Final Report

- EP18BTECH11017
Mohan Chandra Varma

Supervisor
- Joyjit Kundu

Optimizing collective fieldtaxis of swarming agents through reinforcement learning

Introduction

The swarming of animal groups has fascinated scientists in fields ranging from biology to physics to engineering. The complex swarming patterns often arise from simple interactions between individuals to the benefit of the collective whole. Here we show that a machine-learning technique can be employed to tune these underlying parameters and optimize the resulting performance.

In this project we try to simulate a group of fish which try to find preferred regions in a sea or ocean where there might be less light so that the predator doesn't find them. They do it collectively by interacting with each other on an individual level that benefits the whole swarm.

Algorithm and Simulations

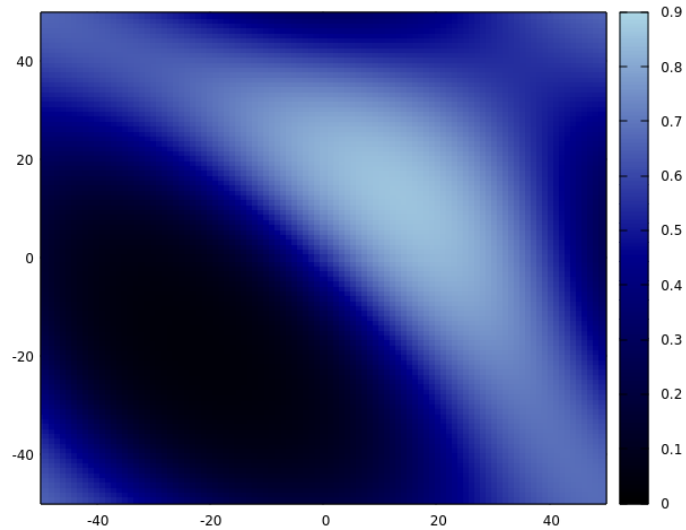
Part - I

Firstly I casted a static light field using the equation

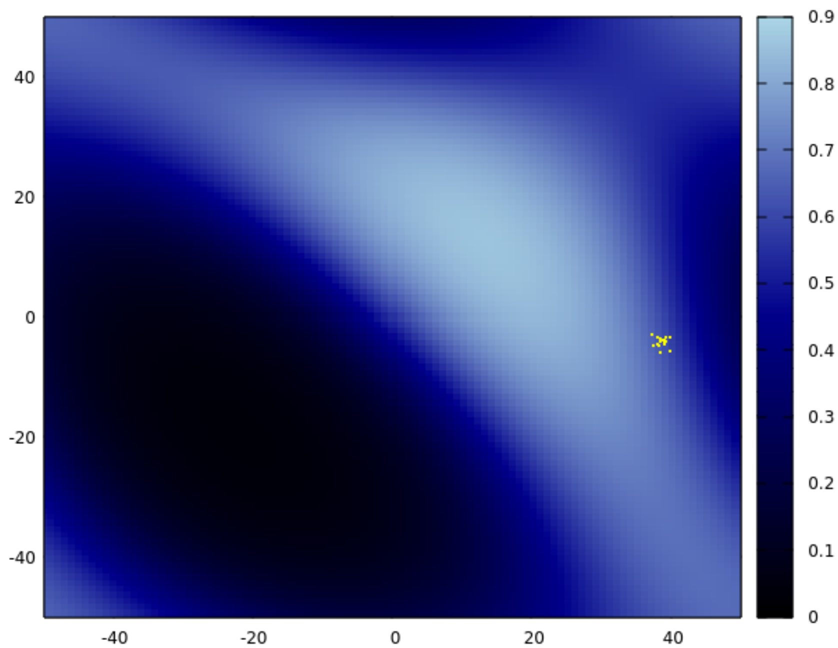
$$F(X) = \sum_k [A_k cos(k.X) + B_k sin(k.X)]$$

