Candidate number
236636

Evaluating the behavioural changes of the OpenAI cart pole task solved with a Full Microbial Genetic Algorithm

Acquired Intelligence & Adapative Behaviour (G6042)

17-05-2022

Abstract

We optimised a Full Microbial Genetic Algorithm (FMGA) to solve the OpenAI cart pole task: finding a genotype that could keep the pole up for 500 episodes. Having found a correlation between the number of epochs, the population size, and the fitness of the best individual, we trained an FMGA with 50 individuals over 1000 epochs to get the most fit controller. We examined how the position, velocity, angle and rotation rate of the most fit controller at different numbers of epochs change. We found patterns in the failures of controllers, such as an increased velocity before falling. We also found that the fittest controllers keep a consistent pattern in each observed behaviour until the last episodes.

1 Introduction

Full Microbial Genetic Algorithms (FMGAs) mimic a population of individual's internal fight to pass on the best genes by a series of fitness tournaments that crosses the loser's genes with the winner's (see papers [3] and [4]). We can define their fitness by their ability to keep a cart pole steady, a balancing task defined by OpenAI where a controller can push the cart to the left or to the right (see reference [2]). The individuals in the FMGA each create neural networks with weights defined by their genotype, the input being the observations returned from the cart pole environment and the output being the action it takes, similar to the method in this paper [5]. To investigate the performance of the FMGA model on this task, we first optimised the size of the population and the number of tournaments to maximise the fitness. Having found a best fit population, we then quantitatively and qualitatively examined how the position and angle of the cart pole changed as the controller tried to keep it steady. We expect a population of 50 genotypes over 1000 tournaments to solve the task, and for the variance of a successful controller's metrics (position, angle, etc.) to stay low.

2 Methodology

After having defined an FMGA, we will optimise its population size and number of tournaments to solve the cart pole task and the effects of these hyper-parameters on its ability to reach maximum fitness. We define the maximum fitness as holding up the pole on the cart for 500 episodes. We will then take the best-fit individual from our optimised population at different numbers of epochs and evaluate how its position, angle, velocity and rotation rate change as it solves the task. The evaluation will be quantitative, as we plot how the observed behaviours change, and then qualitative as we watch the plotted scene's frames of the cart pole through a video.

2.1 Defining the FMGA

This is a brief section to show how we define an FMGA. When training this type of model, we want to extract the fitness the individuals have at each tournament. This permits us to compare how different models evolve when solving the task at hand.

Each individual in the FMGA has a genotype that defines a neural network, which is used with the observations from the cart pole environment to define the next action: push the cart to the left or push the cart to the right. An individual with a high fitness will form a neural network who's outputted action is more effective in keeping the cart pole up right.

2.2 Optimising the FMGA

The first part of optimising the FMGA is finding the population size that reaches the solution first. We recognise that, because of the random nature of the FMGA, large population sizes are likely to have an initial genotype that already maximises the performance of the task. This will maximise the fitness, but it will prevent us to explore other areas of the impact of our FMGA, such as the change in fitness as the number of tournaments increases. This is why we need a population size that is less likely to immediately solve the task, but will in a reasonable number of epochs.

Algorithm 1 Define and train FMGA

Input: number of epochs, population size
Output: fitness of each individual per tournament

```
population = list of random genotypes
populationFitnessPerTournament = []
for t = 1, 2, ..., epochs do
    firstGenotype = getRandomGenotype(population)
    secondGenotype = getRandomGenotypeInNeighbourhood(population,firstGenotype)
    if getFitness(firstGenotype) is less than getFitness(secondGenotype) then
        winner = secondGenotype
        loser = firstGenotype
    else
        winner = firstGenotype
        loser = secondGenotype
        loserReplacement = crossover(winner,loser)
    end if
    if fitness(loserReplacement) is more than fitness(loser) then
        population[loser] = loserReplacement
    end if
    populationFitnessPerTournament.append(getPopulationFitness(population))
end for
return populationFitnessPerTournament
```

Algorithm 2 Evaluating population sizes

Input: list of population sizes
Output: average fitness of final tournament of FMGA models with different population sizes

```
averagePerformanceOfPopSize = [ ]
for i = 1, 2, ..., length(populationSizes) do
    averagePerformanceOfPopPerTournament = [ ]
    for t = 1, 2, 3 do
        performancesPerTournament = getFMGA(populationSize[i])
    end for
    averagePerformanceOfPopSize.append(getMeanFitnessPerTournament(averagePerformanceOfpop))
end for
return averagePerformanceOfPopSize
```

The second part of our optimisation method is the number of tournaments. "Tournaments" and "epochs" are interchangeable names as they are the number of times the FMGA will make two genotypes compete for a crossover. Optimising the number of tournaments lets us know how long we have to train our population for in order to have an individual that can solve the cart pole task.

