The emergence of signaling in populations of artificial agents

Colby Guiterrez-Kraybill
University of New Mexico
colby@kraybill.com

Whit Schonbein
University of New Mexico
71whit@cs.unm.edu

ABSTRACT
Simple forms of communication are interesting not only because they are the germ for the development of more complex syntax and semantics, but also for their own sake: By considering how these systems emerge from self-organizing, local interactions between agents subjected to environmental pressures and opportunities, we may be able to gain important insights into the conditions under which communication arises. In this paper we replicate a previously-developed, agent-based model of communication emergence, and investigate its behavior while varying (i) features of the environment, and (ii) features of conspecific interaction. In contrast to earlier results, populations did not discover optimal communication strategies. In fact, the dominant strategies that emerged eschewed communication completely. Despite these results, we believe this model remains promising.

Author Contributions
Whit Schonbein coded the main model; Colby Guiterrez-Kraybill coded data logging and visualization, as well as optimizing code. Both authors contributed to the contents of every section of the paper. Sections 1, 2, and 3 are primarily Whit Schonbein’s voice, while section 4 is primarily Colby Guiterrez-Kraybill’s voice.

1. INTRODUCTION
Natural language is a paradigm example of the type of collective behavior characteristic of complex systems. Over ontogenetic and phylogenetic time, a syntactically and semantically rich system of communication emerges from the interactions of (broadly) homogenous individuals. This system in turn enables novel forms of adaptive individual and collective behavior, thereby providing impetus for its own persistence (and continued evolution) in the population. [5]

Unlike simpler forms of communication, language exhibits complex syntax and semantics. However, these features are presumably the result of selection operating over more basic systems of communication. Like full-blown language, these more fundamental signaling systems emerge from interactions between individual organisms, and facilitate novel adaptive behavior.

A classic example is threat signaling in populations of vervet monkeys [6]. When threatened by a ground-based predator (e.g., a snake), individuals emit a call that elicits behavior in conspecifics appropriate to that threat (e.g., climbing). In contrast, when threatened by an air-based predator (e.g., an eagle), a different call is used, and the behavior is likewise threat-appropriate (e.g., taking cover). As in the case of more complicated systems of communication, the use of calls enables new forms of adaptive behavior, which in turn promote the persistence of that signaling system.

There are obviously many interesting questions that can be asked about the dynamics of simple signaling systems. For example, how do more complex syntactic structures emerge from mere associations (e.g., case markers)? under what conditions will a stable, shared signaling system break down? How do variations in environmental features (such as number of ground-based predators) influence features of an emergent signaling system (e.g., size of vocabulary). Is there an optimal signaling strategy, and can a population find it?

In this paper we use a previously-developed agent-based model [3] to investigate several of these questions. Specifically, we consider how varying features of the environment and of group interaction influences properties of the signaling system that emerges in the model. Furthermore, we describe a class of optimal signaling strategies and investigate which initial conditions, if any, result in a population locating an optimal strategy.

We find that in our implementation, regardless of initial conditions, optimal strategies are not found. Indeed, the best-performing strategies eschew communication entirely, opting instead for behavior that does better than chance yet worse than what could presumably be achieved if communication were present. This is in contrast to the results given in [3].

The structure of the paper is as follows. In 2 we summarize Grim et. al.’s model and results. In 3.1 we describe our implementation in detail, and in 3.2 we articulate the optimal strategies for the population. In section 4 we present the results of our simulation runs, and discuss them in section
5, along with their relation to those given by Grim et. al. Finally, we conclude with a brief summary.

2. BACKGROUND

There is a long history of using agent-based models to investigate the emergence of communication (e.g., [4], [1], [7], [8]). Very broadly speaking, these models fall into two types. On the 'language game' approach, two agents are selected from a population and forced to play a 'game', e.g., one agent linguistically names an object in the visual field of each agent, while the other attempts to use the utterance as a cue for which object is being referred to; if the listener selects the wrong object, learning occurs so that the listener acquires the speaker’s word [9], [2].

