants. Blackie & Son Ldt: Glasgow https://doi.org/10.1002/mmnd.19890360438

[2] Schmid-Hempel, P. (1992), B. Hölldobler, E. O. Wilson (1990): "The Ants" Springer, Berlin, Journal of Evolutionary Biology https://doi.org/10.1046/j.1420-9101.1992.5010169.x

[3] G. Theraulaz and E. Bonabeau, "A Brief History of Stigmergy," in Artificial Life, vol. 5, no. 2, pp. 97-116, April 1999 https://doi.org/10.1109/SASO.2010.35

[4] Jackson, Duncan E., and Francis LW Ratnieks. "Communication in ants." *Current biology* 16, no. 15 (2006)

[5] Goss, Simon, Serge Aron, Jean-Louis Deneubourg, and Jacques Marie Pasteels. "Self-organized shortcuts in the Argentine ant." *Naturwissenschaften* 76, no. 12, 579-581, 1989

[6] Morgan, E.D., Do Nascimento, R.R., Keegans, S.J. *et al.* Comparative Study of Mandibular Gland Secretions of Workers of Ponerine Ants. *J Chem Ecol* 25, 1395–1409 (1999). https://doi.org/10.1023/A:1020987028163

[7] Bonabeau, Eric & Theraulaz, Guy & Deneubourg, Jean-Louis & Aron, Serge & Camazine, Scott. (1997). Self-organization in social insects. Trends in Ecology & Evolution https://www.researchgate.net/publication/301232920_Self-organization_in_social_insects

[8] L. K. JOHNSON, S. P. HUBBELL, D. H. FEENER, JR., Defense of Food Supply by Eusocial Colonies, *American Zoologist*, Volume 27, Issue 2, pp 347–358, May 1987
https://doi.org/10.1093/icb/27.2.347

[9] D L. Applegate, R E. Bixby, V Chvatal, and W J. Cook. . The Traveling Salesman Problem: A Computational Study (Princeton Series in Applied Mathematics). Princeton University Press, 2007

[10] Nwamae, Believe B. , Kabari, Ledisi G., 0, Solving Travelling Salesman Problem(TSP) Using Ant Colony Optimization(ACO), INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 07, Issue 07, July 2018
http://dx.doi.org/10.17577/IJERTV7IS070129

[11] Jaiswal, Utkarsh and Shweta Aggarwal. “Ant Colony Optimization.” 2011
https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.301.1091&rep=rep1&type=pdf

[12] Dorigo, Marco & Maniezzo, Vittorio & Colorni, Alberto. Positive Feedback as a Search Strategy. Tech rep., 91-016, Dip Elettronica, Politecnico di Milano, Italy, 1999
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.52.6342&rep=rep1&type=pdf

[13] Hassan M. H. Mustafa, Ayoub Al-Hamadi, Mohamed Abdulrahman, Shahinaz Mahmoud, Mohammed O Sarhan. On Comparative Analogy between Ant Colony Systems and Neural Networks Considering Behavioral Learning Performance. *Journal of Computer Sciences and Applications*. Vol. 3, No. 3, 2015
http://pubs.sciepub.com/jcsa/3/3/4

[14] Li, K., Li, X., Ma, S., & Irwin, G. . *Life System Modeling and Intelligent Computing - Communications in Computer and Information Science, Vol. 97*. Springer 2010 https://link.springer.com/content/pdf/10.1007/978-3-642-15597-0.pdf

[15]Collins, Robert J., and David Jefferson. *Antfarm: Towards simulated evolution*. Los Angeles, CA: Computer Science Department, University of California, 1990

[16]Panait, Liviu, and Sean Luke. "Evolving foraging behaviors." In *Proc. of the Ninth Int. Conf. on the Simulation and Synthesis of Living Systems, ALIFE-9*. 2003

[17] Na, Seongin, Yiping Qiu, Ali E Turgut, Jiří Ulrich, Tomáš Krajník, Shigang Yue, Barry Lennox, and Farshad Arvin. “Bio-Inspired Artificial Pheromone System for Swarm Robotics Applications.” *Adaptive Behavior* 29, no. 4, August 2021 https://doi.org/10.1177/1059712320918936

[18] Arvin, Farshad & Krajník, Tomáš & Turgut, Ali & Yue, Shigang. COSΦ: Artificial pheromone system for robotic swarms, 2015 research. DOI:10.1109/IROS.2015.7353405

[19] Seongin Na, Mohsen Raoufi, Ali Emre Turgut, Tomáš Krajník, Farshad Arvin; . "Extended Artificial Pheromone System for Swarm Robotic Applications." Proceedings of the *ALIFE 2019: The 2019 Conference on Artificial Life*. *ALIFE 2019: The 2019 Conference on Artificial Life*. Online. ASME, July 29–August 2 2019 https://doi.org/10.1162/isal_a_00228

[20] Poli, Riccardo & Langdon, William & Mcphee, Nicholas. A Field Guide to Genetic Programming. 2008 https://dl.acm.org/doi/10.5555/1796422

[14] Goldberg, D.E., Holland, J.H. Genetic Algorithms and Machine Learning. *Machine Learning* 3, 1988 https://doi.org/10.1023/A:1022602019183

[15] [13] Poli, Riccardo & Langdon, William & Mcphee, Nicholas. A Field Guide to Genetic Programming, 2008 https://dl.acm.org/doi/10.5555/1796422

[16] Gilbert Syswerda.Uniform Crossover in Genetic Algorithms. In Proceedings of the 3rd International Conference on Genetic Algorithms. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1989 https://www.researchgate.net/profile/Gilbert-Syswerda-3/publication/201976488_Uniform_Crossover_in_Genetic_Algorithms/links/5f84cf27299bf1b53e22ee7c/Uniform-Crossover-in-Genetic-Algorithms.pdf

[17] Schmid-Hempel, P, B. Hölldobler, E. O. Wilson: "The Ants" Springer, Berlin, 732 pp. DM 198.—. Journal of Evolutionary Biology, 5, 1990 https://doi.org/10.1046/j.1420-9101.1992.5010169.x

[18] Greene, Michael & Gordon, Deborah. How Patrollers Set Foraging Direction in Harvester Ants. The American naturalist. 170. 2008 10 DOI: 10.1086/522843

[19] Traniello, James. Foraging Strategies Of Ants. Annual Review of Entomology. 34. 191-210. 2003 DOI:10.1146/annurev.en.34.010189.001203

[20] F. O. Aylward, C. R. Currie, and G. Suen, “The Evolutionary Innovation of Nutritional Symbioses in Leaf-Cutter Ants,” *Insects*, vol. 3, no. 1, pp. 41–61, Jan. 2012, doi: 10.3390/insects3010041

[21] Bullnheimer, B., Kotsis, G., Strauß, C. Parallelization Strategies for the Ant System. In: De Leone, R., Murli, A., Pardalos, P.M., Toraldo, G. (eds) High Performance Algorithms and Software in Nonlinear Optimization. Applied Optimization, vol 24. Springer, Boston, MA. 1998 https://doi.org/10.1007/978-1-4613-3279-4_6