SimAnt Simulation Using NEAT

Erin Gluck and Evelyn Wightman

December 15, 2015

Abstract

In nature, ants in a colony often work together to complete a common task. We want to see if we can simulate this phenomenon. We create a simulation that looks at predator prey interactions based off of the 1991 SimAnt game. In this game a spider attempts to eat individual ants from a colony, but if the ants form a large enough swarm they can work together to eat the spider. The question we are attempting to answer is: Will an ant colony (the prey) learn to swarm together to kill a spider (the predator)? We hypothesize that the ants will learn to swarm and learn to kill the spider as well. To test our hypothesis we run a number of simulation experiments using a hard coded spider and a colony of ten ants who all have the same brain. Their brain evolves over several generations using a program called NeuroEvolution of Augmenting Topologies (NEAT) [6]. The fitness score is determined by how many ants in the colony survive with an added bonus to the fitness score if a large enough swarm is able to kill the spider. The outcome of these experiments supports our hypothesis. After several generations we observe the behavior of ants swarming together and trying to kill the spider. Our numerical evidence shows a positive trend over ten generations of evolution, both for fitness scores and for spider deaths, which further supports our hypothesis. These findings suggest that the prey learn the behavior of working together not only to avoid getting eaten by the spider, but also to kill the spider.

1 Introduction

Collective behavior among multiple agents to achieve a particular goal can be simulated using artificial intelligence and is a large area of research in adaptive robotics. In nature there are many different organisms, such as fish and ants, that form swarms. These swarms are groups of creatures which move together to accomplish a common goal. The goal could be attempting to confuse a predator, getting food, or overcoming a larger predator [3] [4]. Our experiment builds off of a simulation done in 2014 that uses NEAT to look at schooling behavior in fish. In this paper, 'Predation as a Motivation for Schooling Behavior with Simulated Fish' [5], O'Connor and Boninger simulate their prey (the fish) by evolving artificial neural networks with NEAT. In their simulation the prey are safe in a swarm, but can be eaten by a predator if they are alone. O'Connor and Boninger investigate whether prey are able to form 'schools' [5]. We wanted to build off this work and take it one step further to see whether the groups of prey would be able to work together to accomplish a task. Our work investigates prey interactions based off of the 1991 SimAnt game, in which a spider will eat solitary ants, but if enough ants form a swarm together they can overcome and eat the spider. These kinds of swarming behaviors are of great interest in adaptive robotics, and much research is devoted to replicating these natural behaviors in artificially evolved environments.

There are some experiments that look at predator and prey interactions and how multiple agents need to coordinate their behavior to achieve a common goal. One such example is a paper entitled 'Coevolution of Role-Based Cooperation in Multiagent Systems' by Yong and Miikkulainen [8]. In this paper they look at how coordinated behavior between multiple agents can best be evolved. They suggest a powerful approach to do this called Multiagent Enforced SubPopulations (multiagent ESP) where they control agents with neural networks, coevolve them in separate subpopulations, and test them together in a common task [8]. One difference between our approach to our experiments and their approach is that they do heterogeneous evolution and use symbiotic adaptive NueroEvolution [8].

Another piece of work that relates to our experiment is a 2005 study by Von Mammen et al., which works to evolve swarms that work together to build 3-dimensional structures [7]. Their study is similar in structure to ours. They evolve controllers which provide rules for agent behavior, and then test the controllers in simulations where one controller is used for every agent. The task of the agents is to build a 3D structure that matches a target structure as closely as possible. Their evolved agents were able to imitate simple structures, however their model was too simple to effectively construct more complicated shapes. The definition that Von Mammen et al. use for a swarm is different from the definition we use for a swarm. They define a swarm to be the entire population of agents, which makes sense since their task is based on environmental changes rather than on grouping behavior. Since we want our ants to move closely together, we define a swarm as a group of agents in close proximity [7].

Agents moving together is investigated in 'Evolving Mobile Robots Able to Display Collective Behaviors' by Baldassarre et al. [2]. In this paper researchers attempt to evolve robots to find each other in an arena and then move together toward a light source. Their research took an in-depth look into how robots can move together as a group once they have found each other. They observe at least four distinct travel patterns which the robots use to travel together without moving too far away from each other. This ability to travel closely together is important for our task of swarming to kill a spider.