The second type of model is akin to cellular automata in design: A spatial map (e.g., grid) is populated with randomly-initialized agents that interact only with their immediate neighbors. Under certain conditions, the internals of agents are adjusted so that they behave verbally more like their neighbors, either within a generation (e.g., through reinforcement learning), or across generations (e.g., via a genetic algorithm). After some number of iterations, a shared signaling system emerges.

Grim et. al.’s model is of the second type. Here we give a broad overview of the model and their results; details about implementation are given in the next section.

The world is a two-dimensional toroidal grid, where each location contains an agent. Scattered across the world are particles of food and poison, drifting randomly; food and poison are persistent and do not disappear from the world. Agents are in one of two states: feeding or hidden. If an agent is in a feeding state and food lands on its location, the agent’s score increases; if poison lands on the agent while it is feeding, the agent’s score decreases. Similarly, if an agent is in a hiding state and poison lands on its location, the agent accrues no penalty, but if the agent is not hiding, then its score decreases.

Agents cannot move, but they can change states (changing state incurs a small cost). Which state an agent chooses to enter is determined by sounds emitted by its neighbors, and which sound a neighbor emits is determined by the entities (food or poison) currently at that neighbor’s location. For instance, a neighbor could have food in its location, and as a result emit sound A. When the agent hears A, it chooses to enter a feeding state in the hope that the food particle’s random walk will take it into the agent’s open mouth.

Importantly, the mappings from (i) sounds to states and (ii) stimuli (food, poison, or both) to sounds are not fixed. Specifically, agents have two single-layer perceptrons, one for each mapping. These are initially given random weights. In the scoring phase, at each time step each agent’s score is adjusted according to their state and the entities in their location. Then, each agent generates a sound given the entities at their location. Finally, each agent’s state is adjusted according to how they process the sounds of their neighbors, and poison and food particles are moved to a randomly chosen adjacent location.

After some number of iterations, the simulation enters a training phase: For each agent, find the neighbor with the highest score, and train the agent on some subset of the input-output behavior of that neighbor. Continue alternating between scoring and training phases until a shared lexicon emerges. That is, at some point we hope to find that the population has found weights such that (i) all agents generate the same sound for the same stimuli, and (ii) all agents behave in the same (and perhaps optimal) way given the same sound. If this happens, the population has succeeded in finding a shared lexicon, where the sounds generated are ‘tuned’ to the stimulus-appropriate behaviors.

Grim et. al. find that, in a population of 4096 agents (world dimensions of 64-by-64), the proportion of agents that converge to an optimal strategy - i.e., one where poison and food are mapped to distinct sounds, and where each of these sounds is associated with the appropriate behavior - increases at each epoch, at least up to 300 epochs (the limit of the data they provide).

Unfortunately, they do not provide any concrete values (e.g., the y-axis on their graphs is not labeled). Furthermore, they do not give information about average scores, or compare different environmental parameters, or neighborhood sizes, or give an analysis of non-optimal strategies that persist in the population despite the presence of optimal ones. Part of our goal in this study was to rectify this situation.

3. METHODS

In this section we first describe our implementation, and then discuss what constitutes as optimal communication strategy.

3.1 Implementation of the Model

Following Grim et. al., the simulation world is a toroidal grid. For these experiments we used only the 64-by-64 configuration reported in that work.

Food and poison are randomly placed in the world according to a normal distribution. At each time step, these entities move a unit distance in one of the cardinal directions; the direction is randomly selected, and can be varied between 4 (N, E, S, W) and 8 directions. Food and poison are persistent, i.e., they are not destroyed when consumed by an agent. In our implementation, food and poison are discouraged from occupying the same location: If there is a direction to move that avoids a conflict, the particle will choose that direction. Particles of the same type can be stacked on the same location.

