Evolving Pursuit and Evasion Strategies in Realistic Worlds

Richard Garside
Computer Science
University of Leeds

March 2001

Summary

This report details the methods and results of a set of experiments into evolving virtually intelligent agents capable of displaying pursuit evasion behaviour. The advantages of randomness in such behaviour was investigated and how the complexity of the world affected agent behaviour was also looked into. The conclusions drawn were found to be similar to the ones expected but some interesting questions were also raised by the results.

Acknowledgements

Firstly I'd like to thank Jason Noble, my project supervisor. His insight into the field has been invaluable. It didn't matter how long a problem had been bugging me for, he always seemed to be able to solve it quickly. I'd like to thank my family for checking up on me and making sure I was doing the work I was supposed to be doing. Lastly, I'd like to thank my house-mates and my friends for giving me plenty of distractions to stop me going crazy.

Contents

Chapter 1

Overview

1.1 Introduction

Pursuit and evasion is an interesting behaviour exhibited by animals in nature; one essential to their survival. Many animals such as cheetahs and gazelles exhibit pursuit and evasion and many different types of each behaviour exist across many different species. It is not only a behaviour exhibited by animals with a predator prey relationship; similar behaviours are shown by animals competing for the same resource, such as gorillas competing for mates. Essentially variation of pursuit and evasion exist in virtually all animals. Many creatures such as horses will instinctively flee upon hearing an unfamiliar noise before questioning it as in the past this behaviour has helped their ancestors survive. Biologists have spent many years in the field studying animals and speculated about how their behaviours developed. There are many limitations to these methods; you can only watch, you are not free to change their conditions so you can only speculate on the causes of what you observe.

Since the creation of the first computers, researchers have tried to use them to model biological systems to try and gain a greater understanding of their function. The first attempts to model evolution were made in the 1950s. Some of these experiments were concerned with understanding the fundamental processes involved in evolution, while others wanted to harness its power as a search method to find solutions to engineering problems.

To bring greater understanding to the theories developed by biologists, simple computer models have been created of worlds inhabited by animals capable of displaying different behaviour. Evolutionary computer methods have been used to change these behaviours based upon selective pressures present in their environment. The findings from a simulation can never be conclusive because the models are simplified and you can never be sure whether they exactly mimic nature. However, because you are free to change whatever you please, computer models can help us investigate animal behaviour in a way not previously possible. A lot of work has been done in this field but it is by no means exhaustive and there is room to build upon it.

Evolutionary computing covers a huge range of fields. For a general introduction to evolution and biological simulations the reader might care to read (Dawkins, 1986). For a more detailed in-depth understanding of the methods involved the reader might also care to read (Mitchell, 1996).

Cliff and Miller's evolutionary simulation of pursuit and evasion (Cliff & Miller, 1996) produced some interesting behaviours. Prey and predator agents controlled by continuous-time recurrent neural networks were evolved and tested in a world that consisted of an infinite plain and simple physical laws. Prey and predator agents of increasing capability were produced exhibiting a range of realistic behaviours. One point of concern is that their evolution runs took about three weeks on an unloaded machine. Another aspect of their work involved growing neural nets from the genetic encoding. This introduced large genetic gaps between viable solutions and it may have hindered evolution in their system. Evolution works best when genetically similar agents have similar fitness scores.

Wahde and Nordahl have attempted to build upon the work of Cliff and Miller by making the problem more realistic(Wahde & Nordahl, 1998a). In the real world there is rarely an infinite plain; a creature's environment is normally confined in some way. Walls were added to their simulation in the hope that this would cause the agents to exhibit more interesting evasion behaviour. However, neither the predators or prey were given any way of sensing the boundary so they would just bounce off it. This appears to be a fatal flaw in their system and makes their results less interesting because it does not mimic nature. Invisible walls are a very artificial thing to add to a simulation of a natural system.

Wahde and Nordahl furthered there expansion of Cliff and Miller's work by investigating protean behaviour in pursuit and evasion (Wahde & Nordahl, 1998b). Protean behaviour is variable and unpredictable behaviour, prey animals often seem to use unpredictable protean behaviour to confuse chasing predators. For example a mouse being chased by a cat may suddenly change direction and this may momentarily confuse the cat giving the mouse enough time to escape. Wahde and Nordahl wanted to see if they could evolve agents that took advantage of protean behaviour. In their results they get behaviours which could be described as protean but only after tuning the fitness function so that it was favourable for evaders not to revisit locations they had previously occupied. They were still left with the problem of how to measure protean behaviour that Cliff and Miller pointed to at the end of their paper(Cliff & Miller, 1996).

Research is being carried out using evolutionary computer methods to create agents that display minimally cognitive behaviour. Randall Beer in his paper (Beer, 1996) and the follow up paper (Slocum, Downey, & Beer, 2000) evolves agents controlled by neural networks capable of a set of simple tasks that he considers to show minimally cognitive behaviour. These tasks among others include orientation in relation to objects, discriminating between objects and short term memory. The agents of (Cliff & Miller, 1996) were only required to act directly to the stimulus they were currently receiving. (Slocum et al., 2000) was able to evolve agents capable of reacting to past stimuli. It would be interesting to see if predator prey agents could be evolved that would continue to react to an opposing agent that had passed out of their visual range.

Ficici and Pollack are also interested in pursuit and evasion (Ficici & Pollack, 1998). They are concerned that the potential of the game has been unrealised. Some of the pure mathematical formulations of the game are closed-ended while others such as Cliff and Miller's version have an opaque solution space because there is no mathematical way of representing agent behaviour. Their version of the problem was simpler, it involved the predator attempting to predict a bit string produced by the prey who attempted to make the bit string unpredictable.

Several variations of the game were also experimented with. It was felt that these simplifications of the problem maintained the essence of pursuit and evasion. Statistical analysis that wouldn't have been possible with more complicated versions of the problem were then used. This allowed some interesting conclusions to be drawn about the effectiveness of the strategies evolved and the effect of the fitness function using both normal evolution and coevolution.

In straight forward simulated evolution the evolving agent, or solution, is evaluated against a fixed fitness function. It is very easy to see if evolutionary progress is taking place. In nature this is not the way things really are; the environment is changing and animals are in direct competition with each other. In coevolution two or possibly more opposing agents or solutions are evolved and evaluated against each other. A noted difficulty with coevolution is that measuring actual progress becomes difficult when the fitness landscape is constantly being changed by the opposing individuals. This is known as the Red Queen effect. A paper written by Cliff and Miller(Cliff & Miller, 1995) describes the problems involved and goes into several methods of measuring the evolutionary progress in different ways. There methods involved several ways of evaluating the changes occurring in the phenotype and genotype of evolved agents from one generation to the next. An agent's genotype is the collection of all its genetic data and its phenotype is the physical and mental characteristics it possesses due to its genotype. The methods they outlined have not completely solved the problem but they do provide a method for understanding the dynamics of coevolution better.

1.2 Objectives

The project involves many practical objectives (see appendix B.1) that must be achieved to make it possible to accomplish my aims. My aim is to replicate the work of (Cliff & Miller, 1996) and hopefully to build upon it. It is worthwhile to duplicate the work of others and it is something commonly done in most sciences, often at great expense. It is interesting and possibly worrying that computer models are rarely duplicated by people who are trying to build upon findings from those models. These findings could be phenomena caused by arbitrary parameters or by bugs in the programming; without replicating the model for yourself you can never be sure.