In our experiments we use an artificial neural network to model a brain for the simulated ants. Every ant in the colony has the same brain. We evolve the brain using a program called NeuroEvolution of Augmenting Topologies, or NEAT. NEAT is a method for evolving artificial neural networks with a genetic algorithm. A genetic algorithm is a way of solving a problem that mimics the process of natural selection. The idea is to start evolution with simple networks then allow them to become more complex generation after generation [6]. We use the python implementation of NEAT found on Kenneth O. Stanley's website [1] for our experiments. We will provide further details on our methods in the next section.

2 Experiments

2.1 Implementation

We design a graphical simulator with a predator robot - the spider - and prey robots - the ants. In a simulated world that is 1500 pixels wide and 1000 pixels tall, we use a hard-coded spider that moves in a jerky way, mimicking the movement pattern of the spider from the SimAnt game. The spider holds still for 40 time steps, then moves for the next 40 time steps. On each timestep that he moves, he finds the ant closest to him and moves toward it at full speed. If the closest ant is in a group large enough to kill the spider (in our experiments a lethal swarm is a group of three ants or more), he will not move towards the closest ant. The spider has a body radius of 25 pixels, and in the center of his body he has a mouth with a radius of 5 pixels. If an ant touches the spider's mouth it will die. The spider is randomly placed in the simulator for each run.

The simulated colony consists of ten ants at the beginning of each simulation. Each ant has a radius of 5 pixels and is controlled by the same NEAT brain. In order to make it clear how many ants are in a swarm, we print the number in red over the swarm (See Figure 1). Each ant is in its own swarm of size one if it is not within one swarm radius of any other ants (this is described in further detail below).

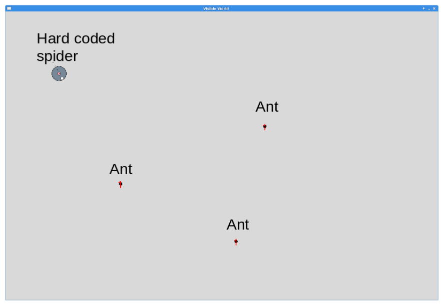

Figure 1: Our simulator with three ants and the spider.

In our experiments there are initially 10 ants which are randomly positioned within the environment. Ants are grouped into swarms on every time step using the following algorithm:

```
for each living ant
    look at the current ant's closest neighbor
    if that neighbor is within the swarm radius
      if the neighbor is already in a swarm
            join that swarm
      else form a new swarm with the neighbor
    else form a swarm with only the current ant
```

For the first half of our experiments the swarm radius is 25 pixels, which is equal to one spider radius. The spider will die when the center of a swarm of size three or more ants comes within his body radius. For the second half of our experiments the swarm radius is $3 * 25$ pixels, which is three times the spider radius. In these experiments the spider will still die when the center of a swarm of size three of more ants comes within the spider's body radius. This change allows for the desired swarming behavior to be rewarded because it does not require the ants to be as close to one another in order to be considered a swarm.

2.2 NEAT

As mentioned earlier, NEAT (Neuro Evolution Augmenting Topologies) is a system which evolves artificial neural networks using a genetic algorithm. NEAT starts out with a simple topology and adds complexity in the hidden layer(s) as it goes, changing both the network topology and the weights of the edges. We used NEAT to evolve neural networks for the brain that we use for all the ants. The brain takes in information about an ant's environment and returns the movement action for the ant to take on that time step.

There are nine total inputs to the network and two outputs. The NEAT inputs and outputs can be seen visually in Figure 2. The inputs are the distances from the ant agent to its nearest three ant neighbors, the angles to those three ant neighbors (this angle is measured relative to the current ant agent's heading), the distance and angle to the spider, and the size of the swarm the agent ant is currently in.

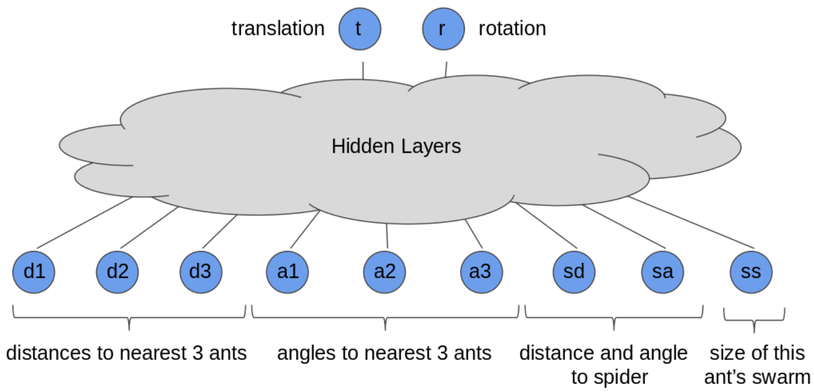

Figure 2: NEAT ant controller.