Algorithm 3 Evaluating the number of tournaments

Input: maxEpochs
Output: average fitness of FMGA model per end of tournaments inputted

```
averageFitnessPerTournament = []
for i = 1, 2, 3 do
    model = getFMGA(maxEpochs)
    averageFitnessPerTournament.append(getMeanFitnessPerTournament(model))
end for
return averageFitnessPerTournament
```

We can visualise the results of algorithm 3 on our population size optimised FMGA with a bar-plot chart that shows the progression of the fitness of our model as the number of tournaments increases. We would, ideally, choose the optimal number of tournaments when a plateau at the maximum fitness starts forming in our fitness data.

2.3 Examining positioning changes quantitatively

Once we have obtained an optimised model, we can explore the behaviour changes in the best individual that leads to maximising the number of episodes of the cart pole task. There are four position trackers we can visualise individually or together to explain the behaviours of the agent:

1. the position of the cart,
2. the velocity of the cart between episodes,
3. the angle of the pole,
4. the rotation rate of the pole.

To understand how behaviour changes with genotypes of varying fitness, we will plot the observations of the fittest genotypes at different number of epochs from our optimised model.

Algorithm 4 Get agent's observations

Input: agent
Output: observations of the agent at each epoch

```
environment.reset()
observations = []
done = False
while done == False do
    observations.append(environment.step(agent))
end while
return observations
```

With the results from algorithm 4, we can plot the observations per episode to observe how they change over time. Ideally, we would find patterns that show how the agent keeps the pole upright, or how it becomes unstable, leading to it falling before the episode 500 mark.

2.4 Examining position changes qualitatively

Using an algorithm similar to 4, we can visualise the actual movement of the cart pole as an agent we have defined is acting on it. This is through a function defined by the OpenAI framework [2]. We will save the frames of the fittest genotypes at different numbers of epochs from our optimised FMGA. The visuals obtained could provide information that numerical changes in behaviours cannot tell us. We will also be able to compare the visuals to quantitative patterns in behaviour to explain how they show a way to keep the pole stable or a failure.

