Neuroevolution of Foraging Behavior: A Pygame Simulation of Bumblebee Traplining

Abstract

This report investigates the foraging behavior of bumblebees, with a greater emphasis on their ability to develop stable foraging routes also known as traplines. Such traplines help Bumblebees to collect nectar more efficiently. The primary objective of the study was to simulate and analyze how bumblebees optimize their foraging strategies using neural networks evolved through NEAT(NeuroEvolution of Augmenting Topologies). The methodology involved creating a simulation environment using Python and Pygame, where Bees forage in a virtual environment and optimize their foraging routes. Various experiments were done using different arrangements of flowers to train and test the bees to develop foraging routes The results demonstrated that the evolved neural networks have the potential to mimic the trapline optimization behavior demonstrated by bees.

Introduction

Bumblebees are known to exhibit remarkable foraging behavior along with the ability to develop stable foraging routes, also known as traplines. Traplines help minimize the travel distances between flowers and allow for efficient collection of nectar. This behavior is similar to the famous optimization problem also known as the Travelling salesperson problem (TSP). Research has shown that bumblebees aren't exactly solving complex equations in their heads. Lihoreau et al. (2013) demonstrated that bumblebees use simple heuristics to develop these routes, adjusting their paths based on the spatial arrangement of flowers and their experiences during foraging bouts. It's something similar to how we might optimize our grocery shopping route over time, learning which aisles to hit first for the most efficient trip. In this study, we aim to simulate the same phenomenon in a pygame simulation in which bees learn using neural networks evolved in real-time using the NEAT (NeuroEvolution of Augmenting Topologies) library in Python (Kenneth O. Stanley and Risto Miikkulainen, 2002). Bumblebees can, therefore, transfer learned patterns between different sensory modalities, without needing to relearn them.

There are different species in existence that also demonstrate trapline behavior for hunting and foraging like hummingbirds and certain species of bats. Understanding this behavior in one species might shed light on other similar species and what evolutionary advantages it affords them.

Simulating animal behavior using computational models like simulation allows researchers a cost-effective way to study, explore, and predict intelligent and complex behaviors that are difficult to study in the wild and simulation is a resource-friendly method to test hypotheses under controlled conditions, providing valuable insights into the crucial factors that influence

decision-making and intelligent behavior in animals. The bees in the simulation can be considered artificial and are essentially agents with multiple sensory parts commonly found in insects like simulated visual fields and pheromone detectors. These bees capture data using the aforementioned sensory parts and learn to detect and react to them in real-time. The experiments in the study are virtual recreations of experiments done using real bees in the lab (Ohashi et al, 2007). These experiments were instrumental in the development of the simulation and the study as a whole. Results from the study will also help add to our understanding and development of optimization problems in general. The study's approach of using NEAT to simulate the foraging behavior has broader implications not only limited to intelligence in animals but also in the larger class of evolutionary algorithms and artificial intelligence in general. By mimicking the way natural system evolves and learn, we can develop more efficient algorithms for complex engineering problems and other domains like robotics.

Methods

The primary objective of this experiment was to simulate the trapline optimization behavior of Bumblebees using neural networks. Pygame was used for real-time visualization and rendering of the simulation along with tkinter library for user interface to interact with the simulation and perform experiments. NEAT(NeuroEvolution of Augmenting Topologies) algorithm works as a core learning mechanism used for the bees, it was designed to evolve neural networks but unlike other neuroevolution methods it doesn't restrict itself with a fixed network topology, NEAT initializes with simple networks but gradually increases the networks complexity by adding new nodes and connections through mutation resulting in not only more optimized weights but also more optimized network topology. We used the NEAT-Python library to evolve neural networks for bees.