All of the inputs are normalized between 0 and 1 in order to work with the NEAT program. The network returns the translation (how far they move on one time step) and rotation (how far they rotate on one time step). A few key evolution parameters can be seen in Table 1.

Parameter	**Setting**
Generations	10
Population size	100
Input nodes	9
Output nodes	2
Number of ants in each simulation	10
Number of steps per fitness test	100

Table 1: Evolution parameter settings for NEAT used in the experiments.

A more extensive list of parameters for NEAT can be found in the appendix (see Figure A).

2.3 Fitness Function

The fitness function we use to evaluate the neural network controllers is based on the ant colony's survival as a group and whether or not the spider is killed. In our experiments we start off with the fitness based 80% on ant colony survival and 20% on killing the spider. Then we increase the incentive for killing the spider to be worth 80% of the fitness. Figure 3 illustrates the different fitness functions with the varying percentages for killing the spider.

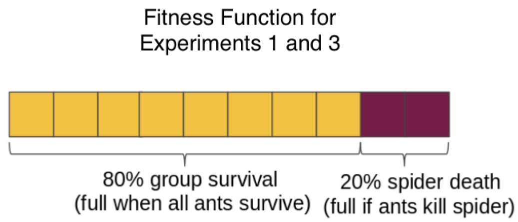

Figure 3: Fitness functions for NEAT. (Left) 80% of fitness score based on colony survival and 20% based on killing the spider. (Right) 20% of fitness score based on colony survival and 80% based on killing the spider (See Table 2 for list of experiments).

Our fitness function sets the fitness equal to the percent of ants who survive until the end of the simulation, multiplied by 0.8 or 0.2 depending on the experiment, plus 0.2 or 0.8 depending on the experiment if the spider was killed. The total possible fitness is 1.0.

2.4 Experimental Procedure

For each experiment NEAT creates a population of 100 neural networks and evolves them over 10 generations. Each ant is controlled by the same NEAT neural network, or brain. During every generation NEAT evaluates each network in the population by using it to run our simulator for 1000 time steps, and then applies the fitness function at the end. After testing each neural network, NEAT records the average fitness of the population, the best fitness found in the population, the number of times the spider died out of the 100 simulations, and the maximum swarm size across all the simulations for that generation.

2.5 Our Experiments

We ran four different experiments. We ran each of the four experiments a number times. We created a table to illustrate the four different experiments in the order that we ran them (see Table 2).

Experiment Number	**Percentage of Fitness Score for Colony Survival (%)**	**Percentage of Fitness Score for Killing the Spider (%)**	**Distance between ants to be considered a swarm**
1	80	20	1 spider radius
2	20	80	1 spider radius
3	80	20	3 * (spider radius)
4	20	80	3 * (spider radius)

Table 2: Experiments we ran.

For experiment number 1, the fitness function is 80% determined by colony survival and 20% determined by killing the spider. The distance between the ants to be considered a swarm is a spider radius. For experiment 2, we increase the incentive for killing the spider so the fitness function is 20% determined by colony survival and 80% determined by killing the spider. The distance between the ants to be considered a swarm is still just a spider radius. For experiment number 3 and 4, we increased the distance between the ants to be considered a swarm from one spider radius to three spider radii.

3 Results

For the first part of our hypothesis we want to show that the ants learn to form swarms. After running our experiments we were successfully able to show this for most experiments. Figure 4 shows the average fitness trend for all four experiments. We can see an approximately linear relationship for each graph for the average fitness over ten generations.