3 Results

3.1 Optimal population size

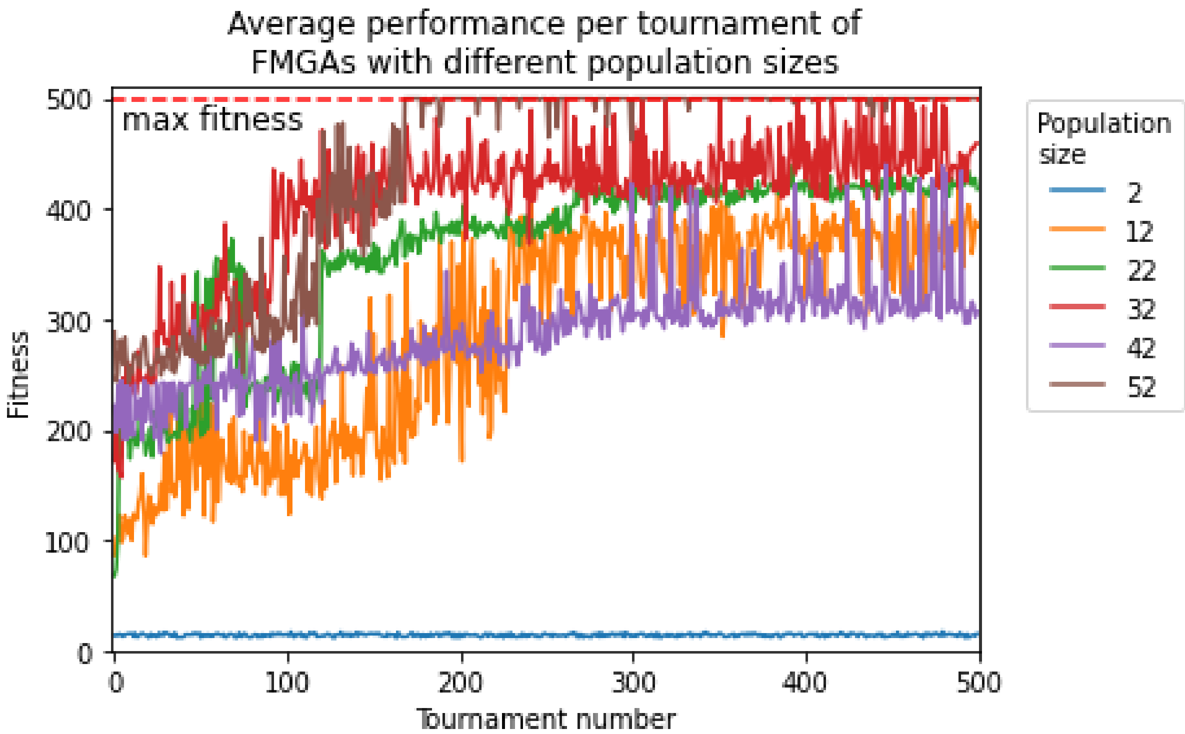

Figure 1: This figure plots the average fitness of FMGA models with varying population sizes, as it changes over the number of tournaments. Each line represents the average fitness of all individuals at that tournament for different population sizes. The red dotted line at the top represents the maximum fitness that can be reached by an FMGA model solving the cart pole task. We notice the largest population, 52, in brown, reaches the maximum fitness at 150 epochs.

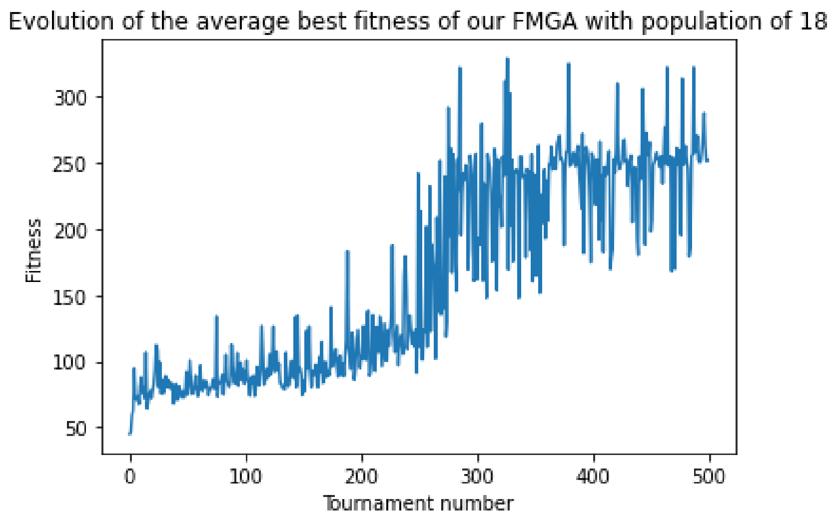

Figure 2: This figure plots the evolution of the average best fitness of an FMGA with a population of 18 individuals. We notice a spiky plateau forming from 300 tournaments. The spikes even seem to reach the maximum fitness, indicating that some individuals sometimes reach the best fitness. This is not a reliable enough model over 500 tournaments for us to guarantee an individual from that population will reach 500 episodes.

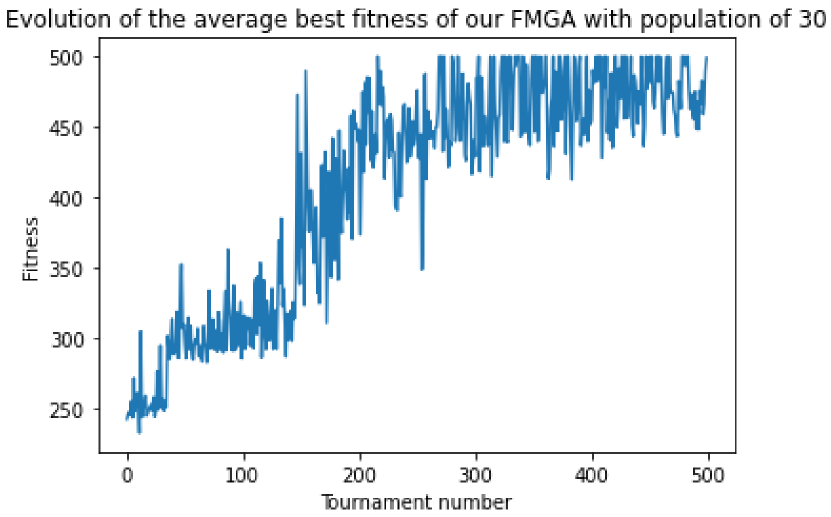

Figure 3: This figure plots the evolution of the average best fitness of an FMGA with a population of 30 individuals. We notice a spiky plateau forming from 250 tournaments. The maximum fitness is sometimes reached by the FMGA model on this plot.

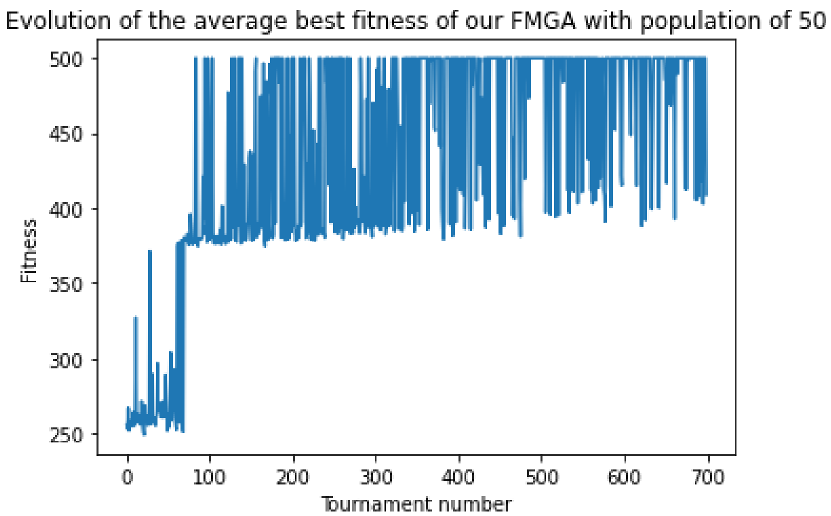

Figure 4: This figure plots the evolution of the average best fitness of an FMGA with a population of 50 individuals. We notice a spiky plateau forming from only 100 tournaments. The spiky plateau very often jumps to a fitness of 500.

We found that, over 500 epochs, the population size that performed best was, as expected, the largest with 52 individuals 1. From 180 epochs, it started plateauing at a fitness of 500, which no other population size did. Some population sizes never even reached the maximum fitness. Low population sizes will grow fast in fitness but lack the diversity to reach the maximum one. We do notice a correlation between the number of epochs and the population size such as in figures 2, 3 and 4. We notice that apart from the population of size 50 (see figure 4), the average best fitness of the FMGAs will start plateauing at a sub-optimal fitness. Therefore, we choose a population of size 50 for our optimal FMGA as it seems to be, in the population sizes we tested, the only one to eventually reach the maximum fitness in the cart pole task without plateauing at a sub-optimal fitness.

3.2 Optimal number of tournaments

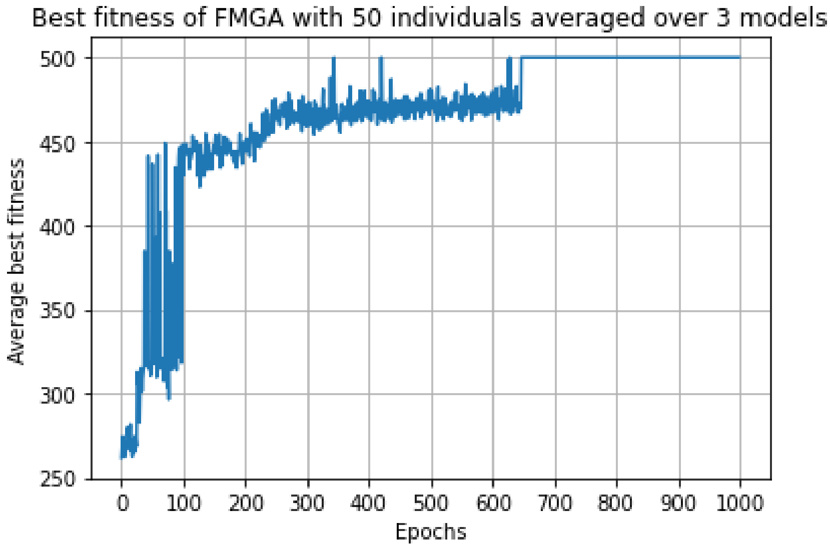

Figure 5: This figure plots the average best fitness of the FMGA model with a population size of 50 over the number of tournaments. We notice that from 650 epochs, the average best fitness of the model is of 500.

