CELLULAR AUTOMATA: USING SIMULATION TO PREDICT REAL-WORLD PHENOMENA

Tyler Sheinberg, tys25@pitt.edu, Mandala 10:00, Pablo Salazar, pas169@pitt.edu, Mena Lora 10:00

Abstract*—When designing buildings, transportation, and other facilities utilized by people, engineers need to take into account traffic-flow and random behavior in order to optimize the efficiency of their models. Cellular automata are just one type of computer simulation that help provide these engineers with a method to test how their designs would function if they were implemented without the need to test them in real-life.*

For every cellular automaton, there exists a grid the contains the individual elements/cells. Each of these cells has an initial state. Before the simulation begins, a set of rules is defined which determine what state each cell will evolve into based on the states of the neighbors of that cell. Some examples of types of rules include division, migration, apoptosis, and differentiation. After the program runs through once and changes the states of all the cells in the grid, the program repeats itself until the grid reaches an equilibrium, where running it no longer alters the states of the cells, or until the researcher terminates it to analyze the results.

This type of technology is crucial to the engineering community because of the broad applications to which it can be applied. The implementation of rules based on real-life situations in a computer program is one way to use logic and accuracy to best predict seemingly random behavior. By giving each cell a set of rules to follow, a simulation can predict how millions of these cells would behave in a given situation. The focus that we will explore in this paper pertains to predicting human behaviors, specifically in evacuation related situations and other traffic-flow based scenarios, and how cellular automata not only optimize productivity, but improve the user's quality of life. We will delve into the benefits to society of simulating human behavior.

Key Words— Behavior, Cellular Automata (CA), Grid, Rules, Simulation, Stephen Wolfram

THE HISTORY OF SIMULATION

Much of science and theory focus on prediction, whether it be prediction of prior events or future ones. Often these predictions are done solely with math, which has proven to be very reliable. The field of statistics has contributed towards more accurately predicting otherwise random outcomes. These calculations, however, were limited to humans' capacity and time to predict a high number of outcomes. In biology, for example, while it seems feasible to predict the future of a single cell, the time and commitment required to predict perhaps hundreds of thousands of cells appears not only overly laborious but perhaps impossible. Thankfully, the creation of machines in the early 1900s began to make computations and other function easier to perform. John Von Neumann realized the potential for machines to make large scale predictions with the right combination of programming, mathematics, and statistics.

Along with his colleague Stanislaw Ulam, Mr. Neumann had created a cell-based model for a machine that was fully capable of reproduction. This was the first example of a theoretical model based on cellular automata. This model became the foundation for the future of this emerging field of study. Von Neumann created the theory of rules that would dictate the behaviors of these automata, which would become the basis for future research in which more complex rules would be applied to simulation. With the implementation of rules, one can see the infinite possibilities of simulation. He began with the creation of cells that could exist in 29 possible states [1]. These cells were connected and affected by other cells above, below, to the left, and to the right of them. Neumann proved that with the different states available and the connection to other cells, machines were able to simulate accurately the process of self-reproduction and evolution that we see in real life.

Von Neumann was revolutionary in the way that he spread information from cell to cell. First, the machine would interpret the information provided to reproduce itself and use it as its instruction. That information would be copied and then given to the newly created cells. This is extraordinary because it parallels with the way that real cells use DNA to copy itself and pass of information to new cells. A real cell only knows what its instructions are with the help of DNA. This was done by Neumann in 1948. The structure and role of DNA was not fully comprehended until 1953, yet Von Neumann created such a similar way to instruct his machines to reproduce.

As von Neumann set the foundation for cellular automata, new scientists would come to create more complex cellular automata. John Conway introduced a cellular automaton called "A Game of Life" in 1970. This "game" was a simplified, interactive modification of Von Neumann's cellular automata. The user controls the number of cells that begins in a grid and with the applied mathematics and rules, the simulation begins in real time and the user may see how the cells reproduce, move, and die [2]. As cellular automata became more applicable in real life situations, new settings were set up in simulation. In 1986, Hugo de Garis used a cellular automaton to accurately build an artificial embryo. Today, cellular automata reach beyond the world of biology. Simulation has moved into physics, chemistry, and other real-world situations. However, in order to understand the history and impact that cellular automata have on the modern world, it is important to understand the basic concepts of this topic.

WHAT ARE THE BASICS OF CELLULAR AUTOMATA?

