Comparing 1-eyed and 2-eyed simulated agents optimised to find light sources

# introduction

I optimised a simulated Breitenberg vehicle using a microbial genetic Algorithm. I did this for both 1-eyed agents and for 2-eyed agents. My goal is to compare their fitness afterwards as well as analyse the types of solutions they developed.

Through this report, I learned the extent on small changes in the fitness function can have vast effects on the agent that the genetic algorithm produced. After comparison of the agents, I edited the fitness function to produce desirable traits in the best solutions produced. This was successful, I was able to control the type of agent produced by slight changes to the fitness function.

# Method

## The GA

I used a microbial GA with tournament selection. When obtaining the best solutions for the different eyed agents. I ran my GA 200 generations with 100 participants each time.

The agent contained 6 genes with wraparound boundaries implemented during mutation. The one-eyed agent had the genes to do with the right sensor deactivated, which simulated one eye using only 4 of the genes. The boundaries for the genes were between -4 and 4. Using a wraparound mutation, this allowed for the occasional big jump in difference and slowed down convergence of the microbes while allowing extra parts of the problem space to be explored when the wraparound occurred.

The tournament performed 10 competitions, whenever comparing microbes. The values of starting position and starting angle where randomised for each of these competitions. Starting position was however capped. This ensures that the agent is responding reactively to the light and reduces the likelihood of a non-optimal agent getting lucky. I did not allow an agent to compete with itself in this tournament in order to prevent lost of highest fitness and maintain elitism.

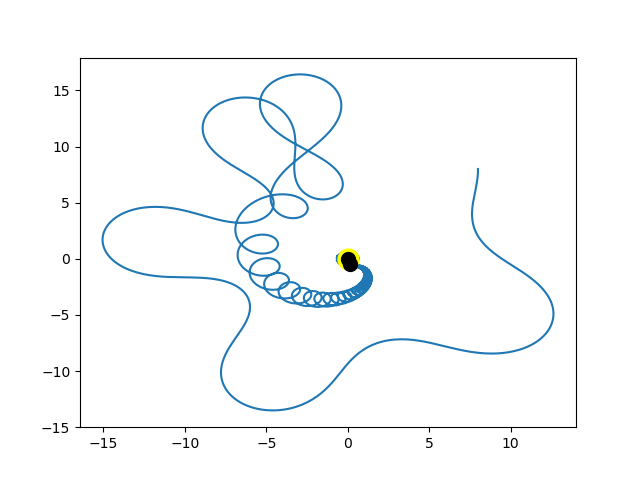
The fitness function I performed was simply distance from the light source at the end of the timescale. My fitness function returns a negative number with the highest fitness possible being zero. I created my fitness function this way because being on the light source is the limit of the fitness.

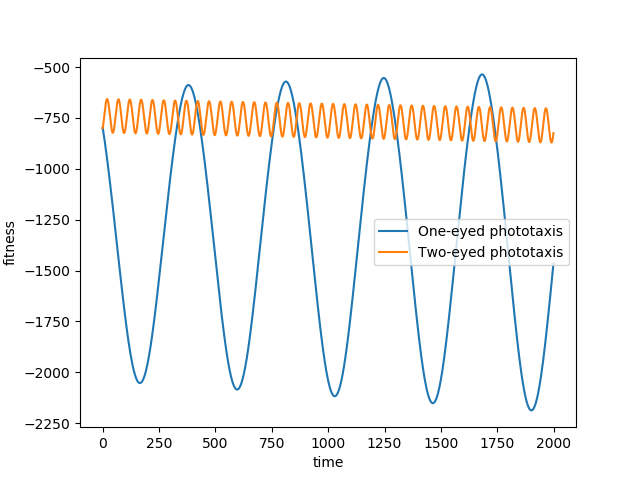
## Pseudocode

:

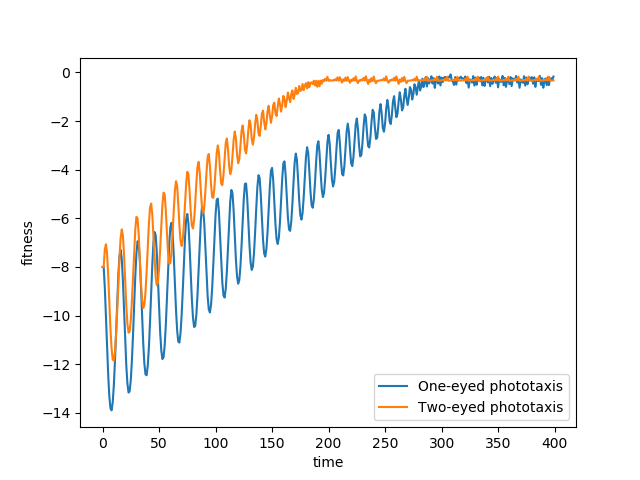
# Findings

Both one and two eyed phototaxis microbe populations were able to produce agents that could move towards the light source.

Below is the path the best performing two eyed agent chose during analysis.

While it is erratic, it did head towards the light source. Interestingly when I tested it in a starting distance that was outside of the range the tournaments generated, the behaviour was always erratic and did not always head towards the light. This same phenomenon occurred with the one-eyed agent. This shows me the importance of the fitness function and the effect it has on an agent it produces when using a genetic algorithm. Below is a graph showing the fitness of both one and two eyed agents when tested outside of their ‘comfort range’.

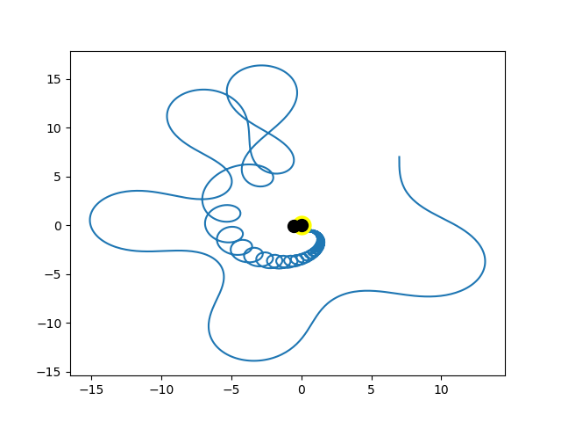
As you can see, when tested out of their comfort range, the performed quite badly, with the two eyed agents progressively getting worse.

When tested within their comfort ranges of variables, the two-eyed agent consistently performed better than the one eyed. The one-eyed agent always achieved the same fitness as the two eyed agent given enough time. Below is the comparison of the two.

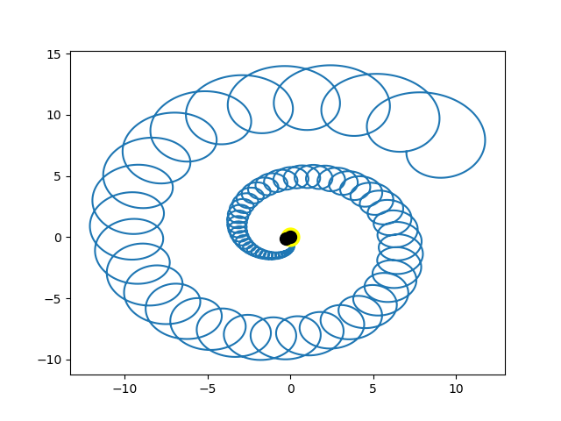
The main difference between the 2 came from how far out the placement was. The difference in performance became increasingly noticeable depending on how far away the starting position of the agent was. With a close starting position being barely noticeable.

## Optimising

My next step in analysis was to attempt to control the erratic behaviour of the agents, to do this, I increased the time allowed during The Gas lifecycle with the hope that this would produce more controlled agents due to the lack of needing to cope with the maximum distance at in a short amount of time. The downside of doing this is it greatly increases the time taken to generate an agent.

Increasing the amount of time in training did in fact lower the franticness and unpredictability of the agent. Seen below is the typical solution provided by the agent before training time was increased.

Then pictured below is an agent developed after allowing more time in the tournament competitions.

While on average the longer timed agent performed better, overall. The difference was not to a large extent. In my case, the very large increase in training time meant that overall the original short timed method was more efficient, since population size or generation count could be significantly increased for the same cost.