Each grid location contains a single, immobile agent. Sound-to-state and stimulus-to-sound mappings are implemented using two independent single-layer perceptrons. Since there are two types of particle, there are four possible stimuli - nothing, food, poison, or both food and poison. These possibilities are encoded as bit strings of length 2 (Table 1).

Similarly, there are four possible state changes an agent can perform: do nothing, feed, hide, (i.e., stay in whatever state it’s already in), or do both (Table 2). In our implementation, the last possibility is treated as equivalent to feeding.
Given these encodings, the perceptrons each have 2 input nodes and 2 output nodes. Stimuli encodings are inputs to the stimulus-to-sound network, and state encodings are outputs of the sound-to-state network.

Weights are randomly initialized at the beginning of a simulation run. Simulations are run for some number of epochs, where each epoch consists of (i) a scoring phase, during which (a) particles randomly walk across the world, (b) agents either feed or hide, (c) agents emit sounds, and (d) agents change state according to the sounds they perceive; and (ii) a training phase, where agents are trained to be more like their neighbors (algorithm 1).

**Algorithm 1 Main simulation loop**

```plaintext
while epoch ≤ numEpochs do
    while trial ≤ numTrials do
        moveParticles
        updateAgentScores
        generateSounds
        updateAgentStates
    end while
    trainAgents
end while
```

For these simulations, we use Grim et. al.’s method for choosing a neighbor on which to train: For each agent, its highest-scoring neighbor is selected. Then, one input-output entry from the neighbor’s sound-to-state mapping is selected randomly, and one is selected from its stimulus-to-sound mapping. The agent’s networks are then trained on a single presentation of each of these pairs, using the standard single-layer gradient descent algorithm (Eq. 1).

$$w_{i,j}(t+1) = w_{i,j}(t) + \alpha x_i (d_j - y_j(t))$$  \hspace{1cm} (1)

In Eq. 1, $w_{i,j}(t)$ is the weight from input $i$ to node $j$ at time $t$, $\alpha$ is the learning rate, $d_j$ is the desired output for the node, and $y_j$ is the actual output of the node. All nodes use a linear activation function.

We performed a series of exploratory runs to determine a suitable learning rate. Adjusting the learning rate did not appear to impact final outcomes so much as time to convergence, so an intermediate rate of $\alpha = 0.02$ was used in all reported experiments.

Environmental pressure is applied to agents by making food, poison, and changing state influence an agent’s score. For these experiments, each food particle eaten increments the score by 1.0, each poison particle that is not hidden from decrements the score by 1.0, and changing state incurs a cost of -0.15.

To explore how different environments affect communication outcomes, we varied the proportion of food and poison in the world. These parameters were divided into three groups: First, the ‘zero’ cases involved situations with either no poison, no food, or neither. Second, the ‘large ratio’ cases involve situations where there is large difference between the amount of poison and the amount of food. Finally, there are the ‘intermediate’ cases, where the ratio of food to poison is closer. Furthermore, to explore how group interaction influences communication, we repeated the intermediate group of runs for a smaller neighborhood. Table 3 summarizes the parameters used for the results reported below.

<table>
<thead>
<tr>
<th>Experiment Group</th>
<th>% Food</th>
<th>% Poison</th>
<th>Neighborhood</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zero</td>
<td>0</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>50</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>10</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>25</td>
<td>8</td>
</tr>
<tr>
<td>Intermediate</td>
<td>10</td>
<td>50</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>50</td>
<td>8</td>
</tr>
<tr>
<td>Large Ratio</td>
<td>10</td>
<td>90</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>90</td>
<td>10</td>
<td>8</td>
</tr>
<tr>
<td>Small Neighborhood</td>
<td>Same as Intermediate</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Simulation parameters. All parameters were run with six different seeds, for 1000 epochs and 100 trials per epoch. For all runs, world size is 64-by-64 and the perceptron learning rate is 0.02.