General Theory

Cellular automata can be boiled down to a few elementary components: the grid, the cells, and their neighborhoods. A grid is a finite area divided into any number of cells that houses the program or simulation. Each cell is given an initial state, typically either on or off (1 or 0). The cells use input based on the surrounding cells in their neighborhoods to determine their next state. The rules that define a cellular automaton determine what input yields what output, or state change for the cell. The vast majority of rules that govern the outcomes of the simulation depend on what we had previously referred to as a cell's neighborhood. The neighborhood of a cell can be defined as the specific cells that surround any given cell whose states determine the fate of the cell they are surrounding. In a square or rectangular grid with cells shaped thusly, a cell's neighborhood is typically comprised of the 8 cells surrounding it, in a similar fashion to the neighboring cells in a grid of a Minesweeper game. Although this is the typical neighborhood that is applied to any given cell, the programmer of the simulation may determine any variation of this to their liking.

Rules are the main component of cellular automata that provides their ability to simulate. A rule is essentially a program that compiles all of the input data regarding the states of any neighboring cells and uses it to determine the outcome state for each cell of the grid. Put in programing terms, the rules utilize a series of IF statements based on the states of the neighborhood to output the state of the cells. One of the most famous and quintessential examples of cellular automata is The Game of Life, created by J.H. Conway in the year 1970. This ruleset takes place on a square grid. The rules that govern this automaton (singular of automata), are as follows. "Cells are living (occupied, black, 1) or dead (empty, white, 0). A living cell survives at the next time step if it has not too many (death because of crowding) and not too few neighbors (death following loneliness): Let s denote the number of living neighbors in the eight neighboring cells. The cell which is alive will stay alive if $s \in \{2, 3\}$, and a dead cell will become alive if $s = 3$. In all other cases, the cell is dead" [2]. Depending on the initial states of the cells in the grid, the simulation will run and produce certain patterns and shapes that can be identified and tracked. A synth is a shape that no longer changes when the cellular automaton evolves. Some of the more common or iconic shapes and synths that the Game of Life is famous for yielding are depicted below.

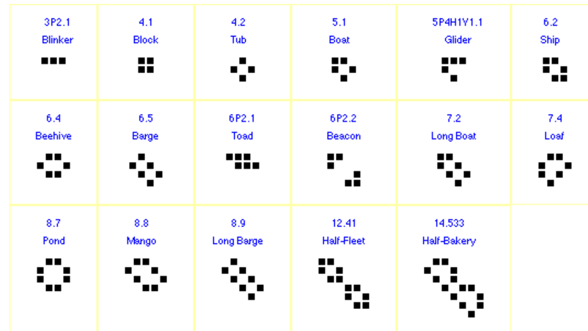

FIGURE 1 [2]
Differing shapes and synths seen in the Game of Life

One of the websites that we have come across in our research demonstrates the game of life, recreated using javascript, which allows the user to alter the initial patterns of the automaton and analyze its evolution. The cellular automata are placed in a large environment so that one can observe the behaviors of the different aforementioned shapes and their individual evolutions. Users can also import their own initial patterns to see their evolutions. Below is a depiction of the simulation.

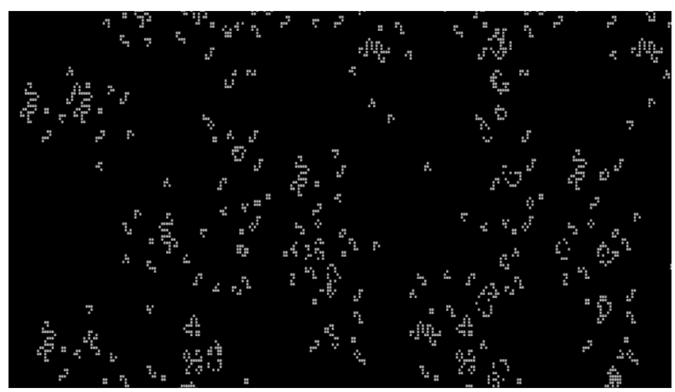

FIGURE 2 [2]
Simulation of Game of Life

Another example of one of the more basic types of a cellular automaton is contact automata. In its most basic form, this simulation emulates the spread of disease or infection. The rules for this automaton state that each cell has two states, either white (uninfected) or black (infected). If any of an uninfected cells' neighbors are infected, that cell becomes infected. This simulation is very impractical but is an excellent and simple example of the application of a simple rule set. If this program was run to completion, every cell on the grid would eventually become infected. A more realistic and useful program would include variance in the data. For example, a new probabilistic rule would be added so that there is a chance for an uninfected cell to remain uninfected even when an infected cell is present in that cell's neighborhood, changing its state to a third option, immune. Rules are useful to researchers in that they can be implemented and modified in order to fit the desired experiment model.