The bumblebees, which are essentially the agents in the simulation learn to optimize their foraging route (traplines), through a process that involves sensing their environment and making decisions on the basis of those sensory inputs. The neural networks are updated according to the current state experienced or perceived by the bees. Every bee in the simulation is controlled by an individual neural network which is initialized with a simple topology consisting of inputs like weather conditions, the nearest flower from the hive(visual landmark), distance to the nearest flower, bees' energy levels, and pheromone levels at the current location, and output neurons which dictate the bee's movement by controlling its direction and speed, there are no hidden layers in the network at the time. The simulation begins with a user-defined population of bees and corresponding neural networks, overtime as bees collect information using their different aforementioned sensory modalities the neural network processes them and outputs decisions on movement while updating their weights. After this, a fitness function analyses the nodes and connections with better fitness scores from bees who are able to efficiently navigate flowers and use them as parents (selection) for the next generation by combining genes(nodes and connections) from them to create better offspring. NEAT then introduces mutations that alter both the structure (node mutation) and weights (weight mutation) of the neural networks often creating new synaptic links (neurons) and destroying old subpar ones. This process occurs continuously in the simulation and updates the neural networks as bees gradually learn to detect flowers and pheromones and start optimizing their detection and decision-making capabilities. As generations pass neural networks become more complex and bees learn more sophisticated behaviors.
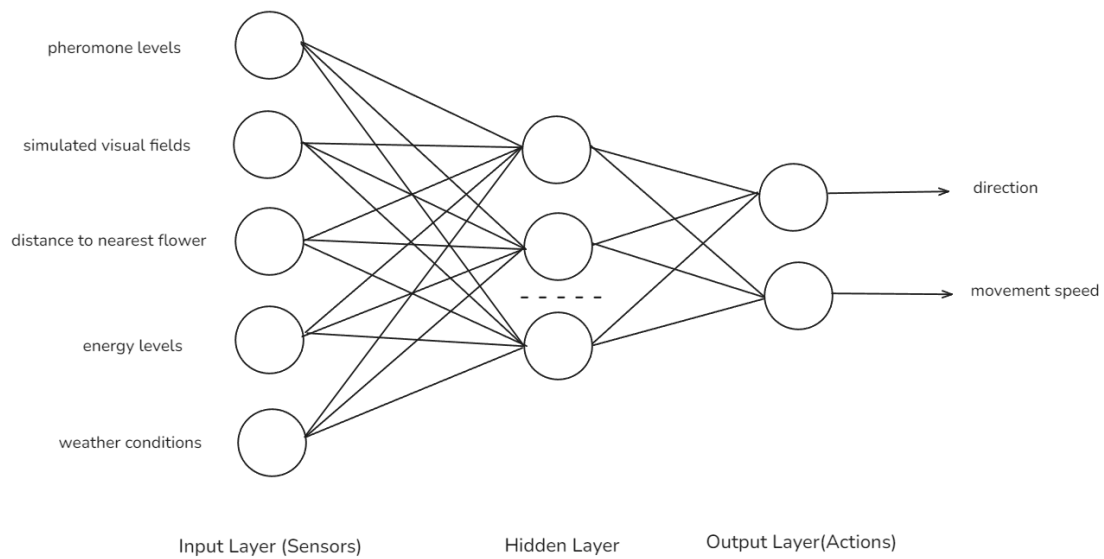
Fig. 1 Neural network architecture of bees

In Fig 1. The inputs and outputs with their corresponding functionalities are illustrated the neural network is a feedforward network which is a good fit for simulation insect brains. The fitness function which determined the best genome was based on the bee's performance in foraging, including nectar collected, efficiency, and other factors.
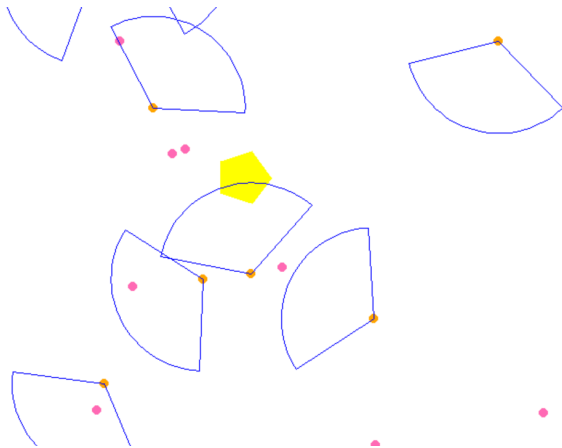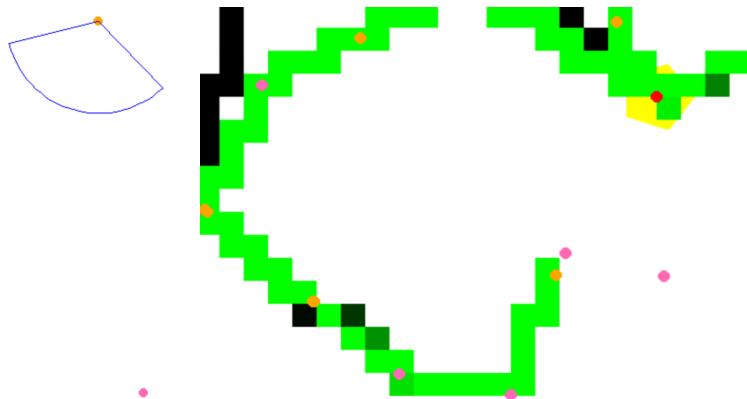