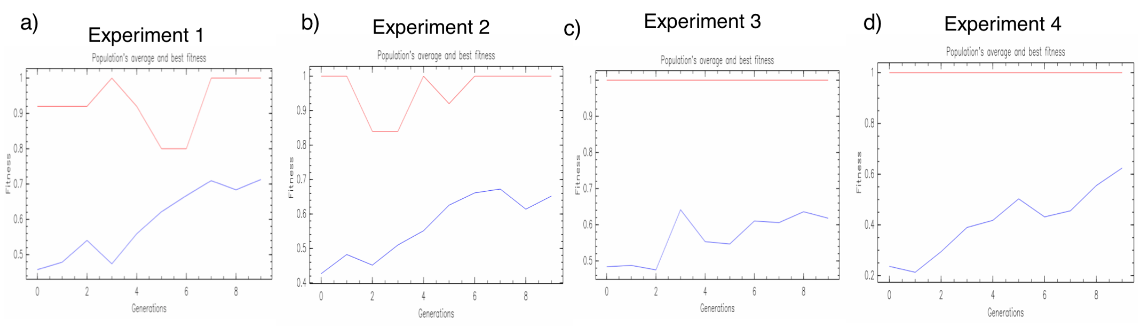

Figure 4: The average fitness (shown in blue) and the best fitness (shown in red) during one run of NEAT. a) Graph for experiment 1 b) Graph for experiment 2 c) Graph for experiment 3 d) Graph for experiment 4.

This indicates that the ants are learning to avoid being eaten by the spider over time. Typically we observed the ants achieve maximum, or close to maximum, fitness in generation zero with the randomly generated NEAT brains. To provide further evidence of swarming behavior, we graphed the maximum swarm size of each generation for all four of our different experiments. These graphs show that the ants form large swarms during almost every generation (see Figure B in the appendix). After observing the behavior from the earlier generations we can attribute the high swarm size to the tendency for the ants to clump up in the corner. The behavior we observed in the later generations shows that the swarms do in fact go after the spider.

For the second part of the hypothesis we attempt to prove that the ants work together in their swarms to go after the spider. We can see from Figure 5 the number of spider deaths for each generation for experiments 1 and 2.

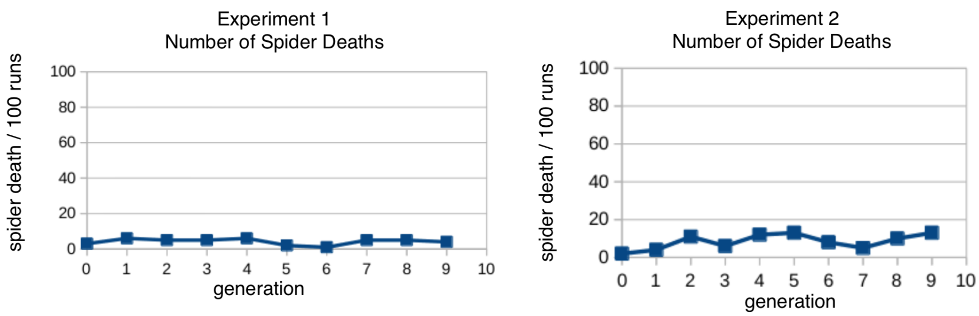

Figure 5: Number of spider deaths in each generation. (Left) Graph for experiment 1 (Right) Graph for experiment 2.

As you can see in Figure 5 (left) when killing the spider was only worth 20% of the fitness score the ants were not able to successfully kill the spider at as high a rate as we had hoped for. We also expected to see an increasing linear trend, so we increased the incentive for killing the spider to be worth 80% of the fitness. As a result, we saw an increase in the number of spider deaths (see Figure 5 (right)), but still did not see as high a rate as we expected. However, the behavior we saw from the later generations for both experiment 1 and 2 illustrated that the swarms of ants did work together to go after the spider most of the time. Figure 6 shows the progression of the behavior of the ants swarming together towards the spider and killing it for one particular run of experiment 2.

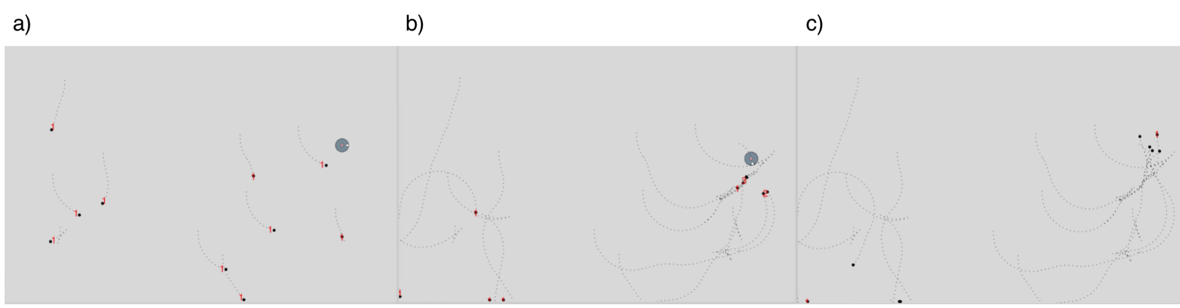

Figure 6: Colony swarming behavior (from experiment 2). a) the ants are scattered at the start of the run b) the ants start to swarm together and go towards the spider c) the swarm successfully kills the spider.

