PSO for Competitive Co-evolution of Neural-Controllers for the Game of Predator vs. Prey

Adam Balint

Supervised by Beatrice M. Ombuki-Berman

Submitted in partial fulfillment
of the requirements for COSC 4F90

Department of Computer Science

Brock University

St. Catharines, Ontario

Contents

List of Tables

List of Figures

1 Introduction

2 Background

2.1 Artificial Neural Networks

Neural networks are a type of model that is based on the way that brains function. Artificial neural networks are made up of components called neurons which are stacked into layers and connected to form a network. Neural network operation consists of providing input data to the input layer of the network, passing that data along to network to the output layer and then interpreting the output in a specific way. In the case of predator vs prey, output neurons will correspond to different actions that the predator and prey can make.

2.1.1 Neurons

As mentioned above, the key components of a neural network is the neuron. The sole job of the neuron is to compute the weighted sum of all of the inputs to the neuron and then proceed to pass the value through an activation function. For the purposes of this paper the logistic function as seen in figure 1 represented by the equation 1.

$$y = \frac{1}{1 + e^(- x)} \tag{1}$$

2.2 Particle Swarm Optimization Overview

Particle swarm optimization(PSO) was developed by Kennedy and Eberhart (1995). PSO takes inspiration from the flocking behavior of swarms of birds. When observing birds there tends to be one leader and the rest of the flock tend to follow that leader. Particles which are analogous to birds are spread throughout a solution space where the particle with the best solution to the problem becomes the leader (global best). Throughout multiple generational

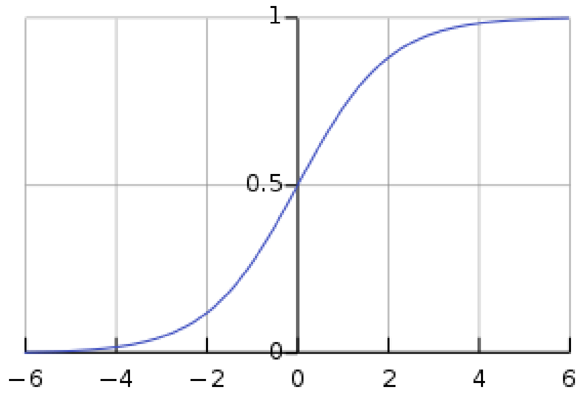

Figure 1: Graph of logistic function

time steps the rest of the particles converge towards the global best. If a new best solution is found, the particle that found the new best solution becomes the global best and the process continues until convergence or a max generation cap is reached.

2.3 Vanilla PSO

2.3.1 Particles and Swarms

As mentioned above, a particle is the base unit of the PSO. Particles consist of a location in the solution space, a velocity and a fitness which is used to determine the best particle. A group of particles make up a swarm. The global best is determined within a swarm, thus if multiple swarms are used each swarm could potentially have a different global best.

2.3.2 Particle Update

An update of a particles location consists of several terms. As convergence is expected, naturally a component of the update is towards the global best. This is called the social term. Another component is the draw towards the particles best seen location (local best). This is called the cognitive term. Each of these terms has a corresponding co-efficient which can be adjusted based on what is required from the PSO. Coefficients tend to be picked based on graph 2 that was proven to be convergent by [1]. A velocity update can be made

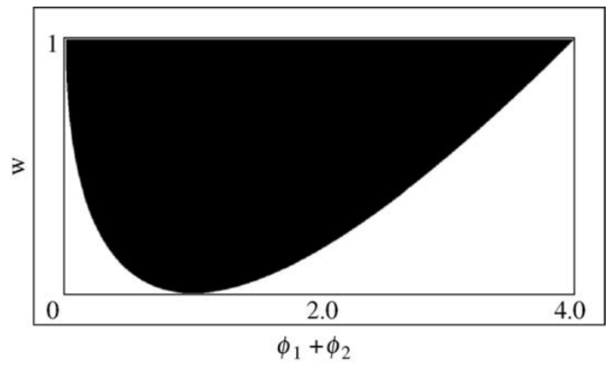

Figure 2: Parameter convergence graph for PSO

by combining the social and cognitive terms using the following equation:

$$v(t+1) = \omega * v(t) + c_1 * \psi_1 * (p(t) - x(t)) + c_2 * \psi_2 * (g(t) - x(t))$$

Where,

v is the velocity of the particle

c_1 and c_2 are the personal and social coefficients respectively

ψ is a random multiplier between 0 and 1

$x(t)$ is the location of the particle at time t

$p(t)$ is the personal best location of the particle at time t

$g(t)$ is the global best location of the swarm at time t

The updated velocity is then used to update the position of the particle. The outline of the steps required to implement the PSO can be found in figure 3.

