Evolving Signals to Solve a Coordination Problem

Matt Tanner
Princeton University, Santa Fe Institute
Advisors: John Miller and Eric Smith

1. Introduction

The ability to communicate is an integral part of how animals cooperate with each other. Communication helps give organisms knowledge of the position as well as information on the future intentions of others. The ability to communicate is necessary in order for social animals such as ants, bees, or even humans so evolve social networks. We wanted to study how communication can evolve out of a system of agents that are only able to use inherently meaningless signals. We gave agents the ability to drop and receive such signals on a two-dimensional grid and gave them a coordination problem to solve. We set up a simple genetic algorithm to allow the agents to evolve the ability to use the signals in ways that would help them solve the problem. We then looked at the forms that the agent's solutions to the problem took.

The most important aspect of the signals is that they do not have any built in meaning. The signals have certain properties which are built in, which govern the ways that the signals act when they are dropped and the way in which the agents are notified that they have received a signal, but none of these include any instructions to agents on how to use the signals. Signal meaning, by contrast, is the way in which the agents use the signals. Initially the finite state machines that govern the actions of the agents are random and the agents have no way to use the signals, hence the signals are meaningless. Eventually, after enough iterations through the genetic algorithm, the agent's governing machines can evolve the ability to use the signals, and so the agents have given meaning to them.

The first outcome of the model that we wanted to look at was the effectiveness of selection on the agents when the agents were able to use signals, and when they were not able to use them. This comparison is important because it allows us to measure the importance that signals have in helping the agents to solve the problem. Agents that can not evolve are able to score some points just because of random noise. Agents that can evolve but can not use signals score higher than the former, but can evolve only simple behaviors that do not allow them to score highly. Since agents that can use signals score the highest of all, the agents must somehow be giving meanings to the signals and using those meanings to coordinate their actions. How this occurs is what we are studying in this project.

The knowledge and reasoning of the agents is severely limited in order to better differentiate the scores of the agents that are able to use signals from those that can not. If the agents know too much about their world, the coordination problem becomes trivial. The first aspect of the agent's intelligence is that their knowledge is spatially local. That means that the agents only know the characteristics of the spot that they are in, and have no knowledge of where other agents are in relation to them, or where other signals might be. The second aspect is that the signals that they come across only give historical information. The signals do not have any built in predictive properties but can only notify the agents that a signal has been dropped on that spot sometime in the past. The third part is that the agent's reasoning is Markov. This means that future states of the finite state machine that gives the agent's actions are dependent only on the present state. Past states are only important in that they are necessary in order to get to the present state. In effect, this means that the agents can not look to the future at all, but they are able have some kind of memory of the past. With these limitations on their knowledge of the world, any ability of the agents to coordinate beyond a trivial solution can be attributed to them giving meaning to the signals and having learned to communicate.

2. Biological Analogies

Often in biological systems, seemingly complex behavior stems from simple rules that are ingrained in the limited intelligence of the organisms displaying the behavior. Cooperative communities, such as those that have evolved in ants, bees, and termites, perform collective actions that are much too complicated for any individual organism to control externally. Ants drop pheromones which allow them to hunt and farm collectively, bees can communicate the location of fields filled with flowers to other bees in the hive, and termites are able to drop pheromones which help them build their huge mounds. These organisms have evolved ways to communicate and cooperate, even though each individual has limited intelligence and only has knowledge of local conditions.

Traditionally, such systems have been studied from the standpoint that the meanings of the signals are known in the sense that the outcomes of the communications are easily studied. The researching biologists have focused on how such complicated behaviors could evolve from the simple rules given by the signals. In this project, we want to focus more on how the organisms give meaning to the signals.

The first part of this is how the type of signal being dropped affects the way that the agents use it. This is to see whether a variety of signals will give a variety of responses to them, and so, by comparing the responses, will give an indication of why the agents are using the given signal in the way that they are. The second part is similar to the traditional studies, which is to see how the behavioral complexity of the agents is related to the meanings that the agents give the signals. This is useful because it allows us to relate behavior and meanings and so get a better understanding of what the meanings are. With a good understanding of the two problems above, the project will hopefully lead into a third part which would be to study language acquisition. This is different from the second part above because studying language involves increasing the complexity of the signals and looking at the interactions of the signals themselves as well as the behaviors of the agents.

3. The Model

The model that we used is an agent/environment model. This means that there is a strict divide between which tasks of the model are performed by the agents and which tasks are performed by the environment. A strict division is necessary so that the agents do not have too much information about what is occurring outside of their immediate vicinity, and so helps avoid building biases into the model that would invalidate the results.

The tasks that are assigned to the agents are divided into the properties of the way that the agents input information and the way that they output it. The only input property of the agents is the ability to receive signals. The agents react to this information with state changes in the finite state machine that governs their actions. The output properties of the agents are what actions to perform in the next time-step. The actions are whether or not the agent drops a signal and also what movement it intends to make. The actions and the state transitions of the agents are governed by a finite-state Moore machine. The actions are contained in the states, which translates to the intentions of the agent. The state transitions correspond to the meanings that the agents are giving to the signals, in the sense that an agent can only use a signal if the ability to use it well is encoded in the agent's state transition wires.

The tasks that are assigned to the environment are chosen so that the agents receive no more knowledge of the overall structure of the world than what we meant to give them. One property of the environment is that it handles the signals, which means that it moves and changes the signals as is given by the properties of whatever type of signal is being used by the agent. It also alerts the agents when they have come across a signal, and removes the signals from the environment when that is necessary.

The environment keeps track of where all the agents are and how they move. This means that it converts between the agent's intentions and the agent's actions. Since the agents cannot move through each other, it's necessary for the environment to keep track of all the agents and prevent two of them from moving into the same square. In order to keep agents from gaining exact knowledge of where another agent is through this method, the agent's state transition occurs as if the agent had moved, and so the agent has no knowledge that it bumped into another agent.

Another necessary component of the environment that keeps agents from gaining knowledge that they are not allowed to have is that an agent's position is solely a property of the environment. This means that the agents themselves do not know where they are on the grid, what cardinal direction they are moving in, or where any other agents are. The environment also keeps score, which keeps the agents from having any knowledge of how well they are completing the task assigned to them.

The basic form of the model is that the agents are all moving around on a twenty by twenty grid. The grid is a torus so that edge effects are eliminated. The agents move around independently according to the machine that governs them and scores are tallied by the environment. The genetic algorithm is run on the agents to give a simulation of evolution and allow us to study how the ability of the agents to cooperate changes.

The agents accumulate points by achieving the goals of their utility function. These utility functions are not the same as an economists utility function in that the agents do not have knowledge about what they are trying to do and so there is no way for the agents to behave rationally. The utility function is instead a simple cooperative task and each time-step that the agent performs that task, it receives a point. The points that the agents accumulate are used by the genetic algorithm function to determine which agents will pass along to the next generation.

Choosing a utility function that will give interesting results is a non-trivial task as a good utility function needs to be a problem that requires communication and coordination among the agents in order to have a good solution. The problem also has to neither be too hard nor too easy. A problem that is too easy allows the agents to evolve a trivial solution to the problem, whether or not they are able to use signals. One that is too hard prevents them from every finding a decent solution and so once again the ability to use signals is not useful. To be able to study the development of communication, it was necessary for us to find a problem that is within the region between problems that are too hard and problems that are too easy.

The utility function that we chose for our agents is a clumping problem. The agents score a single point each time they try to move into a square that is already occupied by another agent. The net effect when agents find a good solution to this problem is that they form motionless clumps distributed throughout the grid. This problem is useful because the difficulty is entirely dependent on the density of the agents within the grid. Too many agents and the solution is trivial because they are always bumping into each other. Too few, and they are not able to find each other whether they have signals or not.

We used a simple genetic algorithm in order to give the agents the ability to evolve. Each generation of agents lasted for 200 time-steps and after each generation we ran a competitive tournament. In this tournament, two agents are picked at random and their scores are compared. The one with the higher score is sent to the next generation with some mutation and that two original agents are sent back into the pot of agents that can be picked at random. This is done until the population of the next generation equals the population of the previous one. One important part of this variation of the genetic algorithm is that there is mutation but no recombination.

The actions and the state transitions of the agents are governed by a finite-state Moore machine. There are eight possible actions that the agents can take in each time-step. The agents can go either forward, backward, left, or right and they also either drop a signal or do not drop one. The order of behavior for the agents is to drop or not drop a signal, then turn in the proper direction and then take one step forward while facing in that direction. After moving, the agent either receives or does not receive a signal, and the state transitions are dependent upon that information. The directions that the agents move in are not cardinal directions but are relative to the direction that the agent is facing in. The movement is not concurrent, although for certain types of signals the agents all drop their signals at the same time.

The machine of each agent has eight states. These states do not correspond to the eight actions in the sense that it is not necessary for an agent's machine to include all eight possible actions. There can be any combination of actions included in the eight states. We chose eight states for the machines because it is a high enough number that all possible actions could be included should that be the behavior evolves, but it is not so high that most states would never be used.

The states are all connected by transition wires that govern the state transitions of an agent's machine. Each state has two transition wires coming out of it connecting to another state. The agent's state changes follow one of the two wires, one if the agent receives a signal, the other if the agent does not. Since there are two possible state changes the agent can make, and there are two possible inputs at each time-step, an agent's state changes are completely deterministic.

The mutations in the genetic algorithm act upon either the state actions of the agent or the state transition wires. One of these is selected randomly, and then it is mutated either so that the wire points to a random state, or the state action is randomized to one of the eight possible state actions. Figure 3.1 shows a graph of a possible Moore machine.

Example of a Moore Machine

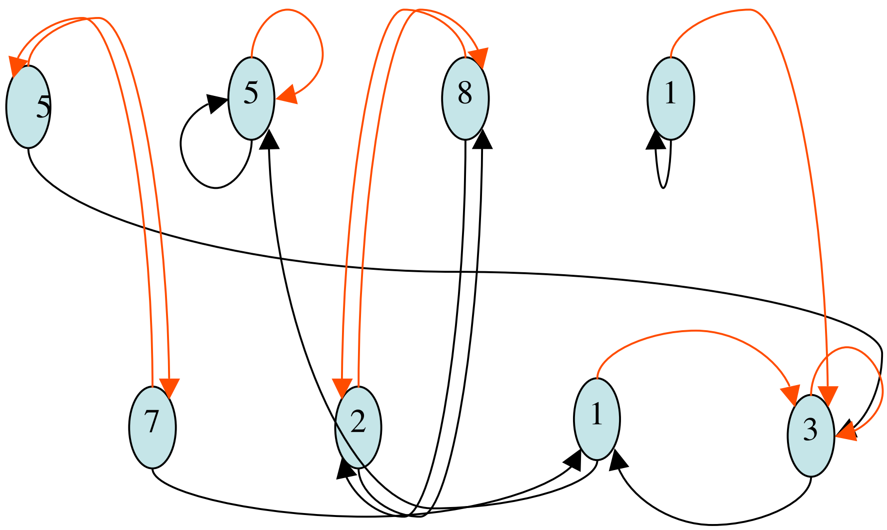


Fig 3.1: Orange arrows are for signals, black arrows are for no signals

The Signals that the agents can use to communicate are a way for their internal states to affect the environment and vice-versa. Without the presence of signals, the environment the agents move around on is constant and so they are unable give each other any information at all. With signals, the agents can change the environment and so give other agents some information about where they might be. The environment changes the agent's internal states because state transitions are dependent on environmental conditions.

The signals themselves are binary which means that they carry no other intrinsic information other than whether they exist or not. The signals are not linked to the agent that dropped them, and there is no way to distinguish one signal from another. The different types of signals that we used to study this problem are solely distinguished by the way in which the environment handles them. Also, the different types of signals that we used all correspond to different experiments, so within each run of the model the type of signal used by the agents is homogenous.

There are four types of signals that we studied. The first type of signal is the base case where the agents are not able to drop signals. This type of signal is important so that we can see how much improvement occurs when the agents do not have any ability to communicate and then compare that to the improvement when the agents are allowed to use signals.

The second type of signal is a dropped signal. It is analogous to a pheromone in the sense that an agent drops it and then it remains in place for a certain amount of time and then it disappears. This type of signal allows an agent that finds it to know that another agent was at that spot within the lifetime of a signal.

The third and fourth types of signals are basically the same. They are analogous to a sound in that the signal propagates out instantaneously and then disappears immediately. All the agents within the radius are notified that a signal has been dropped. The third type of signal propagates out one step in each direction and the fourth type propagates out two steps in all directions. These signals allow agents that come across them to know that there is another agent within one step of them and within two steps of them respectively. The other characteristic of these signals is that in order to keep agents from running across their own signal each time that they move, the order of actions is changed. The agents all signal simultaneously, and are all notified if they have received a signal. Their state transition for the time-step then occurs and they move after that.

4. Results

Figure 4.1 shows the results of running for thirty runs with the four different types of signals. Each generation is averaged over all the runs and the resulting plots are a time-series of these averages. For these experiments, the population of the agents was twenty.

The most important characteristic of this plot is the bottom line which gives the average scores of the agents who are not able to use signals. The three higher lines give the time-series plots of the scores of the agents that can use the different types of signals. These three lines show convincingly that signals allow the agents to score much higher and so give evidence that the agents have learned to assign meaning to the signals and to use them.

The two other parts of this plot that are worth noticing are that it seems that the agents that use dropped signals learn their behavior faster than the other two, and that the agents that use the two step propagating signals seem to score higher than the other two in later generations. We need to do formal statistics on the data in order to see if these two conclusions are actually backed up by significant evidence.

Fig 4.1: score with different signals

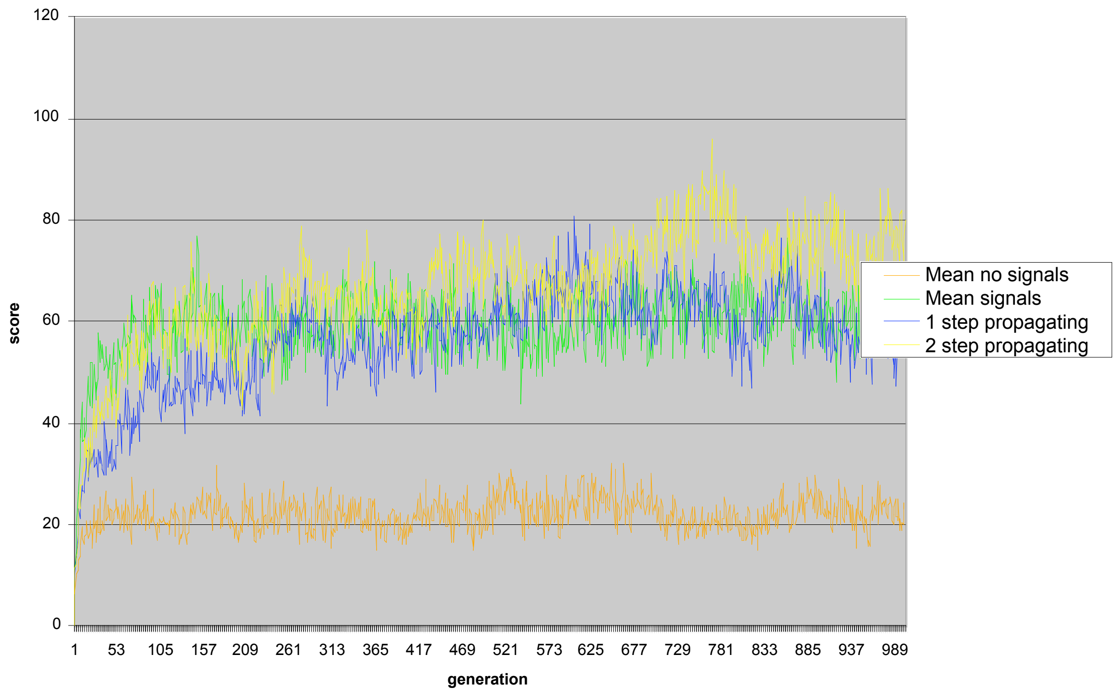
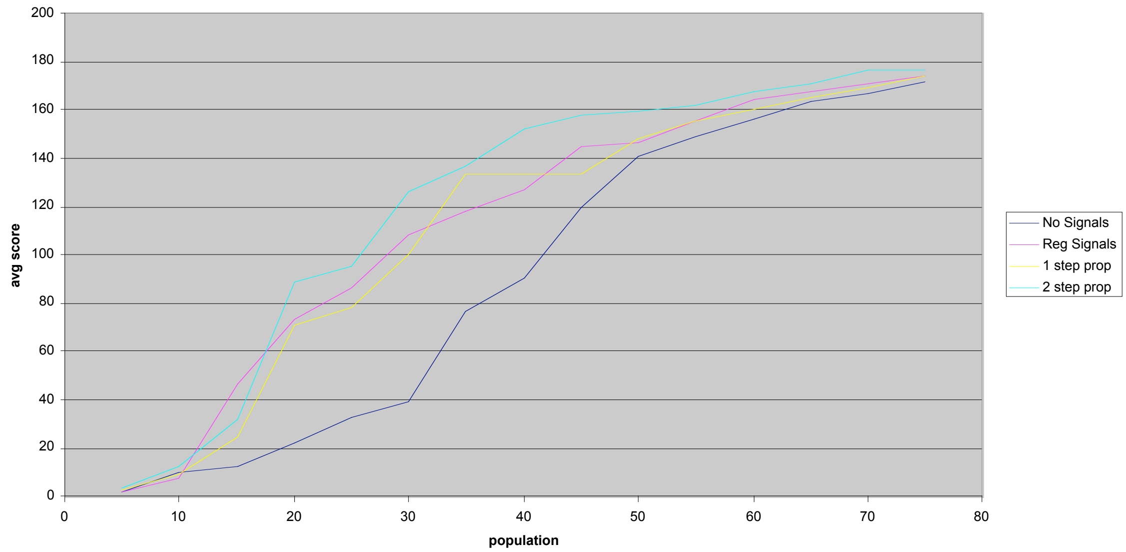


Figure 4.2 gives the plot of average score as it varies according to the population of the agents. In this case, the model was run for each of the four signals and the scores of the agents in the later generations were averaged and plotted as the population was varied.

The most important aspects of this plot are the asymptotes at low and high population levels where the presence of signals do not have any effect on the scores of the agents. The middle region is the area that we were looking for as it is where the problem is neither too hard nor too easy for the agents to solve and so the ability to affect the environment with signals gives interesting results. The other important aspect of this plot is how the agents that use two step propagating signals seem to be able to do better than the other three types of agents throughout the middle zone.

Fig 4.2: effect of population level



Another way to look at how the agents are assigning meanings to the signals is the qualitative behavior of the agents when they have found their optimal solution to the problem. For each of the four types of agents, there is a specific behavior and use of signals that allows them to score the highest. By analyzing the finite state machines that cause these behaviors, we should be able to get a better idea of how the agents are evolving optimal solutions to the problem and from that how they are assigning meaning to the signals.

The agents that can not drop any signals are not able to communicate and so their maximum score is relatively low. The usefulness of studying their solution is that it shows what is possible just through evolving independent behaviors. Comparing the results and the behaviors of these agents, allows us to see better how agents using signals are able to improve their scores. The agents that cannot use any signals move across the screen diagonally; changing direction every time. This behavior allows them to sweep through the largest area, which gives them the best chance of bumping into another agent doing the same behavior.

When they have found the optimal solution, the dropped signal agents display two distinct types of behaviors. One set of agents spins around in a tight circle and drops signals as it moves. The other type of agent heads straight until it hits a signal and then is spins around attempting to bump into the spinning agent. In effect, the spinning agents act as catchers for the agents that move straight. It does not seem to matter whether the agents that move straight drop signals or not. It is not completely clear whether this behavior is the result of a cooperative ecology with the straight moving agents parasitizing off the spinners, or more likely that both behaviors are encoded within a single finite state machine and the random starting state of the machine decides what type of behavior the agent will display.

The one step propagating agents all move straight, some of them dropping signals and some of them not. When an agent comes across a signal, it turns either left or right in an attempt to begin moving in the opposite direction of the other agent on the same plane. This causes the two agents to collide and remain stuck to each other. The two step propagating agents do much the same thing, with agents all moving straight and almost all signaling. The behavior of an agent when it comes across another agent is too complicated to describe qualitatively. This is because the two step propagating agents are able to use their signals the most effectively and so are able to evolve more complicated behaviors in order to find each other.

The next analysis that we performed on the data that we gathered was to compute the effective number of states an agent uses in a generation. We did this by computing the entropy of the number of directions that the agent turned in the generation. This gives us a value between one and four that is equal to the number of states that the agent used on average during that generation. We then created histograms of all these points in order to see what patterns would come up in the overall behavior of the agents. The usefulness of the entropy is that it is a measure of the complexity of an agent's behavior and so the higher the entropy, the more complex the agent's behavior is. The other benefit is that the higher the number of non-integer values for the entropy, the higher effect that the environment is having on the agent's actions. This means that the entropy can give us a measure of how effective the signals are, as the measure of environmental effects is the measure of how effectively the agents are using signals.

Figure 4.3 is of agents whose machines are random each turn. This means they are unable to evolve and the entropy plot is simply a plot of the combinatory probabilities of the different behaviors. Notice how the entropies are all grouped around spikes at the integer values with the highest spike at one. The only non-integer values are clumped around the integers, which shows that the environment is having no effect on the agent's behavior. This makes sense as the agent's are not allowed to evolve or use signals.

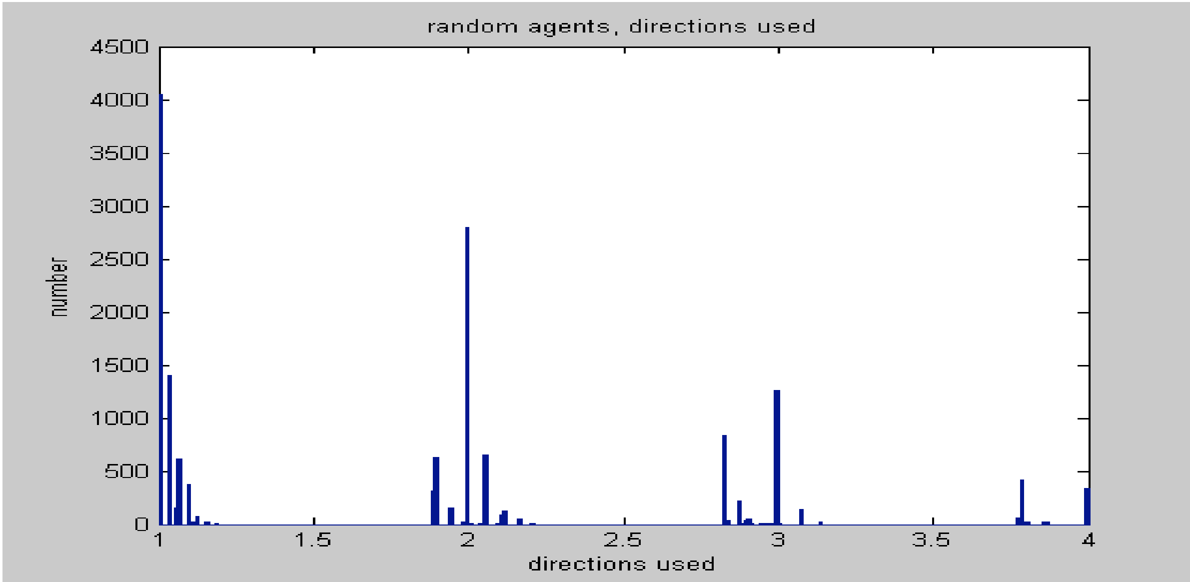
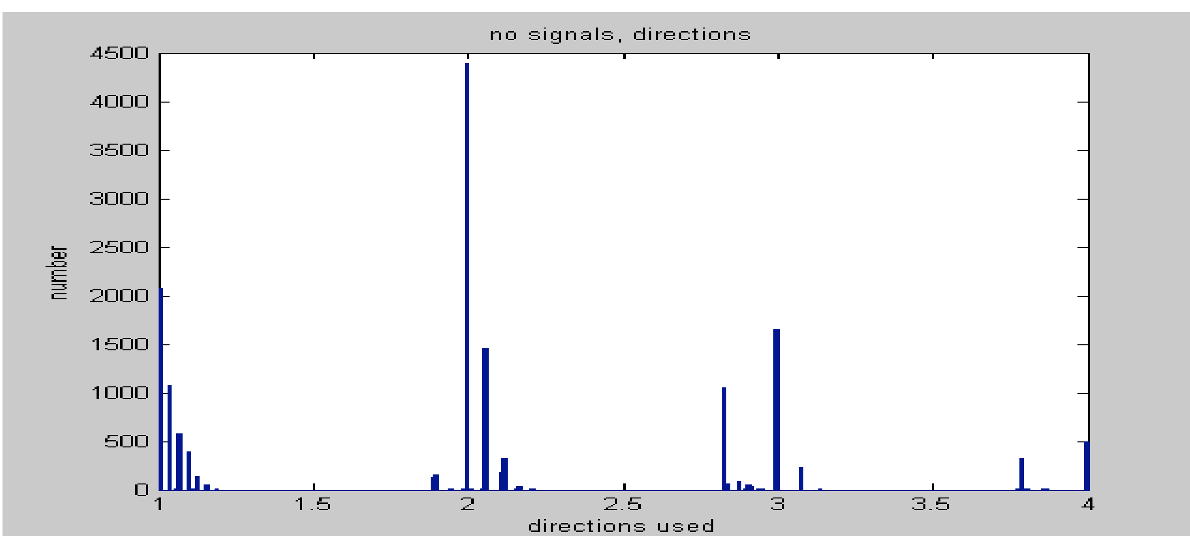
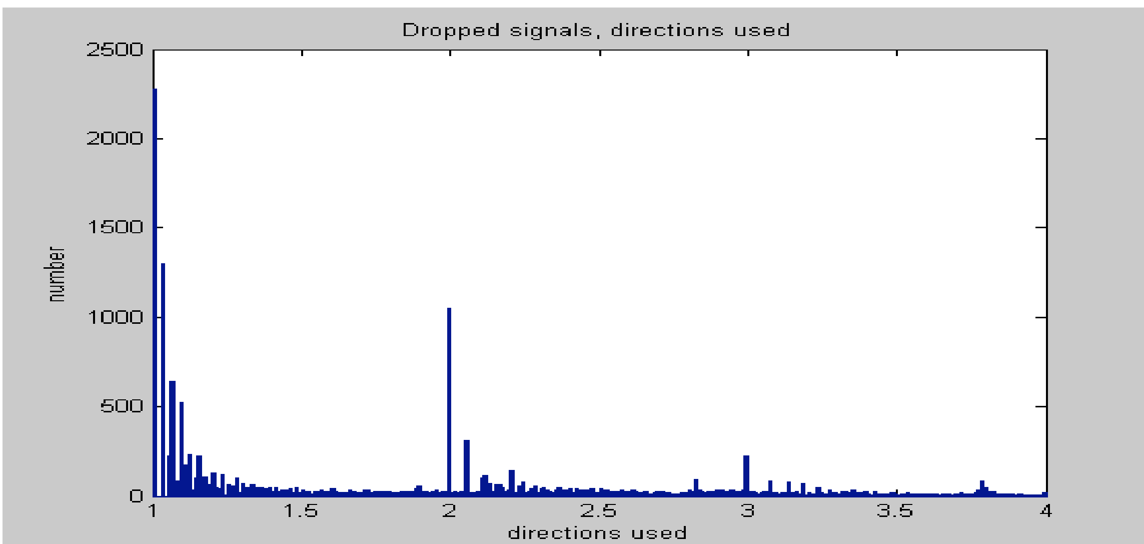


Figure 4.4 shows the entropies of the non-signaling agents. The main characteristics of this plot are that there are still no non-integer values but that the agents have obviously evolved some type of strategy. The spike at one is much lower while the spike at two has taken up the difference. This is a graphical representation of the optimal strategy of these agents, which is to alternate turning left and going straight to make a diagonal.



Figures 4.5 and 4.6 show the entropy of the agents that drop signals and the agents with one step propagating signals respectively. The main results of these are that while the entropy of the majority of the agents are still clumped around the integer values. The environment and hence the signals have begun to have a greater effect on the behavior of the agents. This makes sense as the scores of the agents using these two types of signals was much higher than the scores of either of the two previous types of agents. The only main difference is that it seems that the environment has a slightly higher effect on the one step propagation agents.



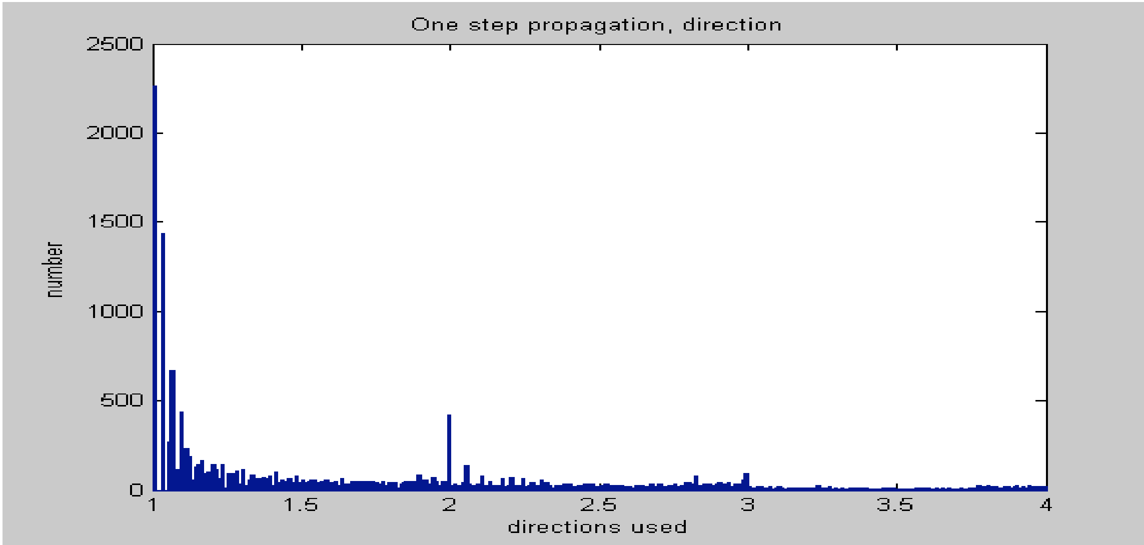
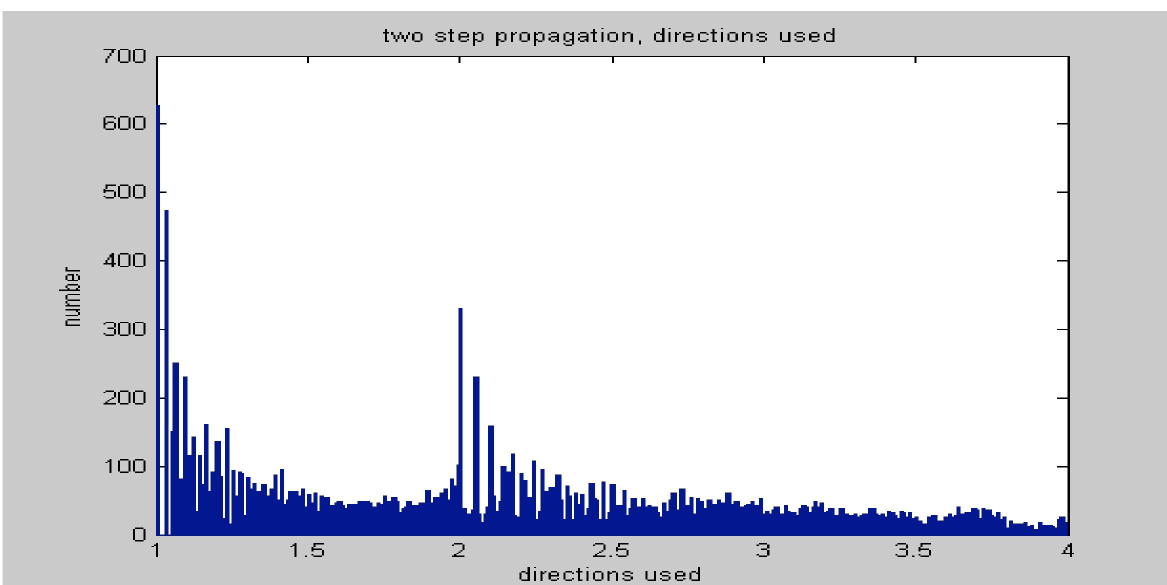


Figure 4.6 shows the last of the entropy plots, which is of the agents that drop the two step propagating signals. The most important aspect of this plot is that the peaks are much lower than any of the other plots and so there is a much larger proportion of non-integer values. This gives evidence to back up the hypotheses that the two step propagating signals score the highest because they are able to affect and be affected by the environment the most effectively.



5. Conclusions

One of the main things that we have learned this far into the project is that the extent of what is learned by the agents is dependent upon their ability to affect their environment advantageously. This means that ability of the agents to communicate is dependent on what type of signal is being dropped, and the way in which they communicate is also dependent on the characteristics of the signals. The rate at which the agents learn the optimal behavior for the type of signal they are using is dependent upon the complexity of their solution. It seems that it takes the agents longer to learn particular complex behaviors such as the optimal solution for the two step propagation signals.

The other main conclusion is that the extent of the environmental effects of the signal are dependent upon the predictive potential of that signal. This means that signals such as the two step propagating ones have a much larger environmental effect and so aid in communication more, because agents are able to better use them in order to predict future behavior of other agents. Signals, such as the dropped signals or the signals that have life-spans of zero, have no predictive potential. In the case of the life-span zero signals, this means that the agents cannot affect the environment and so cannot communicate, while in the case of the dropped signals, the spinning agents evolve in order to give the signals some predictive potential. This behavior is necessary in order for the agents to score higher than in the case of no signals.

6. Future Research

The first problem that we need to resolve is the nature of the agent’s behavior when they are using dropped signals. It is not clear whether there is a cooperative ecology among the spinning agents and the agents that move straight, or more likely, that the behaviors are both programmed into the Moore machines and that the agents are homogenous role-swappers.

The next step is to understand how the signals are being used. The first part of this is increase the number of runs on the experiments and to use formal statistics to make sure that our conclusions are accurate. Then we are going to analyze to individual machines of the agents in order to see how the transition wires give meaning to the signals.

One extension to this project that would be interesting would be to increase the complexity of the signals and study how the signals interact with each other. This would be useful in order to study the development of a language as the interactions could be characterized as a lexicon and grammar.

7. Works Cited

1. John H. Miller, Carter Butts, and David Rode, ``Communication and Cooperation,'' *Journal of Economic Behavior and Organization*, 47 (2002):179--95.

2. John H. Miller and Scott Moser, ``Communication and Coordination,'' Santa Fe Institute Working Paper, 03--03--019, 2003.