Fig. 2 Bee's Field of view

Fig. 3 Pheromone trails

Bees use field of view to detect nearby flowers within a radius of 100px units but when rain(light) is enabled which is another feature for experimental purposes the FOV is reduced to 60px which limits the bee's ability to detect flowers, the bees also tend to move slower and more energy is used for movement. Bees also deposit some amount of pheromones which are tracked by other bees to navigate and optimize their foraging routes, although the pheromones do evaporate after some time to make the simulation as realistic as possible. For debugging and tracking the learning process Field of view(FOV) and pheromone trails were used along with real-time

feedback on the simulation's progress. The pygame window displayed statistics such as the number of successful foraging bouts, the total no of flowers, the average speed of the bees, and the current weather conditions. The bees also have some amount of energy assigned to them and they return to plan their journey according to the energy they have at the moment. If they find themselves short of energy they return to the hive even if they fail to complete their foraging bout.

The behavior of the bees also depends on heuristics like the Euclidean distance between the bee and its target (flower or hive). This distance is calculated using the Euclidean distance formula:

$$\text{distance} = \sqrt{(x_{\text{bee}} - x_{\text{obj}})^2 + (y_{\text{bee}} - y_{\text{obj}})^2}$$

This distance is fed into the neural network which helps the bee to effectively maneuver itself to optimize its foraging or return to the hive efficiently.

Several experiments were designed to test the bees' ability to optimize foraging routes with different spatial configurations of flowers to observe the bees' ability to search the flowers and optimize the corresponding foraging routes. Most of the experiments are recreations of previous experiments done by researchers (Ohashi et al, 2007)

Experiment 1: The first experiment was done by placing the flowers in a positive array (fig. 4) with10 flowers arranged such that choices of nearest neighbors are always consistent with choices of straightest movements; The objective of this and subsequent experiments was to observe if the bees create similar movement patterns and decision-making found in the experiments from Ohashi et al, 2007. The pink dots represent flower orientation in the environment and the yellow represents the hive.
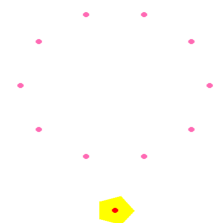

Fig. 4 Positive array

Experiment 2: The second experiment was done by placing the flowers in an independent array (fig. 5) with 10 flowers placed again equidistantly in a triangle with this style of placement bees can choose the distance and turning angle independently (Ohashi et al, 2007). The orange dots in fig.5 are the bees.
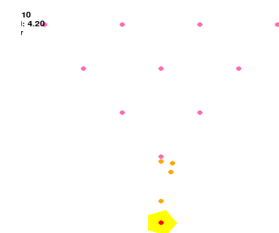

Fig. 5 Independent array

Experiment 3: In the third experiment we used what is known as a negative array (fig. 5) where 10 flowers are arranged in a manner that the

choices of nearest neighbors are often inconsistent with choices of straightest movements. (Ohashi et al, 2007).

Experiment 4: In the fourth experiment we used a grid-style orientation of flowers. This is a variant of the independent array as in this array flowers are also equidistant from each other.
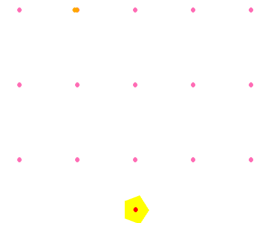
Fig. 6 Negative array

Fig. 7 Grid array

Experiment 5: In the fifth experiment we used a random orientation of flowers to see if the bees were able to locate the flowers and optimize a trapline to collect nectar.

Fig. 8 Random array

Experiment 6: In the sixth experiment we used a random orientation of flowers with rain enabled which limits the visual detection capability of bees by 40% and the pheromones also wash away quicker. We tried to observe how the bees act in such adverse conditions.

During the foraging experiments, the bees also collected nectar from the flowers. The amount of nectar collected was tracked to measure the foraging efficiency of the bees. A higher foraging efficiency indicates that the bees are effectively finding and collecting nectar without wastage of energy. For a bee colony, maximizing foraging efficiency is essential to their survival as it ensures the colony's growth and reproduction in general. After each full hive bout, foraging efficiency is calculated as the amount of nectar collected per bout. This is given by the following equation

$$\text{foraging\_efficiency} = \frac{\sum_{\text{all bees}} \text{total\_nectar\_collected}}{\text{number\_of\_full\_hive\_bouts}}$$

another metric that adds to the observation of the experiments is the search efficiency of the bees. Search efficiency measures how effectively the bees are able to find flowers relative to the distance they travel. In the simulation, the bees primarily rely on pheromone trails and FOV data and learned behavior from neural networks to locate flowers. A high search efficiency indicates that the bee is able to locate flowers without traveling sub-optimal distances, while also minimizing energy expenditure and time. Search efficiency in the simulation is calculated by the following equation

$$\text{search\_efficiency} = \frac{\text{flowers\_visited}}{\text{total\_distance\_traveled}}$$

Both of these metrics are essential to understand the results of the experiments.