We do the sum over wave vectors, k =(k_x,k_y) which runs through k_x,k_y = 0,$\frac{2\pi}{L}$

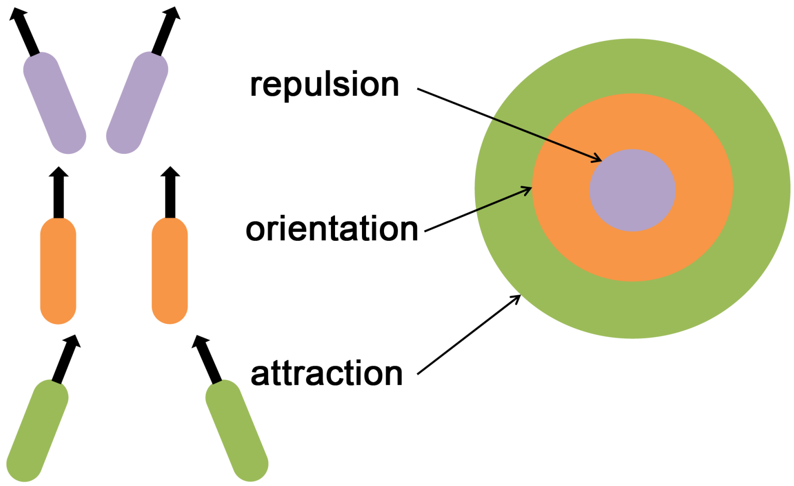
We get A_k and B_k using gaussian normal distribution with mean of 0.5 and standard deviation 0.05



Then I initialized positions of the agents uniformly within a circle of radius R =$\sqrt{\frac{N}{\pi}}$,
where N = number of agents , here N = 16



In fishes we can see a coherent flocking happens by interacting through repulsion at short range r_r, orientation at intermediate range r_o, and attraction at long range r_a.



With each time step Δt, we update the positions and velocities as

$$x_i(t + \Delta t) = x_i(t) + \frac{1}{2}[v_i(t) + v_i(t + \Delta t)]$$

$$v_i(t + \Delta t) = [v_{max} F(x_i(t))] \, R_{noise} \widehat{d}_i(t)]$$

The position of particles changes accordingly with the velocity vector

And the magnitude of velocity of particles changes with the amount light received

And the direction of the particles changes accordingly with the following conditions

i) If there are any neighbors within the zone of repulsion r_r, they need to repel so

ii) If there neighbors in the zone of repulsion but within zone of orientation r_o ,then they need orient in the same direction. If within zone of attraction r_a , then they need to attract so

$$\mathbf{d}_i(t) = \left[\sum_{\substack{j \neq i \\ r_r \leq \|\mathbf{x}_j(t) - \mathbf{x}_i(t)\| < r_o}} \frac{\mathbf{v}_j(t)}{\|\mathbf{v}_j(t)\|} \right] + \left[\sum_{\substack{j \neq i \\ r_o \leq \|\mathbf{x}_j(t) - \mathbf{x}_i(t)\| < r_a}} \frac{\mathbf{x}_j(t) - \mathbf{x}_i(t)}{\|\mathbf{x}_j(t) - \mathbf{x}_i(t)\|} \right];$$