2.4 Charged PSO

Charged PSO differs slightly form the vanilla PSO. Charged PSOs have swarms as well, however in these swarms a particle has a change to be charged. If two charged particles are too close to each other, a repulsive force will be added to the velocity update function. This prevents total convergence allowing for the continuous searching of the search space.

```
Input: ProblemSize, $Population_{size}$
Output: $P_{g\_best}$
Population $\longleftarrow \emptyset$
$P_{g\_best} \longleftarrow \emptyset$
For ($i = 1$ To $Population_{size}$)
  $P_{velocity} \longleftarrow$ RandomVelocity()
  $P_{position} \longleftarrow$ RandomPosition($Population_{size}$)
  $P_{p\_best} \longleftarrow P_{position}$
  If (Cost($P_{p\_best}$) $\leq$ Cost($P_{g\_best}$))
    $P_{g\_best} \longleftarrow P_{p\_best}$
  End
End
While ($\neg$StopCondition())
  For ($P \in$ Population)
    $P_{velocity} \longleftarrow$ UpdateVelocity($P_{velocity}$, $P_{g\_best}$, $P_{p\_best}$)
    $P_{position} \longleftarrow$ UpdatePosition($P_{position}$, $P_{velocity}$)
    If (Cost($P_{position}$) $\leq$ Cost($P_{p\_best}$))
      $P_{p\_best} \longleftarrow P_{position}$
      If (Cost($P_{p\_best}$) $\leq$ Cost($P_{g\_best}$))
        $P_{g\_best} \longleftarrow P_{p\_best}$
      End
    End
  End
End
Return ($P_{g\_best}$)
```

Figure 3: PSO Pseudocode

Charged PSOs are used in dynamic environments where a complete convergence could allow for a non-optimal solution. Thus to ensure that the solution develops along with the changing error landscape total convergence is actively prevented by the use of these charged particles. The updated velocity function for charged particles is as follows:

$$v(t+1) = \omega * v(t) + c_1 * \psi_1 * (p(t) - x(t)) + c_2 * \psi_2 * (g(t) - x(t)) + a_i(t)$$

with

$$a_i(t) = \sum_{j=1, j\neq i}^{n_s} a_{ij}(t)$$

Where,

$$a_{ij} = \begin{cases} \frac{Q_i Q_j}{d_{ij}^3} * (x_i(t) - x_j(t)) & \text{if} R_c \leq d_{il} \leq R_p \\ \frac{Q_i Q_j}{d_{ij}^3} * (x_i(t) - x_j(t)) & \text{if} R_c < d_{il} < R_p \\ 0 & \text{otherwise} \end{cases}$$

Where,

$d_{ij} = ||(x_i(t) - x_j(t))||$

Q is the charge of the particle

R_c is the core radius

R_p is the particle perception limit

3 Literature Review

3.1 Co-operative Coevolution

Cooperative coevolution can be defined as the evolution of models that work together to solve a common goal. Co-operative co-evolution is very prominent in every day life in many work disciplines. As such, to improve the quality of every day life, artificially intelligent co-operative coevolution techniques have been applied to certain real life scenarios. Specifically for simulating emergency response for disaster scenarios. In this type of scenario, there would be three sets of models working together. This would be the ambulance, fire department and the police.

3.2 Competitive Coevolution

Competitive coevolution on the other hand can be see everywhere in games. As an example if two people are playing chess, then if one player begins to play better, then the other player will adapt and reaching the same level if not surpassing the other player's level. Another example of Competitive coevolution that was mentioned by [3] is the RoboCup and the Federation of International Robot-soccer Association (FIRA). These two organizations hold robot soccer competitions, both real and virtual with varying amounts of information. Competitive coevolution in this case would be the two teams of robots improving their techniques in order to counter the other team.