Results and Analysis

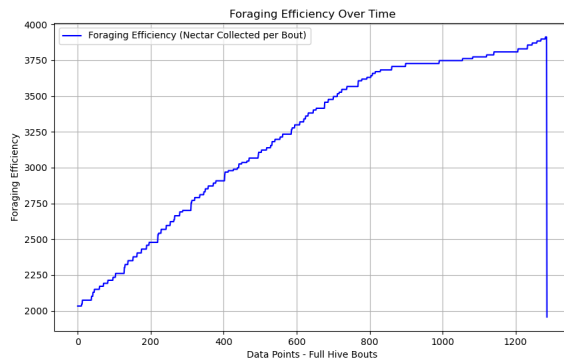The results from the experiments were logged by the simulation and plotted using graphs
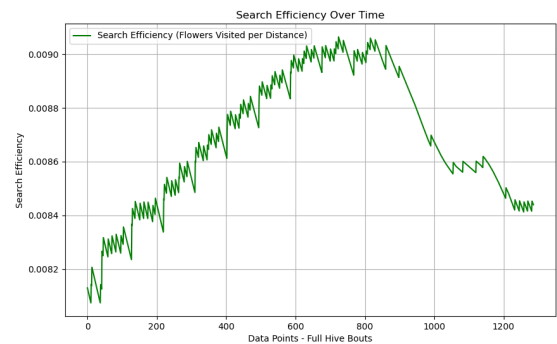

Fig. 9 Experiment 1 FE

Fig. 10 Experiment 1 SE

In the foraging graphs for experiment 1, we observe that foraging efficiency shows a steady increase over time, reaching a peak of 6000 before a sharp drop at the end which is due to the return to hive where the bees unload the collected nectar. The high peak indicates efficient foraging behavior in bees for the positive array which is consistent with the fact that positive arrays are relatively easy to navigate with most choices being straight movements. When we take a look at the search efficiency graph of the foraging bout we see that search

efficiency starts relatively low and experiences a considerable amount of fluctuation before gradually increasing up to 0.0090, suggesting that the bees are becoming more effective at locating flowers as the simulation progresses which again is due to the easy navigation in positive arrays.
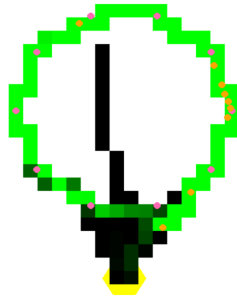

Fig. 11 Experiment 1 Pheromone trail

Fig. 12 (Ohashi et al, 2007 fig 3 b)

The pheromone trails reveal the trapline followed by the bees in Fig. 11 is similar to the trapline in Fig. 12 from the 2007 study (Ohashi et al, 2007) and n=in both cases bees are able to navigate and locate the flowers with relative ease.
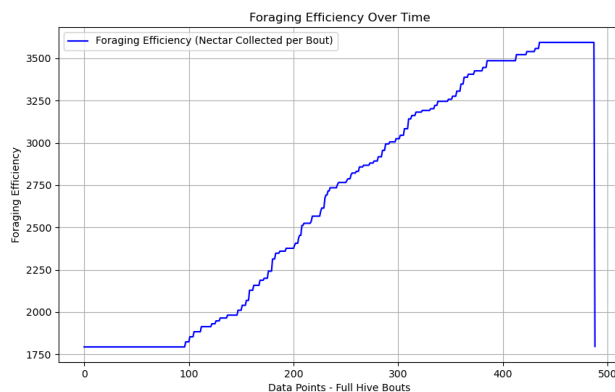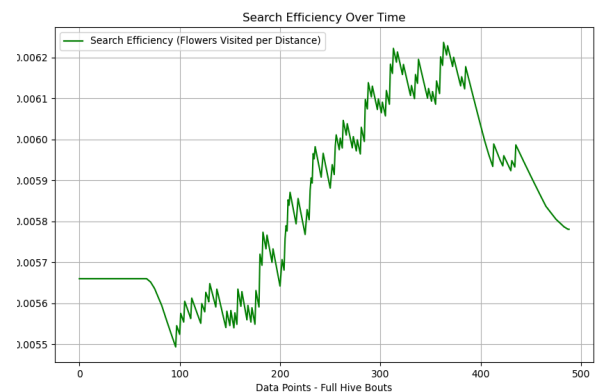

Fig. 13 Experiment 2 FE

Fig. 14 Experiment 2 SE

In Experiment 2 we see that the foraging graph shows an increment for 400 data points in full hive bouts but plateaus at the 3500 mark which is less than the numbers obtained in the positive array which was expected as an independent array was used in this experiment, equidistant placement of flower caused the bees to independently choose the distance and turning angle as there was no clear nearest flower for the bees to choose. The cyclic increment and decrement in the search efficiency graph also confirm that the bees were finding it difficult to navigate the array and the markedly low score of 0.0062 proves that the bees were not following optimal routes for foraging.
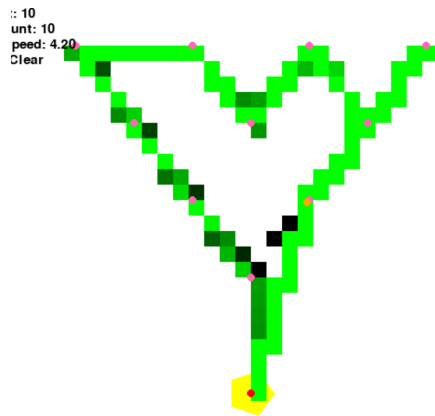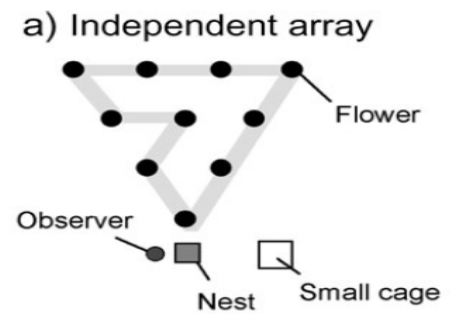
Fig. 15 Experiment 1 Pheromone trail

Fig. 16 (Ohashi et al, 2007 fig 3 a)

When we look at the pheromone trails we see that the bees in both cases largely follow similar routes except the artificial bees falter at the last flower and choose a suboptimal route, unlike the real ones. In Fig. 16 the bees approach the central flower earlier, unlike artificial bees from the experiment who only approach it much later. This indicates the artificial bees have the potential to replicate the behavior.
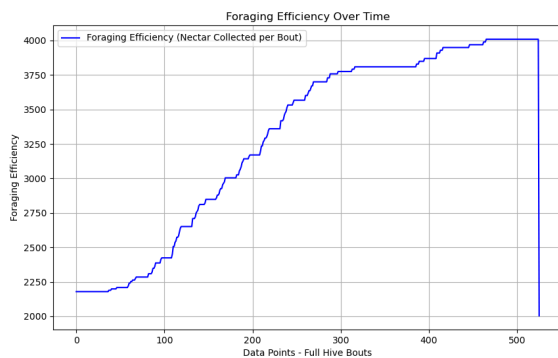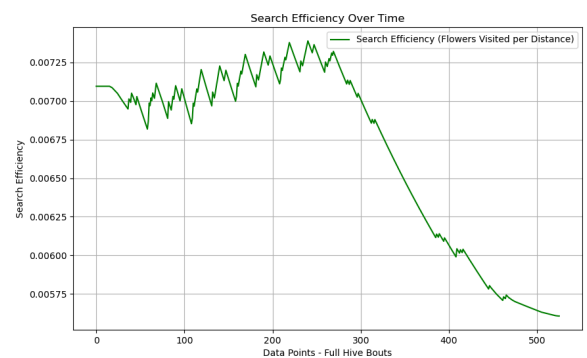

Fig. 17 Experiment 3 FE

Fig. 18 Experiment 3 SE

Experiment 3 results for foraging efficiency show a steady increase up to 4000 which is better than the results of Experiment 2 but there is a sharp decline in the graph of search efficiency which doesn't match with the comparatively efficient foraging graph from the same experiment, again the reason is the orientation of flowers in a negative array which tricks the bees into following the less optimal path for foraging, the bees followed the nearest flower which was visible in their FOV and the other bees followed the first one because of pheromone detection.
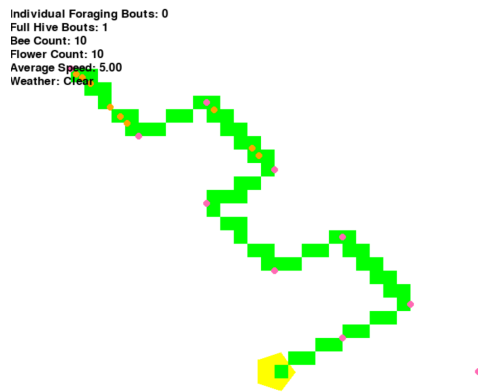
Individual Foraging Bouts: 0
Full Hive Bouts: 1
Bee Count: 10
Flower Count: 10
Average Speed: 5.00
Weather: Clear

c) Negative array

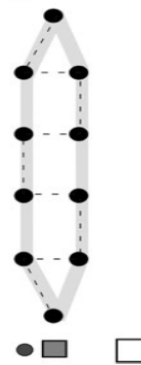**Fig. 19 Experiment 3 Pheromone trail**   **Fig. 20 (Ohashi et al, 2007 fig 3 c)**