Another website that we have discovered, which can be found at http://cubes.io/, is a more visually satisfying simulation of cellular automata. Although most cellular automata are generally programmed in two dimensions, this website models cellular automata in three dimensions. By slightly altering the rules for the Game of Life, the image seen below had been created. While easy to use, the capabilities of this program allow for advanced rule manipulation and observation.

FIGURE 3 [3]
Three-dimensional cellular automaton

Stephen Wolfram, in 1983, had made great advances in the development of cellular automata technology. The programs that he had developed, coined as Wolfram automata, consisted of cellular automata on a one-dimensional grid (in a line), with neighborhood being defined as the cell itself and its two adjacent neighbors, and each cell having two states, either 0 or 1. While most cellular automata are characterized by their constantly refreshing grids that update based on the rules, Wolfram automata would print out the next states of the grid below the previous, on an axis that defines the time at which each grid had been created . One would be able to look at this continuously scrolling chart and observe the patterns that each rule would yield. Mr. Wolfram had observed the long-term behavior of many of his creations and had concluded that any cellular automaton would fit into one of four different groups; "(1) tends to a homogeneous stationary state (2) tends to periodic structures (3) tends to aperiodic, chaotic pattern (4) tends to persistent, complex and localized patterns" [4]. Wolfram had taken the possible permutations of the states of any given cell in a one-dimensional array and the states of its two adjacent neighbors. Those different combinations can be shown as [111,110,101,100,011,010,001,000]. He had altered the evolved state of the cell for every combination of cell and neighborhood state, which lead to his creation of every possible combination of states and evolutions in his 256 different rules.

Creation of the Technology

There is no standard medium in which cellular automata are created. We have elected, however, to focus on the process of writing the programming for a cellular automaton in MATLAB. Below depicts a function written by Iain Haslam, that outputs any of Wolfram's rules. We will walk through and discuss this code in order to demonstrate the logic necessary to create one of these programs.

```
24 - if nargin < 3, nrows=700; end
25 - if nargin < 2 %Use default initial state
26 -     ncols=700; A=zeros(nrows,ncols); A(1,ncols-1)=1;
27 - else
28 -     [unused, ncols]=size(initialstate)
29 -     A=zeros(nrows, ncols); A(1,:)=initialstate;
30 - end
```

FIGURE 4
First 30 lines of code for MATLAB cellular automaton program created by

```
31 - rule=dec2bin(wolfrule,8);
32 - for i=1:8
33 -     ru(i)=str2num(rule(i));
34 - end
35 - for i=2:nrows
36 -     for j=2:ncols-1
37 -         l=A(i-1,j-1); m=A(i-1,j); r=A(i-1,j+1);
38 -         if(( l & m & r & ru(1)) | ...
39               ( l & m &~r & ru(2)) | ...
40               ( l &~m & r & ru(3)) | ...
41               ( l &~m &~r & ru(4)) | ...
42               (~l & m & r & ru(5)) | ...
43               (~l & m &~r & ru(6)) | ...
44               (~l &~m & r & ru(7)) | ...
45               (~l &~m &~r & ru(8)) )
46 -              A(i,j)=1;
47 -         end
48 -     end
49 - end
50 - colormap(gray(2)); image(2-A); axis image; title(wolfrule);
```

FIGURE 5
Last 20 lines of code for MATLAB cellular automaton program

The function, aptly named wolfram, is called with user input of the desired rule number (0-255), the initial states of the cells in a 1 by n array, and the number of iterations that the user wishes to be displayed at the end. Lines 24-30 ensure that the function receives enough inputs to function. In the case that the user does not input enough values, the program inputs its own default values. The following lines 31-34 define the rules of evolution that the initial states of the cells will follow. The command 'rule=dec2bin(wolfram, 8)' converts the Wolfram rule number that the user inputs and converts it into its binary equivalent to the 8th value. For example, the number 177 becomes 10110001. This binary number, in turn, is converted from a string into a number. As previously discussed, there are eight different combinations of states pertaining to each cell and their two adjacent neighbors states, [111,110,101,100,011,010,001,000]. The Wolfram rule that is generated takes these combinations and applies each binary value to each combination. To demonstrate this using the number 177 as 10110001, the leftmost 1 is assigned to the 111 condition, the next 0 is assigned to 110, the next 1 is assigned to 101, and so on. The rest of the program up until line 49 utilizes an IF statement that checks the states of those that are initially provided by the user and assigns them a new value based on their states and the states of their neighbors which in turn corresponds to the rule condition that was determined by the binary number assignment. This process is repeated using the FOR command for the number of times that the user defines when calling the function. The final line of the program creates a figure window that displays the initial values on the top and shows each iteration of their evolutions. The number of rows equals the number of times that the user defines, which corresponds to the number of times the evolutions are iterated. Using the example number of 177 that we had previously discussed, we may call the function wolfram(177, [1 1 1 1 0 1 1 0 0 0 0 1 0 1 0 1 1 0 1 0 1 0 0 1], 70);. For this code, we would like to display Wolfram's 178th rule. Although we had entered the number 177, the range of values starts at zero, so we must add another 1 to that value to properly label which rule we will be displaying. Based on the input, the initial values of the cells are [1 1 1 1 0 1 1 0 0 0 0 1 0 1 0 1 1 0 1 0 1 0 0 1], and the program runs 70 times. The output of this function is depicted below.

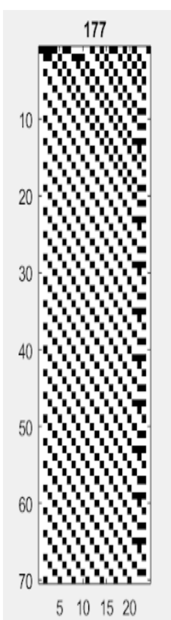

FIGURE 6
Output display of MATLAB wolfram function

By analyzing this photo, we can see that this rule clearly fits into Wolfram's second group of rule classification, meaning it tends to periodic structures. This MATLAB code serves as a great example of one of the most basic and easy to understand forms of cellular automata. While Wolfram's rules serve as an introduction into this field of study, they do not appear to serve any other function or provide any other value to an observer. Perhaps they may derive pleasure from viewing the patterns. Regardless, the utility of this program pales in comparison to the cellular automata the are used currently.

IMPACT ON SOCIETY

Real-life Applications

As stated before, cellular automata can be applied to most processes that involve randomized behavior. To better understand this behavior, cellular automata may provide insight into the prediction of actions, such as crystal growth or cell reproduction. One of the more intriguing applications of a cellular automaton is the prediction of human behavior. Predicting human behavior becomes useful in the design of spaces that may need to be evacuated in the case of an emergency. In order to provide the safest escape routes, hallways, rooms, and exits, must be placed in locations that will provide the safest routes possible for people to leave in an orderly fashion. Engineers and architects take these factors into account during the design of a building. In case of an emergency, the goal of any designer is to make the evacuation of its inhabitants as quick and efficient as possible. Cellular automata may help greatly in visualizing exactly how an evacuation may play out, with a specific floor layout and number of people. This can all be done with the help of statistics, math and simulation.

What makes cellular automata different from other systems of modeling and simulation is its ability to replicate complex systems. "Even if the basic and local interactions are perfectly known, it is possible that the global behavior obeys new laws that are not obviously extrapolated from the individual properties, as if the whole were more than the sum of all the parts" [5]. Cellular automata take systems with inner workings that affect the functionality of the whole system and simplify its mechanics and display it on a larger scale. This makes simulation much simpler. Although the behavior of system relies is based on the combination of all of its mechanisms, big and small, cellular automata strive in these situations. They are capable of providing simplified but concise versions of complicated systems, while also maintaining the functionality to incorporate these finer details and mechanisms in order to attain more refined results.

There are common features in the event of an emergency that can be clearly seen. People move at a faster pace, and only faster as time goes on, and physical interactions increase. Also, exit clogging and the bottleneck effect is likely to occur. The bottleneck effect is "the tendency of the evacuees to haste towards an exit, which results in clogging" [6]. At a first glance, all of this behavior may seem too random to be able to be accurately simulated. This is not the case, however, because everyone shares a common goal in exiting the building or floor. In order to begin to simulate a case in which an evacuation may be needed, the desired floor plan must be laid out. The walls and other items that cannot be traversed first must be placed in a grid. Now that the physical boundaries are set, people may be placed in the grid as individual cells with rules to follow and behavior to be modeled. L.F Henderson and Dirk Helbing are both researchers that applied physics theories into this type of simulation to predict the behavior of human movement as close as possible with factors, such as bottlenecking. Henderson applied gas kinetics to the theory of crowd motion. The gas kinetic theory did not apply well, however, since it had to be assumed that every person in the crowd had the same mass, energy, and velocity. Later Helbing introduced the idea to apply fluid dynamics into the behavior of humans in an evacuation. The physics applied became more realistic outcome [6]. This was especially seen in the situation where people must go through a narrow space. Just like fluid gets "jammed" when passing through a narrow hole, so too will people "jam" while leaving an exit. Finally, R. Colombo and M. Rosini were successful in introducing a model in which the panic effect of humans was accurately portrayed. All of these new calculations and theory applications allow for a smooth, realistic, and useful simulation.