### 3.2 Optimal strategies

A strategy is any pair of stimuli-to-sound and sound-to-state mappings. Given any agent, its current strategy can be extracted by presenting each possible input to the networks and recording their output. Table 4 provides an example.

<table>
<thead>
<tr>
<th>Input</th>
<th>Encoding</th>
<th>Agent’s Output</th>
<th>Encoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Sound</td>
<td>00</td>
<td>Do Nothing</td>
<td>00</td>
</tr>
<tr>
<td>Sound 1</td>
<td>01</td>
<td>Feed</td>
<td>01</td>
</tr>
<tr>
<td>Sound 2</td>
<td>10</td>
<td>Hide</td>
<td>10</td>
</tr>
<tr>
<td>Both Sounds</td>
<td>11</td>
<td>Feed</td>
<td>01</td>
</tr>
<tr>
<td>Nothing</td>
<td>00</td>
<td>No Sound</td>
<td>00</td>
</tr>
<tr>
<td>Food</td>
<td>01</td>
<td>Emit Sound 2</td>
<td>10</td>
</tr>
<tr>
<td>Poison</td>
<td>10</td>
<td>Emit Sound 2</td>
<td>10</td>
</tr>
<tr>
<td>Both</td>
<td>11</td>
<td>Emit Sound 2</td>
<td>10</td>
</tr>
</tbody>
</table>

Table 4: A sample strategy.

In table 4, the first four rows summarize how the agent behaves with respect the sounds it hears, and the second four
rows summarize how the agent behaves with respect to stimuli. In this case, the agent feeds on hearing only sound 1 or hearing both sounds, hides on hearing sound 2, and stays in its current state when all is quiet. Furthermore, it emits no sound when no food or poison is in its location, but it emits the same sound for all other conditions.

Focusing only on the last column, if we read from top to bottom and write down the output patterns so that the first row is most significant and the last least significant, we get a 16 bit binary representation of the strategy: 0001100100101010.

Grim et. al. note that in their simulations there are exactly two optimal strategies corresponding to the two possible assignments of stimuli to sounds: Sound 1 refers to food and sound 2 refers to poison, or vice versa. Which of these options is decided upon determines the optimal strategy for the other half of the strategy: In the former case, sound 1 should be mapped to feeding and sound 2 should be mapped to hiding, and in the latter case, this mapping should be reversed. Table 5 summarizes these possible stimulus-to-sound mappings.

<table>
<thead>
<tr>
<th>Stimulus</th>
<th>Encoding</th>
<th>Optimal Strategies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nothing</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>Food</td>
<td>01</td>
<td>01</td>
</tr>
<tr>
<td>Poison</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Both</td>
<td>11</td>
<td>10</td>
</tr>
</tbody>
</table>

Table 5: Two optimal stimulus-to-sound strategies, A and B.

In our implementation, the issue of which strategies qualify as optimal is not as obvious. First, since we consider cases where there is no poison, the best possible score can be zero, so a strategy of not doing anything at all may be optimal. Second, in table 5 we assume it is optimal to use the sound for poison in the event both poison and food are present. However, whether this assumption is correct may depend on the proportion of food to poison: If there is more food than poison, perhaps it is better overall to have the food sound dominate the poison sound. Third, the way our implementation handles the case of a state being 'both' hiding and feeding creates further possibilities. In particular, since being in both states is treated as equivalent to feeding, it follows that agents may set on mapping a food sound to 11 (both) rather than to 01 (feed).

Finally, the fact that two different sounds may be mapped to the same state implies that some sound-to-state mappings become ‘junk’ insofar as they are never used. For instance, suppose sub-strategy A (from table 5) were dominant. Then two different stimuli, 10 and 11 both elicit sound 10. Since sound 11 is never produced, there is no pressure for agents to settle on an particular rule for their sound-to-state mapping when the input is 11, and the state could be any of the four possibilities.

