Agent Based Model For Civil Unrest

Angus Blance

April 24, 2023

Contents

1 Abstract

2 Introduction

Instances of large-scale disorder happen around the world regularly, resulting in significant damage to property and human life. Given the destructive nature of such events, finding means in which to mitigate the damage is necessary. In this report, we present an Epstein's agent-based model (ABM) that replicates the behaviour of civil unrest, and we demonstrate how the use of smoke by police forces can effectively reduce the number of participants in a riot.

What Causes A Riot

Large scale disorder can be caused by many things such as deprivation, poor police relations, legitimacy of a government or just to many people in one area. Once a riot has started it then gets out of control very fast usually epidemic like(ref 2.). Factor's for this can be knowledge of an ongoing riot through social media or friends and rational choice theory as in if there are shops being looted and police are not doing anything you are more likely to join. Riots can then be continued if there is inadequate policing.

Modelling such events can be done in many ways and takes many disciplines, such as crowd dynamics, mathematics, Computer science and Psychology. In this report the model is based upon Mathematics and heavily uses com-

puter programming to simulate our mathematical model. We also explore environmental criminology theory, which is the idea that someone is morel likely to join in the chaos if they can see or have knowledge of said chaos.

3 My Model

3.1 What is an Agent Based Model

An agent-based model (ABM) is a computational modelling technique used to simulate the behaviour of individual agents and their interactions within a system. Each agent within the model has a set of rules which it obeys by, we then observe how these rules that our agent does affects itself and the other agents around it.

For this study, I employ an ABM to simulate Epstein (2002) Model for civil violence , which consists of three agents: quiet agents (Q) who are not participating in the riot, active agents (A) who are rioting, and police agents (C) who arrest actives. Each agent is placed on an N by N board

Attributes for these agents are given:

Quiet:
-Qalk around the board at random
-Calculates the amount of police and actives within its vicinity
-Becomes active depending on number of Cs and As in vicinity

Active:

-Walk around randomly

Police:

-Walk around randomly

-When near an active they arrest them and take them off the board

3.2 Mathematical Model

How Quiets Chose to Become Active

The main mathematical model that is now going to be shown is taken from Epstein (2002) where we first find the Grievance given by

$$G = H(L - 1) \tag{1}$$

where H is the hardship of each specific non police agent and L is the legitimacy of the government which we set at the beginning of the model.

Now we find the probability (p) to be arrested which each non police agent needs to calculate before they decide whether or not they wish to riot

$$P = 1 - EXP(-k(C/A)_v) \tag{2}$$

Note the subscript v notates the vision which each civilian has. C is the number of police in the vision v and A is the number of rioters in vision v. The civilian counts himself when counting A as he wants to see how the riot

would be once he is added. K is just a constant which we set such that P = 0,9 when C=A=1, so k = -ln(0.1).

For an example, We look to Figure 1 where we have our green dot as our quiet, the red dots are our actives, the blue dots are our police and the black box represents its vision v of our green quiet in the middle. For this example the vision of our agent is 1. We have 2 police and 2 actives within our agents vision.

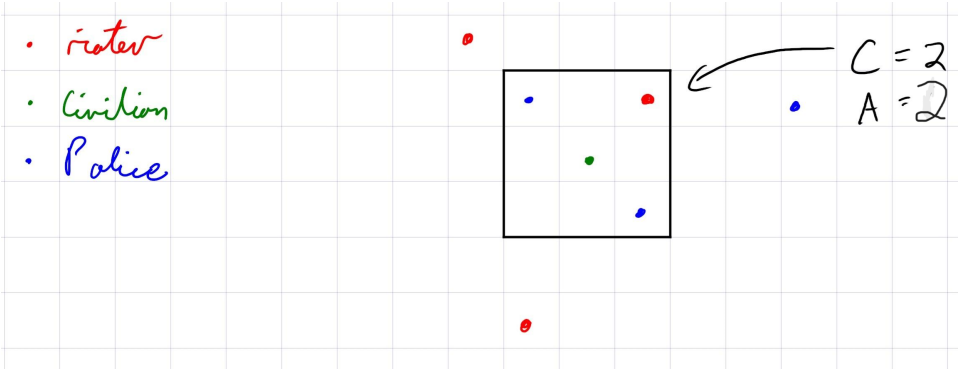

Figure 1: vision example

from this we can now define the net risk $N = RPJ$ where J is the maximum jail time one faces once he has been caught by a police officer. The max jail time is set before the model is run to whatever jail time is deemed fit and the amount of time someone will spend in jail is given from U(0,max jail time).