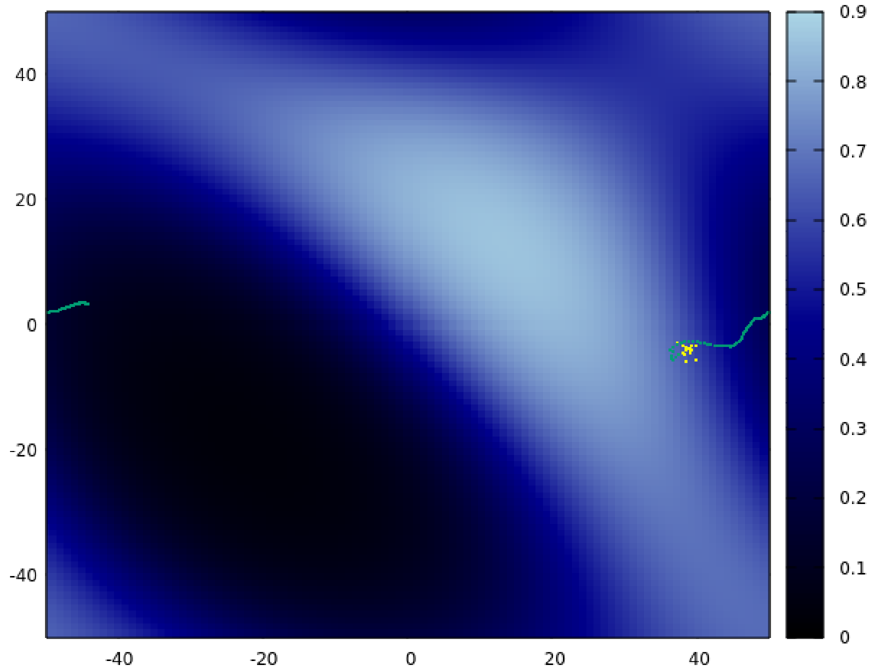
iii) If there are no neighbors within zone of attraction r_a, then there will be no change in the direction so

$$\mathbf{d}_i(t) = - \sum_{\substack{j \neq i \\ \|\mathbf{x}_j(t) - \mathbf{x}_i(t)\| < r_r}} \frac{\mathbf{x}_j(t) - \mathbf{x}_i(t)}{\|\mathbf{x}_j(t) - \mathbf{x}_i(t)\|};$$

$$\mathbf{d}_i(t) = \mathbf{v}_i(t)$$

In training we allow the agents to move up to time t = t_* where t_*= 100 in our case

We can see the following path by the agents



Here we chose r_o = 1.95 , r_a = 2.05 & r_r = 1.00

We need to optimize r_o, r_a using reinforcement learning so that the agents can find the darkest spot easily.

For this we run the training algorithm for N_{train} times indexed by $\alpha = 1, 2, ... N_{train}$.

We take the average field intensity perceived by the agents so that we can use this as reward