There were two additions to their work that I was interested in exploring. The first was the addition of obstacles to the virtual world. Intuitively, we might suppose that these will be a hindrance to the predator and beneficial to the prey. You might expect the prey to attempt to remain as close as possible to the obstacles to hide from the predator. This could be considered similar to (Wahde & Nordahl, 1998a) but their agents were unable to sense their world's boundaries so would never be able to make full use of them. Ideally predator agents might develop short term memories that allow them to continue following prey after the prey goes behind an object that obscures it from the predator's view.

Cliff and Miller had hoped to be able to evolve protean behaviours but were unable to prove they had done so because they had no way of measuring it. Wahde and Nordahl have studied this (Wahde & Nordahl, 1998b) but as they had no definitive measure of protean behaviour their results were not conclusive. It is possible that the complexity of their systems is what made measuring protean behaviour so difficult. Each system had many sources of randomness that may or may not have been linked with any possible protean behaviour. For instance (Cliff & Miller, 1996) used random noise in the inputs of their agents. By removing as much randomness from the simulation as possible and introducing a single random input that can be turned on or off I hope to be able to do comparative tests that tell us whether this input has been beneficial or not. This is a different approach to measuring protean behaviour; instead of trying to quantify the behaviour we are looking at the cause of it. We will be able to see if the random input is beneficial to either predator or prey and we should be able to assess whether this method of measuring protean behaviour is a valid and useful one.

Random inputs were used in (Noble, 2000) as an input for agents evolving fighting and signalling strategies. The input was not found to be widely used but this input may prove more useful for agents in a predator prey situation. If this random input does prove to be useful you would expect prey to benefit from the input more than predators. Typical protean prey behaviour might include dodging randomly left and right or suddenly changing direction.

Chapter 2

Method

2.1 The Simulation Environment

The main concern of the simulation environment is to provide a close to realistic environment for the predator and prey agents to evolve in, while remaining simple enough to minimise computation time. For these reasons the virtual world shall be a simple two dimensional one. It may be interesting as a future development to model a three dimensional world but the extra dimension would make results harder to evaluate and would considerably increase necessary computation time.

The virtual world involved an infinite flat plain where movement was governed by a simple force vector model. A vector representing thrust and momentum were limited by a scalar representing both drag and resistance. This produced realistic looking acceleration, deceleration and turning effects. All objects in the world are simple circles. This simplifies any maths involved, such as calculating the distance between objects.

In this way my simulation environment is very similar to the one used by (Cliff & Miller, 1996). The main addition has been the inclusion of objects that the agents are able to sense. When an agent collides with an object its momentum is halved and its thrust is reversed. If going fast enough they can find themselves in a situation where they are stuck to the obstacle, if this is the case it takes a concerted effort to break free. This makes a collision with an obstacle a considerable disadvantage. The world can be initialised to contain any number of obstacles covering as large an area as required.

2.2 The Agents and their Senses

The sensory system employed by my agents was based on the system used by (Beer, 1996). Each of his agents had an array of distance sensors. When a ray from a distance sensor intersected an object, an input inversely proportional to its distance was injected into the corresponding sensory neuron of the agent. My agents each have an array of five distance sensors that functioned in the same way, each separated by 10 degrees. Each distance sensor has a maximum visual range. If it fails to intersect an object in this range it returns zero. Simple parametric maths were used to calculate if an intersection occurred and the distance to this intersection(web, 2000).

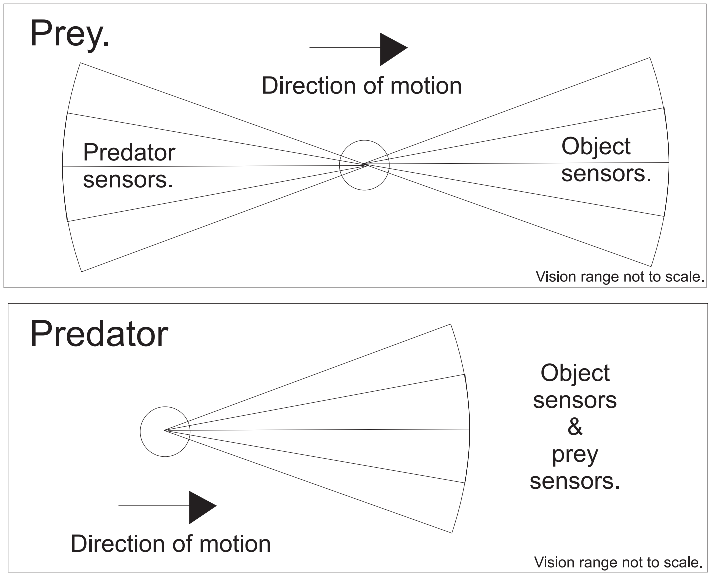

Figure 2.1: An illustration of the placement of sensors on the agents.

The world consists of two different types of objects. One being the moving agents and the other being the stationary obstacles. Agents could have sensed both of them as the same and been left to distinguish the two based on movement. To make their task easier they were given two arrays of distance sensors, one that sensed other agents and one that sensed obstacles. The biological rationale behind this is that many creatures have eyes that consist of sensors sensitive to different colours of light. It is perhaps slightly artificial but as we want to produce interesting behaviour we don't want to make their task too hard.

In similar work by (Cliff & Miller, 1996) the position of the agent's visual sensors has been evolved along with the control system. The idea being to evolve body and mind in unison. This method has not been followed because we want to simplify the evolution task and concentrate on evolving the mind and behaviours. The visual sensors have been kept in a fixed position that would intuitively seem to benefit the task they are being evolved to fulfil.

As there are two types of agents, each evolved to do a different task with different requirements, the placement of the distance sensors varies with each agent (see fig. 2.1). The obstacle sensors of both agents face forwards so they can avoid collisions. However the predator's agent sensors point forwards so they can observe the agent they are chasing and the prey's agent sensors point backwards so they can observe the chasing predator. When (Cliff & Miller, 1996) evolved the position of the visual sensors optimum predators and prey were found to have evolved in similar positions. Their agents only had each other to contend with which is why such specialised eye position could evolve. As my sensors are specialised they too can be in specialised positions. In nature the eyes purpose is more general and its position has to be a compromise that makes it suitable for all its tasks. In predators the eyes are usually located close together facing forwards giving them a sharp but small field of view. The eye's of prey are often spaced much further apart giving them a wider but not as sharp field of view.

In nature predator and prey often have different physical capabilities. I implemented this in my agents. I made the predators faster but less manoeuvrable than the prey. This meant the prey would never be able to leave the predators standing but would be able to out manoeuvre them if they developed that behaviour. I hoped this would lead to more interested pursuit evasion behaviour. A similar method was used by (Wahde & Nordahl, 1998a) to ensure interesting contests.

The agents are driven by three outputs: forward, left and right. This is contrary to a popular approach where agents similar to that used by two-drive-wheeled mobile robots where there are only two outputs: left and right. Equal strength in each output drives the robot forwards, any inequality between the two will drive the agent to the side of the weakest output. This method was used by (Cliff & Miller, 1996). In my three output method, the forward output affects the agents forward force while the difference between the left and right output causes the agent to change its forward facing direction. It is felt that this is a more natural way of thinking about movement, but that it shouldn't affect the results greatly as it allows the same range of movement.

