



University of Nottingham

COMP3004: Designing Intelligent Agents

**“How can frogs evolve to improve  
survival and reproduction”**

Word count: 3934

By Scott Followell

Psysf8 – 20451992

# Contents

1	Introduction.....	4
2	Subject Area Research .....	4
2.1	Predator & Prey.....	4
2.1.1	Green Frog .....	4
2.1.2	Eastern Ribbon Snake.....	4
2.2	Natural Environment.....	4
2.3	Genetic Algorithm (GA) Research .....	5
2.4	Population-Based Consensus Behavioural Evolution (PBCBE) Research.....	5
3	Agents .....	6
3.1	AgentSim Class.....	6
3.2	Frog Agent .....	6
3.3	Snake Agent.....	7
4	Genetic Algorithm .....	8
4.1	Genetic Representation .....	8
4.2	Fitness .....	8
4.3	Selection.....	9
4.3.1	Steady State.....	9
4.3.2	Tournament.....	9
4.3.3	Genetic Algorithm with Simulated Annealing .....	9
4.4	Crossover .....	10
4.5	Mutation.....	10
5	Population-Based Consensus Behavioural Evolution (PBCBE) Algorithm.....	11
5.1	Algorithm Overview .....	11
5.2	Simulation Behaviour Analysis .....	11
5.3	Population Consensus Analysis .....	13
5.3.1	Behaviour Gene Direction .....	13
5.3.2	Trend Override.....	14
5.4	Gene Evolution .....	15
6	Evaluation & Data Analysis.....	15

6.1	Hypothesis.....	15
6.2	Genetic Algorithm Performance Analysis .....	15
6.2.1	Steady State.....	16
6.2.2	Tournament.....	19
6.2.3	Simulated Annealing.....	22
6.2.4	Cross GA Comparison .....	25
6.3	Population-Based Consensus Behavioural Evolution (PBCBE) Performance Analysis	
	27	
6.3.1	Final States.....	27
6.3.2	Gene Evolution .....	29
6.3.3	Behaviour Count .....	32
6.3.4	Population Confidence Consensus.....	35
7	Conclusion .....	37
7.1	Challenges Faced During Implementation.....	37
7.2	Successes & Limitations .....	37
7.3	Future Prospects.....	38
8	Appendices.....	39
8.1	Appendix A: Environment Design, Constraints and Rules .....	39
8.1.1	Environment aspects .....	39
8.2	Appendix B: Genetic Algorithm Overview .....	41
8.3	Appendix C: Table of Figures.....	41
9	References.....	43

# 1 Introduction

This study explores how a Green Frog can evolve to improve its survival and reproduction rate by comparing two evolutionary approaches. It focuses upon developing a simulation representing the vision and cognitive learning aspects of a frog's environment. Two approaches were implemented, firstly a Genetic Algorithm with various selection methods such as steady state, tournament selection, and genetic algorithm with simulated annealing. The second and novel approach was Population-Based Consensus Behaviour Evolution.

## 2 Subject Area Research

### 2.1 Predator & Prey

#### 2.1.1 Green Frog

Lithobates Clamitans, or Green Frog eats any mouth sized animal from insects to spiders and is found in a variety of habitats such as marshes, shallow freshwater ponds, lakes, swamps, and streams [1]. They have 320-degree field of vision with a focal vision of 90 degrees as its eyes are on the sides of its head [2].

Green frogs are very territorial and will reside within an inhabitable environment which they do not leave. They are hunted by snakes, herons, and crows, so to avoid and deter predators, they release a chemical and use natural colour camouflage to hide in grass.

#### 2.1.2 Eastern Ribbon Snake

Commonly known as a Garter snake, it has a smaller focal vision of 90 degrees and a total 180-degree field of vision given its forward facing lateral eye placement. Commonly found in habitats including forests, woodlands, grasslands and lawns. It's never far from water, such as an adjacent marsh, stream or pond as amphibians are a large part of their diet [3].

They primarily eat frogs and tadpoles, but also almost anything it can overpower. To socialise with other garter snakes it releases a pheromone, which also aids in reproduction.

### 2.2 Natural Environment

Given that both the Green Frog and Eastern Ribbon Snake are typically found in areas of wetland, this is why I chose this subject and environment design.

Please refer to [Appendix: Environment Design, Constraints and Rules](#) to understand how these were implemented into my simulation.

## 2.3 Genetic Algorithm (GA) Research

Using AI Methods from year 2 (University of Nottingham), coupled with the paper by R. L. Haupt [4] which furthered my understanding of the subject and enabled the implementation of my ray detection gene values.

## 2.4 Population-Based Consensus Behavioural Evolution (PBCBE) Research

Despite a promising start the poor performance of the Genetic Algorithm (please refer to [Challenges Faced During Implementation](#)) struggled to learn the environment. Therefore, I attempted a new evolutionary method through the use of Population Decision Trends (after reading a paper on Multi-Objective Population Based Training [5]) however it required further development before becoming Population-Based Consensus Behavioural Evolution. The idea was further supported after reading “evolving consensus sequence with a genetic algorithm” by C. Shyu and J. A. Foster on page 1110 [6].

## 3 Agents

### 3.1 AgentSim Class

Parent class for FrogSim and SnakeSim classes, it contains the data structure, how the agent is constructed and its rays along with how it processes what the rays of vision intersect, at what distance and how much the total turn should be depending on the gene strength. The agents are purely reactionary to what they can detect. The `wheel_base` and basic logic was inspired by the light avoiding robot during labs.

The functions and their uses:

`_line_intersection_point()` uses the parametric line intersection formula [7] to allow agents to detect walls.

`_projection_fraction()` determines when the rays of vision intersect with food, snakes, grass and water using the vector projection [8].

`_circle_line_collision()` used to detect when an agent's circular body intersects [9] with a wall, then it should die.

`calculate_turn_response()` is how the gene values are used to determine the strength of the turn based on what it detected. If the gene value is positive, then it will go towards the detected item.

### 3.2 Frog Agent

Building upon the AgentSim class, the frog now has various tracking variables for data analysis such as `final_state`, `fitness` and `successful_behaviours`. Frogs also have a starvation timer that is controlled by `STARVATION_THRESHOLD` which determines the simulation ticks required before a frog dies of starvation due to not eating flies. A frog can be seen with its 320 degree vision in figure 1.

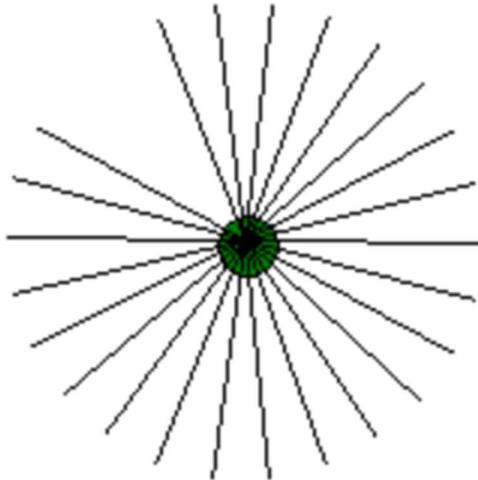


Figure 1: Frog

Frogs have three brain states:

1. Normal – can detect food, snakes, grass and walls but not water
2. Starving – can detect food, snakes and walls but not water or grass. Grass due to it starving and being forced out of safety, controlled by  
`STARVATION_IGNORE_GRASS_THRESHOLD`
3. Reproduction – once a frog has eaten enough flies it can detect water, walls and snakes but can no longer detect grass and food as it does not require either. In this state it cannot starve to death as it has eaten enough food.

### 3.3 Snake Agent

In both the PBCBE and GA the snake is the same. The key differences are the smaller vision of 180 degrees representing their more frontal based vision, but it still uses the same number of rays of vision at 24. The snake can detect frogs, grass and walls. I chose to remove its detection of water, as it did not require the detection of predators given its predators were not implemented into the simulation. The reasoning to remove water detection was to promote the hunting of the snakes, therefore the snake agents have a gene length of five but only three active genes. A snake from the simulation can be seen in figure 2.

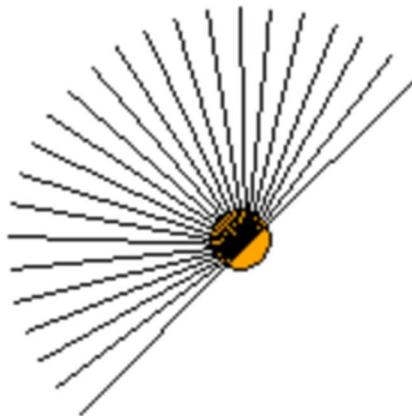


Figure 2: Snake

## 4 Genetic Algorithm

### 4.1 Genetic Representation

The genes are represented as floating point numbers in the range of -1.0 to 1.0 for each of the five detectable environmental elements. Negative values mean it will avoid the element; positive means it will be drawn towards it. The difference from zero to the ends of the range is the strength of the attraction or avoidance increasing and decreasing the turn of agents. Figure 3 shows some potential gene values and what they relate to.

Wall	Food	Snake	Grass	Water
<hr/>				
0.89	-0.90	0.99	0.96	-0.89

Figure 3: Gene Representation

### 4.2 Fitness

A fitness function for GAs help to guide individuals towards local and global optima, in this scenario optimal behavioural combinations. Rewards for behaviours are done in four phases:

- Phase 1 – rewarding time alive
- Phase 2 – eating flies and reaching reproduction threshold

- Phase 3 – proximity-based rewards for water pools after reaching the required fly threshold
- Phase 4 – successfully reaching the water pool and reproducing

Penalties are also applied if a frog died due to a snake, wall, or starvation. I thought the phased based learning would allow the population to learn the environment and different interactions through gene values.

## 4.3 Selection

Initially the entire population is generated with random gene values from -1.0-1.0 through selection which uses the fitness of each individual to decide the most successful and have their genes passed to the next generation.

### 4.3.1 Steady State

Steady state indicates a percentage of the top population performers are retained into the next generation, known as elites, and is determined by `REPLACEMENT_RATE` in my code. The remaining individuals are replaced in the new generation by mutated copies of random elites. Steady state aims to exploit the gene space quickly but often finds a local optimum.

### 4.3.2 Tournament

Tournament selection evaluates the fitness of population subsets randomly picked until the subset reaches `TOURNAMENT_SIZE`. The individual with the best fitness is then chosen and placed into the next generation. This aims to balance exploration and exploitation as moderate fitness individuals also have a chance of their gene being passed into the next generation. However, in my code, the top 10% fitness individuals are retained and immediately passed into the next generation, this is known as elitism to ensure some good performers make it through.

### 4.3.3 Genetic Algorithm with Simulated Annealing

Simulated annealing is a heuristic search technique using a temperature factor which cools over time, the exponential cooling schedule is used. The temperature controls expiration and exploitation.

## 4.4 Crossover

Crossover is where two parents are selected, and their gene data is spliced together to create a child gene which potentially provides a better gene combination. This was not implemented as it broke up successful behavioural combinations.

## 4.5 Mutation

Mutation is the random introduction of new chromosome values and gene combinations into the population based upon a `MUTATION_RATE`. As seen below in figure 4.

```
child[i] += random.uniform(-RANDOM_UNIFORM_NEG, RANDOM_UNIFORM_POS)
child[i] = max(min(child[i], 1.0), -1.0)
```

Figure 4: GA Mutation

## 5 Population-Based Consensus Behavioural Evolution (PBCBE) Algorithm

### 5.1 Algorithm Overview

The PBCBE algorithm manages a gene level consensus and evolution for the frog agents. Each gene has an expected successful behaviour tied to it such as avoiding snakes. The behaviour instances are then tied to the gene sign and to the general trend it's contributing to. If the population has enough observed behaviours and agrees with significant confidence, then that positive or negative direction is used to evolve that gene. This is applied to all genes which subsequently builds inter-gene behaviours as the population consensus grows and stabilises.

This uses the same gene representation format as the GA please refer to [Genetic Representation](#).

### 5.2 Simulation Behaviour Analysis

A behavioural success for this simulation are gene specific and are defined as:

- Wall – once detecting a wall and then no longer detecting a wall equates to successfully avoiding it
- Food – upon detecting a fly a frog should collide with it to ‘eat it’
- Snake – same logic for the wall but for a snake
- Grass – once detecting grass they are successful if they enter the grass
- Water – after collecting enough flies the frogs can detect water and upon detection they should enter the water to be classed as successful

Figure 5 shows the behaviour analysis data collection that occurred during each generation’s simulation in the environment. It is important to note that for the decision nodes “Frog can no longer detect wall?” and “Frog can no longer detect snake” when following the “no” path this is due to the frog dying upon collision with the wall or snake. The final output oval from the decision “All frogs dead or successful: Yes” leads into figure 6 in section 5.3.1.



Figure 5: PBCBE Behaviour Analysis

## 5.3 Population Consensus Analysis

### 5.3.1 Behaviour Gene Direction

With the individual behaviours have been recorded for each gene we need to gauge if the entire population has:

- Collected sufficient behaviours to begin analysis
- We do not want to change a gene trend if there is not sufficient supporting evidence
- Reached a consensus confidence above the `CONSENSUS_THRESHOLD`

This establishes a generation trend for the gene which can be directly used to evolve the genes for this generation and help in directing the future generations.

This builds upon the key principle that even a poorly performing individual may show successful behaviours, this was not captured within the GA harming its performance within this environment. If a gene's analysis resulted in insufficient behavioural data or no confidence for a consecutive number of generations, determined by `NO_CONSENSUS_RESET_THRESHOLD` and `INSUFFICIENT_DATA_RESET_THRESHOLD`, then it will randomise that gene for the entire population to start in a new location in the solution space.

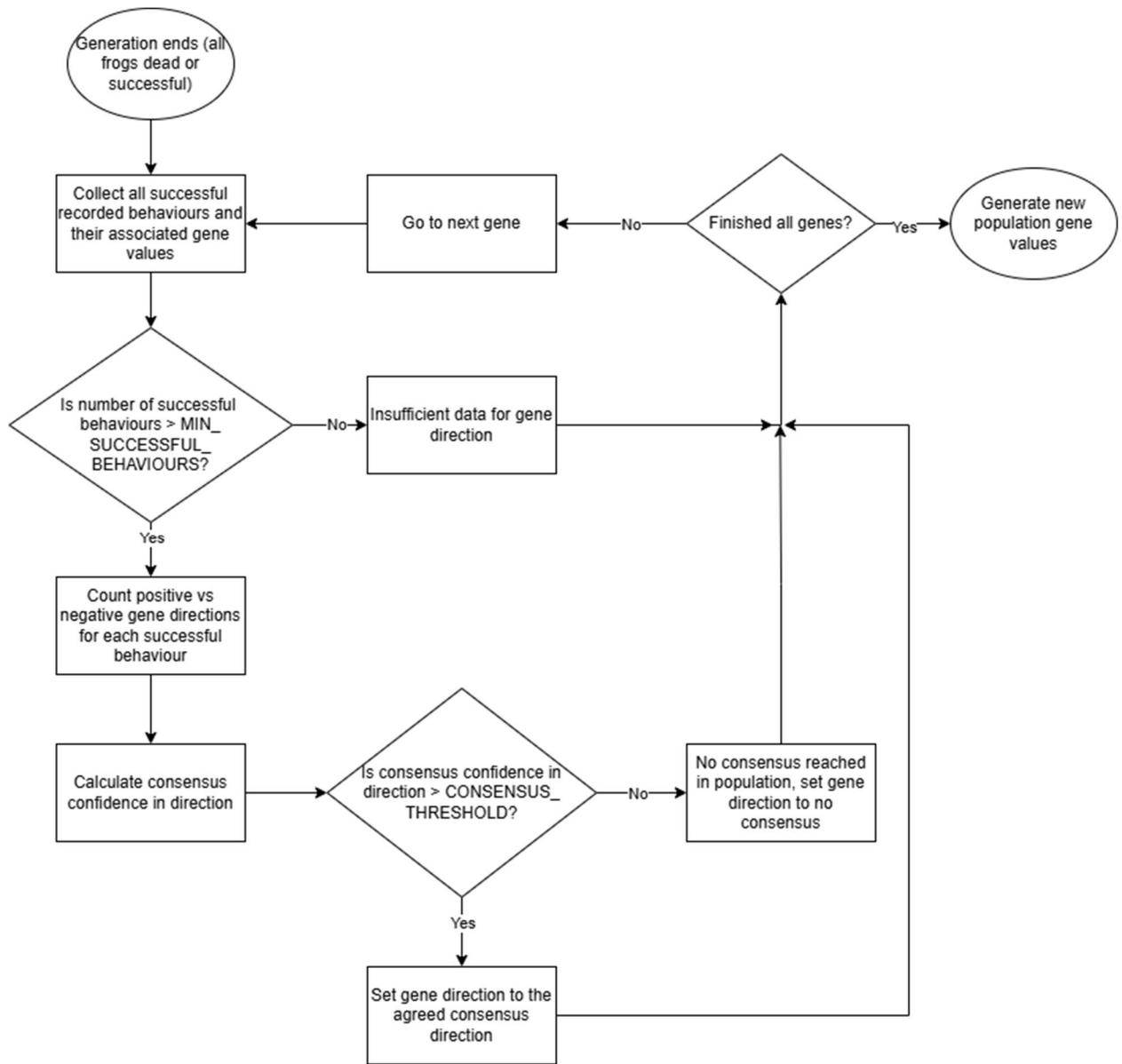


Figure 6: PBCBE Gene Direction

### 5.3.2 Trend Override

This is designed to prevent premature or incorrect behavioural convergence due to early noise or consensus historical momentum. If a new generation does not match the previous number of generation trends (**CONFIDENCE\_WINDOW**), it will require significant evidence of behaviours over a confidence threshold set by **DIRECTION\_CHANGE\_THRESHOLD**, to clear the confidence window of previous historical trends. This will prevent previous historical momentum from affecting the potential change in direction.

## 5.4 Gene Evolution

There are several components determining how the genes evolved. Firstly, the `base_step` is determined by the `EVOLUTION_STEP_SIZE` which is typically set to a low value due to other multiplicative formula used to calculate the `total_adjustment` for a gene. As seen in figure 7 below.

To improve the convergence for generations with high confidence levels (+80%) it will introduce a confidence multiplier. Additionally, there is a further multiplier depending on the number of observed successful behaviours for a gene, further improving convergence.

```
total_adjustment = base_step * confidence_multiplier * sample_multiplier
```

Figure 7: PBCBE Gene Evolution

# 6 Evaluation & Data Analysis

## 6.1 Hypothesis

For both GA and PBCBE algorithms, the frog agents will learn the expected successful behaviours as outlined in [Simulation Behaviour Analysis](#), specifically the essential genes of avoiding walls and snakes while being attracted to food. However, due to the GA's tendency to heavily exploit high fitness individuals, it is anticipated to reach a local optimum and plateau in performance, especially while developing more complex behaviours such as the water gene. In contrast, PBCBE is expected to outperform the GA in the likelihood of achieving global optima due to its ability to retain successful behavioural patterns which may be lost in low fitness individuals.

## 6.2 Genetic Algorithm Performance Analysis

All simulations for the three variations of GA and the PBCBE were run using the same core parameters, the only variation is the data held below `STARVATION_IGNORE_GRASS_THRESHOLD` in their respective json files, as per figure 8.

```

1   {
2     "MAX_GENERATIONS": 100,
3     "SELECTION_METHOD": "GA-SA",
4     "NUM_RAYS": 24,
5     "FROG_POP": 20,
6     "SNAKE_COUNT": 2,
7     "GENOME_SIZE": 5,
8     "LIFESPAN": 1200,
9     "NUM_DOTS": 100,
10    "DOT_RADIUS": 7,
11    "CANVAS_WIDTH": 800,
12    "CANVAS_HEIGHT": 600,
13    "NUM_GRASS_PATCHES": 6,
14    "GRASS_RADIUS": 60,
15    "NUM_WATER_POOLS": 3,
16    "WATER_RADIUS": 40,
17    "FLIES_NEEDED": 5,
18    "STARVATION_THRESHOLD": 500,
19    "STARVATION_IGNORE_GRASS_THRESHOLD": 250,
20    "RANDOM_UNIFORM_NEG": 0.3,
21    "RANDOM_UNIFORM_POS": 0.3,
22    "MUTATION_RATE": 0.1,
23    "INITIAL_TEMPERATURE": 1.0,
24    "TEMPERATURE_DECAY": 0.98
25  }

```

Figure 8: JSON Parameters Example

### 6.2.1 Steady State

A final state is the state a frog was in at the end of the generation which is defined by all frogs dead or successful. Figure 9 shows that across the 100 generations the most likely cause of death was to walls, followed by starvation with a couple of outliers such as generation 6 where majority of frogs were eaten by snakes. However, across the 100 generations being eaten by a snake was scarcely the primary cause of death. Most of the successful or alive frogs occurred during the generation range of 40 to 68.

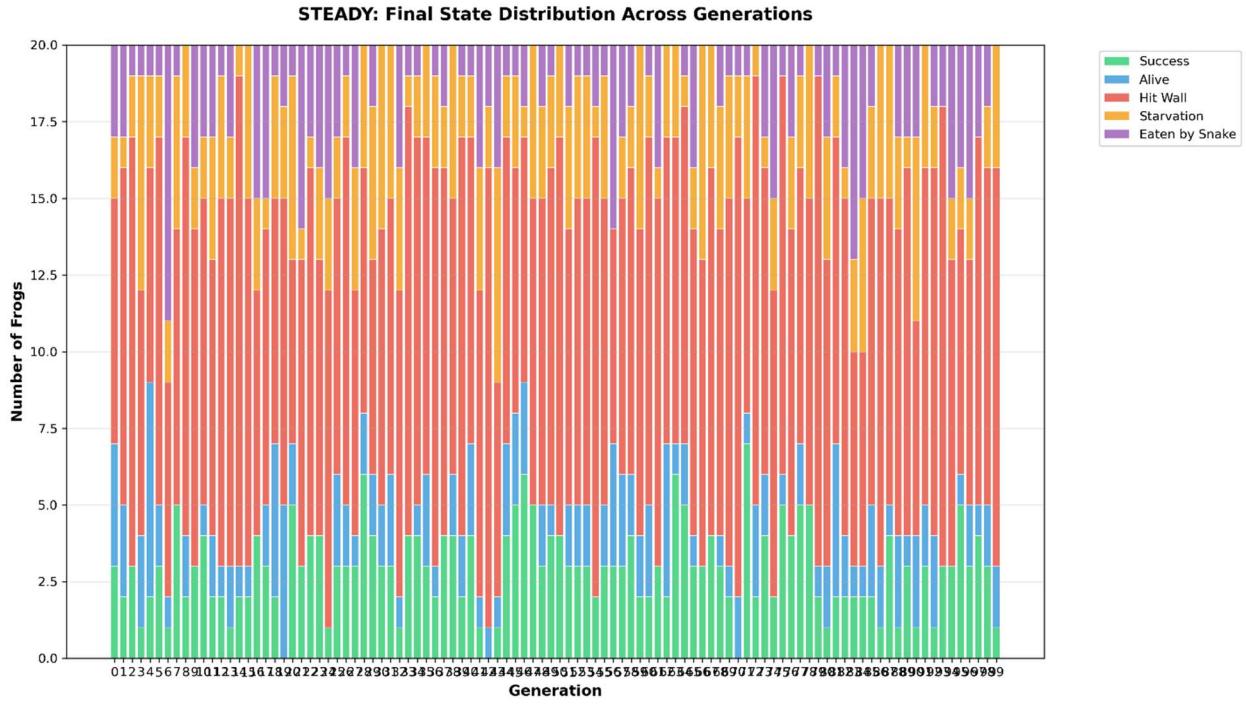


Figure 9: Steady State Final States

With the walls being the most common final state, figure 10 shows how the gene evolved. The black line represents the population average for the wall gene and the red lines are the gene values for each individual frog within the generation.

The graph is very noisy and shows no convergence within the population as to what the gene should be. The population average never leaves the range -0.25-0.25, and on the 100<sup>th</sup> generation marked by the red dots the population's individuals are still very spread throughout the gene space. The other genes also show the same distribution, please see figure 11. To see further individual gene patterns, please refer to /Plots in my code file.

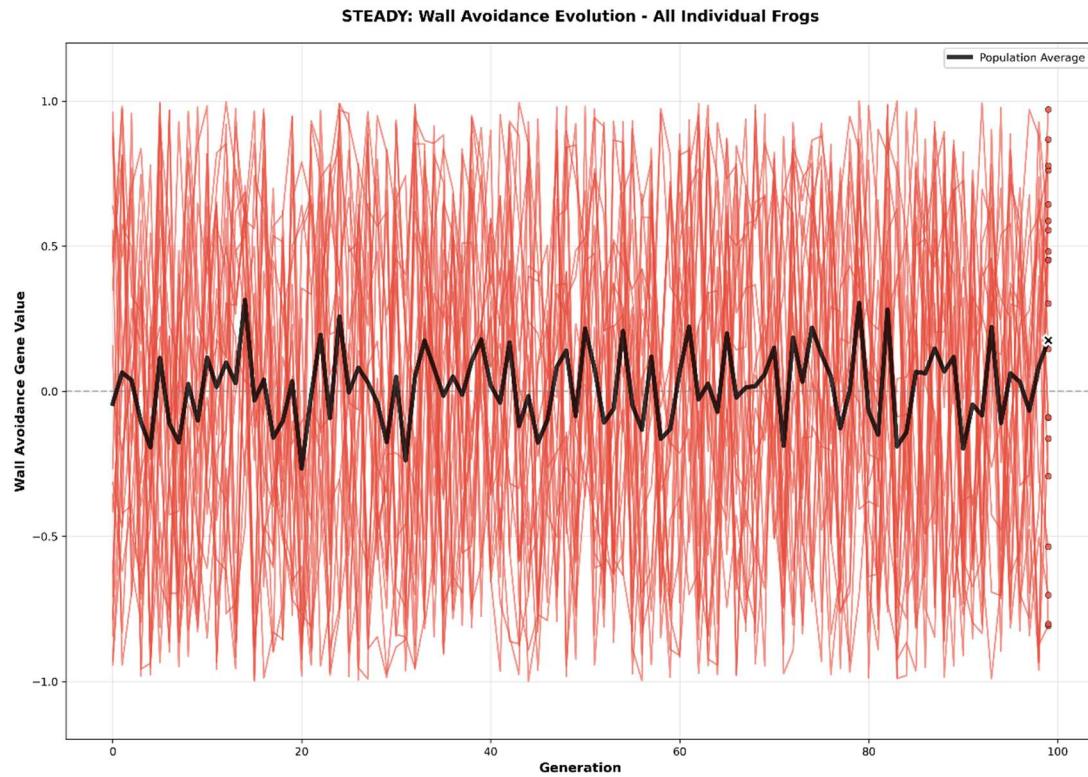


Figure 10: Steady State Wall Gene

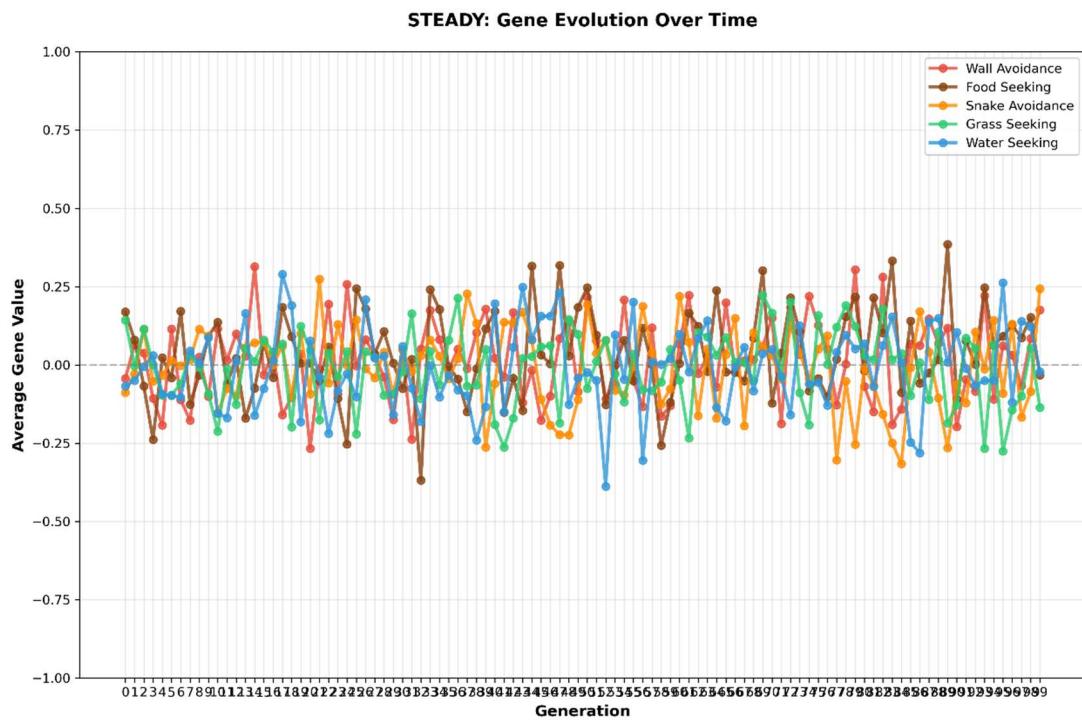


Figure 11: Steady State Gene Evolution

Figure 12 below shows one reason for the lack of gene convergence with a large deviation between the best and average population fitness and worst performing individuals. The fitness average is consistently around 0 meaning the population is barely surviving or dying quickly. This large fitness gap proves there is poor knowledge transfer and high volatility. The volatility is likely due to the random positions for all agents and environment features except walls from generation to generation which prevents them from learning a pattern as they are unable to link behaviours together.

The worst can partially be explained by the random location initialisation which could have them placed in an area lacking flies. However, the majority are at approximately -700 meaning they collected little or no flies and died at a wall.

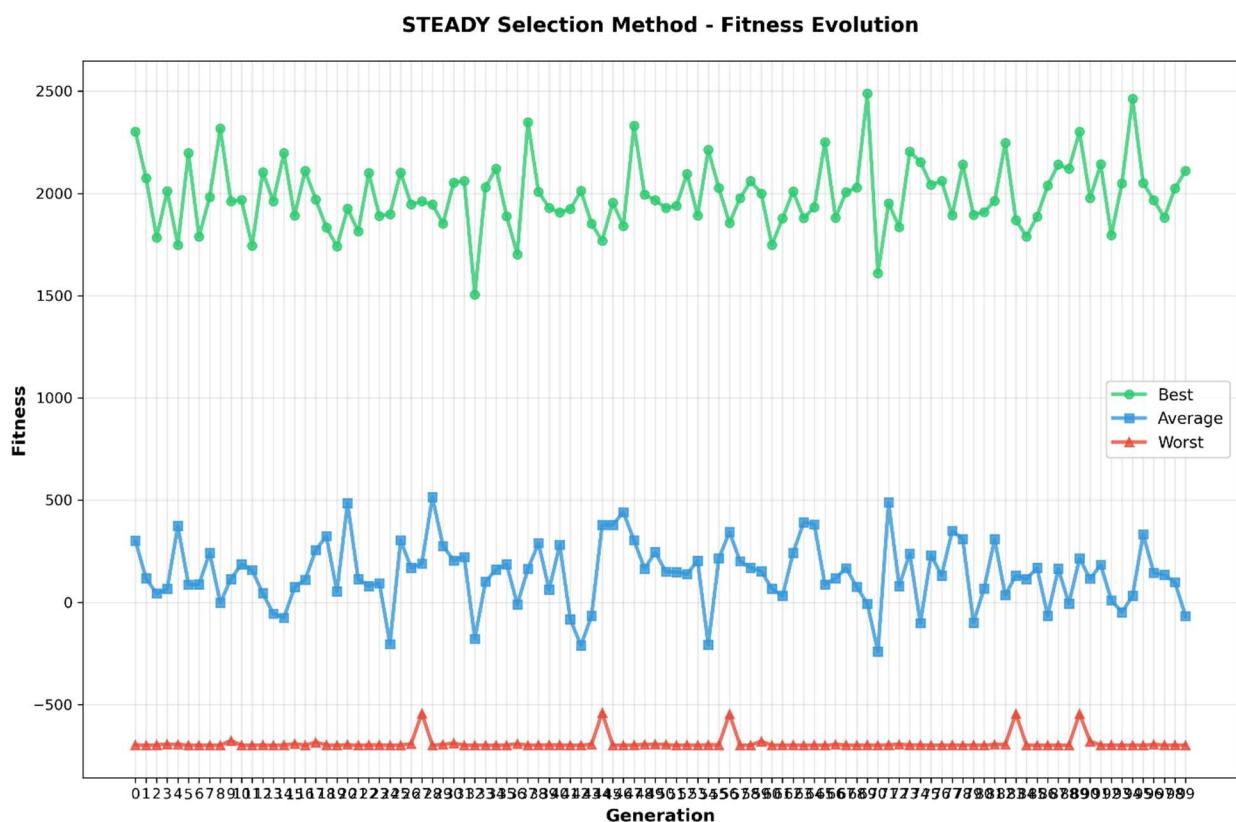


Figure 12: Steady State Fitness

### 6.2.2 Tournament

Tournament selection was set with a tour size of 6 and run for 100 generations. Similarly to steady state, it had a very poor performance, as seen in figure 13. Few successful or alive finishes and a high rate of deaths to walls. Starvation death rate was higher than steady state.

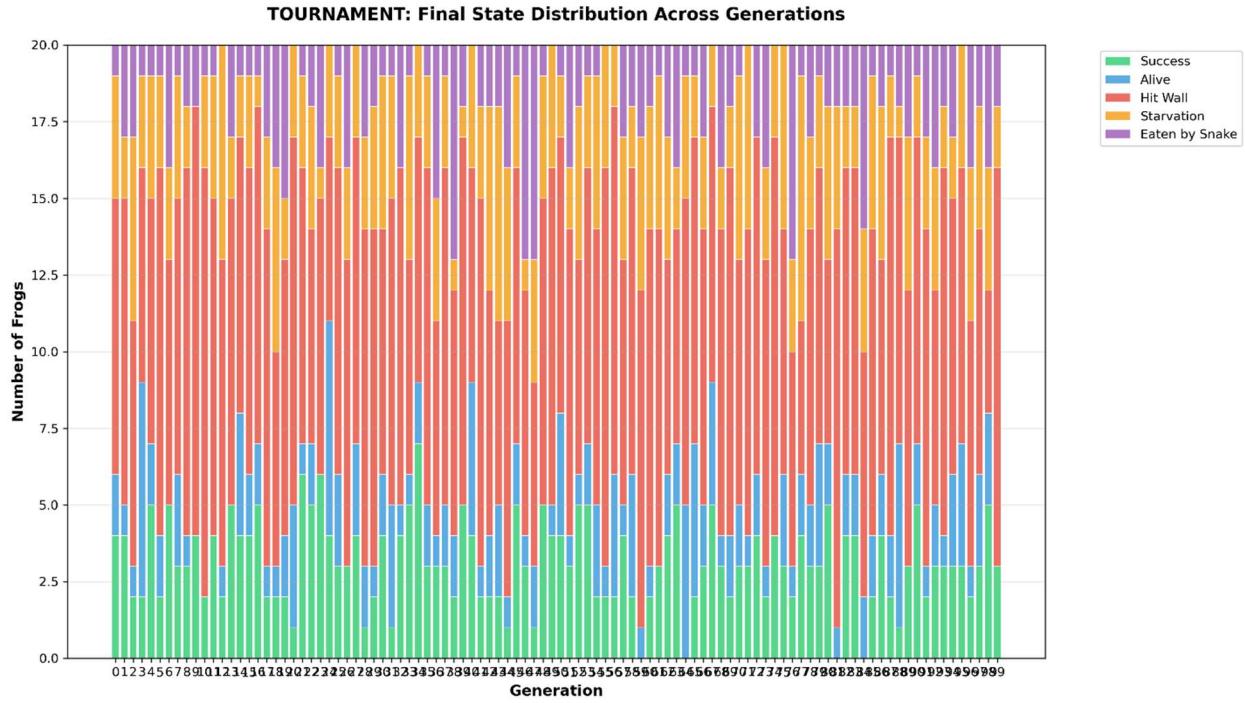


Figure 13: Tournament Final States

High starvation is primarily due to the ineffective evolution of the food gene, per figure 14, with similar volatility and lack of convergence as the steady state wall gene. Again, this is consistent across all of its genes as seen in figure 15.

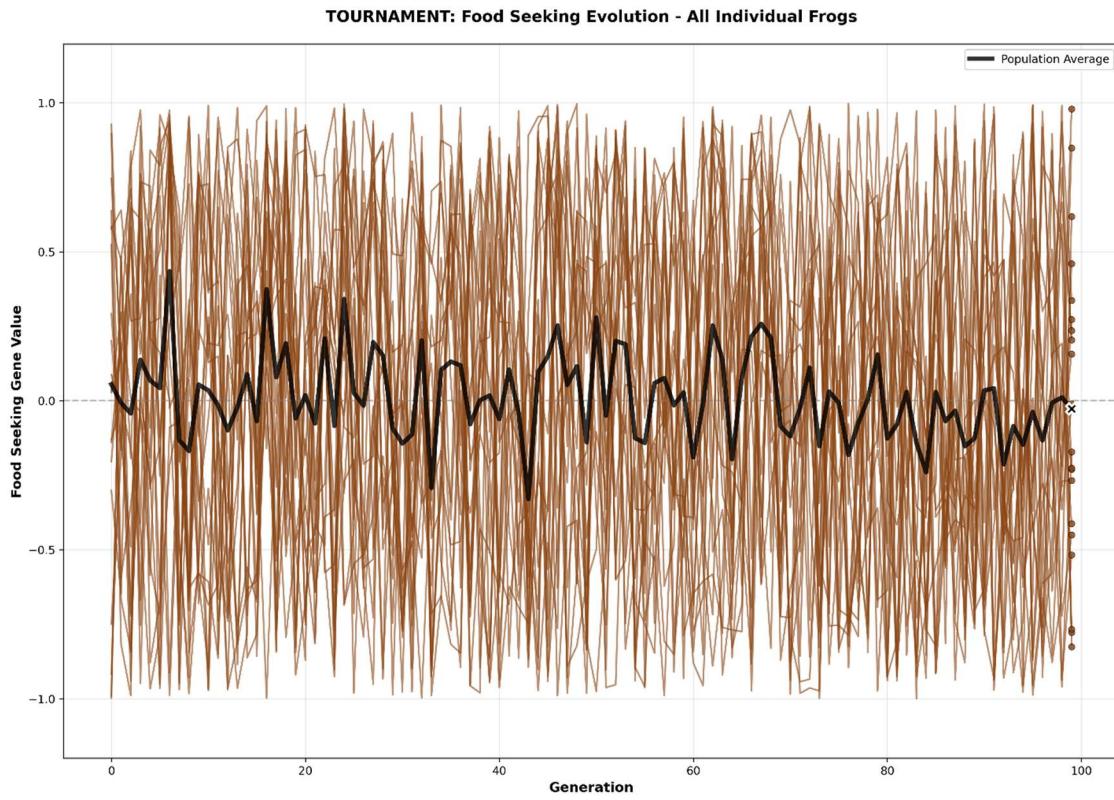


Figure 14: Tournament Food Gene

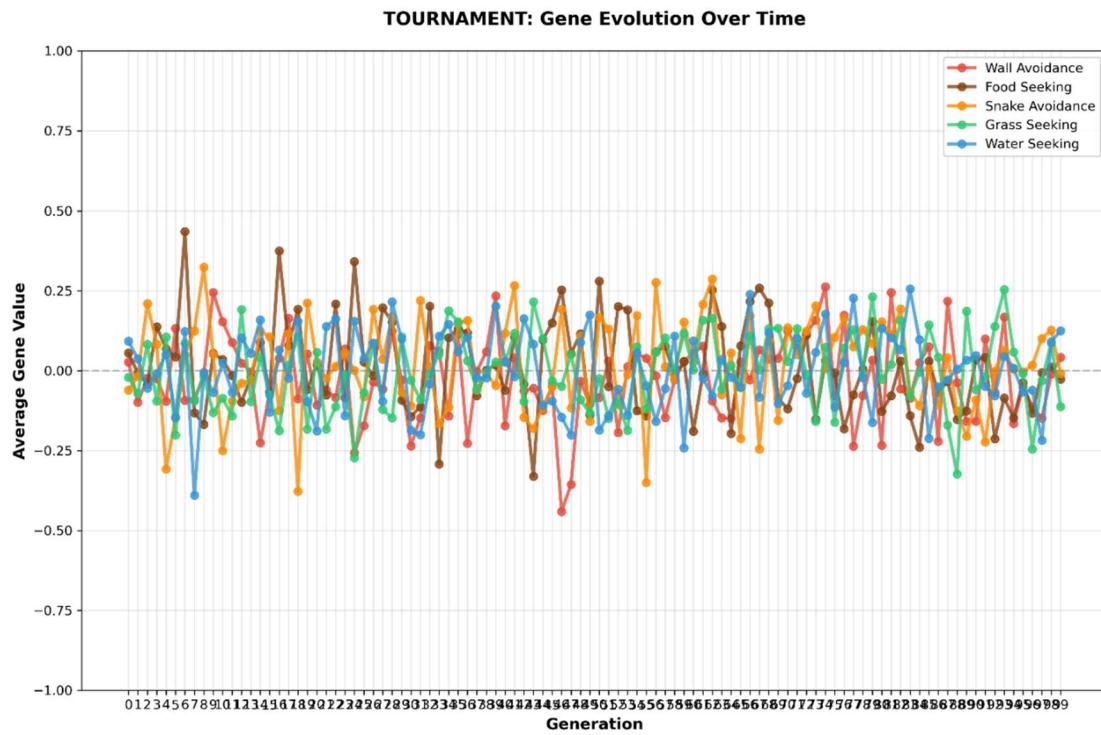


Figure 15: Tournament Gene Evolution

Individuals' fitness scores were similarly poor to steady state and the top performers were likely a combination of random luck and some strong genes, but they were not effectively passed along.

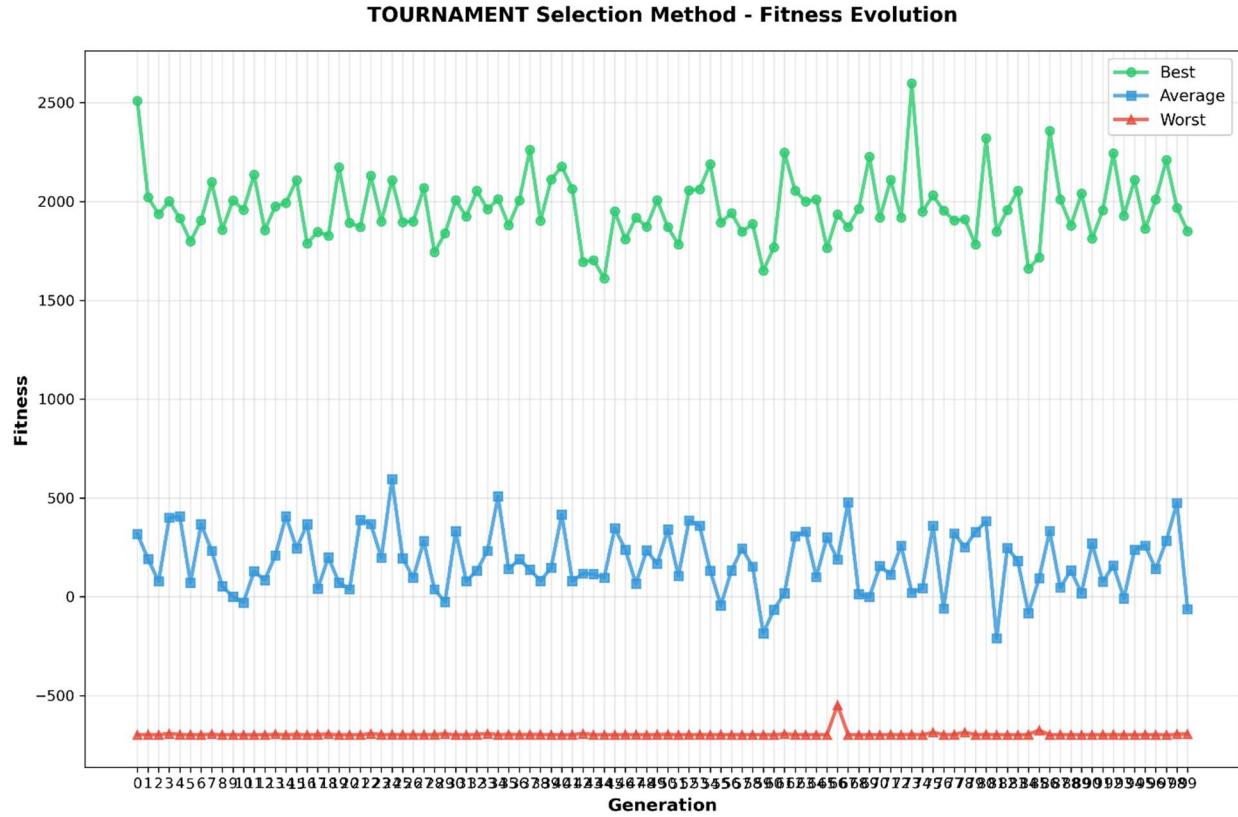


Figure 16: Tournament Fitness

### 6.2.3 Simulated Annealing

Genetic Algorithm with Simulated Annealing (GA-SA) an initial temperature of 1.0 and decay rate of 0.98 did not perform any better and had a similar number of casualties to walls and starvation but slightly lower death rate to snakes, seen in figure 17. The snakes could have been ineffectively evolving due to using the same selection method and not improving their performance hunting frogs.

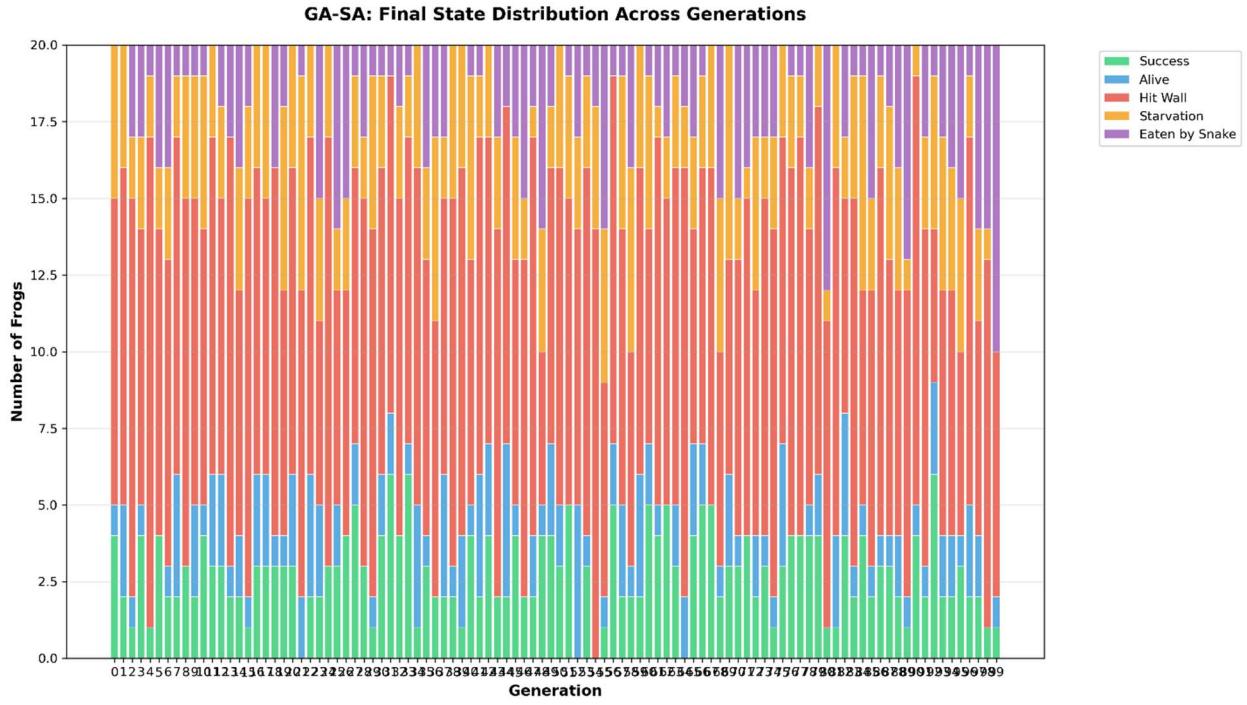


Figure 17: GA-SA Final States

Again, all genes for GA-SA were highly volatile and did not converge seen in figure 18 for wall avoidance. Within figure 19 for the last 15 generations, it did have the lowest average gene variance across the different GA selection methods. This was most likely due to the temperature reducing the mutation rate and the population stagnating. It does not show promising behaviour as they were averaging around the 0.0 mark and the optimal gene should be [-1.0, 1.0, -1.0, 1.0, 1].

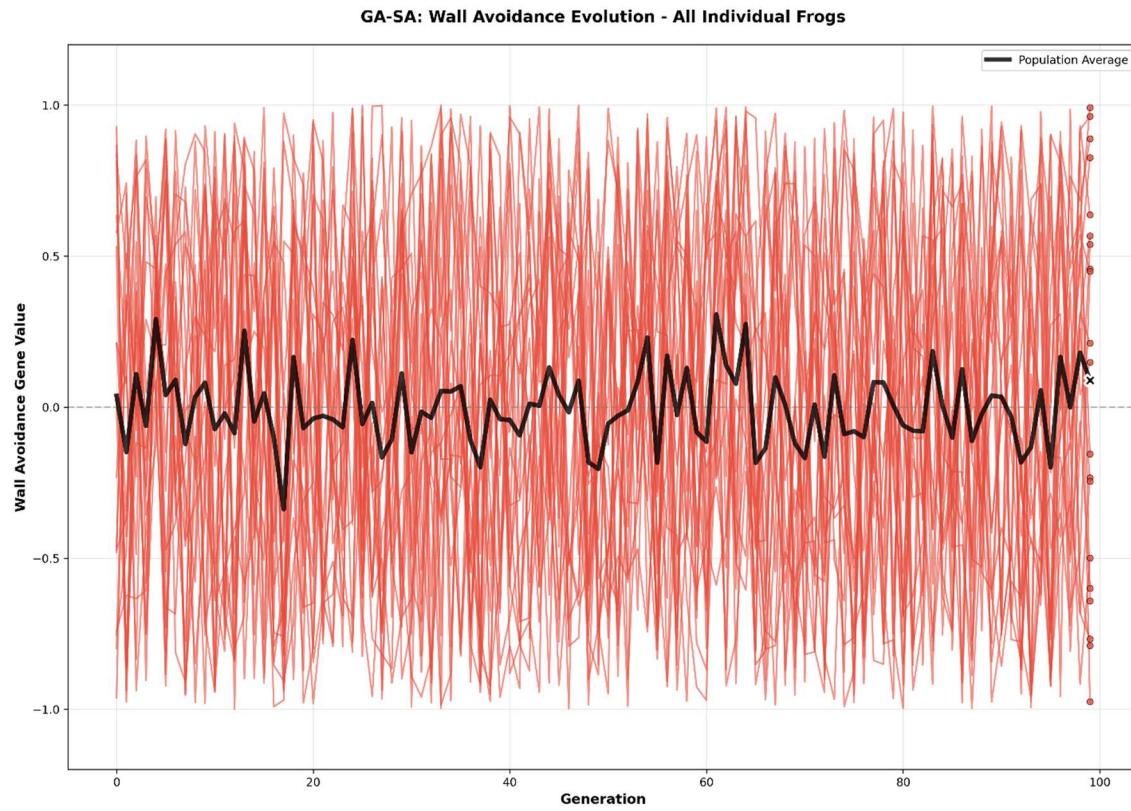


Figure 18: GA-SA Wall Gene

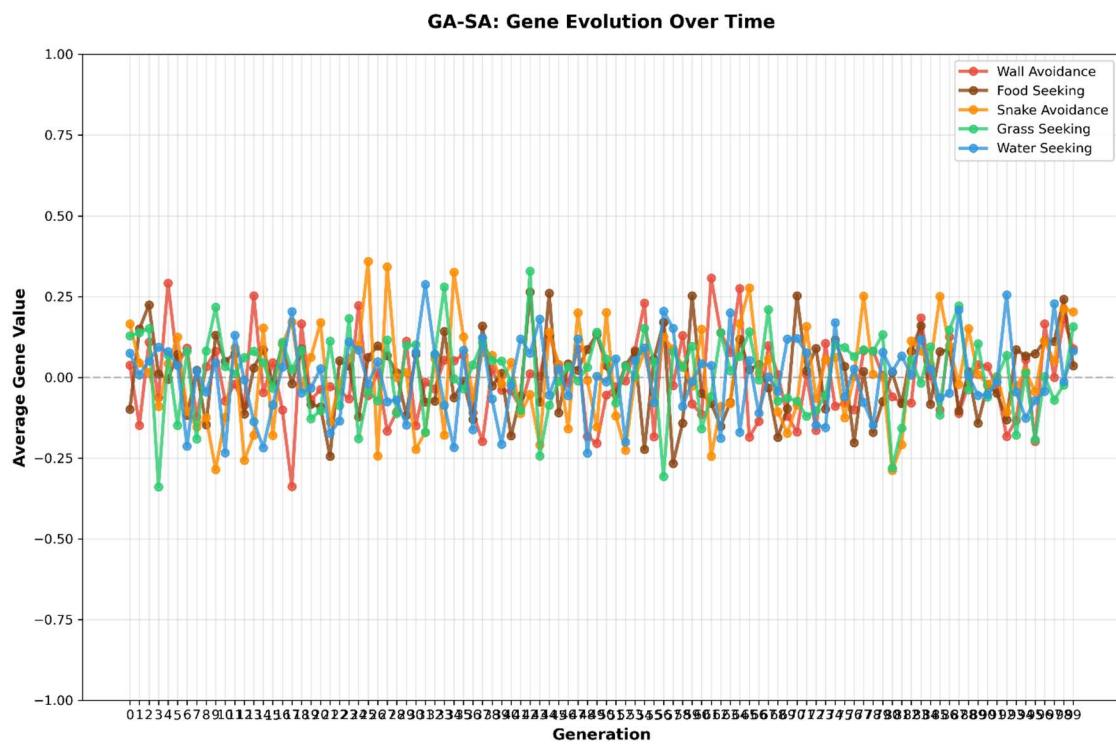


Figure 19: GA-SA Gene Evolution

Following the pattern of steady state and tournament's fitness trend, the average fitness value remained around 0.0 for the duration of the 100 generations. One abnormality was generation 55 which had no successes and subsequently the lowest fitness by a large margin. This shows the importance of collecting enough flies and reproduction has upon fitness.

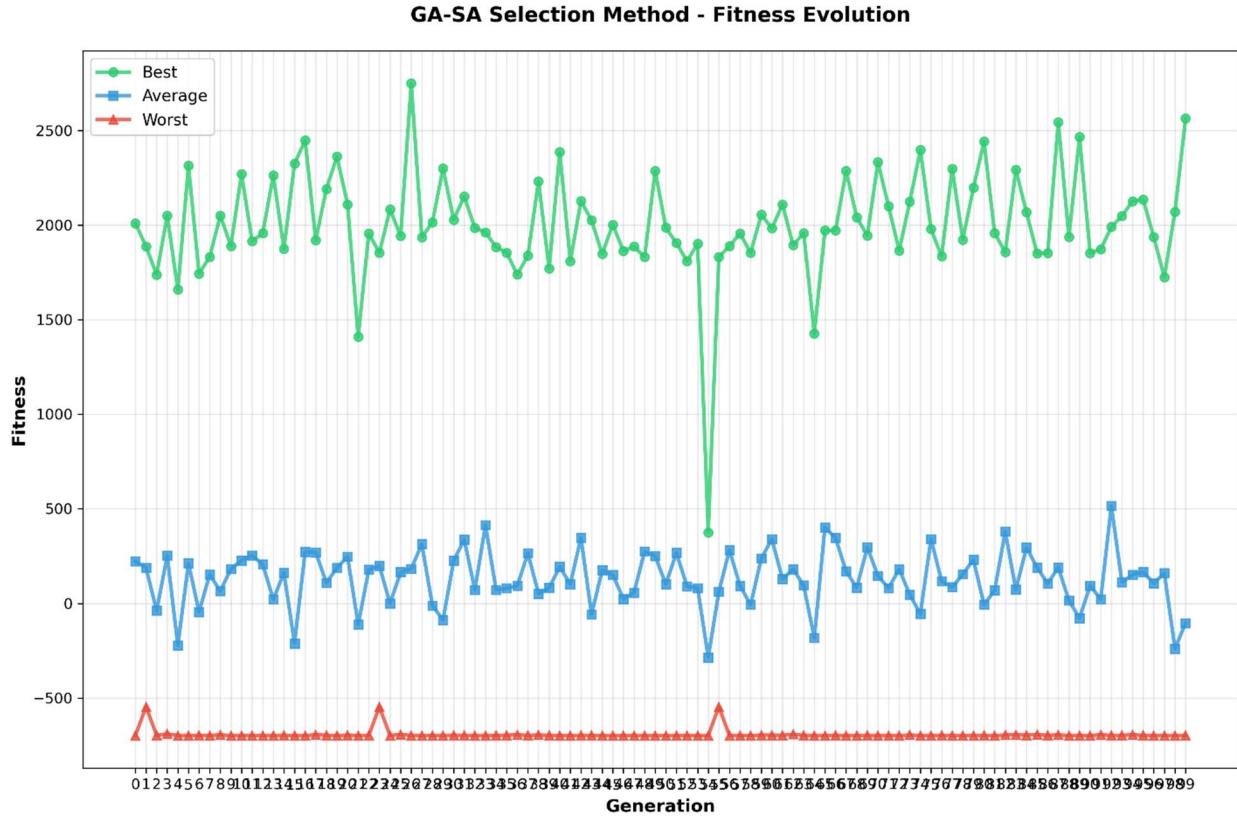


Figure 20: GA-SA Fitness

#### 6.2.4 Cross GA Comparison

Figure 21 shows the food gene average plotted for each GA selection methods over the 100 generations. This still shows no significance and represents their inability to learn the environment and required behaviours. I therefore let the GA algorithms run for 250 generations to see if any pattern could be observed. However, this proved fruitless as seen in figure 22.

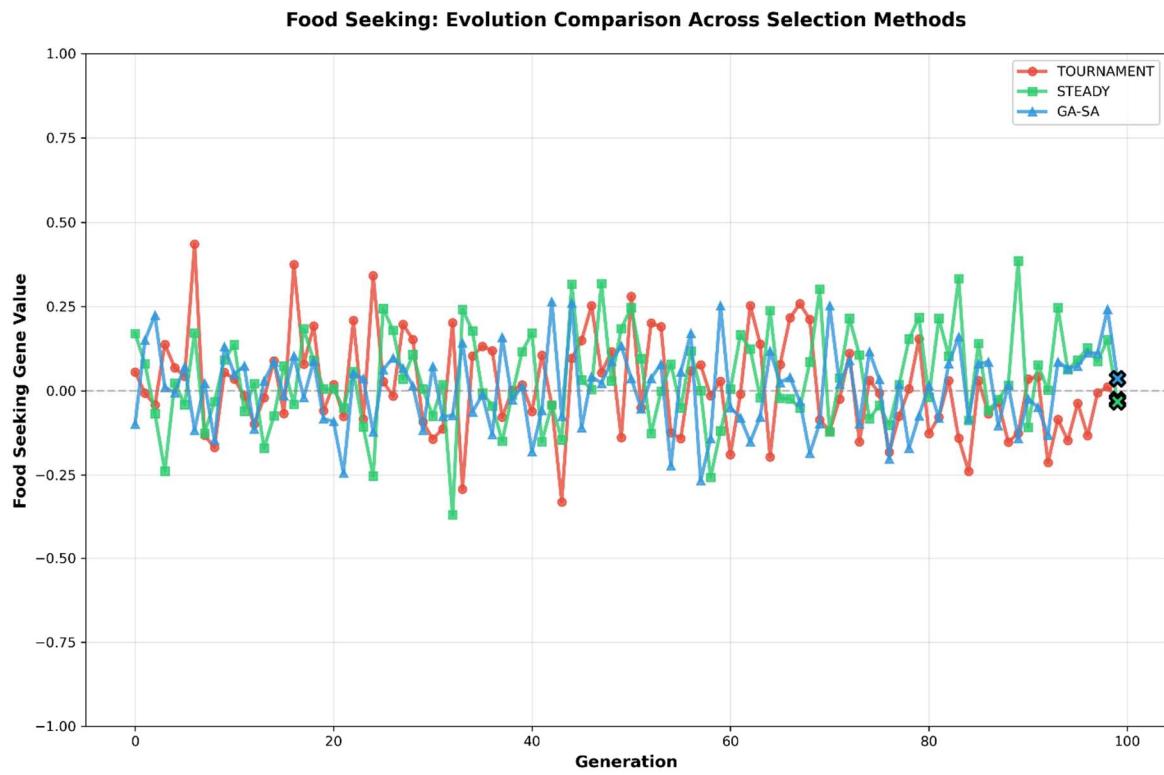


Figure 21: 100 Gen All GA Food Gene

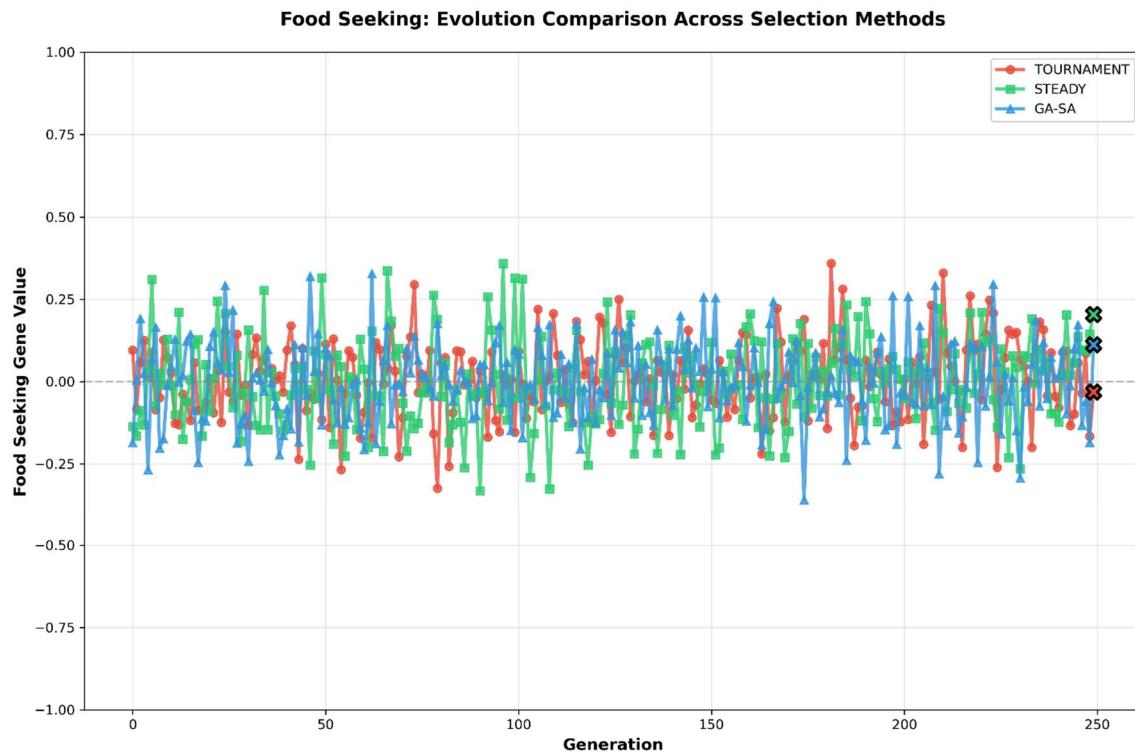


Figure 22: 250 Gen All GA Food Gene

## 6.3 Population-Based Consensus Behavioural Evolution (PBCBE) Performance Analysis

The PBCBE used the same environment parameters and the PBCBE specific parameters can be seen in `ensemble_params.json`. In figure 23 mutation rate and selection method set to tournament can be seen, this is due to the snake agent still using the GA implementation instead of the PBCBE the frogs have been updated to use.

Data for these results are stored in `/training data/consensus`. The data was collected by running five seeded ensemble runs to collect enough data to ensure this new method worked sufficiently and was not susceptible to the random initialisation of the population.

```
1  {
2      "MAX_GENERATIONS": 100,
3      "SELECTION_METHOD": "tournament",
4      "NUM_RAYS": 24,
5      "FROG_POP": 20,
6      "SNAKE_COUNT": 2,
7      "GENOME_SIZE": 5,
8      "LIFESPAN": 1200,
9      "NUM_DOTS": 100,
10     "DOT_RADIUS": 7,
11     "CANVAS_WIDTH": 800,
12     "CANVAS_HEIGHT": 600,
13     "NUM_GRASS_PATCHES": 6,
14     "GRASS_RADIUS": 60,
15     "NUM_WATER_POOLS": 3,
16     "WATER_RADIUS": 40,
17     "FLIES_NEEDED": 5,
18     "STARVATION_THRESHOLD": 500,
19     "STARVATION_IGNORE_GRASS_THRESHOLD": 250,
20     "CONSENSUS_THRESHOLD": 0.7,
21     "EVOLUTION_STEP_SIZE": 0.1,
22     "MIN_SUCCESSFUL_BEHAVIORS": 8,
23     "MOMENTUM_DECAY": 0.9,
24     "MOMENTUM_STRENGTH": 0.15,
25     "ENSEMBLE_RUNS": 5,
26     "ENSEMBLE_SUCCESS_THRESHOLD": 0.6,
27     "ENSEMBLE_CONVERGENCE_SIMILARITY": 0.2,
28     "RANDOM_UNIFORM_NEG": 0.3,
29     "RANDOM_UNIFORM_POS": 0.3,
30     "MUTATION_RATE": 0.1
31 }
```

Figure 23: `Ensemble_params.json`

### 6.3.1 Final States

To prevent lengthening this document further with the 5 plot runs, I will primarily focus on run 4, please see the `plots/consensus` folder to see all data graphs.

All the runs follow a similar trend for final states, in figure 24 there is initially a high death rate to walls for the first 5 generations. After generation 5 the frogs scarcely die to walls. Past this point the second highest cause of death is starvation most likely caused by flies becoming

increasingly scarce as a generation's simulation time elapses. However, in run 3 (figure 25) it has the highest percentage of frog deaths to walls and wall deaths as a percentage of total generations.

The combined alive and success rate drastically increases to generation 14 as the population starts to learn behaviours about eating food, entering grass and snake avoidance. It plateaus until generation 31 once the water gene matures and the alive rate is converted into successes. Generation 39 has a 60% success rate compared to the initial 5%.

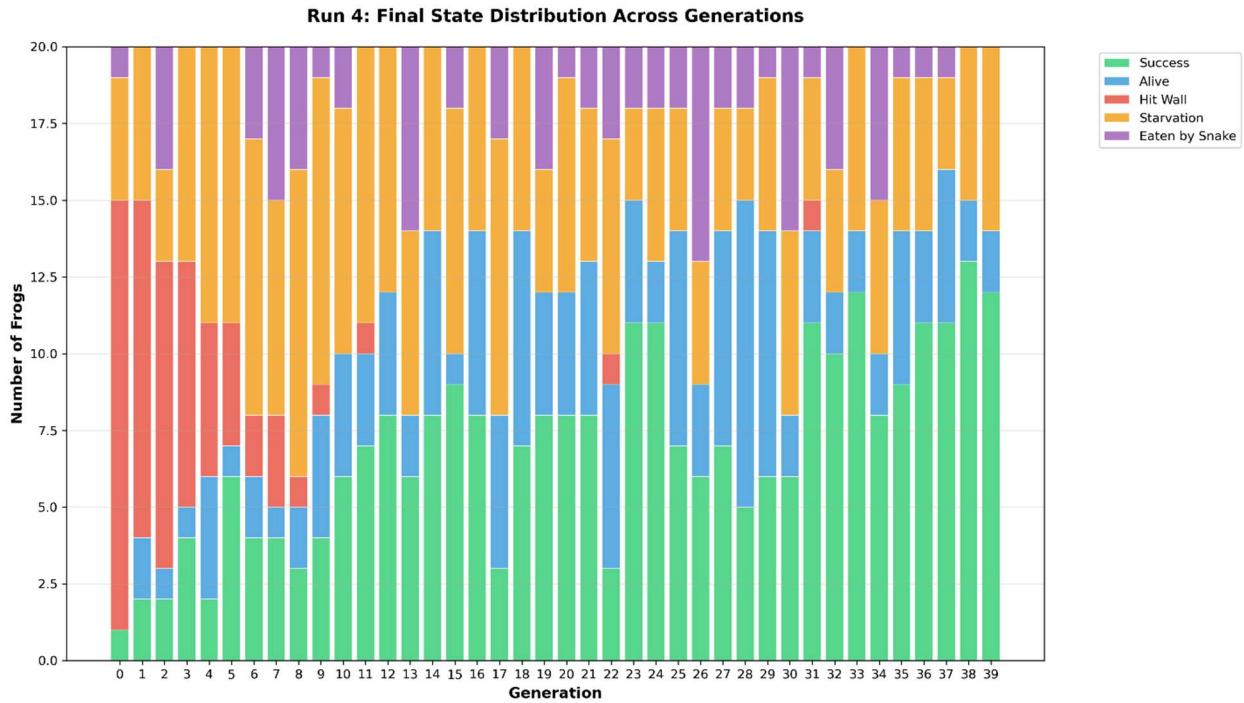


Figure 24: PBCBE Final States Run 4

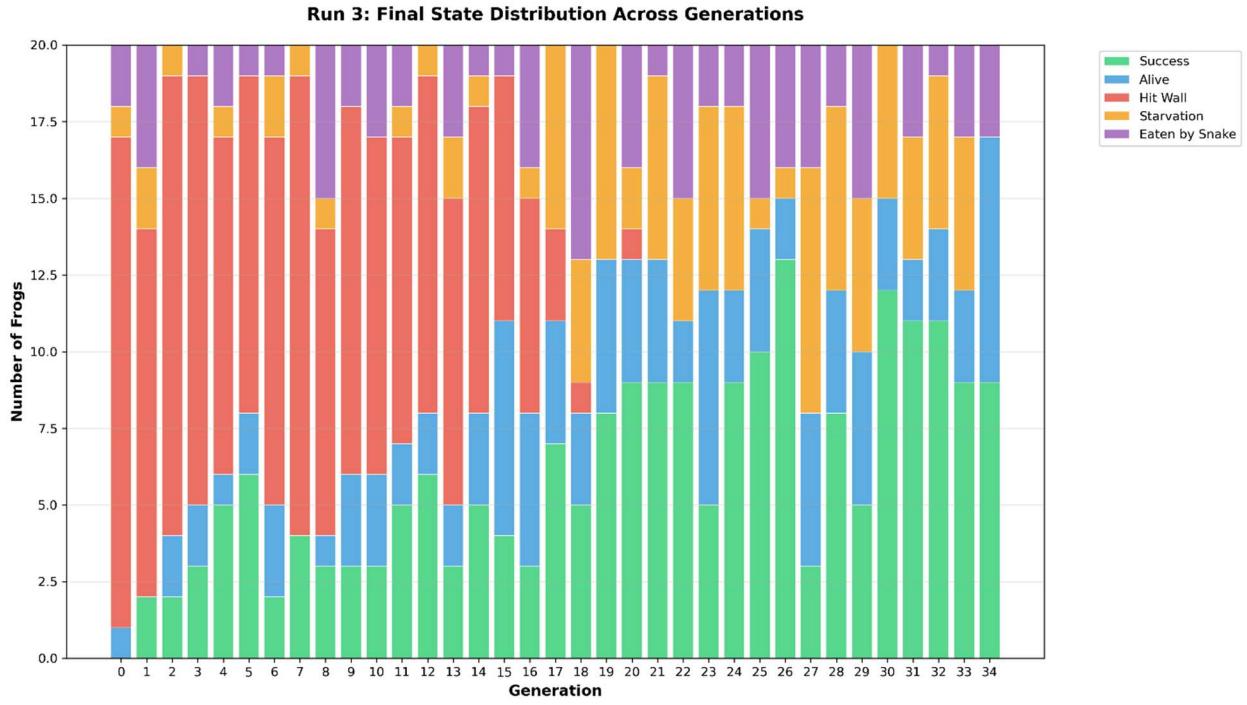


Figure 25: PBCBE Final States Run 3

### 6.3.2 Gene Evolution

Figure 26 depicts the average gene value for run 4 and only 39 generations as my program is designed to stop once the genes converge in the population. The wall and snake genes are the first to develop within the population and fully converging in the 9<sup>th</sup> generation. Food, grass and water gene's average was initially negative, however, the noted population showed a positive trend with the grass gene converging in the 12<sup>th</sup> generation, the food gene in the 18<sup>th</sup> generation and the water gene reaching the require convergence threshold for the simulation to terminate on the 39<sup>th</sup> generation.

**Ensemble Run 4: Gene Evolution Over Time**

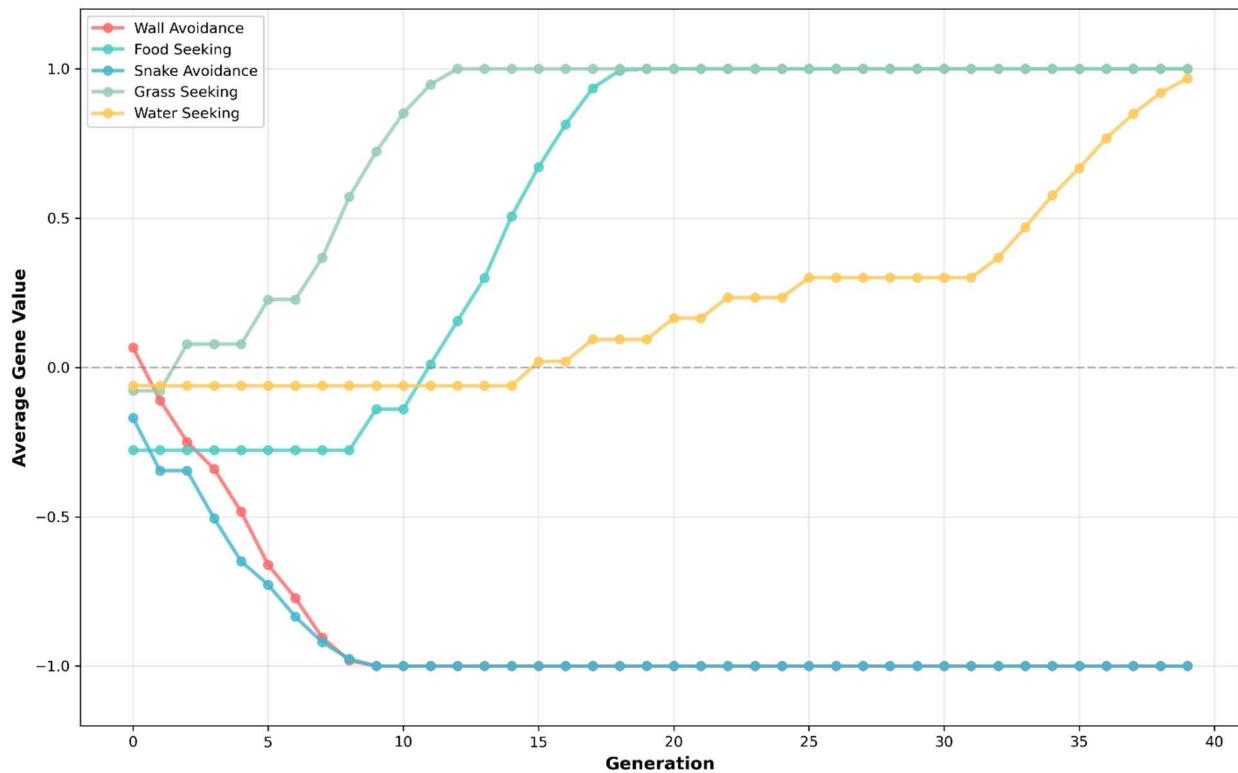


Figure 26: PBCBE Run 4 Gene Evolution

In figure 27, across all five runs, the snake gene converged within the first 15 generations. The 'X' on the graph represents when the different runs reached the convergence threshold for all genes. Figure 28 shows the water gene's evolution and in all 5 runs it is the last gene to converge as represented by the 'X'.

**Snake Avoidance: Evolution Comparison Across Ensemble Runs**

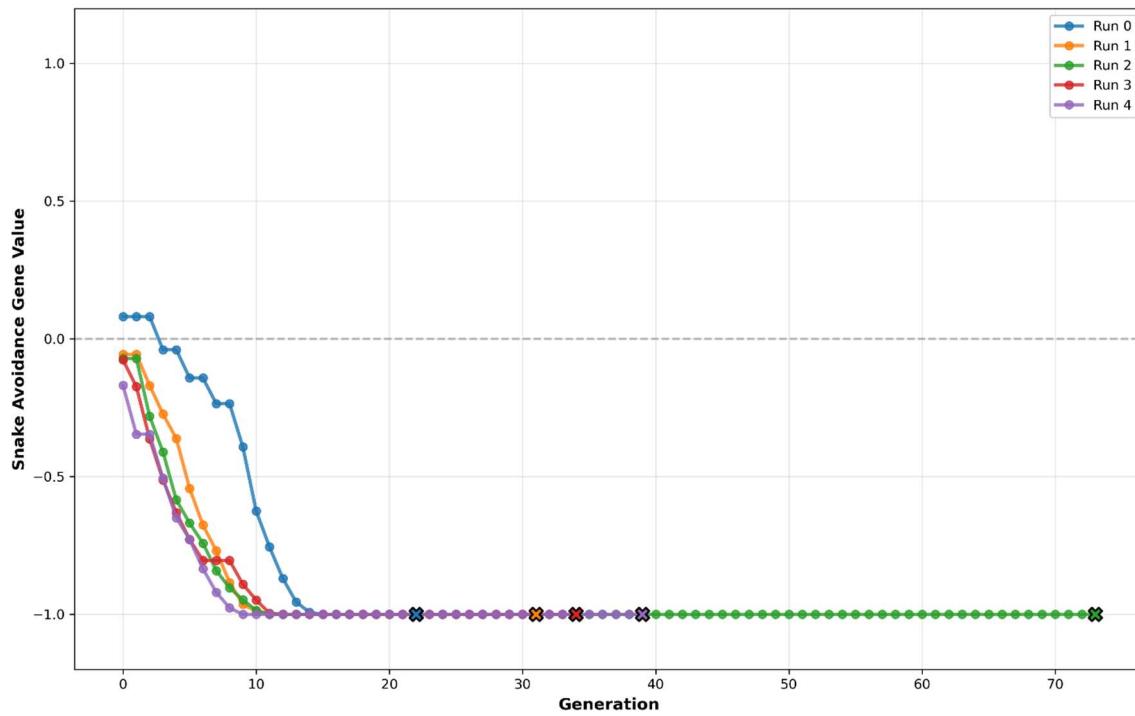


Figure 27: PBCBE Snake Gene All 5 Runs

**Water Seeking: Evolution Comparison Across Ensemble Runs**

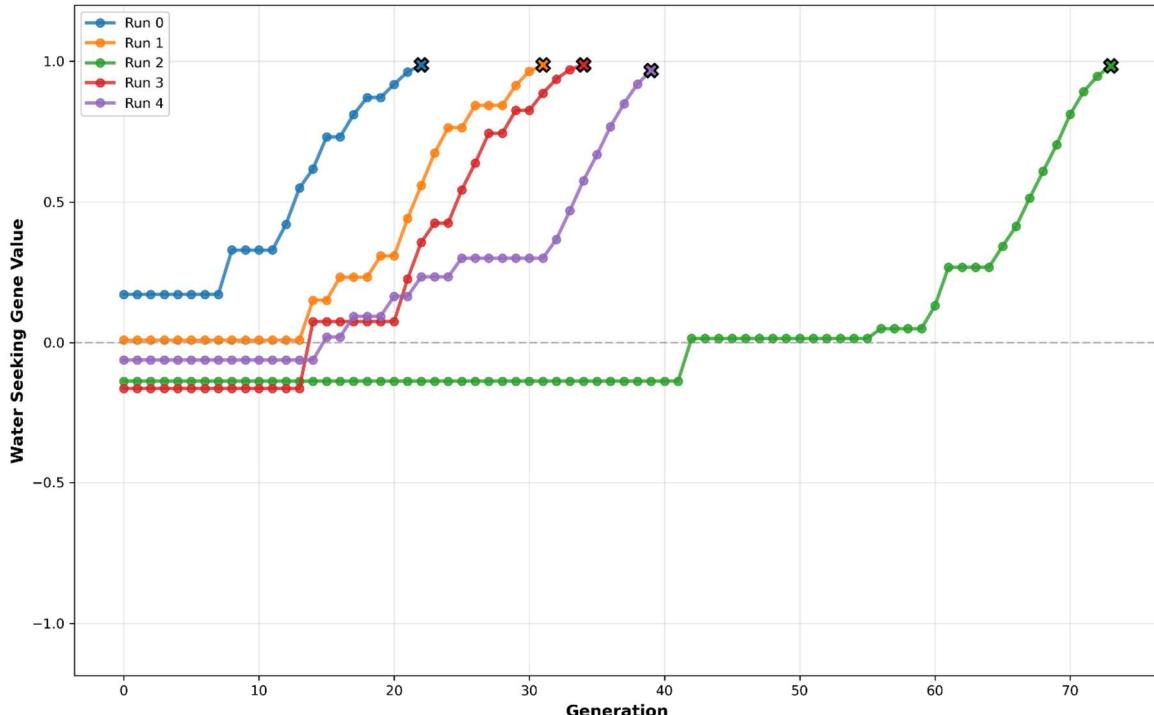


Figure 28: PBCBE Water Gene All 5 Runs

### 6.3.3 Behaviour Count

The water gene is the last to converge as it typically does not meet the required behaviour count for the population to agree upon confidence and gene direction. Figure 29 highlights this averaging around 6.9 compared to the next lowest of snake avoidance behaviours averaging 38.5. Figure 30 also represents the imbalance of recorded grass seeing behaviours due to every tick with the frog inside the grass it counts it as a successful behaviour.

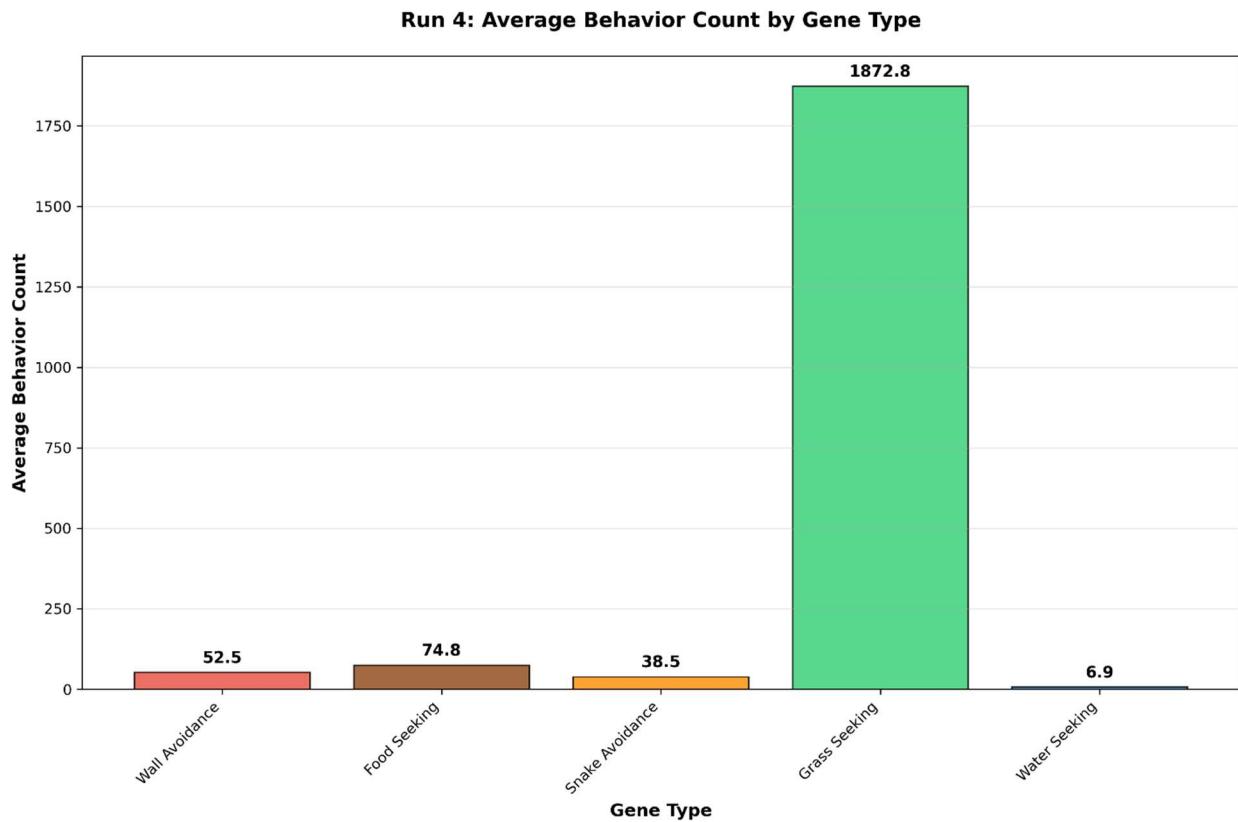


Figure 29: PBCBE Behaviour Count Bar Chart

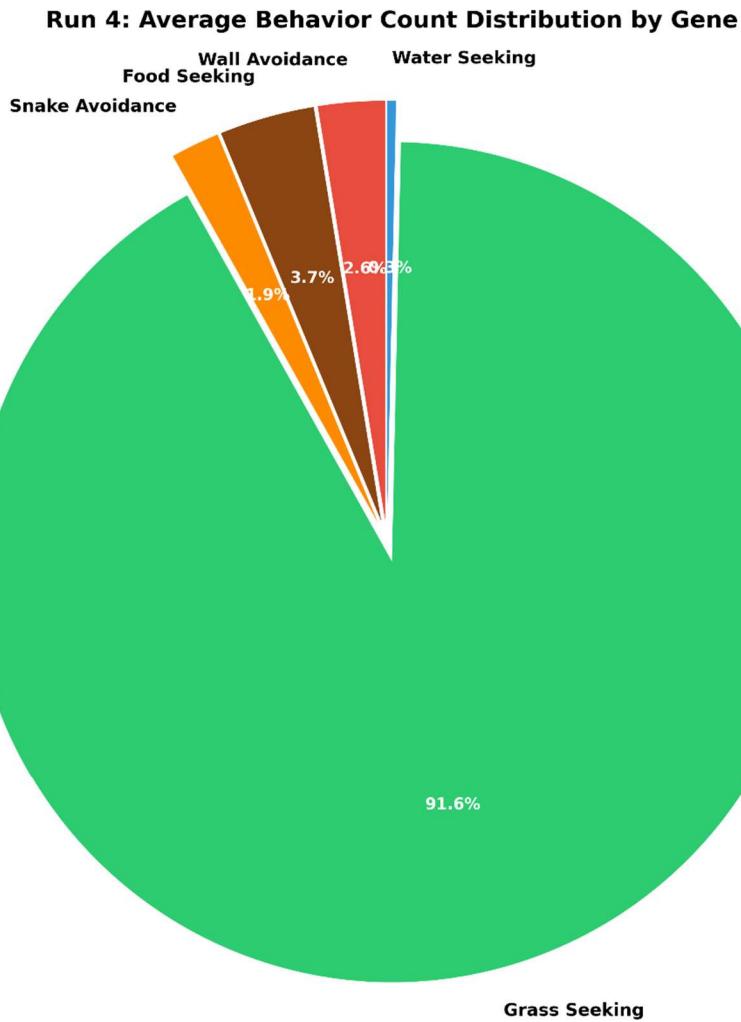


Figure 30: PBCBE Behaviour Count Pie Chart

Figure 31 shows the plotted behaviour count for water in addition to the line of best fit. In comparison to figure 32 of the wall avoidance behaviour, it has a far steeper line of best fit reinforcing the idea that water is a complex behaviour reliant on other behaviours developing first.

