Reza Sarijlou

Mid-Term Report

REU Project Fall 2003

Advisor: Dr. Elena Cherkaev

Social Networks

MID-TERM REPORT

I have been continuing my work on networks from the year before; however, instead of focusing on networks as a whole, I am focusing on one particular network. I chose to discover the basic elements that are concerned with social networks.

In particular I used graph theory as a tool to try to fit the social network to. Graph theory has elements such as nodes and edge's that could represent people and there relationship among them. Then I had to look for the different kinds of social networks and see what kind existed. I wanted to see how it was that a social network could be created. This meant that I had to look at the number of people and how they where link together. I started to look at the network that was made in a university setting, meaning how the people in a University are related. I saw that there where such connection between people that was undirected. I read the work done by Mark S. Handcock from University of Washington and some of the social networks that he had studied such as the use of needle among drug addicts. Once I had some idea as to what could make up relation among people, then I wanted to implement this again using graph theory in matlab. Using the matlab gplot function I looked at some random graphs. My goal was to use the gplot function in a program and then try to implement the relations among the people in the network. Soon I realized that it was much difficult to implement this and I started to read the work of Rainer Hegselmann and came across the use of Cellular Automata.

Cellular automata is a machine which includes the state of neighboring automata in it's transition rules. Some of the features of a CA are D-dimensional lattice, time is advancing in discrete steps, we have a finite number of states, the cells change their states according to local rules, the system is homogeneous (meaning that the set of all possible states is the same for all the cells).

This method is an efficient way of modeling social networks. This could be a network that is random or one that is based on some rules. Each cell or person is making a transition meaning going to one of the other possible states depending on what his/her neighbors is doing. I have been working on a program that I am going to modify in the coming week to have different rules and I want to look at the output and study the structure of the network. I want to come up with some social rules and implement them to see how the different rules can affect the structure of the social network and what will not affect it both in a major and minor way.

I have written a matlab code Index A that will create a plot of a 2D lattice that will follow cretin rules. I hope to modify this code by adding several other rules and then again looking at the structure of the output and look for some patterns. I am interested to see if some patterns might develop or if a particular cell does something that the other cells are not doing.

As a guide I will be using the paper of Rainer Hegselman, "Modeling Social Dynamics by Cellular Automata," I think this paper has a lot of the ideas that I want to use in my simulation. Also this paper will help to see how I can interoperate the results that I get from my simulations.

One thing that I must consider is that how complicated I want to make the transition rules in the in the program. How important is this to finding out what effects the structure of the network. I will start with simple cases and try to make them more complicated interactions and see if this is making a big or small effect on the network. I will do some playing around with this to see what is a ideal relation for understanding the structure. I will start with simple transition rules and make them complicated as I go on. One thing that I want to look at is how random networks are different then a network with cirtin transition rule.

In conclusion CA is a good method to use to have a better understanding of how the structures of these networks are built and what affects them in a big or small way. My main goal in using CA to run these simulations is to see what is happing to the structure. As you can see in Fig1-4 these are some outputs of a social network using the CA method. I hope to look at some more of these kinds of graphs for different transitions to see if I can learn more about what is happing to the structure.

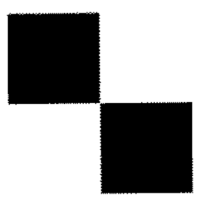

Fig1

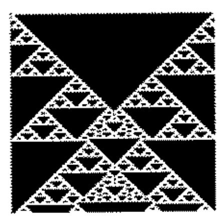

Fig2

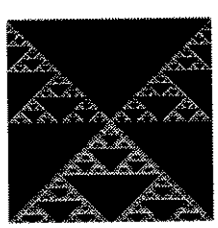

Fig3

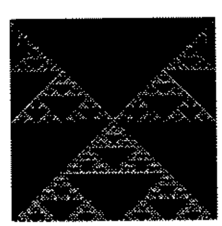

Fig4

Index A.

```
quitVar = 1;

while quitVar > 0

  clear all;
  close all;

  picWidth = input('Enter number of cells: ');
  disp('Enter the cells that are initially off, cells are referenced
from (1,#Cells)');
  disp('Example input [5 3] will turn off cells 5 and 3');
  initOff = input(' ');

  firstRow = ones(1,picWidth);

  for k = 1:length(initOff)
       firstRow(initOff(k)) = 0;
  end

  n = length(firstRow);

  pic = zeros(n);

  pic(1,:) = firstRow;

  for i = 2:n
   for j = 2:(n-1)
       if(pic(i-1,j-1) == 0)                    %if left neighbour is off
        pic(i,j) = pic(i-1,j+1);                %cell is opposite of right
neighbour
       else                                     %if left neighbout is on
        pic(i,j) = 1;                           %initialize to on
        if(pic(i-1,j) == 1)                     % and cell was on
         pic(i,j) = mod(pic(i-1,j+1)+1,2);      % cell is opposite of
right neighbour
        end
       end
   end

        %Consider the boundry cells seperatly
        %Wrap our world around when we reach the horizontal limit
        %In other words, let pic(k,n) also be pic(k,0)

        if(pic(i-1,n) == 0)                     %if left neighbour is off
         pic(i,1) = pic(i-1,2);                 %cell is opposite of right
neighbour
        else                                      %if left neighbout is on
         pic(i,1) = 1;                            %initialize to on
         if(pic(i-1,1) == 1)                      % and cell was on
          pic(i,1) = mod(pic(i-1,2)+1,2);       % cell is opposite of right
neighbour
```

```
         end
        end

        if(pic(i-1,n-1) == 0)                    %if left neighbour is off
         pic(i,n) = pic(i-1,1);                %cell is opposite of right
neighbour
        else                                     %if left neighbout is on
         pic(i,n) = 1;                           %initialize to on
         if(pic(i-1,n) == 1)                     % and cell was on
          pic(i,n) = mod(pic(i-1,1)+1,2);      % cell is opposite of right
neighbour
         end
        end
  end

  colormap(gray);             % Change the colour to grayscale
  imagesc(255*pic,[0,1]);
  axis equal;
  axis off;

  disp('________');
  quitVar = input('Try another setting? (1 to continue, 0 to exit)');

end
```