3.3 Predator vs. Prey

The type of competitive coevolution that this paper focuses on is the predator vs prey game. This game is similar to the robot soccer in the way that both the predator and the prey attempt to out manouver each other in order to emerge victorious. This paper implements and expands on a variation of [2] paper on evolving behaviour through competition for the predator vs prey game.

4 Experimental Setup

4.1 Predator vs Prey

The predator vs prey game consists of two types of creatures (predators and prey) competing on a nine by nine board. The goal of the predator is to capture the prey and the goal of the prey is to avoid the predator for the 20 turns. Both the predator and prey move simultaneously in this implementation of the predator vs prey game. This should cause both the predators and the prey to learn to predict the movement patterns of the other type of creature.

4.1.1 Predator

As stated before, the goal of the predator is to catch the prey. In order to interact with the board, the predator has a neural network that is associated with it. This neural network's input layer consists of 81 input nodes which take in the board state. There is a certain number of hidden nodes that was varied between runs, and there are 7 output nodes. The output node actions are arranged as seen in table 1. Some output nodes in the neural network are connected. These nodes are listed together in the table and when the action for the set is decided, the node with the higher activation is used. The predator has two unique output nodes that the prey does not have access to. These output nodes apply a two times multiplier to the movement in either the horizontal or vertical direction. This could be

Output Node	Action
1	Move diagonally (yes or no)
2 and 3	Horizontal movement left or right
4 and 5	Vertical movement up or down
6	Chase horizontally
7	Chase vertically

Table 1: Predator output node to action mapping

Output Node	Action
1	Move diagonally (yes or no)
2 and 3	Horizontal movement left or right
4 and 5	Vertical movement up or down

Table 2: Prey output node to action mapping

though of similar to a sprint or pounce action from the predator's point of view. If there is no diagonal movement, then the largest of the 2, 3, 4 and 5 nodes will be taked to determine the direction of the movement.

4.1.2 Prey

The actions of the prey are set up similarly to the actions of the predator, however the prey have a slightly smaller neural network. The prey neural network consists of the same 81 input nodes which take in the board state followed by a certain number of hidden nodes that was varied between runs, but unlike the predators, they only have 5 output nodes. The actions that the prey can perform are summarized in table 2. The prey neural network is similar to the predators, except without the special nodes. For node set of 2 and 3, and set 4 and 5, the node with the higher activation is taken. If there is no diagonal movement, then the largest of the 2, 3, 4 and 5 nodes will be taked to determine the direction of the movement.

4.1.3 Game Progression

As the game is played and the learning of both the predator and prey progresses, some large gains and falls in fitness should be observed as one type of creature outsmarts the other and vice versa. The runs are expected to reach an equilibrium when averaged over time.

4.2 NN Implementation

On an implementation level, the neural networks were implemented using an array of matrices which are multiplied together to calculate the output of the network. Each matrix represents the connection weights between 2 layers of the neural network.

4.3 PSO Implementation

4.3.1 Calculating Fitness Values

Fitness values for particles are calculated by instructing each particle to play 400 games against particles from the other swarm. There were three different point systems that were experimented with. These are outlined in table 3. The fitness value calculated was divided by the number of games that was played by the particle in order to reduce it to a range between -2 and +1. As seem in table 3, game V1 is heavily stacked in favor of the prey as if the prey or predator fall off then the predator does not capture the prey and the prey gains a point. Game V2 is a balanced match between the two types of creatures while game V3 provides a slight advantage to the prey because if the prey falls off then the predator loses one point, while if the predator falls off then the prey gains one point as it successfully avoided the predator.

4.3.2 Simulation

To set up the simulation, an opponent is selected randomly from the opposing swarm. The predator and prey are then placed on the board randomly such that the predator is on the upper half of the board and the prey is on the lower half of the board. Each turn consists of the predator and prey taking in the board surroundings and then deciding a move to make. The predator and prey then simultaneously make moves for 20 turns. The game ends if the

Game Version	Scemario	Predator Score	Prey Score
V1	Predator catches prey	+1	-1
	Prey avoids predator	-1	+1
	Predator and prey fall	0	+1
	Predator falls	0	+1
	Prey falls	0	+1
V2	Predator catches prey	+1	-1
	Prey avoids predator	-1	+1
	Predator and prey fall	-2	-2
	Predator falls	-2	0
	Prey falls	0	-2
V3	Predator catches prey	+1	-1
	Prey avoids predator	-1	+1
	Predator and prey fall	-2	-2
	Predator falls, prey does not fall	-2	+1
	Predator does not fall, prey falls	-1	-2

Table 3: Game Rules

Board State	Encoding
Empty Square	0
Same Species	1
Opposite Species	-1

Table 4: Game Board Encodings

predator catches the prey, or if the 20 turns have been made.

4.3.3 Board Encoding

As mentioned, the neural network takes as input the board. Therefore, the board has to be encoded before it is given to the network. This board encoding encodes the 9x9 board as seen in table 4.

4.3.4 Hall of Fame

Both the predator and prey swarm have their own hall of fame which stores prior global best values of that swarm in order to provide some backtracking over time to see if a previous set of weights is more efficient than the current best weights. Each hall of fame can hold a maximum of 25 prior best fitnesses. The hall of fame is updated every time a new global best fitness is determined and the new global best weight value is stored in the hall of fame. The values in the hall of fame are re-run every 10 epochs in order to determine if any older values have a better fitness than the current best as these could change as the swarm of the other creature type evolves.

4.4 Run Summary

Tables 5 and 6 contain all of the parameter sets that were tested. Throughout the experiments, The percentage of charged particles, the core radius and perception limit and the number of hidden nodes were varied. The ability to fall off the board was also changed in order to see if any behavioural changes would develop. If a closer look at the game rules is taken, then it can be seen that if the ability to fall off is removed, then all three game rules simplify the the same values. That is, if the predator catches the prey, then the predator gains one point while the prey loses a point. If the prey avoids the predator, then the opposite takes place. Further on in the report, the game rules version number will be prepended to the experiment number excluding when the creatures cannot fall off the board. For each of the parameter sets, 5 runs we conducted. Each run consited of 50 predators and 50 prey being evolved over 500 iterations. Each game that way played took place over 20 turns unless the predator caught the prey, or either the predator or the prey fell off of the board.

5 Results

5.1 Walled Game

When the arena is walled off and the creatures are not able to fall off of the board, it is expected that the prey will figure out some sort of move pattern to avoid the predator quickly, and the predator should be able to quickly adapt to the prey and capture it. This would show on fitness graphs as lines representing the predators and prey crossing over each other as they keep adapting to capture or escape respectively. This type of behaviour is not

Experiment	Can Fall	Hidden Nodes	Core Radius	Perception Limit	Swarm Charged (%)
1	False	15	2	40	0
2					33
3					67
4					100
5			5	30	0
6					33
7					67
8					100
9		30	2	40	0
10					33
11					67
12					100
13			5	30	0
14					33
15					67
16					100
17		60	2	40	0
18					33
19					67
20					100
21			5	30	0
22					33
23					67
24					100

Table 5: Experiment Summary

Experiment	Can Fall	Hidden Nodes	Core Radius	Perception Limit	Swarm Charged (%)
25	True	15	2	40	0
26					33
27					67
28					100
29			5	30	0
30					33
31					67
32					100
33		30	2	40	0
34					33
35					67
36					100
37			5	30	0
38					33
39					67
40					100
41		60	2	40	0
42					33
43					67
44					100
45			5	30	0
46					33
47					67
48					100

Table 6: Experiment Summary Continued

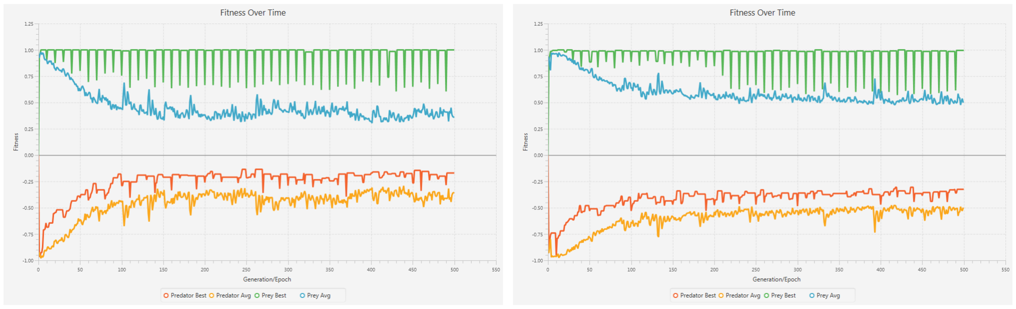

Figure 4: Experiment 13 run 0 (Left) and run 2 (Right) fitness over time

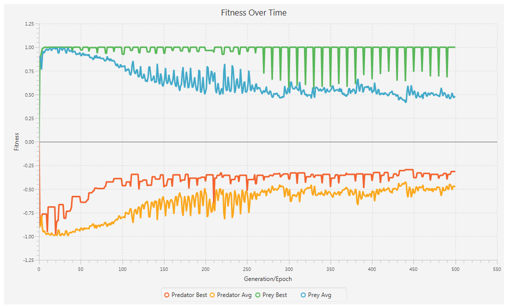

Figure 5: Experiment 1 run 2 fitness over time

seen in any of the walled off games, although the fitnesses do fluctuate in tandem with each other, just not as extremely as expected. Furthurmore, the prey always outperforms the predator in the walled off game. The general pattern of all of the graphs tends to be a sharp improvement in the beginning of the run, and then the curve flattens out as seen in figure 4 which both came from the same experiment. In most cases the fitness of the particles at the end of the run are determined by the gain/loss that occured at the beginning of the run. A more promising run that seems like it would have improved if training time was longer can be seen in figure 5.

In the walled game, although the overall population performance for the predator is not great, there tends to be at least one strong particle that is much higher than the rest of the swarm. This can be seen in figures 7 and 8. An example of the types of games this type of particle plays can be seen in figure 6. In this case, the prey always seems to want to leave the stage, however since it cannot it seems to get stuck and then the predator just approaches and captures the prey. Although this is a very simple scenario, the path of the predator is

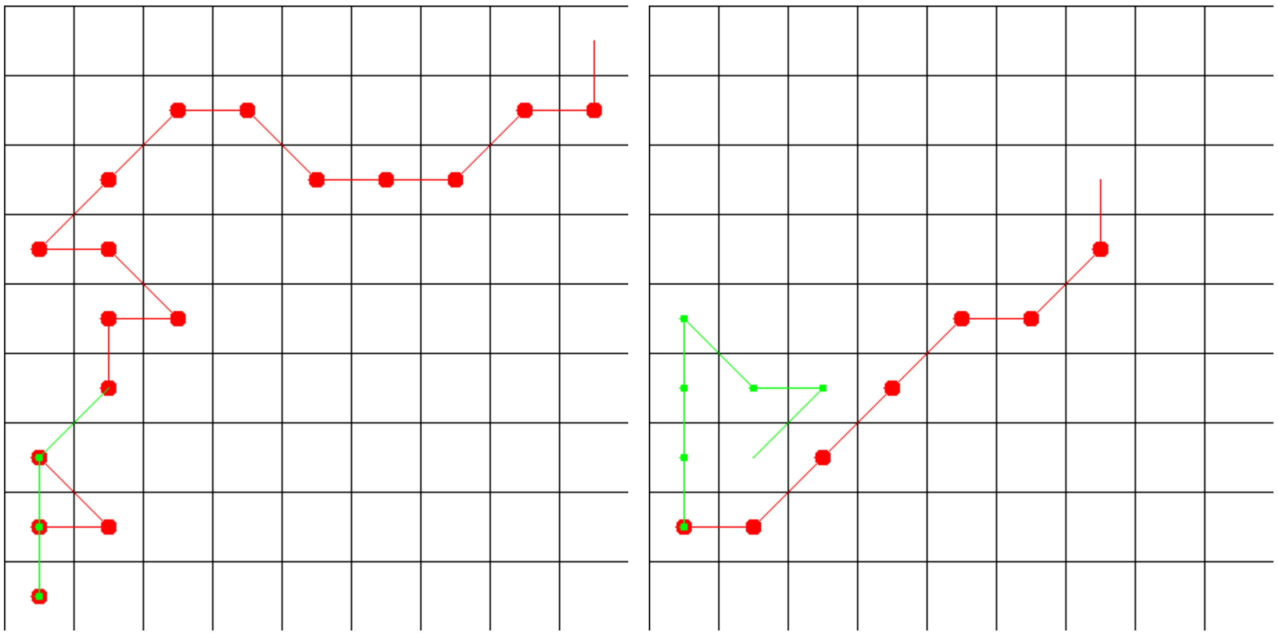
Figure 6: Experiment 15 run 2 example games