In experiment 3 the pheromone trails seem to follow a similar pattern from Fig. 20 again, It is to be noted that the orientation of the array and position were not accurate enough which could be the reason why one of the flowers was not detected and the search efficiency scores plummeted during the experiment. In all three experiments, exactly the same number of flowers (10) were used to ensure that the scales remained the same for better comparison mainly in terms of nectar collection. The rest of the experiments are of different scales and are thus are not comparable.
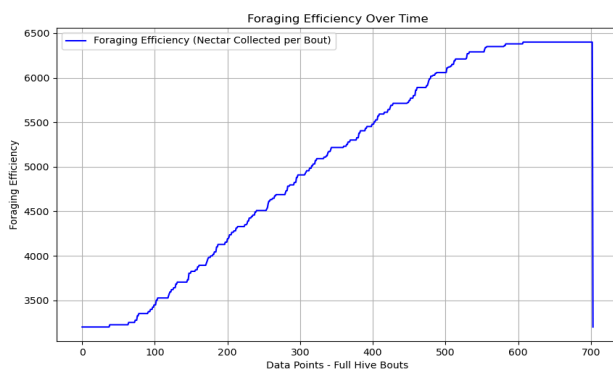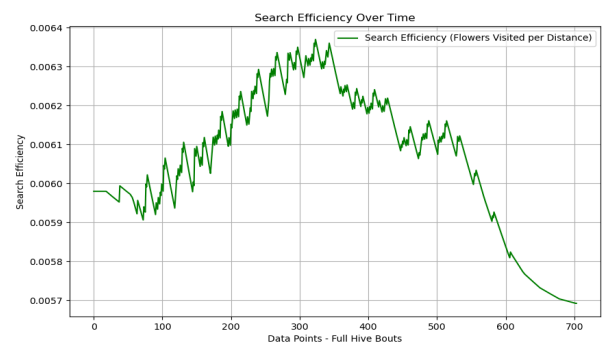

Fig. 21 Experiment 4 FE   **Fig. 22 Experiment 4 SE**

In Experiment 4 another version of the independent array is used which is just a grid of equidistant flowers. In the foraging efficiency graph, we can observe that the nectar per bout reached a peak of 6500, and the search efficiency similar to results from the previous
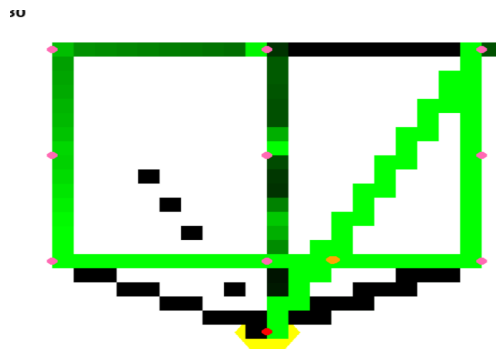
independent array shows a lower score and approximately similar scores capped around 0.0060.

Fig. 23 Experiment 4 Pheromone trail

The pheromone trails show the formation of a suboptimal trapline emphasizing the fact that Independent orientation of any kind leads the bees to form suboptimal traplines. The number of flowers used here was reduced to
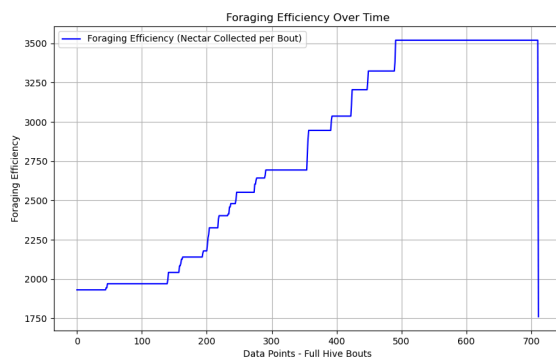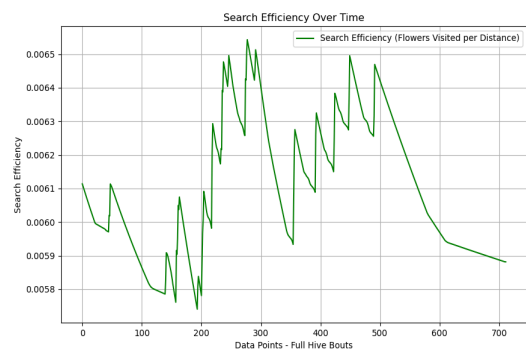