Consequently, we assume there more than two optimal strategies in our model. The overall simulation, then, can be interpreted as a search for one or more of these optimal strategies out of the entire set of \(2^{16}\) possibilities.

---

![Figure 1](image.png)

**Figure 1:** Average scores by epoch for different food and poison parameters. The top graph shows results for parameters drawn from the 'zero' and 'large ratio' sets, and the bottom graph shows results for parameters from the 'intermediate' set. Scores are averaged across six runs. Note that not all parameter settings are shown. In all cases, average score improves over time.

### 4. RESULTS

The problem space we have defined is combinatorially large, with parameters for varying pseudo-random seed, food, poison, learning rate, neighbors to visit during learning, trials, and epoch length. We first performed an initial exploration of this space by examining the results of many food, poison, and pseudo-random seeds (table 3).

Figure 1 shows the average scores for populations run under different initial conditions. The top graph shows results from the 'zero' and 'large ratio' parameter sets, and the bottom shows results from the 'intermediate' set. All scores are averaged across six runs on different seeds. As the graphs show, all populations improved their score over time, regardless of parameters. As expected, runs with zero food cannot get a score above zero, and those with more poison do worse, indicating that some agents still feed despite there being no food. In the bottom graph, 50% food and 25% poison is especially interesting because it begins below zero and adapts so that the score is positive; this is also the parameter setting explored by Grim et. al.
Figure 2: Comparison between the median scores of agents and the Shannon entropy of the entire grid vs Epoch. These plots exhibit the expected behavior, that entropy is high (in bits) at the start of a simulation with maximum randomness of strategies, while descending in complexity as agents tend toward similar or identical strategies. (a) When food=10% and poison=25%. (b) When food=25% and poison=25%. (c) When food=50% and poison=25%.

Figure 3: Strategy maps for each agent. The full strategy, represented as 16 bit values are mapped to an 8 bit color range. (a) shows the initial randomly distributed strategies. (b) shows the strategies after 500 epochs of learning and (c) 1000 epochs of learning.

Figure 4: Hamming distances are computed based on the local neighbors, in this case n = 8.
4.1 Entropy
To quantitatively analyze the individual simulations, measuring the entropy (in terms of Shannon Information) of the strategies of all the agents allows a numerical measure of how strategies are coalescing into groups (figure 2). The entropy measurement provides a broad measure of the number of groups in the simulation that are interrelated.

In figure 2(a), food is relatively low and poison is relatively high. The overall strategies learned converge to a 0 median score value over 100 trials. Gradually the Shannon entropy (measured in bits), decays to 0, as there is a singular unifying strategy or monoculture. For figure 2(b) the amount of food and poison are the same, with a similar median score response, while the measure of entropy indicates a monoculture is not reached within 1000 epochs. In the 2(c) case, the median scores move past reaching 0 and the entropy indicates there is not a monoculture reached with a higher complexity indicated by the entropy measure of 2 bits. This re-inforces our decision to focus on simulations with food 50% and poison 25%.

4.2 Strategy Distributions
Visually representing the different strategies provides a qualitative method for determining if the simulations are leading to learned behaviors. In figure 3(a), the initial setup of agents, with food 50% and poison 25%, should be uniformly random, and this image shows what appears to be a random distribution of different strategies. After 500 epochs of learning (figure 3(b)), visible groupings of agent strategies are clearly defined and then progress to more similarity after another 500 epochs in figure 3(c). This corresponds to the the measure of complexity as entropy in the plot in figure 2(c). As these maps were started at epoch 1, there is already an indicator that some learning has taken place based on some small clusters of similar color.

Another way to visualize these changes are to plot the differences in Hamming Distance from individual agents to the agents in their immediate neighborhood (n = 8). These are shown in figure 4 with the same progression from randomly distributed distances to larger groupings of similar agent strategies.