2.3 The Agent Control Architecture

I considered three different control architectures; the first being an instruction list based approach. This would involve a list of actions matched with a list of events. The advantage of such a method is that it is easy to understand the agent's evolved behaviour. The disadvantage is that the set of behaviours is limited and it is not similar to nature. Also, how do you decide upon the list of actions and events? Any decision made at this stage is going to greatly influence the evolved behaviours. This method was used by (Noble, 2000) where it was important to understand the strategies of evolved agents. Selective pressure was built in to keep instruction lists short. This linked with both the idea that cognition is expensive and also made the lists easier to understand by making it expensive for agents to keep instructions that weren't useful.

The second method was to use a genetic programming approach in a language such as lisp. Such an approach leads to a greater range of possible behaviours while still maintaining the ability to look openly at the agent's strategy. The down side is that the strategy encoding will be harder to understand than a simple instruction list. This method was used by (Reynolds, 1994) in a symmetrical pursuit evasion problem in the form of a game of tag where agents took turn at being predator or prey depending on who was 'it'.

The third method was to use a neural net. The disadvantage of such a system is that their evolved internal function is difficult or can be impossible to understand. There are many advantages; they have been found to produce a wide variety of behaviours, a greater range than instruction lists or genetic programing. They can also be encoded in a way amenable to evolution and mimic the nervous system of animals in nature. There are many types of neural nets but the most effective for this task have been found to be continuous-time recurrent neural networks (Slocum et al., 2000). This is partly because their recurrent connections make it possible for them to maintain state. This potentially allows them to have a short term memory. This control method was chosen. The code written for the neural nets was based on that used by (Yamauchi & Beer, 1994).

2.4 The Genetic Encoding Method

Bit strings are a prevalent method of encoding information for genetic algorithms. This is partly due to the similarities to both the 4 chemical code that DNA consists of and the two state code used by computers. It does however separate the genotype from the phenotype and introduces problems with crossover and mutation. For these reasons a more direct mapping was used. A neural net can be represented by a list of floating point numbers that represent the weights in the network. To make the nets easier to analyse any weight whose magnitude was less than 0.5 was made to be zero.

2.5 The Genetic Algorithm

Agents compete in contests where they are evaluated by a fitness function related to their task. This score is used to select which agents reproduce to form the next generation. To save time in the evaluation of agents each agent is evaluated against the highest scoring opponent from the previous generation. This technique is known as LEO (Last Elite Opponent) evaluation and was used by (Cliff & Miller, 1995).

There are many different selection methods (for details see (Mitchell, 1996), chapter 5.4). The roulette wheel method was used in the experiments presented, this was found in preliminary runs to be a simple but effective method. The chosen fitness function was found to have a great bearing on the success of an evolutionary run. Fitness functions were evaluated using short test runs to determine which would promote rapid coevolution of pursuit and evasion. Several were evaluated before a successful one was found.

In one fitness function I rewarded the predator for approaching the prey and gave it a bonus if it caught it, the prey was rewarded for the length of time it managed to avoid capture. This was based on the fitness function used by (Cliff & Miller, 1996). It was not completely clear from their explanation how exactly their fitness function worked , my version must have been subtly different because mine failed to produce the expected results. Predators found that by following a small circular path they spent half their time approaching the prey. This behaviour gave them a higher fitness score than other behaviours present at the time and caused the simulation to converge on a local optimal peak in the fitness landscape, one they never escaped from. As all the predators performed badly, most of the prey survived for the entire contest time so they were not strongly influenced by any selective pressure.

The most successful fitness function was as follows. Predators were awarded marks for every time unit they were near the prey. This score was the reciprocal of the distance between the agents squared and then multiplied by a constant. This gave them highest marks for being right next to the prey with the score rapidly decreasing as they got further away. The prey's fitness function was similar except you scored maximum points for being a safe distance away with the points rapidly decreasing as the distance got less.

Each agent was evaluated five times and their average fitness score was taken. The predator started in random position in the starting area and the prey started in the middle (see fig 2.2). To minimise fitness variations in a population being caused by randomness in the simulation, a set of five random positions were generated at the start of each generation. Each predator then began from these positions in each of the trials.

A separate population of predators and prey were maintained, each population consisted of one hundred individuals. Each new individual in a population was created from two parents using one point crossover with a 0.25% chance of any weight being mutated. The size of this mutation was plus or minus an exponential number. I experimented with several different mutation rates, higer ones were found to cause the population to vary randomly with no control while lower ones caused the population to converge too quickly. This rate was found to allow a steady convergence with the chance of finding new solutions in the surrounding search space (see Appendix C for further details).

In each evolution run typically of 1,000 generations every agents genetic data was saved along with their fitness score. The mean, minimum and maximum fitness of each generation was also saved along with the standard deviation of the genetic data in a population. The standard deviation was worked out using the average weight of each agent's neural net. It is recognised that this is not a definitive measure of the genetic spread but it was easy to calculate and gave a good indication of the level of convergence in a population.

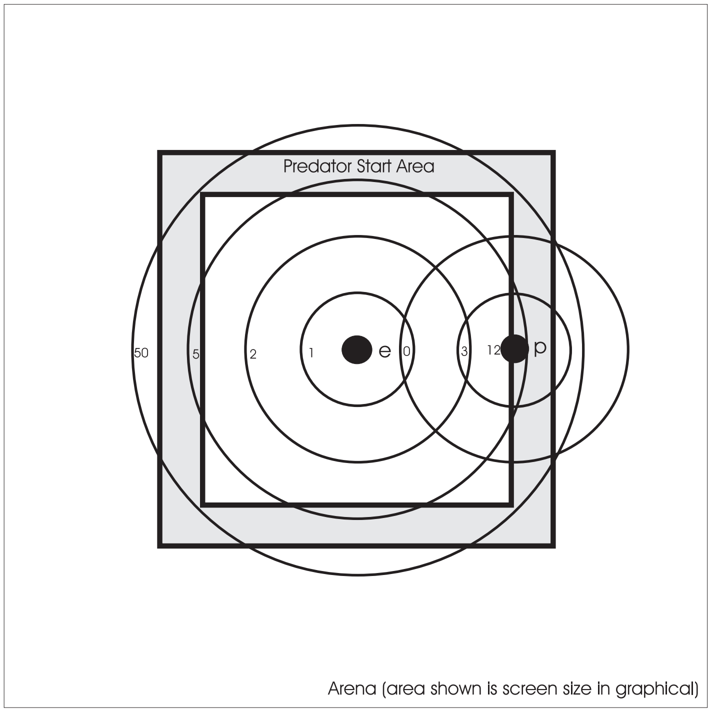

Figure 2.2: This shows the arena where all contests took place. The predator is marked with a p and the prey is marked with an e. The circles surrounding them show there scoring regions and the numbers show the points associated with those regions as they were awarded in the simulation. The actual numbers are dependant on the constants chosen but as this is relative it is not important.

2.6 The Contests

A contest lasts for 20,000 time units. One hundred time units can be thought of as simulating one second of time, meaning that a contest lasts for twenty seconds. A contest is also terminated if the predator collides with the prey.

The predator is placed at a random position in the square donut starting region (see fig. 2.2). This places it toward the edge of the prey's avoidance area and means the prey is out of the predators scoring area. The location and position of the starting region was found to give both agents a reasonable chance weighted slightly toward the predator as the most interesting contests were found to be the ones where the prey struggles to escape from the predator.