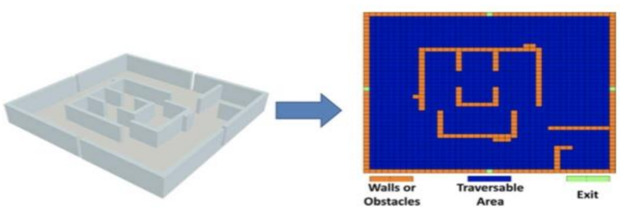

FIGURE 7 [6]
Model of grid layout

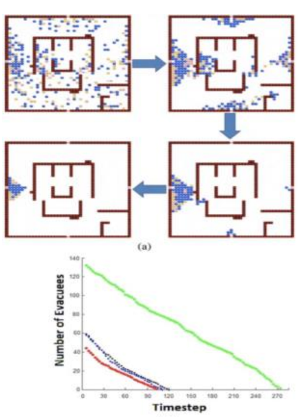

FIGURE 8 [6]
Simulation of pedestrian flow and its data

Yet another instance in which cellular automata have become a useful tool is in the research of pedestrian flow on a busy sidewalk. This may not seem like an important topic to simulate, but several considerations must be taken. First, sidewalks are a part of any functioning city or town. Using sidewalks are inevitable and sustain the order necessary in an area of large population density. Functioning sidewalks are not only convenient for the people using them, but also the roads that they surround. Similar to the jamming that may occur from an evacuation, it is also important to try to avoid jams in sidewalks, especially when considering that this overflow might conflict with traffic flow. Emergencies may also occur in the open. In such a case, it may be important to attempt to simulate what a type of event would cause people to do. Simulation can offer a look at how the design of sidewalks may be enhanced without a large-scale project. Such simulations may also provide insight into the behavior of humans themselves. One phenomena observed in pedestrian flow is the tendency for humans to reach for organization and order. This may be seen as humans staying on one side depending on the way that they are walking, whether towards or away from something.

For any type of simulation, special factors that will affect the overall prediction must be considered. For example, with the phenomena of organization in pedestrian flow, it should be considered that cultural differences will affect it. In China, it is preferred to walk on right side of a path, while in Japan it is preferred to walk on the left. These conditions were closely examined in the study of bi-directional pedestrian flow by researchers Hao Yuea, Hongzh Guan, Juan Zhang, and Chunfu Shao. In their research, they looked at three types of bi-directional pedestrian flow that the classified as bi-directional pedestrian flow, laterally-interfered pedestrian flow and evacuation pedestrian flow. Some pedestrian flow situations may be ignored because of the natural organization of humans. This led to the mentioned researchers to conclude that "the study on unsymmetrical pedestrian flow system should deserve more of our attention" [7]. This will allow the simulation to solely focus on situations that would be most beneficial to work on, such as the case of an evacuation or emergency.

In order to begin, rules and limitations must be set for humans, in which this case will be the objects occupying a cell in a simulation grid. Yuea and his colleagues applied an extended gas model, called the Lattice Gas Model to represent the subconscious behavior of humans when walking. Humans are similar to gases in the way that they have a tendency to "overtake slower ones from the left and to swirl the face-to-face pedestrians from the right is considered" [7]. In this simulation, a two-dimensional grid is needed to portray the sidewalk. Within this grid, "people" are placed in each cell and are now bound by the rules set. The person in each cell may only move into another cell that is only one cell away. This provides a realistic sense of movement, since obviously people move one "cell" at a time and cannot teleport. Another rule set forth in the simulation is that only one person may occupy one cell at a time. While the simulation is running, it is possible that two people will reach for one cell. This is when parameters come in to play. Parameters are a special kind of variable in computer programming language that is used to pass information between functions or procedures [8]. With this type of tool, the people in each cell are able to "speak" to each other and let one another know that they aim to occupy that space and "only one of them will be chosen randomly with equal probability. The selected pedestrian moves to the corresponding cell and the unselected pedestrians stay at the original position and will not move to any other cell" [9]. Once these simulations run, specific data are collected in order to reach the goal of making movement as efficient as possible. These data include the speed at which the pedestrians are flowing, the density in which they are packed in, and the direction in which people end up moving after a specific amount of time. This data may later be passed on to other engineers or designers so that they may create efficient sidewalk spaces, signs, and obstacles on the street.