We found that our optimised model, found through our experimentation in the previous section, reaches the solution on average at around 650 epochs (see figure 5). From 0 to 50 epochs, there is a straight increase in fitness from 275 episodes to 325. Then there is a very spiky plateau for 50 epochs where the fitness jumps from 325 up to 450, we can attribute this to an increase in the search space of the individuals. Following this, there is a much less spiky plateau that, on average, increases up to a fitness of 475. The final plateau is constant at 500 after 650 epochs, where we can confidently say that the population will always have at least one individual with a fitness of 500.

3.3 Quantitative changes in observations

Supported by the experimentation to find the optimal FMGA model, we first generated an FMGA with a population of 50 over 1000 epochs, and returned the individuals with the maximum fitness at each epoch. We then put them through agents and made them run over the 500 episodes of the cart pole task. We plotted their observations as they change through the number of episodes in the following figures of this section.

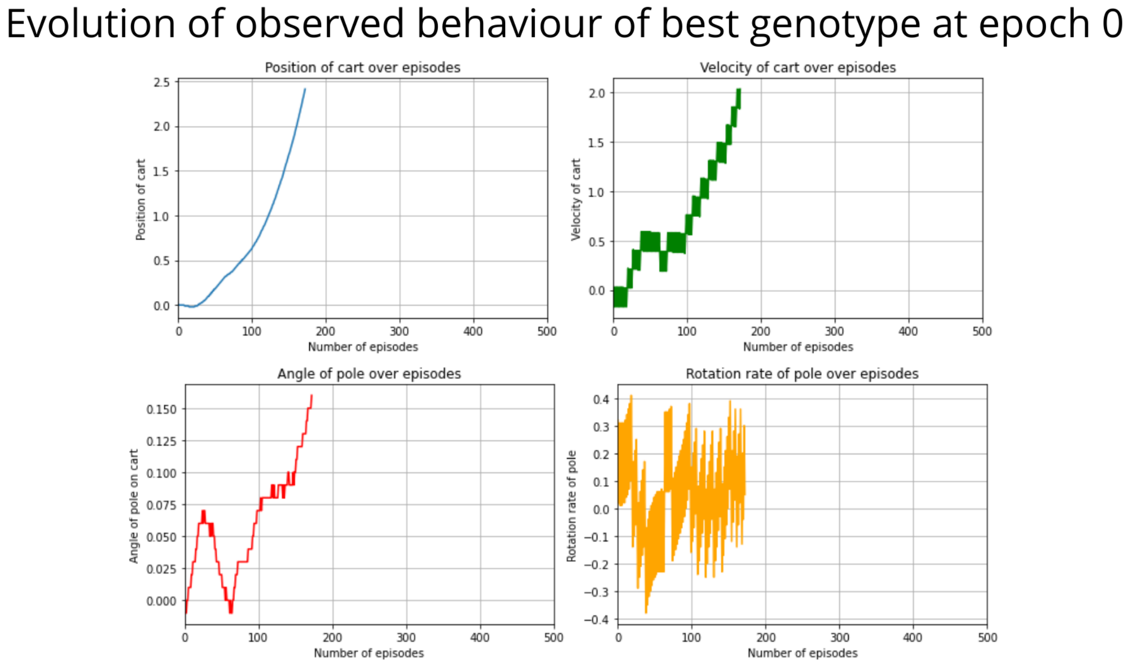

Figure 6: We ran our FMGA over 1000 epochs. At epoch 0, we took the genotype with the highest fitness and plotted its observations here. Where the lines stop is where the pole on the cart fell, therefore the agent stopped.