2.7 Evaluation Programs

A number of programs were written to analyse the results created in the evolution runs.

graphical The graphics program is the main analytical tool. It is possible to load any agent produced from any population, view their fitness scores and watch a contest in a graphical representation of their world. Contests can be viewed from a number of different views. There is a two dimensional view from above; this view is useful for analysing the contests because everything can be clearly seen. There are also several three dimensional views of the world from various positions. This does not add to the model but it does make the agents look more interesting, its main purpose is to demonstrate the possible entertainment applications this technology could have. OpenGL was used to produce the graphical program. OpenGL is a portable 3D graphics library originally produced by silicon graphics. Readers interested may care to read (Wright & Sweet, 2000).

The graphics program also uses gnuplot to produce graphs of the various statistics produced by the program.

A selection of other small programs have been written that create more statistics that can be viewed in gnuplot.

plotXY Produces graphs of the paths taken by agents in a contest along with graphs of their speed and turning speed. This is a useful tool in analysing the strategies of evolved agents.

checkRandom Produces graphs of the average weight to the random input for the experiments into protean behaviour.

Chapter 3

Results

A set of four experimental trials were carried out. Each experiment consisted of five identical runs starting with a different random population, each run lasted one thousand generations. The simplest experiment run took eight hours to complete, as the complexity of the test conditions increased the length of a run escalated to just over ten hours. The four experiment conditions were as follows.

Experiment A The first experiment was the control. There were no obstacles in the world and agents had five distance sensors able to sense the opposing agent as their only input.

Experiment B In this experiment agents were given an additional input to those of experiment A. This input provided the agent with a random number stream with weights initialised to zero in the initial population.

Experiment C In this experiment the world included five obstacles that were randomly positioned at the start of each generation. Agents were able to sense objects with an extra five distance sensors specific to this task. Agents in this experiment did not have a random input.

Experiment D This experiment is a combination of experiment B and C. The world included obstacles as described in C and agents had access to a random input as described in B.

3.1 Experiment A

A huge amount of data has been produced in all of these experiments and it has not been possible to analyse all the individuals created. However, it was found that population convergence meant that a population would exhibit a very small range of similar behaviours. This meant that it was only necessary to view a small select range of individuals in detail.

The predators from all the runs in experiment A expressed very similar behaviour. This behaviour developed very quickly (usually in less that 40 generations) and didn't change considerably for the rest of the run. It was found to be sufficient and predators would catch the majority of the prey in the allowed maximum 2000 time units.

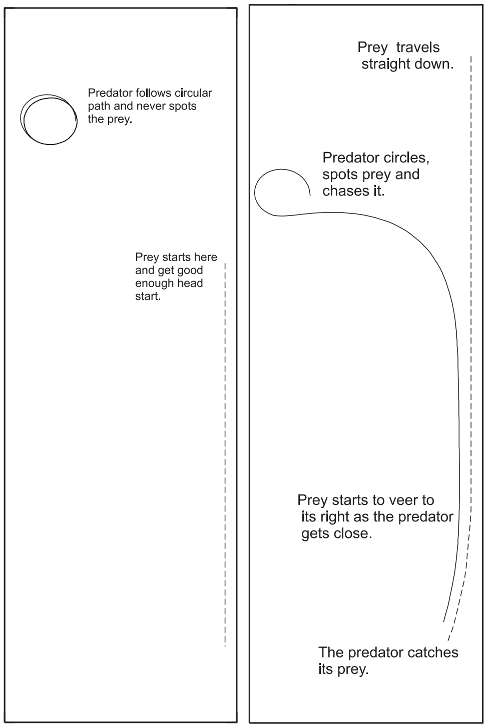
Predator follows circular path and never spots the prey.
Prey starts here and get good enough head start.
Prey travels straight down.
Predator circles, spots prey and chases it.
Prey starts to veer to its right as the predator gets close.
The predator catches its prey.

Figure 3.1: The path of a typical predator following a prey agent with a simple go straight then turn when threatened strategy. In the first plot the predator fails to spot the prey, in the second it spots and catches the prey.

The typical predator behaviour (see fig. 3.1)consisted of following a small circular path until the prey was spotted. The direction of this path never varied in an individual but did sometimes across the population and often in different runs. In this first stage the predator is not aware of the prey's position and is trying to locate it. When and if the predator sees the prey it chases it. It is able to correct its heading if the prey changes direction and will eventually catch the prey due to the predator's speed advantage. Over evolutionary time the only noticeable change was the size of the predators searching stage circle. They tended to evolve to use smaller circular paths that would allow them to scan their whole surroundings quickly and therefore locate their prey sooner.

From this it would seem that there is one, or a family of close to ideal solutions for predators that are easy to find using a genetic algorithm. These solutions appear to be capable of remaining stable from generation to generation for long periods of time. The same can not be said for the prey behaviours that were evolved. There were a far greater range of different behaviours, some of the better ones remained in a population while others were often not present as long and were replaced.

One of the most common starting prey behaviours was to run in a straight line from their starting position. This in three quarters of cases was a good move because it gave them a good head start before the predator located them. Whether it was a beneficial move or not depended on the predators starting location and which way round the predator circled when looking for it's prey.

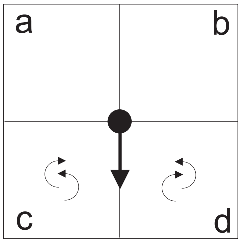

Figure 3.2: The quadrants of the arena.

If the predator begins in either quarter a or b then the prey normally gets a good enough head start and can escape from most early predators. If the predator begins in either c or d then in depends whether it rotates toward the centre or not; if it does then it will spot the prey and catch it quickly. This is the worst case for the prey. If the predator rotates away from the centre then the prey will get a reasonable head start and may outrun early predators. This starting approach was common at the start of runs but didn't always survive for long because it was not general to all predator positions.

A second, less common, but more successful starting strategy was for the predator to spin in a small circle until it saw the predator and then to run in a straight line away from it. The delay caused by locating the predator made the possible head start less but meant that there always was a head start and that you never ran into the predator. If this method evolved then it was likely to survive to the end of the run.

Some prey just followed the path dictated by their starting strategy, others developed a cornered strategy that they employed when the predator got too close. This normally involved veering to one side, increasing the turning angle as the predator got closer. Such strategies were common in predators that went in a straight line from the start but rare in those that located the predator.

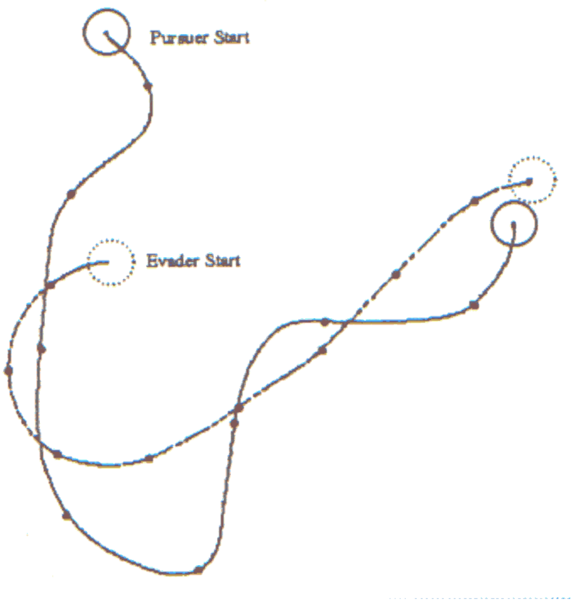

Figure 3.3: Here is an example of a plot of agents taken from Cliff & Miller, 1996.

Paths similar to those seen in (Cliff & Miller, 1996) were never seen (see fig. 3.3). Their prey agents followed a curved path turning both left and right seemingly randomly to evade the predator. If my agents used turning to evade agents they would only turn one often predefined way. The differences seen here could be because (Cliff & Miller, 1996) used inputs with random noise or it could be a behaviour caused by the differences in our vision systems. My agents were seen to wobble very quickly and very slightly when they got closer to an opposing agent; this would help them scan the agent's position more accurately. The larger wobble seen in (Cliff & Miller, 1996) could be similar to this.

3.2 Experiment B

In experiment B a random input was introduced. This was not found to cause any new behaviours such as random changes in direction. All the behaviours found had been seen in a similar form in experiment A. An early prey agent was found to randomly change velocity but this was not beneficial as it meant the predator simply caught up when it slowed down.

Although there is no visible change in agent behaviour due to the random input, it is obviously getting used as can be seen from (fig. 3.4). Observational experiments were run with and without the random input and the prey showed no noticeable change in behaviour. It is not clear what this input is getting used for but it is obviously beneficial to prey and not to predators .

3.3 Experiment C

The inclusion of obstacles made understanding the agent behaviour far more difficult. It was necessary to observe them far more times than had been needed in experiments A and B. This was because of the difficulties with associating behaviour with the correct stimulus. It was often not immediately clear if an agent turned because it could see the opposition or because it could see an obstacle; several observations would usually clear this up.

The fitness function did not make any mention of the obstacles so any behaviour related to

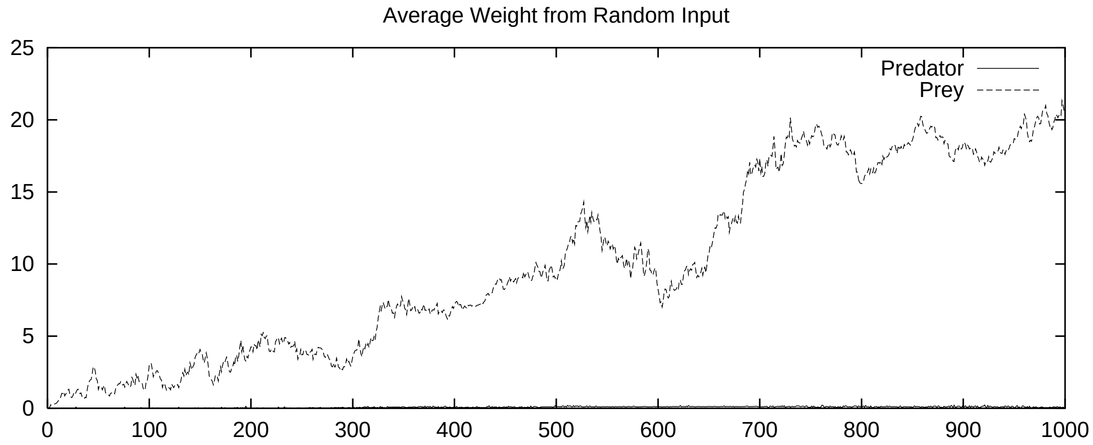

Figure 3.4: This graph shows the increasing use of the random input in prey agents over evolutionary time. The input is hardly used at all by the predators. This is a graph of trial three, but all five showed similar results.

them would be because of how interacting with them affected their performance in relation to pursuit and evasion. A great range of behaviours related to the objects was observed. Most of the strategies included elements that were similar to those seen in experiments A and B.

The conditions of this experiment made it much harder for predators to find a successful strategy. They would eventually find the same circle then chase strategy found commonly in experiments A and B. Often their apparent visual range was quite small but would grow with time slowly. If they lost site of a prey agent that was obscured by an object they would start circling again. This served two purposes; it stopped the predator crashing into the object that was blocking its view and it gave it a chance to find the agent it had lost sight of. It didn't always work but it was reasonably successful. Some predators showed slightly less crude object avoidance but none of them were universally successful; they were quite often one sided and only able to avoid objects that occurred on one side.

The difficulty suffered by the predators in the object infested contest area made things a little too easy for the prey. A virtually universal starting behaviour was just to go straight from the starting position. Many of the prey completely ignored both the predator and any objects.

The lack of selective pressure made this behaviour sufficient.

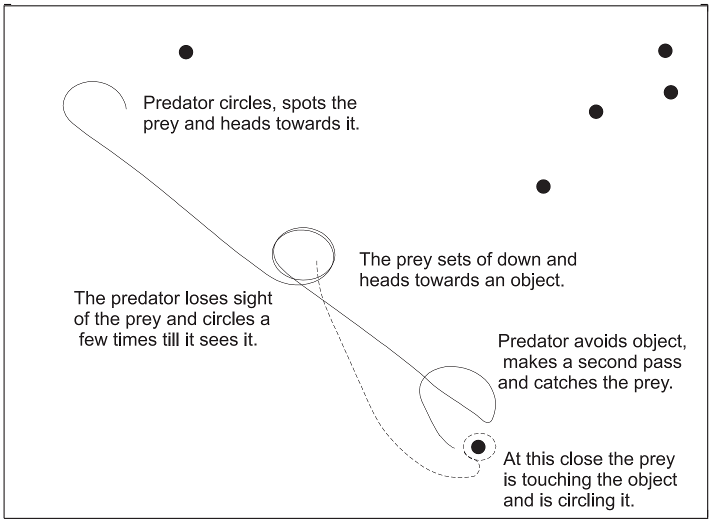
Predator circles, spots the prey and heads towards it.
The prey sets of down and heads towards an object.
The predator loses sight of the prey and circles a few times till it sees it.
Predator avoids object, makes a second pass and catches the prey.
At this close the prey is touching the object and is circling it.

Figure 3.5: Plot of a reasonably well evolved predator against a prey with object homing tendencies.

Some prey agents exhibited object homing tendencies(see fig. 3.5), they would turn and head toward objects and then crash into them. Some of these would then free themselves while other would force themselves into the object; this gave them a good chance of being completely hidden from the predator. Some prey were able to avoid the obstacles from some approach paths, this worked well if a predator was chasing them because there was a chance the predator would crash into the object.

Not many prey were able to react sensibly to the predator. Some entered circles as predators approached, similar to agents had in experiments A and B but this was rare. The most impressive behaviour found was a predator that when chased by a predator would head toward an object and then quickly turn. In many cased the predator was unable to react quick enough and would crash.

Although successful solutions were found they were not always found and some runs went through many successive generations of very low predator scores. Any species in nature they did so poorly would probably go extinct. As runs where this happened never managed to find very successful solutions, extinction in the form of early program termination would be a sensible way to deal with this situation.

3.4 Experiment D

The inclusion of a random input to agents in the world with obstacles had no visual effect on their behaviours but the random input was obviously been used (see fig 3.6). This situation was the same as occurred in experiments A and B. Most of the behaviour was similar to that found in experiment C and it was again the case that the difficulty of the problem was making it hard for the predators which in turn was meant the selective pressures on the prey were quite small.