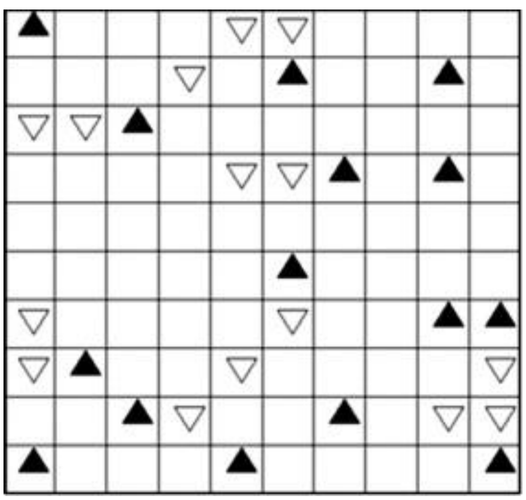

FIGURE 9 [8]
Schematic illustration of the bi-direction pedestrian flow model with two types of walkers

Sustainability:

Developing a sustainable future has been a growing concern for engineers in recent years with a growing awareness of our impact on current and future generations. While the general populous would define sustainability as having positive environmental benefits, one of the less common, yet undoubtedly important, definitions of the word describes something as being sustainable if it successfully improves what can be considered one's quality of life. According to the UN, sustainability extends past environmental factors in that it "requires meeting the basic needs of all and extending to all the opportunity to satisfy their aspirations for a better life" [10]. Under this definition, we can clearly see how useful cellular automata are in improving conditions of life. For example, while simulating the flow traffic in pedestrian heavy areas may help increase the movement on a large scale, individually, each of the pedestrians have had their qualities of life improved.

The potential that cellular automata provide for predictability with numerous factors at play have given urban planners the opportunity to design and plan in introspect new urban areas while keeping several issues in mind, such as the amount of land used, the traffic flow of the designed roads, and the quality of land left after the development is finished. In the simulation of urban development, researchers for the University of Hong Kong considered four factors in the sustainability of life and the environment during the urbanization of areas, them being "(1) not to convert too much agricultural land at the early stages of development; (2) to decide the amount of land consumption based on available land resources and population growth; (3) to guide urban development to sites which are less important for food production; and (4) to maintain compact development patterns" [11]. With these factors in mind and the manipulation of the rules and constraints in cellular automata, researchers are also able to prioritize some land, either assigning certain areas to be prime for urban development or too sensitive environmentally to be worked on. This is crucial in maintaining the sustainability and quality of people, the demand for new homes, and the area affected.

One's quality of life depends on not only providing basic needs, but also optimizing tasks and making life easier. Cellular automata aid in this aspect in the way that they allow for the study and analysis of different systems and mechanisms in order to improve quality and streamline processes.

Pros and Cons:

Now that the proper rules and physics are applied, and a proper grid layout is set, simulations can provide the best design for the fastest evacuations possible with the suggested designs. This is a great tool to have for engineers, architects, and designers for many reasons. For any corporation that they may work for, money and resources are always on the list of things of which use should be minimized. With a simulation, it is not necessary to build physical models to test an evacuation. A designer just needs the right software and skills to test their design. This type of research as compared to others will save time, money, and resources. A computer simulation will not require research to go further than the lab in which the computer for simulation is set. In addition, real physical models will not be needed for testing. Without cellular automata, if a study needs to be done for the most efficient area that provides optimal evacuation routes, a real physical model would be the most likely candidate to perform tests. This could create a big hole in a budget for research and design. Some situations that may need to be tested may also be so dangerous or obscure that they cannot ethically be carried out. If research and testing is needed to examine an intense emergency, such as a major catastrophe, real people will of course not be allowed to participate as subjects in the test. The dangers are far too great to put peoples' lives at risk. Not only is it unethical to do this for their physical but also their mental health. Even if a real catastrophe does not occur, a person may still be troubled psychologically, and that cannot be allowed. Cellular automata are completely digital and will not require the use of real people or resources. Economically, it is far better to hypothesize and predict with machines rather than perform a live test.

Cellular Automata are advantageous in the way that it can take a scenario that looks unpredictable and form a structure on it. This level of order makes prediction a lot easier to comprehend. Wolfram proposed that cellular automata has four classifications, which are "those which rapidly tend to equilibrium regardless of initial conditions, those which settle into oscillations, those whose output appears to be random, and those which are able to propagate complex structures forward in time" [12]. When we narrow down our unpredictable behaviors into categories that they may fall in, calculations and predictions are much more feasible, which is why cellular automata are so useful in the prediction of behaviors. When predicting humans' behaviors, it is impossible to look into their mind and determine what they will do, but cellular automata focuses on the fact that in a given situation, a main goal is found by everyone, and thus they will all follow an algorithm to reach that goal. Most, if not all, will intend to exit an area that poses danger, and most people walk in similar manner on a sidewalk. This is how cellular automata are able to look at systems on a macro level while modeling and proposing its intricate rules on a micro level. Cellular automata offer a balance between randomness and uncertainty with order and mathematics. Dr. Xin-She Yang, a senior researcher of Oxford University said that "computational modeling and numerical computation have become the third component [in analysis], bridging the gap between theoretical models and experiments" [13]. Machines are becoming smarter and smarter every day, being able to learn at a higher rate than any human can. Cellular automata have the potential to perform complex calculations that can narrow down the precision of accuracy in future research and testing.

The use of cellular automata is not perfect, however. One of the primary drawbacks that one must consider is the fact that while simulations have the capacity to accurately model the systems that they are programmed to represent, it is difficult to replace real-life, experimental data with a computer simulation. Although cellular automata are capable of receiving large amounts of input conditions and running simulations thusly, they are only capable of calculating the outcomes that the programmer is aware of and are unable to predict unknown events and interactions. This is why, as we had discussed in our real-life applications section, that cellular automata excel in predicting traffic flow and pedestrian movement. Both of these situations are well-documented and well-known. Therefore, it can be determined that cellular automata are better fitted for analyzing data to predict similar situations based on different initial conditions, rather than being used for experimental research simulations. Additionally, cellular automata provide a great way to observe large scale systems and mechanisms. As soon as there is a need to examine the inner workings of the systems, the program must be remade to include these conditions. While researchers and analysts save time and money by not setting up the experiments in real-life, the sheer amount of work required to accurately simulate more complicated mechanisms may present researchers with a dilemma. Should they spend time and money to hire programmers to make an intensely complex cellular automaton, useful for a certain subset of situations, or should they abstain and proceed to conduct the observations, experiment, and analysis experimentally in real-life is a question that will strongly need to be considered before taking action. Finally, it must be considered that different automata are needed when a new situation is presented, and new variables and parameters must also be presented. There is not one universal program for every type of automata. Each automaton comes with its specific layout and rules. A cellular automaton intended for a biological model will not work for an automaton intended for the crystal growth of a chemical compound. Every situation calls for a new layout and new rules for the objects in that grid. This costs time and hard work. New research will need to be done for a long time to expand the capabilities of cellular automata so that it may be possible to place a specific scenario in an automaton without the need to start from scratch.

LIVING UP TO ITS POTENTIAL

The work of Von Neumann, Stephen Wolfram, and other computational statisticians have provided society with an efficient method of conducting studies and predicting events, and have provided the world with a new, emerging field of study. Cellular automata pool data about individual elements of a defined region of space, as well as the data of other elements surrounding the primary focus and apply that data to different rules that produce the next evolution for each of the elements in the system. This interaction between adjacent elements models the interactions that we see in real life: how different chemical compounds react with each other based on each of their make-ups, and how an infection may spread through physical contact, or how a forest fire spreads. Cellular automata allow for the simulation and study of these real-world events without the need to expend valuable resources in order to gain information, regardless of how obscure or dangerous the studies may be. Cellular automata are a vital resource that allow not only the field of engineering, but fields of all sorts to begin to predict the future and improve on it.

Tyler Sheinberg
Pablo Salazar

SOURCES

[1] P. Sarkar. "A Brief History of Cellular Automata. University of Central Florida. Accessed 2/20/2018. https://www.cs.ucf.edu/~dcm/Teaching/COT4810-Spring2011/Literature/CellularAutomata.pdf

[2] H. Koening. "The Game of Life News". PentaDecathlon. Accessed 1/14/2018. http://pentadecathlon.com/lifenews/2006/06/single_glider_collision_survey.html

[3] "cubes.io: 3d cellular automata." bigblueboo labs. Accessed 1/28/2018. http://cubes.io/

[4] V. Ya. Shevchenko, S. V. Krivovichev, and A. L. Mackay. "Cellular Automata and Local Order in the Structural Chemistry of the Lovozerite Group Minerals". SpringerLink. Accessed 1/14/2018. https://link.springer.com/content/pdf/10.1134%2FS1087659610010013.pdf

