Artificial Bees Simulation v2.0

1.0 About

This project is to simulate a simple artificial bee colony with 40 bees in 3D space. Like in previous simulation (http://inf0warri0r.blogspot.com/2013/01/artificial-bees-simulation-using-neural.html) the function of a bee is to get honey from a flower and bring it to the hive which is in the middle. When the bee isn't carrying honey it is in color red and when it get near to a flower it became blue. Then it has to get back to the hive to store the honey.

2.0 Implementation

A bee containing a small brain consisting of two neural networks.

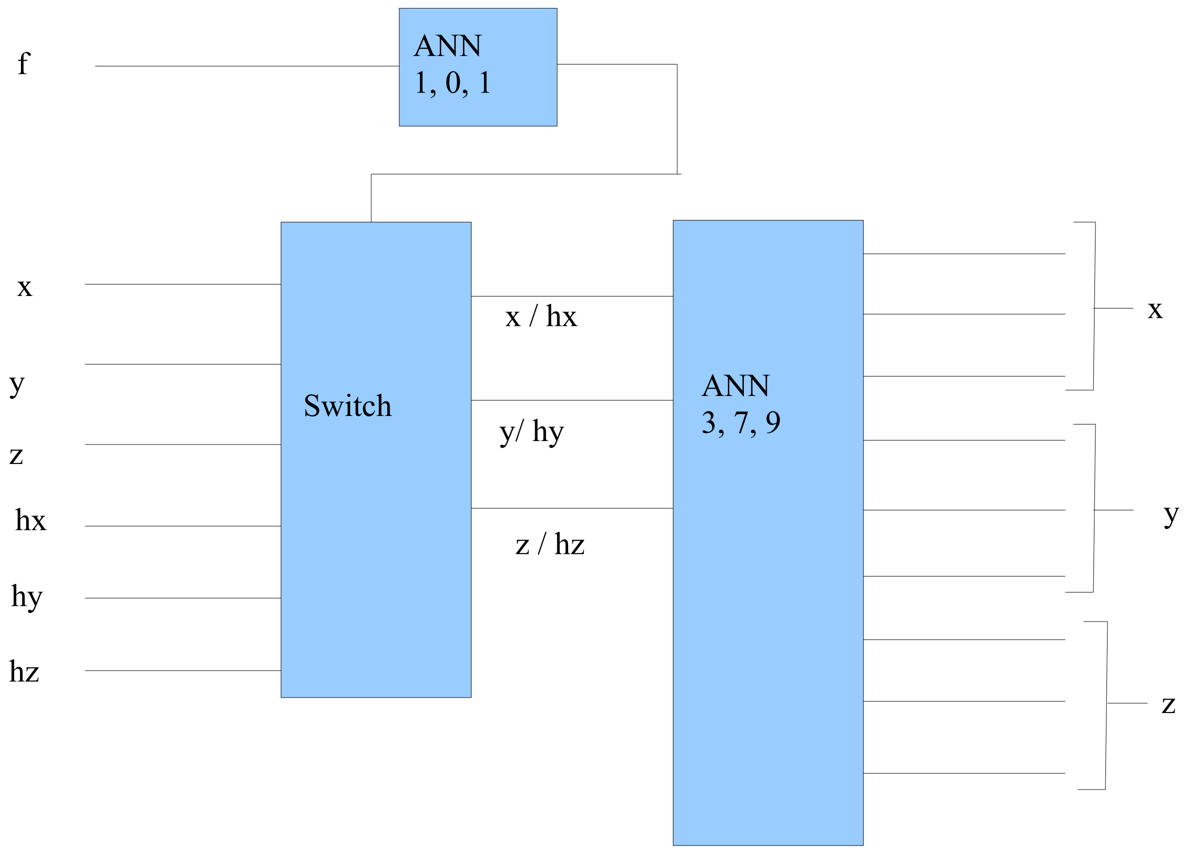

Figure 2.0 – brain of a artificial bee

The small neural network has 2 layers with one input and one output neurons. And other one is consisting of 3 input neurons 7 hidden neurons and 9 output neurons. The input of the first neural network is a flag saying if the bee is carrying honey. The second neural network get the coordinations of the hive or the nearest flower according to the output of the first neural network.

Weights of the neural networks are updated using a genetic algorithm. First it make the swam with random weights and the after each generation (600 iterations) new swam is made by crossovers and mutations of old swam.

The two bees with highest fitness will be in the next generation without mutations . The rest is generated by crossing over the two third of the highest fit bees randomly. The probability of a bee to select for crossing over is happening according its fitness.

Fitness function for the bees,

```
def fitness(self, b):

    x = self.bee[b][0]
    y = self.bee[b][1]
    z = self.bee[b][2]
    f = self.bee[b][3]

    for i in range(0, self.flower_count):
        d = (x - self.flower[i][0]) ** 2.0
        d = d + (y - self.flower[i][1]) ** 2.0
        d = d + (z - self.flower[i][2]) ** 2.0
        d = d ** 0.5
        if d < 20.0 and d > -20.0:
            if f == 0:
                self.flower[i] = (self.flower[i][0], self.flower[i][1],
                self.flower[i][2], self.flower[i][3] + 1)
                self.bee[b] = (x + 10, y + 10, z + 10, 1)
                return 0.0
            else:
                return 0.0

    d = (x - self.hive[0]) ** 2.0
    d = d + (y - self.hive[1]) ** 2.0
```

```
d = d + (z - self.hive[2]) ** 2.0
d = d ** 0.5
if d < 60.0 and d > -60.0:
        self.bee[b] = (x, y, z, 0)
        if f == 1:
                self.total = self.total + 1
                return 100.0
        else:
                return 0.0
return 0.0
```

Figure 2.1 fitness function

The flowers are in random locations. When a flower is visited by more than 5 bees it turns in to turns in to light blue and when it is visited by 8 bees it dies and another flower is born in another random location.