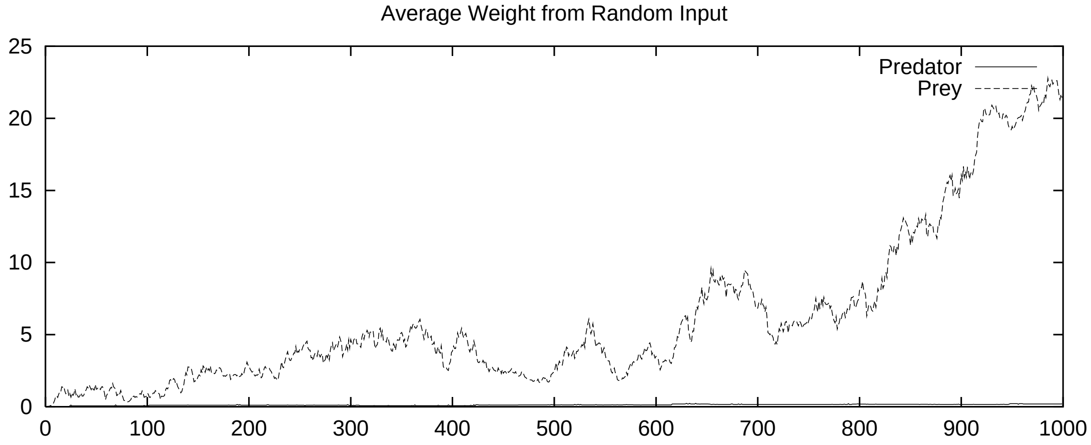

Figure 3.6: This graph shows similar usage of the random input as of that found in experiment B. This is a graph of trial three, but all five showed similar trends.

Some more advanced object avoidance was observed in predators in one run but it is unlikely that this was caused by the random input. When they approached an object they would slow down, turn the amount necessary to avoid it and then continue. Several similar but not quite successful approaches were observed.

One prey agent was found that behaved very strangely. It seemed to have slight object homing tendencies but would occasionally quickly change direction with the cause of this not being clear. This was not always done in a sensible way. Sometimes it would turn 180 degrees to head straight toward a chasing predator. This agent was found midway through a generation, its species was not found to survive for many generations even though its behaviour was interesting and could have lead to a more successful strategy. When it was denied the random input it acted in a similar way although it seemed to spin more as if the random input was acting as a limiting input. No other agents acted noticeably different without the random input.

Chapter 4

Conclusion

This report has detailed the methods used and has presented a small selection of the results obtained. Many things have been learnt and all the questions originally posed have been addressed. These in turn have raised new questions that it is felt might be worth investigating.

The work of (Cliff & Miller, 1996) was successfully replicated and some interesting differences in agent behaviour was observed. There are many possible reasons for this; such as the different vision systems, the different propulsion systems, random noise inputs in (Cliff & Miller, 1996) or the fitness function used. Further tests would be necessary to confirm this. Both systems have successfully evolved pursuit evasion behaviour in agents and this is what was hoped.

From the increased use of the random input in the prey's genotype we can conclude that protean behaviour can be used to the advantage of prey. From its lack of use in predators we can also conclude that it is not beneficial to prey. These are the sort of results we would expect but the extremely large difference between the two was surprising. It is unfortunate that no conclusive evidence of this input was seen in the observable behaviours. It is thought that it is probably present but that more sophisticated methods of viewing and classifying agent behaviour are required.

Adding obstacles to the environment did indeed make the world more realistic but it also made the problem of developing pursuit evasion behaviour more difficult. It is particularly interesting that the lack of selective pressure on the prey meant that they developed very weak behaviours. Some interesting behaviours were found. Predators that could find the prey and sometimes avoid the obstacles would eventually evolve and the prey showed a range of interesting although not very robust behaviours. Some prey found the objects to be beneficial and would try and stay as close to them as possible, some would avoid them and some would try to lure the predator to crash into them. Due to the difficulty of the task a more successful evolution strategy might be to increase the complexity of the world over evolutionary time. Start with no objects, let the agents develop pursuit evasion strategy then slowly add objects and see if they manage to evolve additional strategies to deal with them.

The obstacles were not included in the fitness function leaving agents to deal with them as they found to be most beneficial to their task. It is interesting to note that different ways of using the objects evolved in the prey agents.

Chapter 5

Evaluation

5.1 The Simulation Environment

A very neat, simple and efficient simulation environment was created. It met the minimum objectives and went on to include obstacles to make it more realistic. It was well planned and its simple object orientated design made it easy to make changes that were found to be necessary after initial test runs. More time could have been spent on making it more efficient but it was found that the planning involved had already made it quite efficient and any more time spent on speeding up the code would have been a false economy.

5.2 The Agents Control Architecture

A number of different architectures were evaluated and the one chosen was found to be successful for the chosen task. As suspected the neural networks have been too difficult to understand to be analysed but this was not the focus of the project. There is a vast amount of data stored that could be analysed further and a greater understanding of the neural nets evolved would be very interesting and could prove useful.

5.3 The Evolutionary Algorithm

The evolutionary algorithm chosen was found to work successfully and the large range of interesting results generated are proof of this. The nature of coevolution means that there was no obvious steady increasing fitness for individuals. More work on measuring the coevolutionary dynamics of the system are necessary but as the scientific aims of the project do not rest with that it was felt that it would be wasteful to spend too much time on it.

The selection method used was found to work fine but more initial experimentation could have been done using different methods to see if any were more suited to this task. Much work has been done by others in this field and there is no obvious advantage to any particular method. The method was chosen because it could be implemented very efficiently.

5.4 The Graphical Package

The graphical package met all it's minimum requirements and more. It was an invaluable aid when analysing the results. The object orientated nature of all the programs meant that most of the code needed to run it was also used by the evolution program. This meant that the graphical program was also an invaluable testing tool to see if the simulation environment was working as it should.

Although the graphical program also has features for viewing the world in three dimensions, this aspect of the program could have been enhanced. The third dimension did nothing to help with analysing the agent behaviour and this view was not used. The purpose of this enhancement was always for entertainment purposes and more time could have been spent perfecting this.

5.5 The Scientific Aims

The questions originally posed have all been investigated successfully. Not all the answers are conclusive but there are several interesting points that could be investigated further. With the evolutionary algorithm producing so much data(approximately six giga-bytes), it is felt that a lot more information could be found in it and that this is worth investigating. There are so many questions that this data and similar could help answer and I don't think the problem is making the data, it's analysing it effectively and in a empirical way.

Chapter 6

Further Work

There were many questions raised by the results obtained in my experiments and it is felt that some of these questions require further investigation. In particular doing more runs using agents with different physical properties to explain the differences between the behaviour evolved in this piece of work and that evolved in (Cliff & Miller, 1996). It is felt that the physical properties of agent's bodies and how they see the world could greatly affect their behaviour and that this is worth investigating.

A lot of time was spent watching agent contests of elite evolved agents trying to make sense of their behaviour. Being able to watch them in real time was a great tool but further enhancements to the tool such as a better way of visualising the neural net could be developed. Also as many agents displayed very similar behaviour under set circumstances it might be possible to use this as the basis of an agent behaviour classifying system along with methods developed in image processing for describing general shapes and paths. This is an area that could be researched further.

The quality of behaviours evolved in the world with objects was disappointing and further work could be carried out to make this task one that developed in difficulty, rather than dropping the agents straight in the deep end without being able to swim.

Also, the results obtained by the protean experiments were certainly very compelling but without understanding the use of the random input we can not draw any definite conclusions.

Further work into analysing the function of the neural nets needs to be carried out and for this more tools are needed to visualise them.

Lastly it would be nice to make the graphical display program more elegant. It serves it's purpose but it is a work in progress and not really complete.