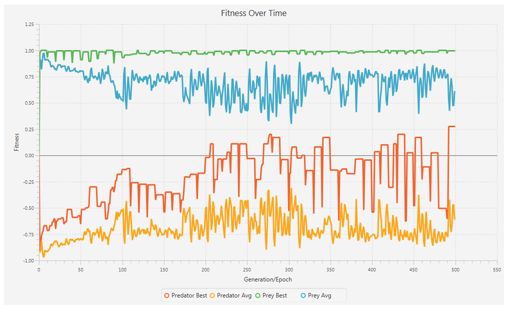

Figure 7: Experiment 15 run 2 fitness over time

quite direct when going to capture the prey, as such it demonstrates that the predator is capable of learning to capture the prey.

5.2 Game V1

As seen in table 3, the game V1 rules are heavily stacked in favour of the prey. The effect of this can be seen in all of the fitness over time graphs that were produced. Even the experiments that included charged particles performed poorly with this version of the game. An example of a graph from 25 and 41 can be seen in figure 9 and an example of a graph from experiment 28 and 44 can be seen in figure 10. In figure 10 it is expected that at least

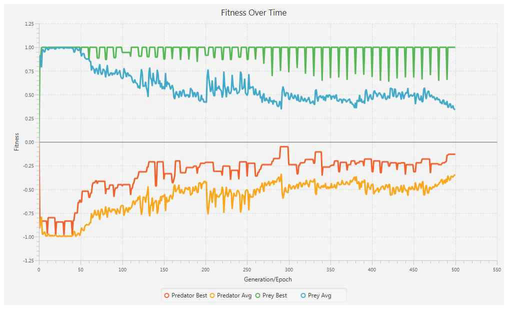

Figure 8: Experiment 6 run 1 fitness over time

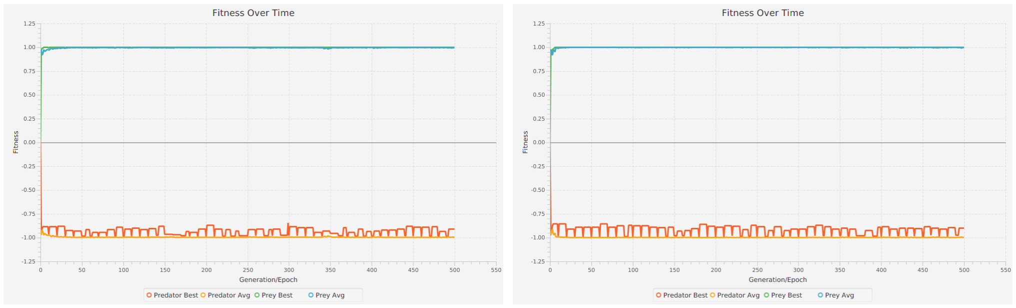

Figure 9: Game V1 Experiments 25(Left) and 41(Right)

one particle should have a high fitness value, but as it can be seen, the full charged swarms from experiment 28 and 44 do not perform better than those from experiments 25 and 41.