Fig. 24 Experiment 5 FE

Fig. 25 Experiment 5 SE

In the random placement experiment, the foraging bouts are not impressive and plateau at 3500 and the peak search efficiency of 0.0065 is also very low suggesting the bees find it difficult to find flowers when flowers are fewer in numbers and their high reliance on pheromones than visual fields.
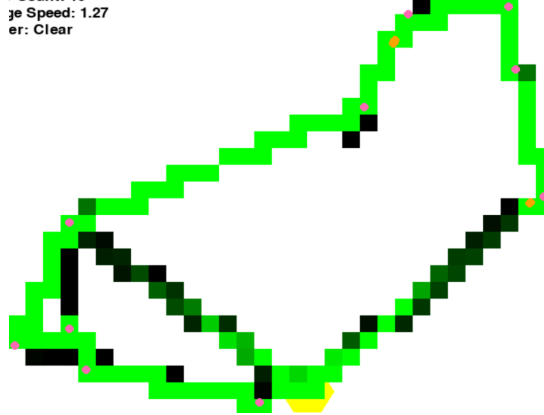
Fig. 26 Experiment 5 Pheromone trail

Traplines were formed (Fig. 26) but the bees were not confident enough to traverse them with reliability and often slowed down and went into exploration mode more often.
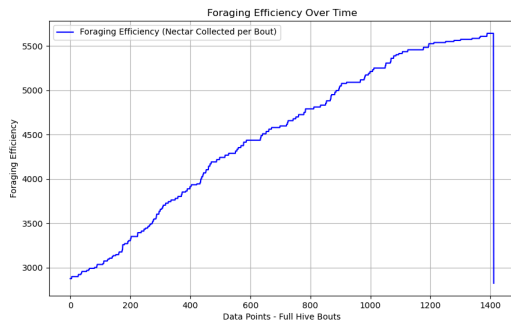
Fig. 27 Experiment 6 FE
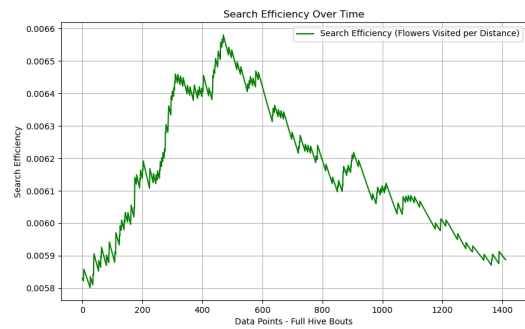
Fig. 28 Experiment 6 SE

In experiment 6 the no of flowers was increased to 15 and rain was enabled to limit the FOV of bees and pheromones also vanished quickly, as a result, the bees managed to forage and navigate and search for flowers evidenced by the high nectar count but the search efficiency graph which peaked at 0.0060 with multiple fluctuations shows that the bees were constantly searching for information in the unfavorable conditions and trying to adapt.

Fig. 29 Experiment 6 Pheromone Trail

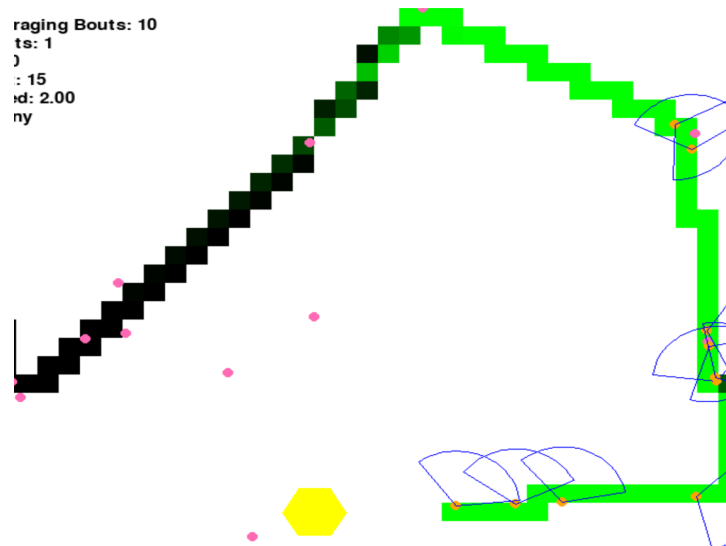Pheromone trails seem to wash away quickly due to rain in Fig 29 and bees' FOV is reduced, they also stopped their FOV movements to conserve energy and returned to the hive.