Appendix A

Personal Reflections

I have thoroughly enjoyed doing this project and I am very pleased with my results. My only real regret is that because the programming and evolution runs took so long I did not have as much time as I would have liked to analyse my results. This was a lot more involved than I had anticipated and I should have put more time aside for it.

It was very difficult and frustrating to find a successful fitness function and I was too keen to believe that one was working based on very weak examples of behaviour. A lot of time was wasted on long runs testing useless fitness functions. I soon learnt that short test runs were very important.

Runs take a long time. I spent approximately 130 hours on a GHz processor machine doing useful experiment runs. That doesn't include failed experiments with unsuccessful fitness functions. If I had been able to start these sooner I would have been a lot better off. Runs can also generate a lot of data. I managed to completely fill my hard disk and crash my computer. There was much panicking when it wouldn't boot up and much relief when I finally fixed it. Lastly I wish I'd kept more backups, they were never necessary but they almost were and it would have been a nightmare if I really had lost everything.

This project has been a challenge. I'm quite sad that it's over and hope I will be able to continue the research I have started in my own time.

Appendix B

Practical Objectives and Deliverables

B.1 Practical Objectives

- Create a simulation environment for the agents where they can move around and sense their surroundings.

  This will be a simple virtual world following a simple physics model in which the agents display their behaviour. As contests in this world will be run repeatedly many thousands of time, efficiency will be very important.

- Develop a control architecture that allows a good range of possible strategies and is amenable to evolution.

  For the agents to display behaviour they must have some sort of control mechanism. Possibilities for this include human control, a list of instructions and conditions or a neural net. The chosen method must be amenable to evolution and allow a good range of strategies.

- Design and run an evolutionary algorithm that selects higher agent fitness and assess several different evolutionary algorithms.

The main part of my project will be to design an evolutionary algorithm that creates predator and prey behaviour. To make this effective I will have to assess several different algorithms. Of particular interest will be the chosen fitness function. I may decide to start with a simple fitness function such as obstacle avoidance before progressing onto the co-evolutionary predator prey algorithm.

- Develop a graphical package to display the evolved behaviours and statistical information.

  This objective ties in with the graphical modules I have taken at university. The program will allow you to view the statistics generated in an evolutionary run and to view contests between saved individuals to see how effective the latest run has been.

B.2 Deliverables

The deliverables for this report were previously agreed to be.

- This report.
- All source code.
  This can be found at http://www.garsonix.co.uk/pursuit-evasion/
- The graphical display program with examples of genetic data to show.
  Again, this can be found at http://www.garsonix.co.uk/pusuit-evasion/

Appendix C

Mutation Rates

A number of preliminary runs were done using a simple avoiding obstacles fitness function. The following graphs were obtained and used to decide upon the mutation rates used in the final program.

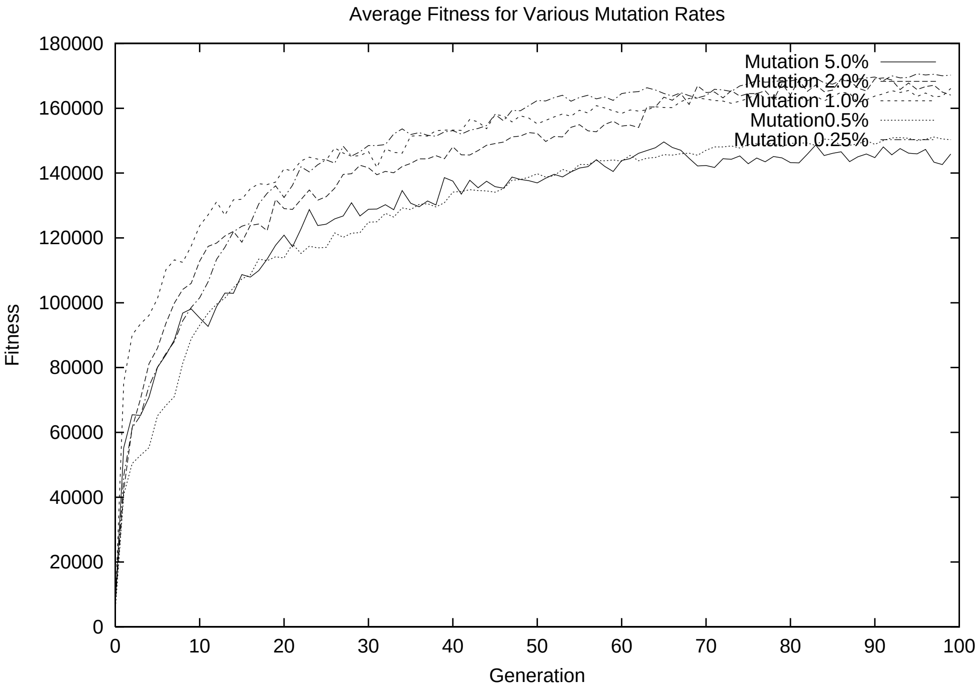

Figure C.1: A comparison of average fitness for various mutation rates. There is little difference between them. A mutation rate of 5.0% performed slightly worse but this was closely followed by 0.5%. The difference involved was not seen to be significant.