When observing the games being played by the particle with the best fitness over all of the runs an interesting pattern emerges. In most cases, the game ends with the prey running off of the board. This is because the easiest way for the prey to win if for it to fall off. While the predator also falls off about one third of the time, the predator tends to stay on the board for a longer time than the prey. An example of this technique can be seen in figure **??**. Observing the prey behaviour, this can be reinforced as the best prey had 324 games of

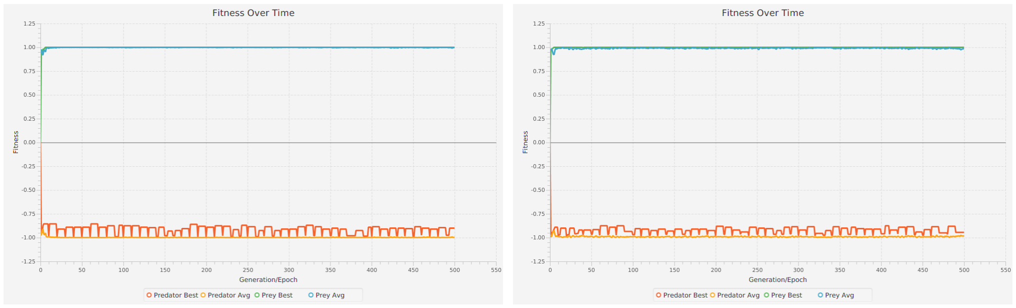

Figure 10: Game V1 Experiments 28(Left) and 44(Right)

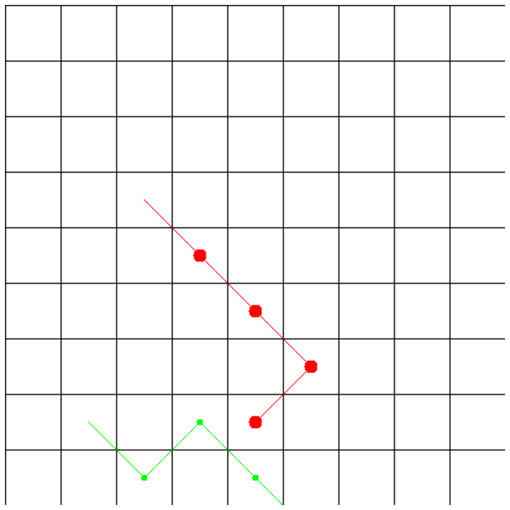

Figure 11: Experiment 25 example predator game

400 ending with the prey falling off the board. An example of several games can be seen in figure 12. Unfortunately, since it is extremely easy for the prey to win and for the predator to lose, the type of behaviour that evolved is quite basic. In general, both creatures tend to just move in a straight line until one of them falls off the board.

5.3 Game V2

Game V2 is the game with the even set of rules for the predators and the prey. This game mode should be the mode that provides the closest set of results between the predators and the prey. Unfortunately, the system still seems to provide an advantage to the prey as in every experiment the prey are the creatures with the highest fitness. However, unlike in the walled off game and the V1 game, in this game mode, there are times when the predators have a better score than the prey. Unfortunately for the prey, they always get outperformed by the prey.