We observed this desired behavior (from Figure 6) in the later generations for all four experiments. Based on this behavior we can say that the ants do in fact learn to go after the spider and kill it as time goes on.

In the later generations we noticed on a few runs the ants would all swarm together, but they were too far away to be considered a 'swarm.' Thus the group of ants never formed a large enough swarm to be considered lethal (as a reminder in all of these experiments the lethal swarm is a group of three ants or more). In order to fix this, we increased the distance an ant can be from another ant in order to be considered a swarm. We did this by changing the distance from being the radius of the spider, to being three times the radius of the spider. By increasing the radius, the desired behavior we saw is rewarded more often with spider deaths and increased fitness scores. Figure 7 shows the graphs of spider death for experiments 3 and 4 in which we increased the swarm radius.

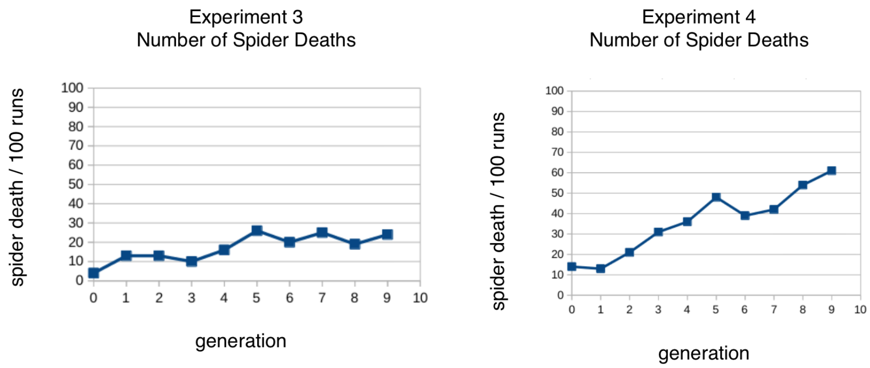

Figure 7: Number of spider deaths in each generation. (Left) Graph for experiment 3 (Right) Graph for experiment 4.

As we can see, the number of spider deaths increases and as we predicted we see a more increasing linear relationship. From our numerical results and the behaviors we observed, we conclude that our original hypothesis is correct.

4 Discussion

Based on previous work by O'Connor and Boninger [5], we were confident that the ants would form swarms. We hypothesized that these swarms would then learn to work together and kill the spider. We observed the behavior we expected to see and thus we claim our original hypothesis is correct.

Our predator, the spider, was hard coded to head towards the closest ant. At first we found that even when the closest ant was in a large enough swarm to kill the spider, the spider would still go after the ant and kill itself. This behavior is something we wanted to fix, so we made the spider smarter. If the closest ant was in a large enough swarm to kill the spider, he would stop moving until the closest ant was no longer in a lethal swarm.

After running our initial set of experiments, where killing the spider was worth 20% of the fitness, we noticed that the number of times the ants killed the spider was quite low. As a results we decided to increase the incentive for killing to spider to be worth 80% of the fitness. We did see an increase in the number of spider deaths but the highest in one generation was only 13 deaths out of 100 runs. After observing numerous different runs, we noticed that the behavior of the ants swarming together towards the spider occurred many times in the later generations but it did not always result in actually killing the spider. Based on this observation we decided to increase the distance between ants in order for them to be considered a swarm from one spider radius to three spider radii. From this adjustment we saw a large increase in the number of spider deaths.

From what we observed the ants came up with unique strategies to avoid being killed by the spider. One strategy was rushing to the corners all at once. This could account for the high max swam size in the beginning generations. Another strategy we saw in later generations was circling around the spider to try to kill it. Unfortunately this did not result in a spider death because of how we considered swarms. There are a number of limitations in the way we assigned ants to a swarm. The first limitation of our swarm function is that ants have to be extremely close to each other in order to be considered a swarm. This was easy to correct, however there are still a few issues with the overall swarm construction. Because of the way we assign ants to swarms, the order in which ants are added can affect the size and number of swarms that are formed. As an example, say we have four ants in a line, A, B, C, and D which are arranged such that A is within a swarm radius of B, and B is within a swarm radius of C, and so on down the line, but no other pairs are within that radius (so A and C are not within a swarm radius). If we add the ants to swarms in the order A, B, C, D, then A will form a swarm with B, and C will join that swarm through its proximity to B, then D will join the same swarm through its proximity to C resulting in one swarm of four. However, if we add the ants to swarms in the order A, D, C, B, then A will form a swarm with B, and D will form a swarm with C. C is then in a swarm, and B will join whichever swarm it is closest to, resulting either in two swarms of size 2 (AB and CD), or one swarm of size 1 (A), and one swarm of size 3 (BCD). This means that if there were six ants grouping together to kill the spider, they may not have been counted as a single swarm but rather as three separate swarms of size 2. While the effects of this problem were somewhat mitigated by increasing the swarm radius, the swarm construction algorithm remains an area for improvement in future studies.