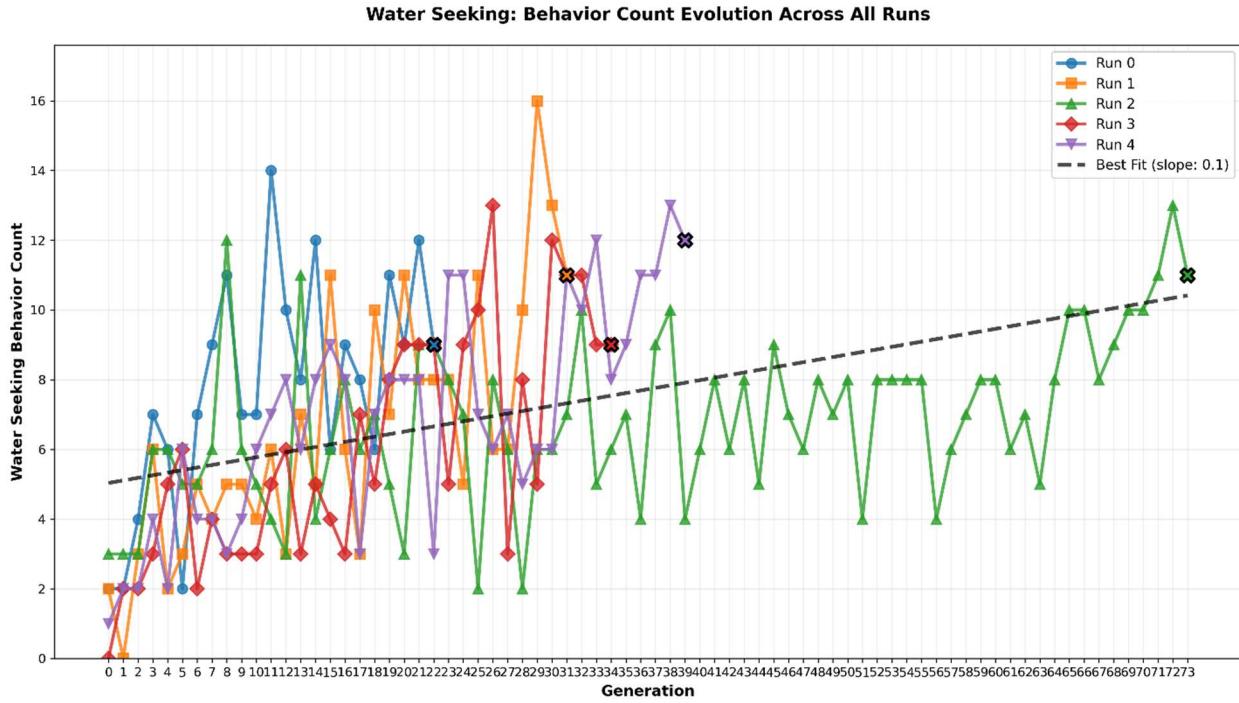


Figure 31: PBCBE Water Behaviour Count Line Graph

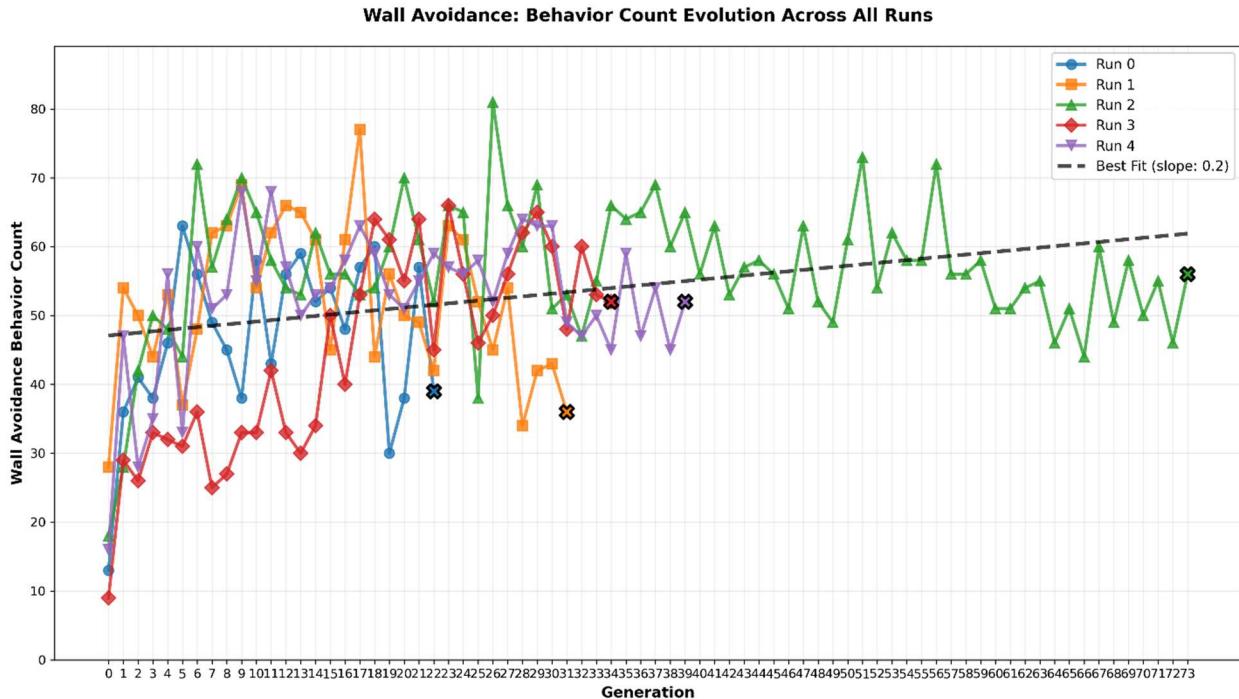


Figure 32: PBCBE Wall Behaviour Count Line Graph

#### 6.3.4 Population Confidence Consensus

The population confidence threshold is 0.7 and needs to be surpassed to reach consensus. Figure 33 shows that runs 0 and 2 reached the consensus threshold in the initial generation closely followed by run 3 in the 2<sup>nd</sup> generation. It took until the 10<sup>th</sup> generation for runs 1 and 4 to reach the consensus threshold, after that their confidence only increased as the generations continued.

Figure 34 highlights the need for a consensus threshold so that erratic behaviours are not taken into the wider population harming the gene pool, this is especially prevalent in run 2. Please note that some generations do not have point due to them not meeting the required recorded behaviour to consider confidence in a direction.

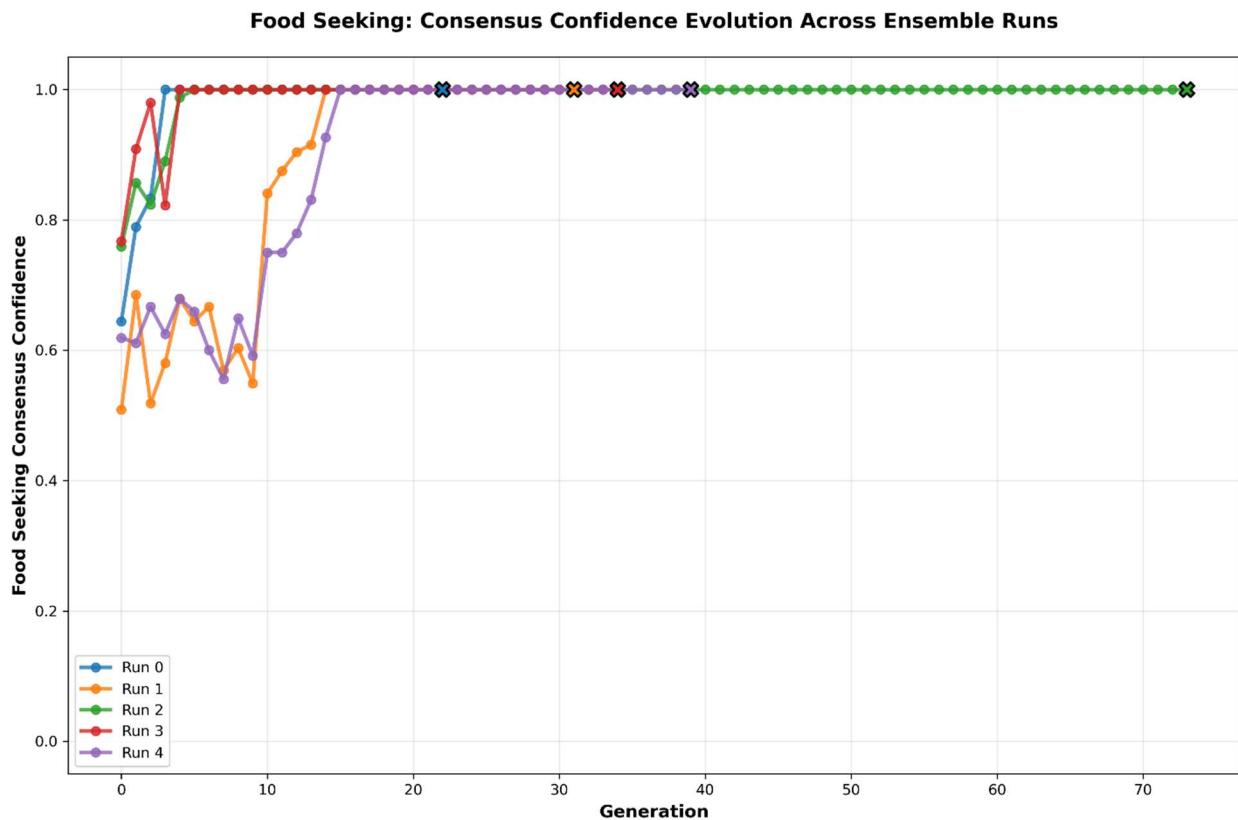


Figure 33: PBCBE Food Consensus Confidence

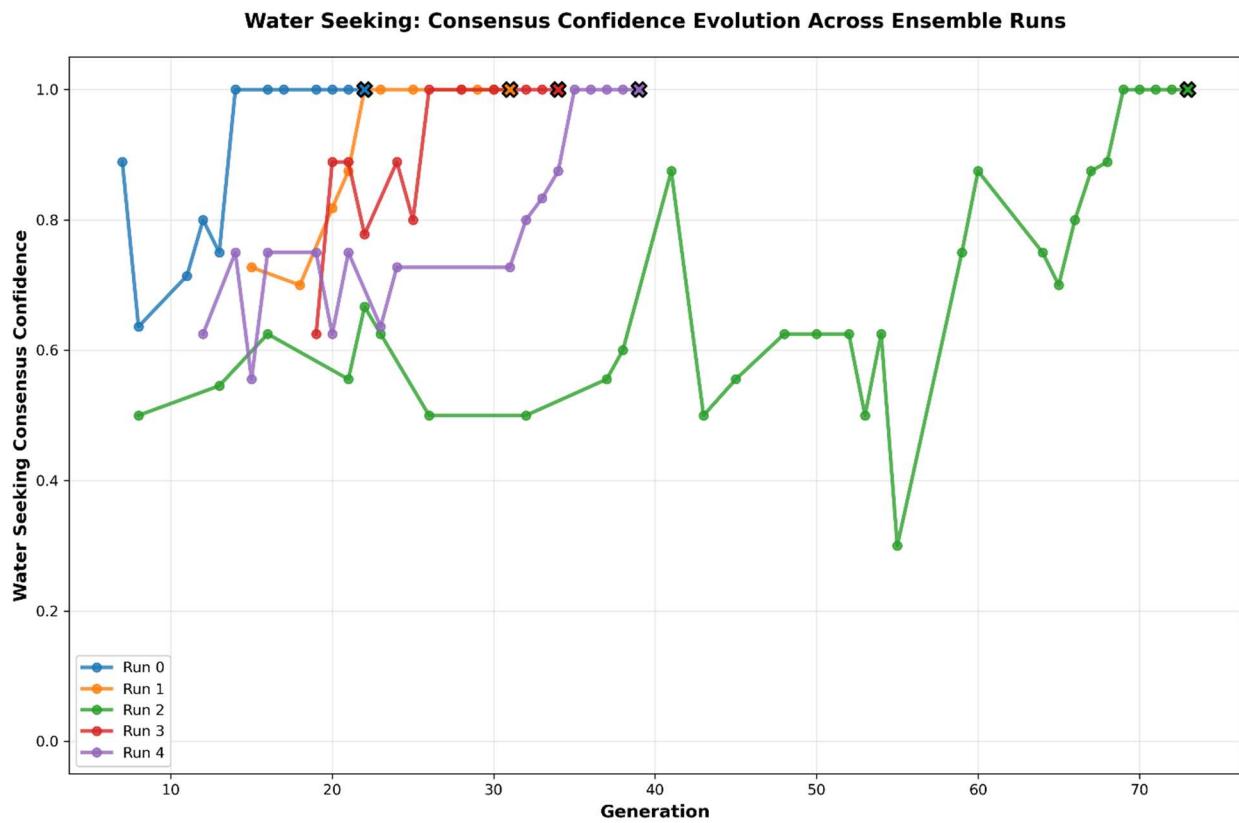


Figure 34: PBCBE Water Consensus Confidence

## 7 Conclusion

This study explored how a Green Frog can evolve to improve its survival and reproduction rate by comparing two evolutionary approaches. A traditional Genetic Algorithm (GA) to a novel approach Population-Based Consensus Behavioural Evolution (PBCBE). The GA variants of steady state, tournament and simulated annealing failed to develop stable survival behaviours over 100 and 250 generations. Whereas the slowest PBCBE run reached an all-gene successful convergence in 73 generations and run 0 converged in only 22 generations. The successful genes were as I hypothesised at [-1.0, 1.0, -1.0, 1.0, 1.0] representing wall and snake avoidance while being attracted to food, grass and water. PBCBE's gene level consensus learning of behaviours allowed weaker individuals to contribute to population knowledge, overcoming the limitations of fitness-based methods. The simulations results suggest that a population driven, behaviour focused evolution is more effective than individual fitness selection for evolving complex survival and reproductive behaviours in a dynamic environment.