The visual analysis is complicated by high-order bit changes leading to large changes in color values when mapping from $2^6$ possible values to $2^8$ colors while those changes may only indicate a single bit difference in strategy. Binning strategy values can provide a less noisy visualization of the strategy data and these histogram plots are shown in figure 6.

The histogram of strategies in the figure 6(a) show an apparently random distribution as is expected. Within 100 epochs, 4 different strategies are already being favored in figure 6(b). The following 6(c) shows that on epoch 200 the number of favored strategies has narrowed down to 3. After this, the trend towards 2 dominant strategies slows as the third strategy at around $5 \times 10^3$ to $6 \times 10^4$ slowly decays from epoch 300-1000. Running our simulations for 5000 epochs showed that this trend continues, favoring the retention of 2 strategies. This indicates that some form of learning occurred but does not provide definitive proof that language learning took place.

Figure 5: The top plot shows a sample of simulation results when the number of neighbors used to train against is 4 (n = 4). These appear very similar to the n = 8 results, but with more noise in the scores over the length of the simulation. The bottom plot shows a comparison between the stddev of the scores showing that the n = 8 simulations indeed have quantitatively less noise during the learning process, as there are more neighbors available to re-enforce the locally favored strategies.

4.3 Varying Neighborhood Size
In the majority of our simulations we used a neighborhood size of 8. Early in development, we focused on neighborhood sizes of 4 and noted that the mean of scores were noisy and decided that working with a neighborhood size of 8 should reduce this noise in the learning algorithm over length of the simulations.

As shown in the plots for figure 5, the general results for simulations run with n = 4 appear to follow the same trends of simulations with n = 8. However, the score range is narrower: 10 versus 20 for n = 4, n = 8 respectively and the n = 4 score means show more variance, as shown in the second plot for figure 5.
Figure 6: Histograms of strategies for simulations run with 50% food and 25% poison
5. DISCUSSION

The results indicate that our implementation of the model succeeded in self-organizing to some extent: Average scores increased over time, entropy decreased, and histograms show that some strategies come to be favored over others.

However, it is not obvious that the resulting strategies are optimal in the sense described in section 3.2. In this section we first discuss the strategies that were found, and then briefly relate the results back to the study we were attempting to replicate.

At the end of each simulation run, scores were reset and the simulation was run for 5000 time steps, at which point the strategy of the agent with the highest score was extracted. Table 6 shows the winning strategies for a subset of our simulation run parameters.

As can be seen in the table, when the amount of food is less than or equal to the amount of poison, strategy 0 is dominant. According to this strategy, (i) the agent never produces any sound, and (ii) the agent always persists in maintaining whichever state it was in at the previous time step. In this case, we hypothesize that the preceding state is simply hiding: At some point during the execution of the simulation, the agent avoided both (i) poison (by hiding) and (ii) the costs of switching states (by always choosing action 00, i.e., 'stay in the same state').

What fails to occur is a transition to the more optimal strategy of adjusting one's state to take advantage of the occasional arrival of food. Indeed, we hypothesize that once the population finds strategy 0 it is unlikely or impossible to do better, because the training algorithm will eliminate variation in the sounds produced by nearby agents - there is no sound at all - and hence no information is being transmitted about nearby food. The population becomes stuck in a local minimum.

In contrast, when the amount of food is greater than the amount of poison, there are multiple best strategies. However, the majority of these share the prefix '174'. Table 7 shows the 16 bit representations of the full strategies in this group.

<table>
<thead>
<tr>
<th>Strategy</th>
<th>sound-to-action</th>
<th>stimulus-to-sound</th>
</tr>
</thead>
<tbody>
<tr>
<td>17408</td>
<td>01 00 01 00</td>
<td>00 00 00 00</td>
</tr>
<tr>
<td>17477</td>
<td>01 00 01 00</td>
<td>01 00 01 01</td>
</tr>
<tr>
<td>17488</td>
<td>01 00 01 00</td>
<td>01 01 00 00</td>
</tr>
<tr>
<td>17489</td>
<td>01 00 01 00</td>
<td>01 01 00 01</td>
</tr>
<tr>
<td>17492</td>
<td>01 00 01 00</td>
<td>01 01 01 00</td>
</tr>
<tr>
<td>17493</td>
<td>01 00 01 00</td>
<td>01 01 01 01</td>
</tr>
</tbody>
</table>

Table 7: Full strategies for the '174' group.