Discussion

In the study, we explored multiple scenarios using the simulation particularly the foraging behavior of bumble bees. The simulation based on the neural networks evolved through the NEAT(NeuroEvolution of Augmented topologies) algorithm was an attempt to essentially create an artificial bee brain in order to better understand the nuances of their highly optimized foraging routes. By allowing the bees to develop their own foraging routes also commonly known as traplines we aim to further the pursuit to decode their tiny brains and their mechanisms.

The results from the experiments reveal several key insights such as how bees gradually optimize their foraging efficiency and search efficiency, depending on the configurations of the flower in and around their hives. The positive array experiment (Ohashi et al, 2007) which was originally done in a lab with real bees was conducted on virtual or rather artificial bees. The bees quickly learned to follow the pheromones and later learned to search better and choose the most optimal path which is the straight lines, this led to high efficiency in nectar collection and search. This outcome supports the hypothesis that simple, direct paths allow the bees to optimize their foraging path better

In the independent array experiment which consisted of an equidistant triangular configuration of flowers the bees struggled to maintain high efficiency and almost always chose the most suboptimal paths. The fluctuation in the search efficiency graph also suggests that bees were forced to make more complex decisions when choosing between multiple equidistant targets. Their inability to optimize an optimal trapline asks for more modifications and experimentation to make them more adaptable to such situations as the real bumblebees. The Negative array experiment showed a similar movement pattern as was found in the 2007 study (Ohashi et al,) but the faulty orientation of the array was an

issue that skewed the results somewhat as the search efficiency graphs were rendered inaccurate. Grid array experiment which was just a different version of the independent array showed similar results as the bees often used sub-optimal paths, although the linear nature of the array allowed for faster movements and as a result high foraging efficiency. This experiment highlights the difficulty that bees face when no single route has the certainty of being consistently better than others, forcing them to rely more on exploration.

The last experiment which involved the Random placement of flowers in the simulation with environmental stressors like rain, demonstrated the bees's adaptability. The formation of traplines despite the adverse conditions suggests that the bees are able to navigate and optimize traplines better when more flowers are found at random as they are able to differentiate the relative distance between flowers and themselves to make better foraging decisions which results in more optimized traplines.

The simulation is still in its infancy and has several bugs that need to be fixed and worked on. Future versions can have features like dynamic obstacles that will make the simulation more realistic and prove a good challenge for the bees to adapt to. But before any new addition, the current version and the learning behavior of the bees need to be fine-tuned, some relevant changes could be the modification of simulated visual fields to detect unique patterns and visual landmarks along with improvements in the maneuverability of the bees to make them more adept in navigating their search for flowers. Other Potential fixes could be in the decision-making mechanisms of the bees we will have to ascertain that the bees are able to navigate independent and negative arrays optimally. Extensive testing with the neural network architecture is also required as it's possible that adding more input layers may improve the bees' ability to make decisions. New experiments can also be devised like the problematic array experiments which will test the bee's decision-making abilities in new ways. The simulation is a good tool for understanding the behavior of bees and also understanding both natural and artificial systems and it could be extended to incorporate other insects like ants and spiders to simulate and understand complex intelligent behavior displayed by them. Evolutionary stable strategies and competition between multiple species could also be studied using similar tools which will require a more nuanced approach to simulation. One limitation of this study is the use of simplified environmental models that may not capture all the complexities of real-world foraging. Further work could involve a more detailed simulation environment with multiple variables at play, such as the aforementioned competition with other bees or varying flower nectar levels. Although in this iteration The feed-forward network was used which is a good and widely used architecture for simple behaviors it could be replaced with a more fitting Spiking neural network which is biologically more plausible in this use case as they are adept in capturing the temporal dynamics of a biological neurons, Spiking neural networks are capable of emulating the timing and frequency of neuron firing which is a better fit than feed-forward neural networks.

The study with multiple experiments and its results show that simple heuristics like nearest neighbor, pheromone detection, and energy conservation evolved through neural networks, can replicate the foraging strategies observed in bumble bees. This study holds potential for more improvements and helps us understand intelligent behavior like optimizing traplines with more clarity.

References

1. A Simple Iterative Model Accurately Captures Complex Trapline Formation by Bumblebees Across Spatial Scales and Flower Arrangements ,Andrew M. Reynolds ,Mathieu Lihoreau,Lars Chittka https://doi.org/10.1371/journal.pcbi.1002938
2. Evolving neural networks through augmenting topologies Kenneth O Stanley 1, Risto Miikkulainen 10.1162/106365602320169811
3. Trapline foraging by bumble bees: IV. Optimization of route geometry in the absence of competition Kazuharu Ohashi, James D. Thomson, Daniel D'Souza https://doi.org/10.1093/beheco/arl053