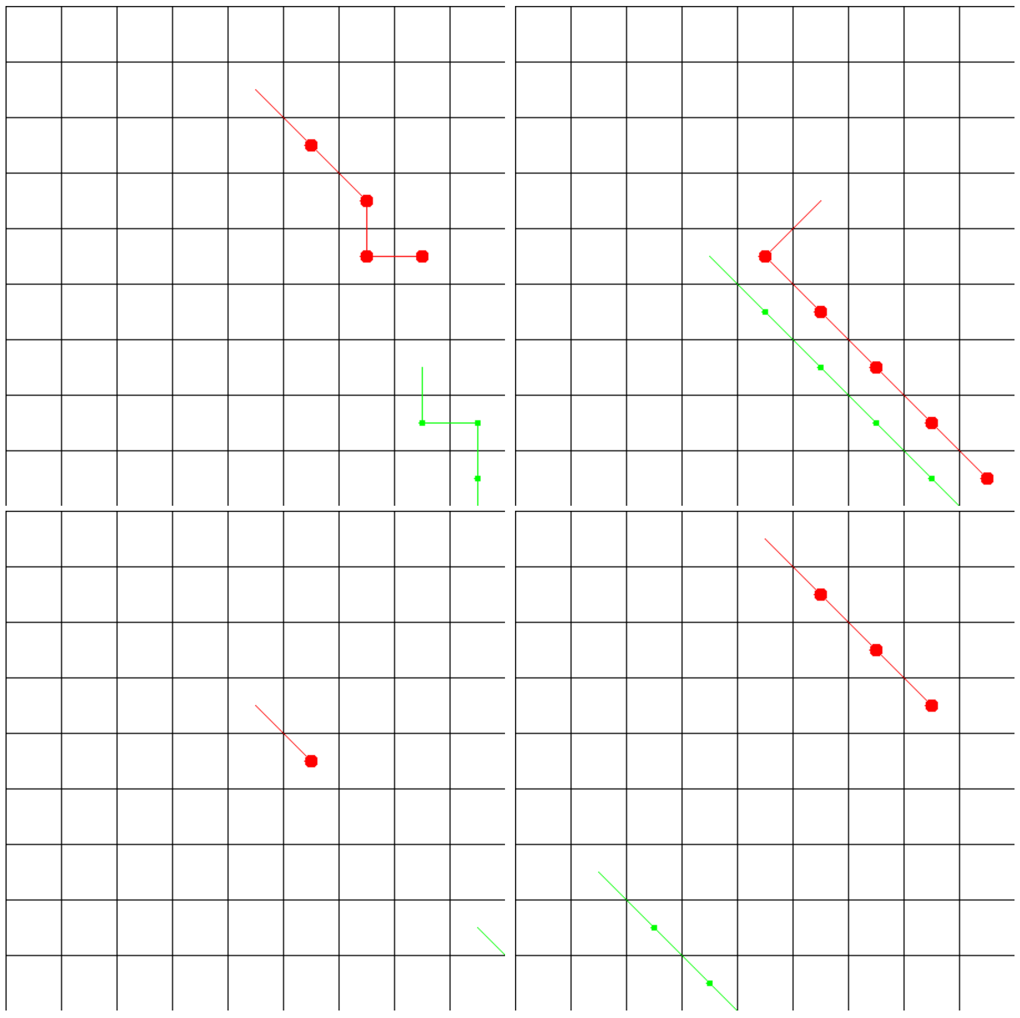

Figure 12: Experiment 25 examples of prey games

5.4 Game V3

6 Future Work

Future work could be summarized into three categories. Creature expansions, Board expansions and System Expansions.

6.1 Creature Expansions

Creature expansion to the system could take place in several forms. An expanded neural network system for the creatures should be tested. This could help both species of creatures develop more complex behaviour. An additional change that could be made to creatures is tho provide each creature a line of sight that would replace the entire board as an input. This way the simulation is more realistic to real life and both the predator and prey would have to learn to situations without full knowledge of the board. Another set of experiments could be to simplify the predator's movement and remove the chasing in the x and the y direction. This would reduce the complexity of the network and potentially provide a more even evolution between the two creatures.

6.2 Board Expansions

Board expansions that should be tested include expanding the board for multiple sizes in order to increase or restrict the freedom to move as the smaller the board size, the more interaction there will be between the creatures. Another expansion should add a border to the board which can be used by the creatures as input to give a warning that the edge of the board is there. That way, the behaviour that is developed in V1 of the came could be more direct as both the predator and the prey would know exactly where the board ends. Some type of terrain could also be added. This could include mountains which could take place as a wall where the creatures cannot move, or traps which would cause the creatures to lose if it was stepped on.

6.3 System Expansions

The obvious next steps for this system expansion is to introduce a team dynamic. That is, add multiple predators and prey competing against each other. This could introduce interesting teamwork mechanics such as working together to corner a prey. The neural networks for these predators could come from the same swarm, or more swarms could be introduced. If more swarms are introduced, then sub-specializations could potentially evolve. In case of predators, there could be one that could become the chaser and another one that could be be ambusher. Similarly, for prey there might be one swarm that evolves to be sacrifices for the benefit of the group.

References

[1] F. van den Bergh, A.P. Engelbrecht - http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.142.22

[2] LH. Langenhoven, Evolving Behaviour through Competition Special Project, 2006

[3] C. Scheepers, Coevoution of Neuro-controllers to Train Multi-Agent Teams from Zero Knowledge, 2013

[4] http://www.robocup.org

[5] http://www.fira.net/main/

[6] http://www.cleveralgorithms.com/nature-inspired/swarm/pso.html