All agents in the 174 group of strategies share the same sound-to-action sub-strategy. This sub-strategy takes the agent into the feeding state and keeps it there, regardless of which sounds are emitted by neighbors. In this case the stimulus-to-sound component of the full strategy becomes irrelevant, which is why there is variation in that sub-strategy.

Finally, we can also see from table 6 that when there is slightly more food than poison (e.g., 25% food, 10% poison), different random seeds result in one or the other of the outcomes just described: Strategy 0 or a member of the 147 group.1

Consequently, in our implementation there appears to be a significant bias towards settling on less-than-optimal signaling systems; indeed, in both cases signaling fails to emerge. That is, in both strategy 0 and the 147 group, sounds are irrelevant to the level of success the agents were able to achieve.

It is interesting to note that we also implemented the model using an alternative agent architecture, and found similar results. In this alternative, rather than use perceptrons, we used probabilistic lookup tables: Each input pattern was associated with each output pattern by a probability; training an agent on a given input-output pair involved increasing the probability of that output and reducing the probabilities of the others.

This alternative architecture was less stable on account of being probabilistic (e.g., even if one output was highly likely, there remained a chance that a different output could be chosen). But the trajectories for average scores were quantitatively and qualitatively similar, suggesting that these agents were also not locating optimal strategies.

Because of these results, we hypothesize that Grim et. al.'s model is constrained in ways that ours is not. As noted in section 2, the authors claim to be searching for only two optimal strategies, and suggest that both are found on every run. We believe this is highly unlikely unless there exist additional constraints on the simulation to narrow the search. What these constraints might be is currently unknown.

6. CONCLUSION

Simple signaling systems are interesting not only because they are the germ for the development of more complex syntax and semantics, but also in and of themselves: By considering how these systems emerge from self-organizing, local interactions between agents subjected to environmental pressures and opportunities, we may be able to gain important insights into the conditions under which communication arises.

In this paper we replicated the agent-based model of communication emergence given in [3], and investigated its behavior by varying (i) features of the environment (percentages of food and poison), and (ii) features of conspecific interaction (number of neighbors). In contrast to Grim et. al., our populations did not discover optimal communication strategies. In fact, the dominant strategies that emerged failed to capitalize on the opportunities offered by the fact agents report the status of their local environment, instead finding strategies that eschewed communication completely. Despite these results, we believe this model remains promising.

7. REFERENCES

1Strategies 1024 and 1280 are small variations on strategy 0.
Table 6: Winning strategies over six runs for each set of food and poison parameters. Strategy 0 is dominant when amount of poison is greater than or equal to the amount of food. Strategies with the prefix '174' are dominant when the amount of food is significantly greater than the amount of poison.

<table>
<thead>
<tr>
<th>% Food</th>
<th>% Poison</th>
<th>Best strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>10</td>
<td>0 0 0 0 0 0</td>
</tr>
<tr>
<td>10</td>
<td>50</td>
<td>0 0 0 0 0 0</td>
</tr>
<tr>
<td>25</td>
<td>10</td>
<td>17489 0 0 17493 17492 17408</td>
</tr>
<tr>
<td>25</td>
<td>50</td>
<td>0 0 0 0 0 0</td>
</tr>
<tr>
<td>50</td>
<td>10</td>
<td>17488 17493 17477 17489 17477 1024</td>
</tr>
<tr>
<td>50</td>
<td>50</td>
<td>0 1280 0 0 0 0</td>
</tr>
</tbody>
</table>