[5] B. Chopard. "Cellular Automata Modeling of Physical Systems". Accessed 2/14/2018. https://link.springer.com/referenceworkentry/10.1007%2F978-0-387-30440-3_57

[6] E. Boukas, I. Kostavelis, A. Gasteratos, G. Sirakoulis. "Robot Guided Crowd Evacuation". IEEE Transactions on Automation Science and Engineering. 2015. Accessed 1.28.18. http://ieeexplore.ieee.org.pitt.idm.oclc.org/stamp/stamp.jsp?tp=&arnumber=6824875

[7] L.F Henderson. "On the Fluid Mechanics of Human Crowd Motion". ScienceDirect. Accessed 2/14/2018. https://www.sciencedirect.com/science/article/pii/0041164774900276

[8] H. Yuea, H. Guan, J. Zhang, C. Shao. "Study on Bi-direction Pedestrian Flow Using Cellular Automata Simulation". ScienceDirect. Accessed 2/14/2018. https://www.sciencedirect.com/science/article/pii/S0378437109007882?via%3Dihub#b7

[9] G. Maira Barbosa de Oliveira, S. Regina Cardoso Siqueira. "Parameter Characterization of Two-Dimensional Cellular Automata Rule Space". Accessed 2/14/2018. https://www.sciencedirect.com/science/article/pii/S0167278906000777

[10] G. H. Bruntland. Chapter 2. Our Common Future: Report of the World Commission on Environment and Development. 3 3.20.1987. Accessed 3/27/2018. http://www.un-documents.net/ocf-02.htm

[11] X. Li, A. Gar-On Yeh. International Journal of Geographical Information Science. Modelling sustainable urban development by the integration of constrained cellular automata and GIS. Accessed 3/27/2018. https://www.tandfonline.com/doi/abs/10.1080/136588100240886

[12] J. Merritt. "Cellular automata as emergent systems and models of physical behavior". U. of Illinois. Accessed 2/15/2018. http://guava.physics.uiuc.edu/~nigel/courses/569/Essays_Fall2012/Files/merritt.pdf

[13] X. Yang, Y. Young. "Cellular Automata, PDEs, and Pattern Formation". University of Oxford. Date Accessed 2/21/2018. https://arxiv.org/pdf/1003.1983.pdf

ADDITIONAL SOURCES

A.C.J. de Korte, H.J.H. Brouwers. "A cellular automata approach to chemical reactions; 1 Reaction controlled systems." Chemical Engineering Journal. Accessed 1/28/2018. https://ac.els-cdn.com/S1385894713005809/1-s2.0-S1385894713005809-main.pdf?_tid=4dcd08f8-f974-11e7-9b45-00000aab0f6b&acdnat=1515966546_795177c246dc3e26722acd62a97e1229

Jing Xun, Bin Ning, Ke-ping Li, Wei-bin Zhang. "The impact of end-to-end communication delay on railway traffic flow using cellular automata model." Transportation Research Part C. Accessed 1/28/2018. https://ac.els-cdn.com/S0968090X13001307/1-s2.0-S0968090X13001307-main.pdf?_tid=4be55fd4-f976-11e7-9616-00000aacb35d&acdnat=1515967393_ee06a3e3d183ebca47bcf4367654d3ca

Karl-Peter Hadeler and Johannes Müller. "Cellular Automata: Analysis and Applications." SpringerLink. Accessed 1/14/2018. https://link.springer.com/content/pdf/10.1007%2F978-3-319-53043-7.pdf

M. Rickert, K. Nagel, M. Schreckenberg, A. Latour. "Two Lane Traffic Simulations using Cellular Automata". 2008. Accessed 1.28.18. https://arxiv.org/pdf/cond-mat/9512119.pdf

r10aldinho. "ca code for matlab". 2006. Accessed 3.27.18. https://www.mathworks.com/matlabcentral/fileexchange/39492-ca-code-for-matlab?focused=3770289&tab=function

Weng, W.G., Chen, T., Yuan, H. Y., Fan, W.C.. "Cellular Automaton Simulation of Pedestrian Counter Flow with Different Velocities". Center for Public Safety Research. 2006. Accessed 1.28.18.https://journals.aps.org/pre/pdf/10.1103/PhysRevE.74.036102

ACKNOWLEDGMENTS

We would like to thank our co-chair, Elizabeth Lyman, for guiding us through the processes of creating this research paper. We would also like to thank the library for providing us with the necessary resources to make our research possible. Finally, we would like to thank our chair, Ryan Lauer, for providing us helpful incite from the perspective of someone working in this field.