Then we define the reward as

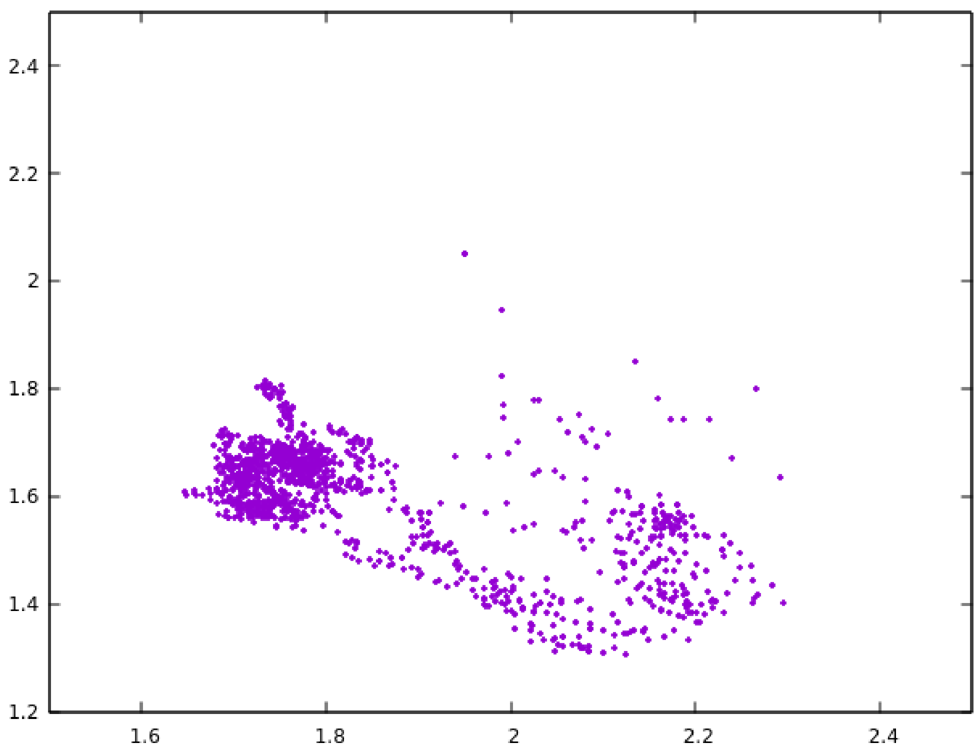
$$Q = \max\{ 0 , f(0) - f(t_*)\}$$

So, if f(t) decreases over time t_* then the reward is going to increase.

At α-th training session with initial positions and velocities of agents, we evaluate the rewards at three nearby points: Q_0 at $(r_o^{(\alpha)}, r_a^{(\alpha)})$, Q_1 at $(r_o^{(\alpha)} - \delta, r_a^{(\alpha)})$ and Q_2 at $(r_o^{(\alpha)}, r_a^{(\alpha)} + \delta)$ with deviation $\delta = (\alpha + 1)^{-1/4}$. And we update the parameters as

$$r_o^{(\alpha+1)} = r_o^{(\alpha)} + \gamma(Q_0 - Q_1)/\delta$$
$$r_a^{(\alpha+1)} = r_a^{(\alpha)} + \gamma(Q_2 - Q_0)/\delta \qquad \text{where } \gamma = (\alpha + 1)^{-3/4}$$

After many iterations we can see how r_o, r_a are changing and get the optimised r_o, r_a.

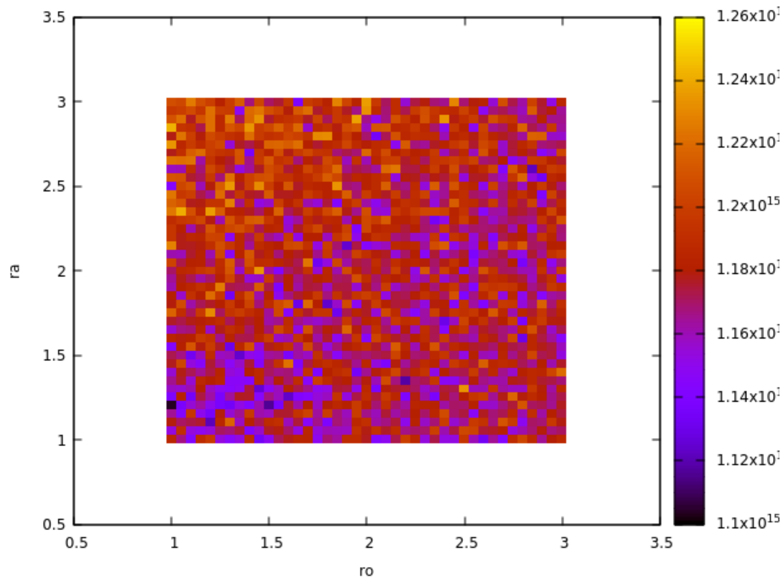


We can see that r_o, r_a are approaching 1.75, 1.68.

Part - II

We need to train our agents with allowed continuum values of ro and ra i.e., [1,3] .
We need to train on each and every pair of ro and ra values and take the average reward over different static fields.

After training over 1600 fields we got



In this we can see that ra greater than ro will give us more reward.

Conclusion

In this project I have simulated a static light field and introduced agents to swarm around and find the darkest place. We can see how the simple interactions between agents benefits the whole swarm. I have implemented an algorithm to optimize r_o, r_a using reinforcement learning which was given in the paper. We can see how optimizing r_o, r_a can help in effectively sensing the gradient.