In the future it would be interesting to run our experiments over more than ten generations and see if the trends we observed continue. Another interesting component to add is a graph of the average swarm size over each generation (as opposed to the max swarm size) to see if the average swarm size actually increases with each generation. It could be interesting to add two hard coded spiders and see if over a number of generations the colony of ants could learn to kill both spiders. If we had more time we would have like to run some experiments with an increased colony size and possibly some experiments with an increase in the lethal swam size as well.

Among the predator prey simulation studies we surveyed, our work was significant in that our prey has the ability to kill the predator. This created a somewhat deceptive problem for the prey, as approaching the spider while in a swarm smaller than three was dangerous, but approaching the spider while in a swarm greater than or equal to three was desirable. In the field of adaptive robotics as a whole we examined a number of different experiments that deal with cooperative behavior [8] [7] [2]. Our paper provides evidence that further supports the outcome of these papers in that multiple agents can coordinate their behavior to achieve a common goal. Further we have extended the work of O'Connor and Boinger [5], by showing that the swarms of prey can actually work together to achieve a goal. We successfully evolved simulated robots to display a natural goal-oriented swarming behavior.

Acknowledgments

We would like to thank Professor Lisa Meeden for all her help with this project and throughout the entire semester. We would also like to thank Andy Lee for his joint work on the original simulator that we used to build off of, and Kenneth O. Stanley for his NEAT implementation.

References

[1] The NEAT users page. www.cs.ucf.edu/∼kstanley/neat.html. Accessed: 2014-05-01.

[2] Gianluca Baldassarre, Stefano Nolfi, and Domenico Parisi. Evolving mobile robots able to display collective behaviors. *Artificial life*, 9(3):255–267, 2003.

[3] Eric Bonabeau, Guy Theraulaz, Jean-Louls Deneubourg, Serge Aron, and Scott Camazine. Self-organization in social insects. *Trends in Ecology & Evolution*, 12(5):188–193, 1997.

[4] C Ronald Kube and Eric Bonabeau. Cooperative transport by ants and robots. *Robotics and autonomous systems*, 30(1):85–101, 2000.

[5] Melissa O'Connor and Joe Boninger. Predation as a motivation for schooling behavior with simulated fish, Sep 2014.

[6] Kenneth Stanley. Competitive coevolution through evolutionary complexification. *Journal of Artificial Intelligence Research*, 21, 2004.

[7] Sebastian Von Mammen, Christian Jacob, and Gabriella Kókai. Evolving swarms that build 3d structures. In *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, volume 2, pages 1434–1441. IEEE, 2005.

[8] Chern Han Yong and Risto Miikkulainen. Coevolution of role-based cooperation in multiagent systems. *Autonomous Mental Development, IEEE Transactions on*, 1(3):170–186, 2009.

Appendix

Figure A: NEAT parameters

```
input_nodes          = 9
output_nodes         = 2

max_weight           = 30
min_weight           = -30
feedforward          = 1
nn_activation        = tanh
hidden_nodes         = 0
weight_stdev         = 0.9

[genetic]
pop_size               = 100
max_fitness_threshold = 1.1

# Human reasoning
prob_addconn           = .1
prob_addnode           = .05
prob_mutatebias        = 0.2
bias_mutation_power    = .5
prob_mutate_weight     = 0.9
weight_mutation_power = 1.5
prob_togglelink        = 0.01
elitism                = 0

[genotype compatibility]
compatibility_threshold = 3
compatibility_change    = 0.0
excess_coeficient       = 1.0
disjoint_coeficient     = 1.0
weight_coeficient       = .4

[species]
species_size         = 10
survival_threshold  = 0.2
old_threshold        = 30
youth_threshold      = 10
old_penalty          = 0.2
youth_boost          = 1.2
max_stagnation       = 15
```

Figure B: Graphs of the maximum swarm size for experiments 1 - 4