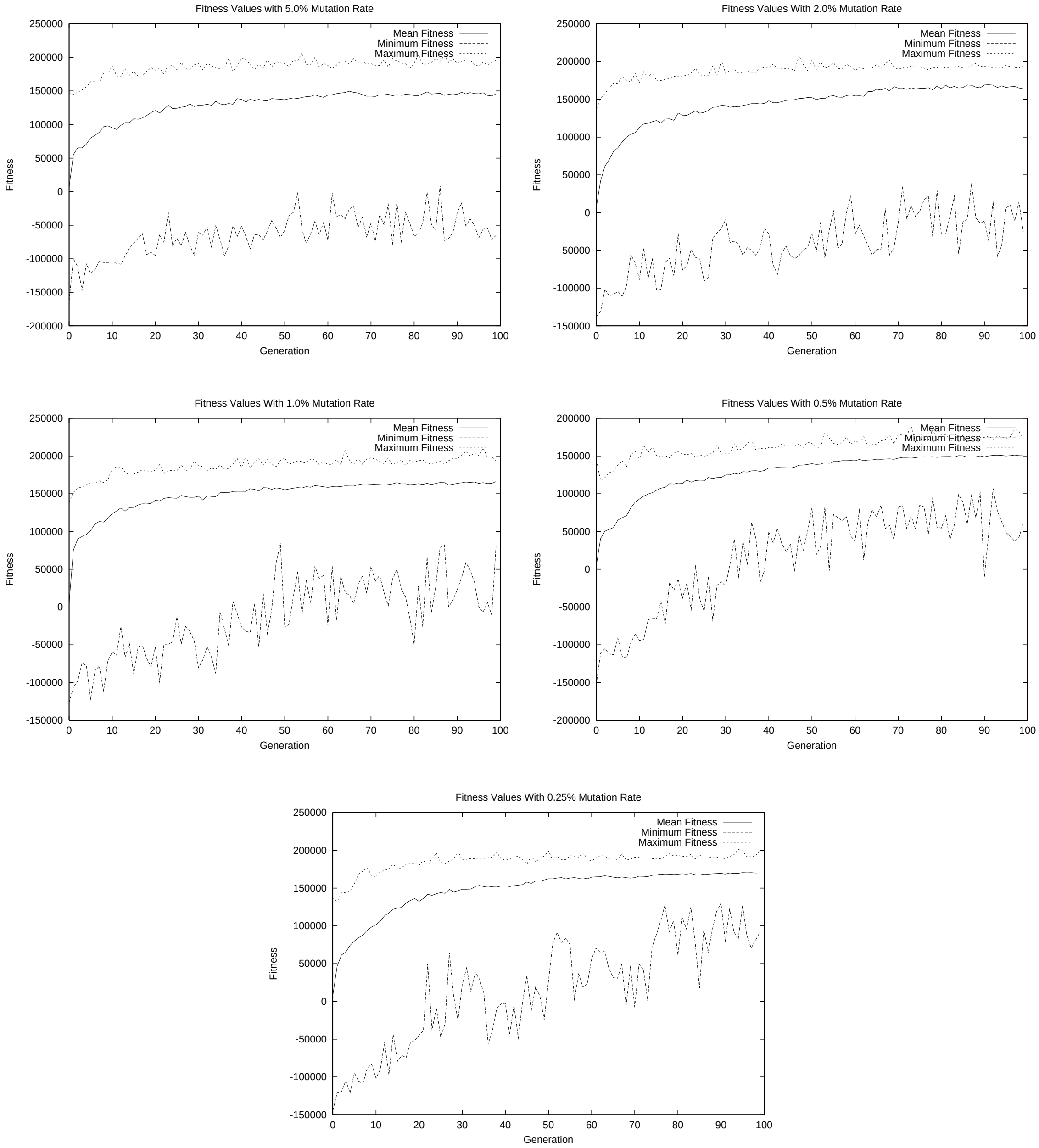

Figure C.2: Graphs detailing the fitness spread of populations using different mutation rates. All values are averaged from five runs. As the mutation rates get lower the range of different fitness values decreases and the worst agent in a population get generally higher fitness scores.

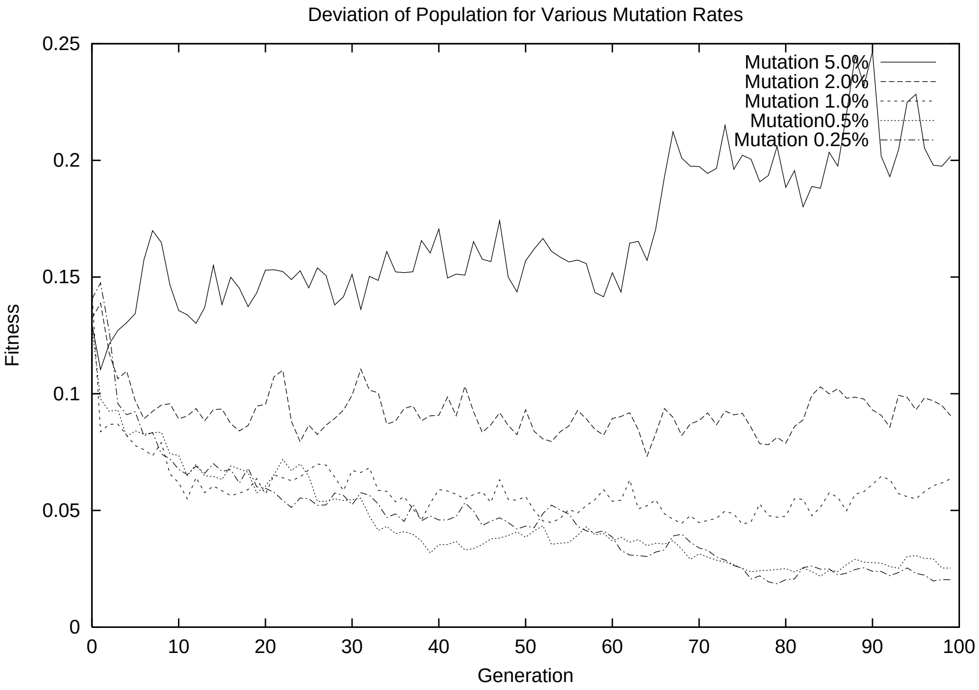

Figure C.3: The average weight in a agents neural net was taken as a rough indication of their genetic position. The standard deviation for each population was calculated to estimate the population's spread. As the mutation rate is lowered the population convergence increases. With a mutation rate of 5.0% the population appears to be quite unstable.

Bibliography

Beer, R. (1996). Toward the evolution of dynamical neural networks for minimally cognitive behavior. In Maes, P., Mataric, M., Meyer, J., Pollack, J., & Wilson, S. (Eds.), *From Animals to Animats 4*, pp. 421–429. MIT Press / Cambridge, MA.

Cliff, D., & Miller, G. (1995). Tracking the red queen: Measurements of adaptive progress in co-evolutionary simulations. In F. Morán, A. Moreno, J. M., & Chacón, P. (Eds.), *Advances in Artificial Life: Proc. Third Euro. Conf. Artificial Life*, pp. 200–218. Springer - Verlag.

Cliff, D., & Miller, G. (1996). Coevolution of pursuit and evasion II: Simulation methods and results. In Maes, P., Matarić, M., Meyer, J.-A., Pollack, J., & Wilson, S. (Eds.), *From Animals to Animats 4*, pp. 506–515. MIT Press / Bradford Books, Cambridge, MA.

Dawkins, R. (1986). *The Blind Watchmaker*. Longman, Essex.

Ficici, S., & Pollack, J. (1998). Coevolving communicative behavior in a linear pursuer-evader game. In Pfeifer, R., Blumberg, B., Meyer, J.-A., & Wilson, S. (Eds.), *From Animals to Animats 5*, pp. 505–514. MIT Press / Bradford Books, Cambridge, MA.

Mitchell, M. (1996). *An Introduction to Genetic Algorithms*. MIT Press / Bradford Books, Cambridge, MA.

Noble, J. (2000). Talk is cheap: Evolved strategies for communication and action in asymmetrical animal contests. In Meyer, J.-A., Berthoz, A., Floreano, D., Roitblat, H., & Wilson, S. W. (Eds.), *From Animals to Animats 6: Proceedings of the Sixth International Conference on Simulation of Adaptive Behavior*, pp. 481–490. MIT Press, Cambridge, MA.

Reynolds, C. W. (1994). Competition, coevolution and the game of tag. In Brooks, R. A., & Maes, P. (Eds.), *Artificial Life IV*, pp. 59–69. MIT Press, Cambridge, MA.

Slocum, A., Downey, D., & Beer, R. (2000). Further experiments in the evolution of minimally cognitive behavior: From perceiving affordances to selective attention..

Wahde, M., & Nordahl, M. (1998a). Co-evolving pursuit-evasion strategies in open and confined regions. In C. Adami, R. Belew, H. K., & Taylor, C. (Eds.), *Artificial Life VI: Proc. of the Sixth Int. Conf. on Artificial Life*, pp. 472–476 Cambridge, MA. The MIT Press.

Wahde, M., & Nordahl, M. (1998b). Evolution of protean behavior in pursuit-evasion contests. In Pfeifer, R., Blumberg, B., Meyer, J.-A., & Wilson, S. (Eds.), *From Animals to Animats 5*. MIT Press / Bradford Books, Cambridge, MA.

web (2000). http://www.swin.edu.au/astronomy/pbourke/geometry/pointline..

Wright, R., & Sweet, M. (2000). *OpenGL Super Bible*. Waite Group Press, Indiana.

Yamauchi, B. M., & Beer, R. (1994). Sequential behavior and learning in evolved dynamical neural networks. *Adaptive Behavior*, *2*(3), 219–246.