### 7.1 Challenges Faced During Implementation

I experienced several challenges during development, the first was behavioural. Frogs were detecting grass and never leaving it to eat food and later reproduce. This was fixed by implementing starvation, when the frog dropped below a certain starvation threshold it would ignore the safety of grass and hunt flies.

The gene length of the agents originally was done for each five weights of wall, food, snake, grass and water for each of the 24 rays. This led to an enormous solution space of 120 chromosomes per individual that could not effectively be searched. This was reduced to a gene length of 15 where each weight had a left and right bias. The turn calculation in AgentSim was then redesigned to only require the five weights as explained in this report.

As seen in the results, the GA was not working effectively to distinguish successful behaviours which is why I designed and implemented decision trends. However, this was far too volatile for the majority even when it was in the wrong, and was then improved to PBCBE.

### 7.2 Successes & Limitations

Successes	Limitations
Implemented 3 Genetic Algorithms	Problem domain specific implementation of PBCBE
Implemented Population-Based Consensus Behaviour Evolution a novel algorithm	Snake agent uses GA with tournament selection also harming its effective ability to learn
Implemented ray based vision for agents	Complex code

Complex multi agent environment with staged behaviours for the frog	
Ensemble system design to ensure results are accurate by performing multiple runs	

### 7.3 Future Prospects

For major future work, I would like to remove my direct listing of successful behaviours in PBCBE and instead design a way in which the frogs can determine what a behaviour is and if it was successful or not. This would make it less problem domain specific. To improve the comparison of the GA I would implement a gene-based fitness mechanism so the genes can be more effectively evolved.

Other future work could include frogs which are currently invisible and immune to snakes in the grass which could be changed to a camouflage for a better dynamic between predator and prey. Frog and snake vision mentioned in section 2.1 both have a focal vision of their frontal 90 degrees which could be implemented as longer rays improving their reaction to the environment. The agents are reactionary and implementing a memory system could develop more profound inter-behavioural interactions.

# 8 Appendices

## 8.1 Appendix A: Environment Design, Constraints and Rules

### 8.1.1 Environment aspects

Non-tidal marshes are defined as frequently or continually inundated with water and soft stemmed vegetation adapted to the saturated soil conditions. They have a mix of water from a few centimetres to a meter deep with vegetated areas surrounding it of varying density.

All aspects of the environment were chosen to represent a marshland as the common habitat of the Easter Ribbon snake and the Green frog.

#### 8.1.1.1 Walls

The simulation window walls, being detected by a frog, (figure 35). This represents the edge of habitable space for both frog and snake. Frogs and snakes cannot simply enter back in from the other side of the simulation space when reaching the edge. This correctly represents an agents understanding of where it can viably live and not leave.

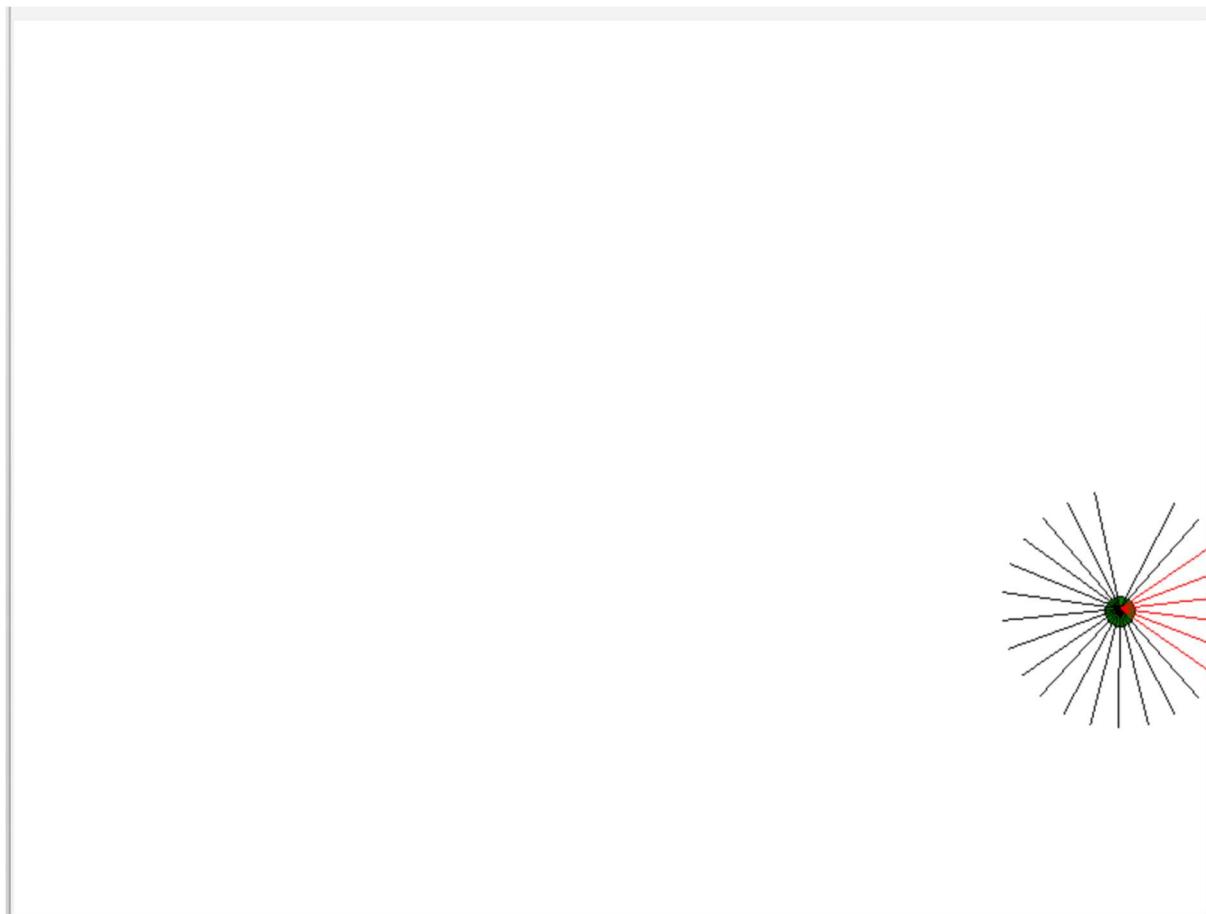


Figure 35: Environment Walls

#### *8.1.1.2 Flies*

The brown dot in figure 36 represents a fly, a food source for the frogs. During the simulation you can see the fly is comparable in size to the frog, but flies are obviously much smaller. Due to the ray-based vision, if the fly was any smaller, the rays would have a harder time detecting it accurately improving the reactions of the frogs towards food.



Figure 36: Fly

#### *8.1.1.3 Grass Patches*

The light green circles are areas of dense grass where frogs can hide from snakes. In addition to testing the predator and prey dynamic, the simulation is also testing hide and seek actions for further complexity. Please refer to figure 37 for their appearance.

Once a frog enters the grass a snake can no longer detect it with its ray-based vision. It is therefore invisible and even if a snake collides with a frog, it will not kill it.

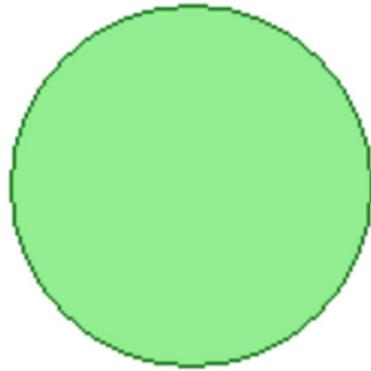


Figure 37: Environment Grass Patch

#### *8.1.1.4 Water pools*

Water pools represent possible areas for frogs to lay their eggs. For the simulation, this is marked as a “success” for frogs if they reach a water pool. However, frogs are required to find a certain amount of food specified by `FLIES_NEEDED` before they can detect it with their ray-based vision and enter it to succeed. The food requirement is simulating the energy requirement for reproduction.

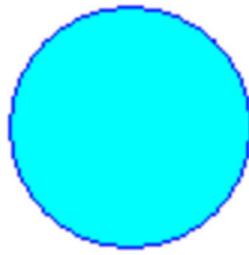


Figure 38: Environment Water Pool

## 8.2 Appendix B: Genetic Algorithm Overview

A genetic algorithm is a population-based search and optimisation technique inspired by natural selection, survival of the fittest and evolution. The solution is represented in a gene structure and will have different methods of manipulating the chromosomes through selection of individuals based upon fitness then deciding which genes will be carried to the next generation. To increase exploration within the solution space crossover, mutation factors can be applied to create new gene combinations.

## 8.3 Appendix C: Table of Figures

FIGURE 1: FROG .....	7
FIGURE 2: SNAKE .....	8
FIGURE 3: GENE REPRESENTATION .....	8
FIGURE 4: GA MUTATION .....	10
FIGURE 5: PBCBE BEHAVIOUR ANALYSIS.....	12
FIGURE 6: PBCBE GENE DIRECTION .....	14
FIGURE 7: PBCBE GENE EVOLUTION .....	15
FIGURE 8: JSON PARAMETERS EXAMPLE .....	16
FIGURE 9: STEADY STATE FINAL STATES.....	17
FIGURE 10: STEADY STATE WALL GENE.....	18
FIGURE 11: STEADY STATE GENE EVOLUTION .....	18
FIGURE 12: STEADY STATE FITNESS.....	19
FIGURE 13: TOURNAMENT FINAL STATES .....	20
FIGURE 14: TOURNAMENT FOOD GENE.....	21
FIGURE 15: TOURNAMENT GENE EVOLUTION.....	21
FIGURE 16: TOURNAMENT FITNESS .....	22
FIGURE 17: GA-SA FINAL STATES .....	23
FIGURE 18: GA-SA WALL GENE .....	24
FIGURE 19: GA-SA GENE EVOLUTION.....	24
FIGURE 20: GA-SA FITNESS .....	25

FIGURE 21: 100 GEN ALL GA FOOD GENE .....	26
FIGURE 22: 250 GEN ALL GA FOOD GENE .....	26
FIGURE 23: ENSEMBLE_PARAMS.JSON.....	27
FIGURE 24: PBCBE FINAL STATES RUN 4.....	28
FIGURE 25: PBCBE FINAL STATES RUN 3.....	29
FIGURE 26: PBCBE RUN 4 GENE EVOLUTION .....	30
FIGURE 27: PBCBE SNAKE GENE ALL 5 RUNS .....	31
FIGURE 28: PBCBE WATER GENE ALL 5 RUNS.....	31
FIGURE 29: PBCBE BEHAVIOUR COUNT BAR CHART .....	32
FIGURE 30: PBCBE BEHAVIOUR COUNT PIE CHART .....	33
FIGURE 31: PBCBE WATER BEHAVIOUR COUNT LINE GRAPH .....	34
FIGURE 32: PBCBE WALL BEHAVIOUR COUNT LINE GRAPH .....	34
FIGURE 33: PBCBE FOOD CONSENSUS CONFIDENCE .....	35
FIGURE 34: PBCBE WATER CONSENSUS CONFIDENCE.....	36
FIGURE 35: ENVIRONMENT WALLS .....	39
FIGURE 36: FLY .....	40
FIGURE 37: ENVIRONMENT GRASS PATCH .....	40
FIGURE 38: ENVIRONMENT WATER POOL .....	41

More data plots were done than used in the report, please see /Plots within the code submission folder.

## 9 References

- [1] ‘*Lithobates clamitans*’, *Wikipedia*. May 25, 2025. [Online]. Available: [https://en.wikipedia.org/w/index.php?title=Lithobates\\_clamitans&oldid=1292242211](https://en.wikipedia.org/w/index.php?title=Lithobates_clamitans&oldid=1292242211)
- [2] ‘Frog’, A.Z. Andis Arietta. [Online]. Available: <https://www.azandisresearch.com/animal-vision/frog/>
- [3] ‘*Thamnophis saurita*’, *Wikipedia*. Apr. 05, 2025. [Online]. Available: [https://en.wikipedia.org/w/index.php?title=Thamnophis\\_saurita&oldid=1284012813](https://en.wikipedia.org/w/index.php?title=Thamnophis_saurita&oldid=1284012813)
- [4] R. L. Haupt, *Practical genetic algorithms*, 2nd ed. Hoboken, N.J: John Wiley, 2004.
- [5] A. Dushatskiy, A. Chebykin, T. Alderliesten, and P. Bosman, ‘Multi-Objective Population Based Training’, in *Proceedings of the 40th International Conference on Machine Learning*, PMLR, Jul. 2023, pp. 8969–8989. [Online]. Available: <https://proceedings.mlr.press/v202/dushatskiy23a.html>
- [6] C. Shyu and J. A. Foster, ‘Evolving Consensus Sequence for Multiple Sequence Alignment with a Genetic Algorithm’, in *Genetic and Evolutionary Computation — GECCO 2003*, E. Cantú-Paz, J. A. Foster, K. Deb, L. D. Davis, R. Roy, U.-M. O'Reilly, H.-G. Beyer, R. Standish, G. Kendall, S. Wilson, M. Harman, J. Wegener, D. Dasgupta, M. A. Potter, A. C. Schultz, K. A. Dowsland, N. Jonoska, and J. Miller, Eds., Berlin, Heidelberg: Springer, 2003, pp. 2313–2324. doi: 10.1007/3-540-45110-2\_124.
- [7] ‘Intersection of a Line and a Plane’, Mathematics LibreTexts. [Online]. Available: [https://math.libretexts.org/Bookshelves/Calculus/Supplemental\\_Modules\\_\(Calculus\)/Multivariable\\_Calculus/1%3A\\_Vectors\\_in\\_Space/Intersection\\_of\\_a\\_Line\\_and\\_a\\_Plane](https://math.libretexts.org/Bookshelves/Calculus/Supplemental_Modules_(Calculus)/Multivariable_Calculus/1%3A_Vectors_in_Space/Intersection_of_a_Line_and_a_Plane)
- [8] ‘Vector projection’, *Wikipedia*. May 08, 2025. [Online]. Available: [https://en.wikipedia.org/w/index.php?title=Vector\\_projection&oldid=1289415674](https://en.wikipedia.org/w/index.php?title=Vector_projection&oldid=1289415674)
- [9] ‘Distance from a point to a line’, *Wikipedia*. May 16, 2025. [Online]. Available: [https://en.wikipedia.org/w/index.php?title=Distance\\_from\\_a\\_point\\_to\\_a\\_line&oldid=1290751448](https://en.wikipedia.org/w/index.php?title=Distance_from_a_point_to_a_line&oldid=1290751448)