3.0 Compiling and Running

This script is written in python. Install 'python' , 'openGl for python' and 'pygame' to run the script. (use synaptic package manager in Ubuntu to install them) .

use command ,

```
python main.py
```

to run. Use 'a' and 's' in keyboard to to turn the simulation around y axes. Use 'd' and 'f' to move up and down and 'g' and 'h' to move back and fourth.

4.0 Output

In the beginning the bees act randomly. But after some generations bees start to make formations. And the formations brakes up get together again according to the pattens of the spreading of the flowers.

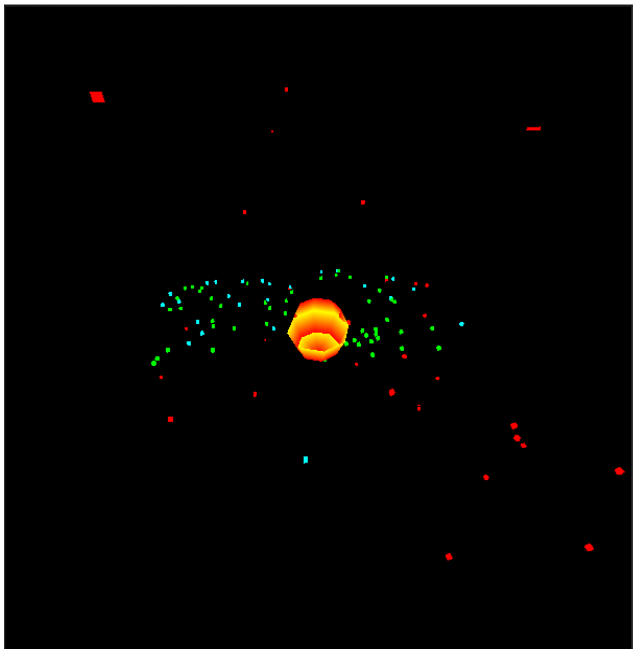

Figure 4.0

The to amount of honey collected vs. generation count graph is shown in Figure 4.1

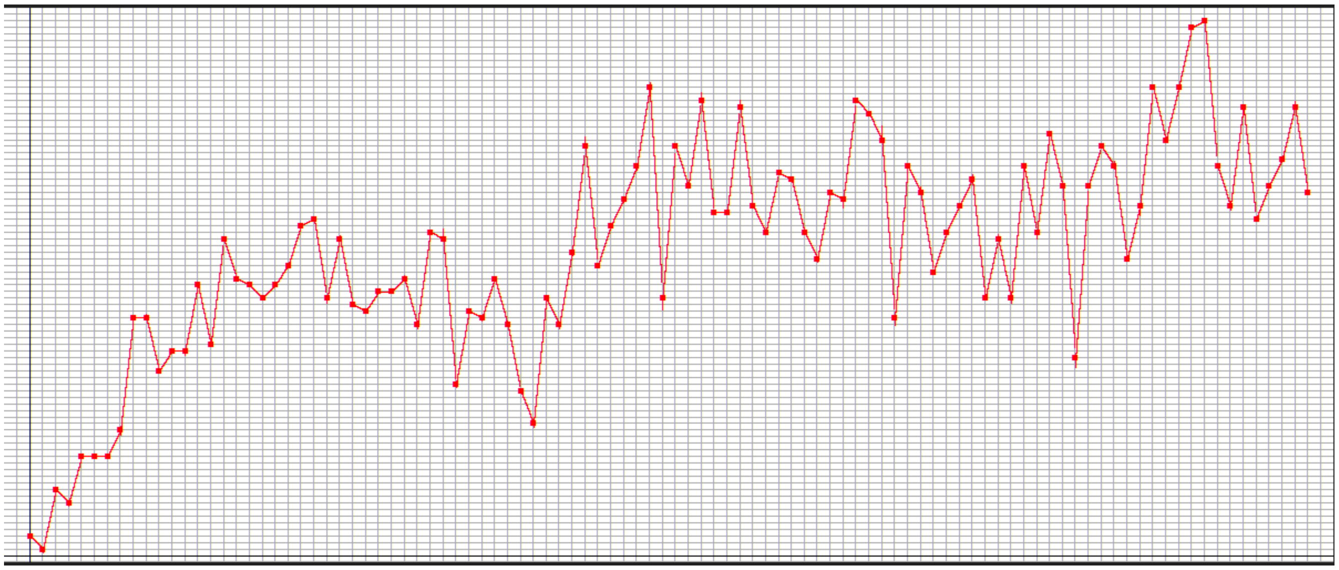

Figure 4.1 (x range 1 – 100, y range 5 - 85)

5.0 More Inf0:

- *tcg.galahena@gmail.com*
- *http://www.inf0warri0r.blogspot.com*

6.0 Licenses: