

# Niche Construction Within a Cellular Automata

by

Gobindroop Mann

Supervised by Dr. Chris Marriott

A senior thesis submitted in partial fulfillment of the departmental honors requirements for the degree of

Bachelor of Science  
Computer Science & Systems  
University of Washington Tacoma

June 2020

Presentation of work given on Niche Construction within a Cellular Automata

The student has satisfactorily completed the Senior Thesis, presentation, and senior elective course requirements for CSS Departmental Honors.

Faculty advisor:  Date 6/16/20

CSS Program Chair:  Date 6/16/20

## Abstract

This experiment takes a stochastic cellular automaton and through an abstraction of its cells as agents and food, creates an ecosystem. This ecosystem mimics behaviors found in nature such as niche construction. It was learned that when the agents within this automaton survived the full length of the experiment (10000 cycles), they always broke into distinctive groups.

## Introduction

The goal of this project was to create a stochastic cellular automaton that would mimic behavior found in nature. A cellular automaton is a discrete model which was originally created in the 1940s by Stanislaw Ulam and John von Neumann (Chopard and Droz 2). In this model there is a grid of cells, each of which contains a finite number of states. For each cell within the grid there is a set of cells which are defined as neighbors to the cell. The model starts with each cell within the grid being assigned a state (this is the initial state of the grid where time  $t=0$ ). The model then advances to a new generation (time  $t=1$ ) and every cell within the grid is updated simultaneously according to some fixed rule. When a new generation is created every cell within the grid is updated according to the state of its neighbors (Wolfram 602). An evolution of this model is what will be used in this experiment, a stochastic cellular automaton. The difference between a stochastic cellular automaton and a normal cellular automaton is that the latter has a rigid rule that does not change over time, is applied to all cells, and does not depend on chance. The former, on the other hand, has a rule that can change over time, may not need to be applied to all cells, and does depend on chance. This means that if the same initial state were to be run for, arbitrarily, 100 runs for the same amount of time, there is no guarantee that the end state for each of the runs would be the same.

A more specific goal of this project is to observe a phenomenon found in nature known as niche construction. Niche construction is a process where some organism alters its environment to better suit that organism's survival needs (Laland and O'Brien 303). A prime example of niche construction are lemon ants found in Peru; these ants use a chemical within them known as formic acid to kill off the trees that are not suitable homes for them. Through this process, the lemon ants *construct* an environment in which they can thrive. For this type of behavior to be observed within a stochastic cellular automaton, several abstractions must be made. We must designate some cells within the grid to be agents (something which will construct its environment), and then we must designate other cells to be the environment itself (something for the agents to construct to their needs). This will be explored in more detail within the experiment itself.

Previous work within the field of cellular automata include Conway's Game of Life and Stephen Wolfram's study into elementary cellular automata. In Conway's Game of Life, the neighbors of a cell are defined as the eight cells that directly touch the cell within the two-dimensional grid. There are two states defined for each cell in this cellular automaton, dead and alive. The rules are very simple as well. If a cell has two or three neighbors that are alive and the

cell itself is alive it will remain alive in the next generation, if a cell is dead and has exactly three alive neighbors it will become alive in the next generation, and if neither of these conditions apply the cell will be dead the next generation regardless of whether it is alive or dead currently (Adamatzky 2). Based on just these simple rules many interesting states were discovered. On a simple level, there were many states discovered in which a group of alive cells seemed to migrate together and move in some direction, and, if left uninterrupted, would continue forever (these are known as spaceships) (Eppstein 433). On a more complex level, on May 18, 2010, Andrew Wade announced he had discovered a state which was not only a spaceship but would also create a copy of itself while destroying its parent (Goucher). This extremely intricate state takes 34 million generations to fully replicate itself and destroy its parent.

In Stephen Wolfram's study of elementary cellular automata only a one-dimensional grid was used (which is why the cellular automata were known as elementary). Once again, each cell had only two states dead or alive. The neighbors of a cell were defined as cells that directly touched the cell. The major difference in Wolfram's study was the rule. Wolfram's experiment came from testing every possible mapping from one generation to another (Wolfram 601).

This project will expand upon previous work done with cellular automata by introducing chance into the cellular automata and also by using the automata to create a form of artificial life which will demonstrate niche construction.

## Architecture

In this experiment we will be designating certain cells within the automaton as "food" (these will be cells that do not eat anything and do not move) and then we will be designating other cells within the automaton as "agents" (these will be cells that move positions within the grid and consume the food). To collect data there is an experiment manager which creates an automaton, lets the automaton run, downloads files containing the data from a particular run, and then creates a new automaton to start the process over. The job of the experiment manager is to automatically change the value of our experimental variables. The specifics of what is changed is further elaborated upon in the experiment section.

Upon creation, the automaton creates a 100x100 board, and then adds 50 agents to random places on the board. On every update cycle, the automaton updates each of the agents and then updates all of the food. After this it checks if any of the agents died and if they did, they are removed from the board. On every update cycle the automaton also collects data for all of the genomes of the agents and all of the genomes of the food and stores them in an array (the genomes will be discussed in further detail). Also, the automaton displays this data in real time using graphs. Every 10000 update cycles, if the option is selected, the data gets sent to a database, and if the option is selected, the data also gets downloaded as a CSV file. After each update cycle there is also a draw phase, in this phase the automaton draws each of the foods onto the board and then also draws each of the agents onto the board, with their updated location. This is used to visualize what is happening in real time.

Each food contains only one genome which determines its color on the board, this affects what it can be eaten by. When the experiment starts, each square in the 100x100 board is given a .05 percent chance of having food in it, every other cell will be black (think of these as dead). Once this initial creation of food is complete no other food can be created, they can only grow from existing food. Those that are not black will be given a color based on an HSL value where the S is held at 100 and the L is held at 50. The hue will be determined by the genome of the food (in other words it will be random). On every update cycle the food will have a chance of growing. If this is the case, the food will pick a random cell within its direct surrounding eight cells (its neighborhood) and create new colored food in that square. The color of the new food will be the color of the old food with an increase or decrease of hue. Should the food be against a wall its child food will grow on the opposite end of the board, as if the wall were looping to the other side (a toroidal arrangement). If the food's hue goes above the limit a hue can be, it wraps back around to the lowest possible hue value, and if the food's hue goes below the limit a hue can be it wraps back around to the highest possible hue value. For every given update cycle there is also a chance that the food will just die.

After the board is created, 50 agents are put on the board in random locations, after this the only way that another agent is created is if it is the child of an existing agent. Every agent has five genomes, a genome for what color food they can eat, a genome for what color food is poison to them, a genome for what food they are attracted to, a genome for what kind of food they try to avoid, and then a genome for the range of values for which they can be healed or poisoned. Agents also have seven hit points. The color of the agent in the experiment is determined by its food genome using an HSL value where the S is held at 100, the L is held at 50, and the H is determined by the food genome. When the agents are initially created all of these genomes are set to random values, but if the agent is a child of some other agent then the values are based on the value of the parent's genomes. The child's color can be up to 1.8 higher or lower hue than their parents, the exact value being a random value within this range. The way that an agent reproduces is by passing over a white cell. A white cell is created when an agent successfully eats food. When this happens, there is a 50 percent chance that the agent will reproduce.

On every update cycle, an agent will determine its action based on the cell within the grid that it is currently located on. If the agent is on a white square it will have a chance at reproducing, if the agent is on a black square nothing will happen, and if the agent is on some colored square (food) it has a chance to eat that food and turn the food white or to fail to eat the food and turn the food black. When an agent lands on a colored cell (food) a distance calculation between the food's genome and the agent's food and poison genomes will be calculated. A random value between zero and the agent's heal poison range genome will be chosen and if the difference between the food genome and the agent's food genome is less than that value the agent will eat the food and then turn that cell white. If the food is not eaten then another random value between zero and the agent's heal poison range genome will be chosen and if the difference between the food's genome and the agent's poison genome is less than that random value the agent will fail to eat the food and then turn that cell black. If the agent fails to eat the food, it will lose one hit point. However, if the health of the agent is not at the full seven and the agent successfully eats a food it will gain a hit point. When the agent runs out

of hit points it will die. If the agent successfully eats food, there is no chance that it can be poisoned by it. After this there is a random 1.5 percent chance of an agent dying. When an agent dies it is removed from the simulation.

After the action of the agent is determined (eat, poisoned, reproduce, etc.), it will move to a new square. The agent can move to any of the surrounding eight squares. The way that an agent chooses to move is based on the agent's attract and avoid genomes. The smaller the difference between the genome of a potential next cell and the attract genome the higher chance the agent has of moving there. After this calculation, the avoid genome's distance from a potential next cell will be measured and the same logic will be applied. Black and white cells are given a score of 10 and 180, respectively. The score of every other cell will be somewhere between these values, based on the calculation mentioned before. After every one of the surrounding cells is given a score, each cell will be given a chance to be the next location of the agent based on how high the score is. A next location is chosen and then the agent will move to that location.

## Experiment

Apart from their genomes, food and agents have other parameters that also affect them. The main difference between a parameter and a genome is that a genome will change based on its parent while a parameter will remain the same and does not have the ability to change. The majority of these parameters were used as control. For the agent, the first of these parameters is the spawning population, which determines how many agents spawn at the very beginning. This was set to a value of 50. The next parameter is the maximum hits for the agent. This is the amount of times that agent can eat poison before it will be killed. This value is set to 7. Next, we have two toggle parameters these are agent AI and agent healing. The former of these determines whether the agents will move towards food that they like, if this is toggled off the agent will move completely randomly. The latter determines if the agents can heal when they eat food that is good for them. For this experiment these were both toggled on. The next parameter is the death chance for the agent. This is a random chance, regardless of health, of dying every tick. This is set to 0.015 (maximum being 1 meaning every agent dies 100 percent of the time each cycle). The next parameter is reproduction chance, this is the chance that the agent can reproduce after it has stepped on a white cell. This is set to 0.5 meaning 50 percent of the time. Next, we have offspring volatility. This determines how far off the genes can possibly be of a child from their parent. This is set to 0.01 meaning children can alter by a percent from their parents. The final parameter is a range parameter similar to the heal poison range. This is called the attract avoid range. This determines how far away an agent's food genome should be from a food's genome in order to be attracted to it. This also determines how far an agent's poison genome should be from the food in order to avoid it. This is set to 36 (out of a possible 180).

For the food, all of the parameters were varied in order to simulate different types of environments. The first parameter is the food growth rate. This determines the chance a food has of growing to one of its neighboring areas. This is set to 0.4 (out of a possible 1 meaning every food will grow every tick) for a "medium" environment, it is set to 0.8 for a "dense" environment, and is set to 0.2 for a "sparse" environment. The next parameter for the food is

the food decay rate. This determines the chance each food has of dying every tick. This is set to 0.005 (out of a possible 1, which means every food would die at every cycle) for a “medium” environment, it is set to 0.01 for a “dense” environment, and it is set to 0.0025 for a “sparse” environment. The death rate was increased with the growth rate in an attempt to give the agents some hope of being able to survive (if random cells are being killed at a higher rate there is a chance some other color could come into the area and save the agents. For example, if there was a small group of red agents trapped within some green food, with a high death rate there would be a chance for some red food to bleed in and save the population). The next parameter is food spawn. This is a chance that is given to each of the cells of the 100x100 board of hosting a food within it at the start of the simulation. This is set to 0.0005 (out of a possible 1 meaning that every cell would have a random food within it) for a “medium”, it is set to 0.001 for “volatile”, and is set to 0.00025 for “stable”. The final parameter for the food is the food offspring volatility. This determines how different a food is from the one that it grew from. This is set to 0.01 (meaning there is a possible 1% variance in genomes from a parent to a child) for “medium”, it is set to 0.02 for “volatile”, and it is set to 0.005 for “stable”. A parameter that applies to the overall simulation rather than a group in particular is the wait turns, which are used to see how many turns the agents have to wait before they can start moving and reproducing. This is set to 300 ticks and is kept as a control. The purpose of this parameter is to allow the food to grow before the agents are released to eat it.

The experiment manager cycles through every possible combination of volatility and growth presets (Dense-Volatile, Dense-Medium, Dense-Stable, Medium-Volatile, Medium-Medium, Medium-Stable, Sparse-Volatile, Sparse-Medium, Sparse-Stable). For each of these nine “environment” settings the parameter heal poison range was also altered. The settings that were used for this were 20, 25, 30, and 33. These values represent the distance (maximum being 180) that the agent can be from the food in order to be affected by eating the food. Each of the specific experimental and control variables can be seen in Table 1, Table 2, Table 3, and Table 4. This experiment was run 15 times on each of these 36 settings and the data was sent to a database for further investigation.

*Table 1: Experimental settings for Food Spawn Chance, Food Spawn Value, and Food Offspring Volatility*

	Stable	Medium	Volatile
Food Spawn Chance	.00025	.0005	.001
Food Offspring Vol.	.5%	1%	2%

*Table 2: Experimental settings for Food Growth Rate and Food Decay Rate*

	Sparse	Medium	Dense
Food Growth Rate	20%	40%	80%
Food Decay Rate	2.5%	5%	10%

Table 3: Experimental settings for Heal Poison Range

Heal Poison Range	Value
	20
	25
	30
	33

Table 4: Control Variables

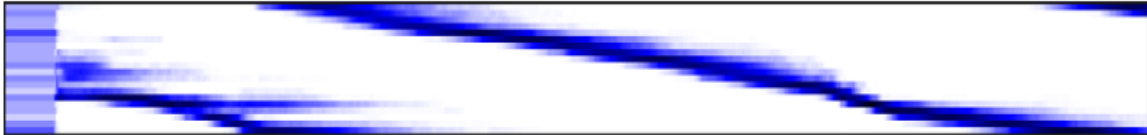
Spawning Population Agents	50
Offspring Volatility Agent	1%
Agent Decay Rate	1.5%
Agent Reproduction Chance	50%
Hit Points Agent	7
Wait Cycles Before Releasing Agents	300

## Results

The results of this experiment varied, there were results that show promise but require future experimentation (perhaps a recreation of the model) and then there were results that outright failed. To be specific, the purpose of this experiment was to show cases of niche construction, base conditions for niche construction are that the organisms make a change to the environment and that this change benefits the organisms. In this case we will measure benefit by the amount of time the agents lived therefore results that outright failed would be runs in which the agents did not survive the duration of the experiment. For the results that outright failed, the cause was that all of the agents would die out, all of the food would die out and therefore the agents would die out, or all of the food and agents would die out at roughly the same time and we were left with a blank screen. I believe that the cause of these problems was the harshness of the environment. For the majority of cases in which the death of all agents occurred, the settings were either set to “volatile” and “dense” or they were set to “sparse” and “stable” settings. The problem with the first group was that the food was growing too quickly, while the agents would try their best to adapt so that they can eat the food, the volatile nature of the food allowed it to evolve faster than the agents were able to keep up with. In this situation agents would be surrounded by food that was poison to them quickly because after they finished eating the food that was “edible”, food from other areas would quickly grow to take the eaten food’s location. This would continue until eventually the agent was surrounded by food which is poisonous to the agent. Since the agents were not able to evolve at the same

rate they would often get trapped in regions of food that was poisonous to them and they would die of poisoning. The problem with the second group was the opposite, there was no food for the agents to eat and when the agents did find food it was not of great variety due to the stable nature of the food. This resulted in the agents starving to death.

*Figure 1: A series of three histograms representing, from top to bottom, the food genome for every agent within the experiment, the poison genome for every agent in the experiment, and then the genome for every food in the experiment.*



*Figure 1A: Food Genome for Agents*



*Figure 1B: Poison Genome for Agents*



*Figure 1C: Food Genome for Food*

In the other, “successful”, runs there were several events which consistently took place and may point towards niche construction. One of these events was that the genomes of all of the agents and the genomes of the food all tended to group up with their respective group. This means that the food genome for all agents seemed to group together, the poison genome for the food all seemed to group together, and the genome for the food all seemed to group together. This can clearly be seen in Figure 1 (the x-axis represents time while the y-axis represents the value of the genome being graphed, also the darker the shade of blue the higher the percentage of agents or food within the experiment with that value), the top most histogram represents the food genome for every agent within the experiment, as can be seen after an initial period of randomness the food genome for the agents seems to group together, the same could be said about the poison genome for the agents (the middle histogram), and then the genome value for all of the food (the bottom histogram). The ordering of these events was also consistent. The agents would always group first and then the food would follow. Another interesting pattern that was observed was that, often (82.7% of the runs that lasted the full 10000 cycles), the food genome for the agents seemed to chase after the genome of the food. In this event the food can be seen adapting in a certain direction and then the agent adapting right behind it to be able to eat the food. Since both of these genomes are looping (once they hit the maximum of 1 if they increase their genome it is set to the minimum of 0 and vice versa) this chasing was something that continued throughout the 10000 ticks of time allotted per experiment.



The original goal of this experiment was to take a stochastic cellular automaton and modify it such that it displayed niche construction; a phenomenon found in nature. While I do not believe that this experiment fully succeeded, I believe that there was a lot of progress made towards the goal. Niche construction is a difficult thing to demonstrate especially within the constraints of our experiment, but I believe that certain behaviors exhibited by the agents does resemble niche construction. For example, it was shown in our experiment that when left alone the food would grow randomly (this can be seen in the wait time of 300 cycles before the agents are released) but when the agents were released into the environment they caused the food to group up rather than be dispersed randomly. This may be an example of the agents molding their environment to better suit their needs. Perhaps the agents thrive more easily when they have one solid mass of food to evolve to eat.

## Conclusion

The main takeaway from this experiment is the newfound use of a cellular automaton. While the results of experiment itself were not conclusive, as there was not enough evidence to back up whether or not the results we achieved were some form of niche construction, the use of a stochastic cellular automaton still stands as a new way to represent events in nature and even other fields such as chemistry. Perhaps by abstracting the cells within a cellular automaton as some role and not treating each cell as the same we can observe numerous patterns within biology and even other fields.

This project was limited, I feel, in its randomness. There are several cases where, through sheer bad luck, all of the agents would die without a chance to show any behavior. Even worse, sometimes there was behavior (such as the agent's food genome surrounding the food's genome) that could not be recreated as it occurred through luck. Another limitation of our project was the lack of proof for theories related to the model. For example, it was observed that the food's genomes almost always collapsed down to a single genome value and then evolved as a collective to avoid the agent's food genome. The difficulty comes in proposing why this occurred. Was this a case of niche construction where the agents are molding the food to their liking? Did this occur because the agents ate every other genome value for food and the survivors reproduced until they became the entire population? Or is there some other variable in play that is not being seen? It is very difficult to say.

To improve upon this project for future work, I would try to create a model that was consistent in its creation of the agents. For example, perhaps we create "species" of agents where certain species have certain values for each of their genomes (with some variance). This would much better represent organisms in the wild, and if the same set of agents (5 of species A, 34 of species B, etc.) was released into a similar environment I feel the results would be much more consistent. What this would also allow for is recreation of past events. If we observed some unusual pattern we could create a species that mimics the genomes of the species at that time and run the experiment on the same environment.

## References

Adamatzky, Andrew, ed. *Game of life cellular automata*. Vol. 1. New York, NY, USA:: Springer, 2010.

Chopard, B., and M. Droz. *Cellular automata*. Vol. 1. Berlin, Germany: Springer, 1998.

Eppstein, David. "Searching for spaceships." *MSRI Publications* 42 (2002): 433-452.

Goucher, Adam P. "Game of Life News." *Game of Life News: Oblique Life Spaceship Created*, 21 May 2010, [pentadecathlon.com/lifeNews/2010/05/oblique\\_life\\_spaceship\\_created.html](http://pentadecathlon.com/lifeNews/2010/05/oblique_life_spaceship_created.html).

Laland, Kevin N., and Michael J. O'Brien. "Niche construction theory and archaeology." *Journal of Archaeological Method and Theory* 17.4 (2010): 303-322.

Wolfram, Stephen. "Statistical mechanics of cellular automata." *Reviews of modern physics* 55.3 (1983): 601.