In figure 6, we plot the genotype at epoch 0 with the best fitness's observed behaviours. We notice the the position of the cart, the velocity and the angle all increase over the last 100 episodes. This shows the instability of the pole as it is pushed to one side. This is a very unstable cart, unsurprisingly since its genotype is random.

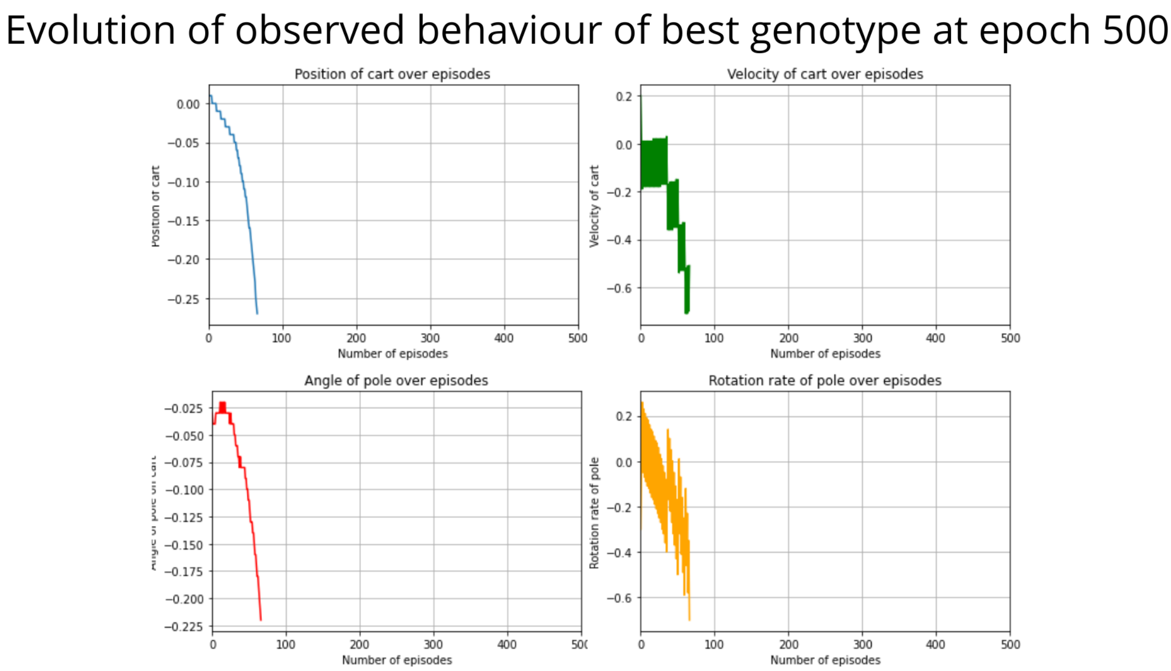

Figure 7: At epoch 500 of our optimised FMGA, we took the genotype with the highest fitness and plotted its observations here. Where the lines stop is where the pole on the cart fell, therefore the agent stopped.

At epoch 500, the best genotype's observed behaviours (figure 7) surprisingly perform worse than the best genotype at epoch 0 (figure 6). It is very unstable at it simply pushes the cart to one side at an even higher velocity.

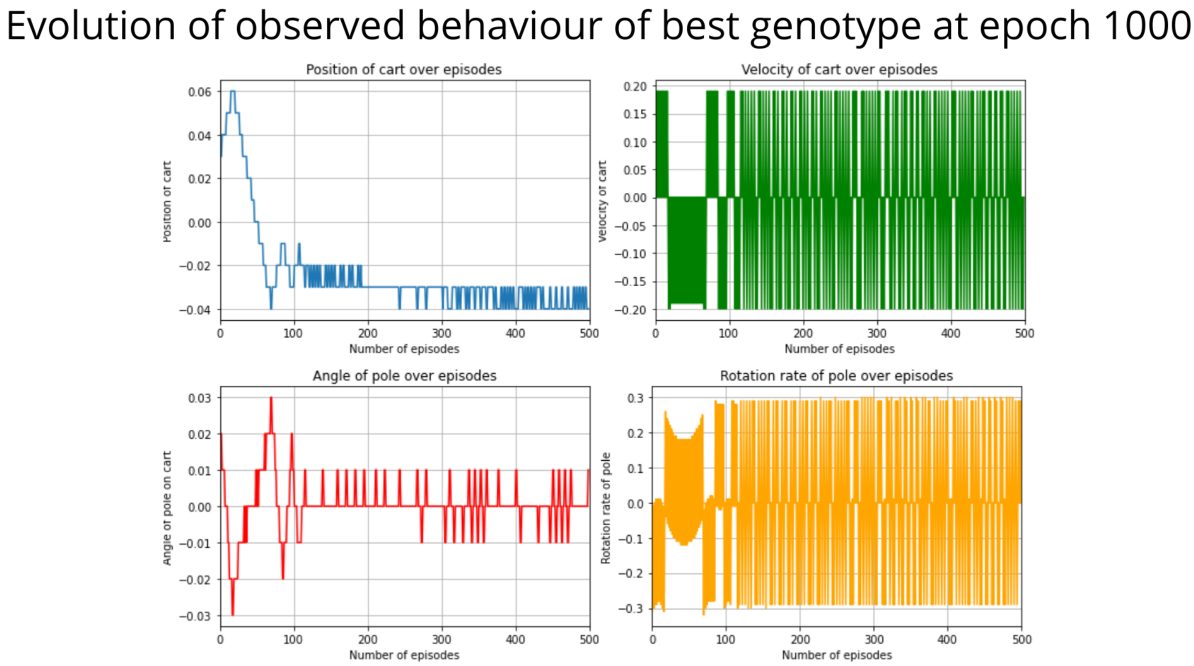

Figure 8: We ran our FMGA over 1000 epochs. At epoch 1000, we took the genotype with the highest fitness and plotted its observations here. This agent ran held up the pole for 500 episodes.

In figure 8, we plotted the change in behaviours of the best genotype at epoch 1000. We find that it goes up to the last episode, and that its behaviours differ very much from the other genotypes who's pole fell over (see figures 6 and 7). Mainly, each metric jumps in values for the first 80 epochs, and then finds a pattern that it mostly holds towards the end. The position of the cart for example varies from 0.06 to -0.04, but later on only varies from -0.03 to -0.04, indicating that it is stable as it does not move a lot. The velocity and the rotation rate consistently rapidly jump from high to low values. Compared to 6 and 7, there are patterns and no convergence towards a value in each behaviour, which is what keeps the pole upright.

We had found in our optimisation results that more epochs meant finding more fit individuals. The comparison we made of the behaviours of the best genotypes at different numbers of epochs proves this, as our last genotype 8 was the one that performed best. This is also subject to the randomness of the population as we have found in earlier results (see figure 5) that at epoch 500, there should be on average an individual that can keep the pole stable for 500 episodes. Overall, our main finding here is the emergence of patterns from the best fit genotypes to keep the pole stable, and the lack thereof prevents the pole to stay upright for 500 episodes.

3.4 Qualitative changes in observations

Evaluating the behaviours qualitatively can give us information about how a genotype acts on the cart pole task that the quantitative evaluations may not. Therefore, we extracted the animations of 3 agents with different genotypes, taken from a trained FMGA model's best genotypes at each epochs, at epoch 0, 500 and 1000. The videos of the animations are available here [1]. In this section, we will examine the animations.

The first agent's genotype was extracted from 0 epochs. We find, when watching the animation [1], that the cart gets pushed to the left for most of the animation. Staying stable for 231 episodes indicates that it did also get pushed from the right side, as we notice that the pole stays straight for most of the animation. For a randomly generated genotype that did not go through any tournaments, getting through almost 50% of the episodes is impressive. Still, we notice that it exhibits some random behaviour as we would expect for it to try and keep the cart at the center.

The second agent's genotype was extracted when the FMGA was trained over 500 epochs. We already notice some signs of evolution as now the cart is pushed to try and stay in the middle for the first 80 episodes. The pole stays mostly straight but the cart still ends up getting pushed to the left.

Therefore, we do notice changes in the behaviour of the agent that indicate some evolution happened.

The third agent's genotype was extracted when the FMGA was trained over 1000 epochs. We notice now that the agent keeps the cart mostly at the center, which is an evolution trend we had noticed appearing already at 500 epochs. This agent is able to keep the pole up for 500 episodes, effectively reaching a 100% accuracy. We even notice, as in the quantitative examination, that it exhibits patterns in its pushing, as we even see the pole making rapid small movements from left to right. We therefore notice that the genotypes have evolved to produce repetitive patterns in their pushing to keep the pole standing.

4 Discussion

Our initial hypothesis for the optimisation of the FMGA model, population of 50 with 1000 tournaments, turned out to be true, but the exploration that found it turned out more revealing as we found out about the common fitness evolution patterns. We found that a small population, limiting diversity, is a hard limit on the fitness of the FMGA model that will see its fitness plateau much before the solution, even if it grows rapidly at the start. On the contrary, large populations with a very big diversity will take longer to train and have a slow growing fitness, but are less likely to settle for a sub-optimal solution. With more computational power available, we would have explored the best individuals from large populations and compare their solutions proposed and their behaviours to the ones that emerged from smaller populations. We could find that a more diverse population, having its best individuals find a solution to the max fitness, may be less robust that the ones that took many tournaments to evolve.

The genotype we found with a fitness of 500 in the FMGA of population 50, trained over 1000 epochs, had very consistent behaviours 8. It seemed to show that consistency in behavioural patterns is the way to keep the pole stable on the cart. Comparatively, the genotypes found at 0 and 500 epochs had behaviours that did not swing but went straight for bigger or smaller values with increasing velocities (see figures 6 and 7). This meant that they would fail in the first 200 episodes, not being able to keep the pole balanced, which we also observed in our qualitative examination. For now, our report indicates that more epochs for an FMGA means more robust genotypes proposed.

Since we used FMGAs, we could have explored the effects of its varying hyper-parameters when the population is evolving towards a solution. For example, the mutation rate in FMGAs is partially responsible for the diversity of the population, increasing the amount of search space it covers. We might find, when increasing the mutation rate, that smaller populations do not fall as easily into sub-optimal solutions, and the spiky plateaus they visually form in their fitness when plotted would appear at higher values of fitness. This is one example, but there are many hyper-parameters of the FMGA model that could be explored to change how populations would evolve to sole the cart pole task.

Our FMGA implementation did produce an important drawback: the variance in fitness. We know from our algorithm's implementation that the fitness of the population cannot decrease from one tournament to the other 1. However, we noticed that the best genotype at each tournament, when evaluated on one population from an FMGA, did decrease in some tournaments. This can even be noticed in the average best fitness plotting 4. This is due to the randomness in the agent's actions on the genotype as the observations may differ from one scene of episodes to another, even if the agents are the same. This means that the same agent acting on the cart pole task in two separate events, may hold the cart pole up for a different number of episodes, and therefore have a different fitness. To overcome this, we could have changed the implementation of our FMGA algorithm to get the average fitness of the loser's replacement's genotype before deciding if we are using it to replace the loser. An averaged out fitness of the possibly new individuals in the FMGA population at each tournament could have prevented our FMGA model's population fitness' random decreases at each tournament.

5 Conclusion

This report explored the optimisation of the FMGA model to solve the cart pole task, and examined the quantitative and qualitative results that it produced. We were able to optimise our model, with a population of 50 trained over a minimum of 1000 epochs, to produce, on average, at least one genotype that solves the cart pole task by keeping the pole upright for 500 episodes. We found that consistent patterns in the observed behaviours of our best agent is what kept it stable, in comparison with the lack of patterns in the observed behaviours of agents with a lower fitness. We were also able to show an improvement in observed behaviours of stability as the number of epochs used to train our FMGA model increased. In all, further exploration in this area could look at bigger populations in FMGA models, if it is computationally possible, as to find if early 'random-finding' is as robust as finding the solution through evolution with lots of tournaments. Furthermore, the parameters of FMGAs could have been experimented with to find if allowing more diversity in smaller populations would prevent them from finding sub-optimal solutions too quickly.

References

[1] Fmga best genotype at different epochs' behaviour on the openai cart pole task result youtube playlist. Available at https://youtube.com/playlist?list=PLG2p0BQ1eHPP5ZPDcx3zI80jUHque9SaI.

[2] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

[3] Inman Harvey. The microbial genetic algorithm. In *ECAL*, 2009.

[4] Melanie Mitchell, Stephanie Forrest, and John H. Holland. The royal road for genetic algorithms: Fitness landscapes and ga performance. 1991.

[5] David E. Moriarty and Risto Mikkulainen. *Machine Learning*, 22(1/2/3):11–32, 1996.