we now have all we need to create the agent rule:

$$Agent\ rule: if\ G - N > T\ riot;\ otherwise\ stay\ a\ civilian \tag{3}$$

so the difference in G and N is the expected utility of expressing ones private grievance whilst T is the utility of not.

4 Computational Approach and Implementation

First of all we have our N × N board say N=10 we the have figure 2

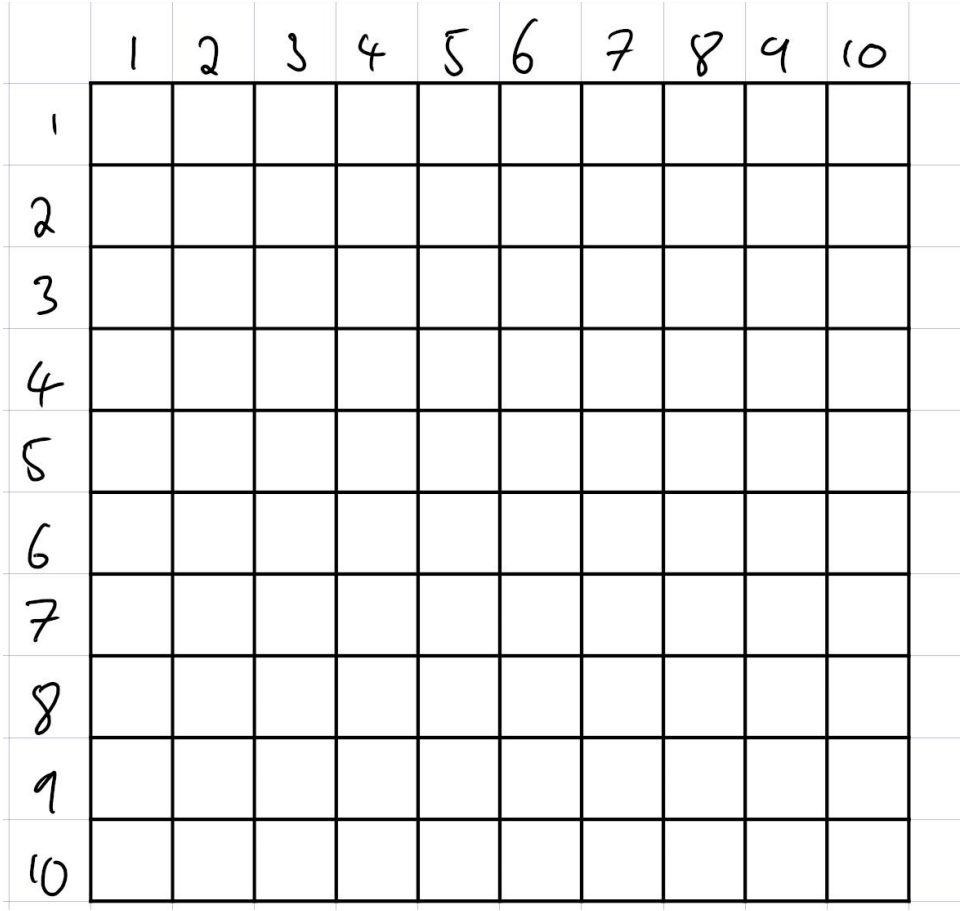

Figure 2: Grid

Now agents are introduce to the model with attributes as described prior, the Actives are denoted by the colour red, our quiets green and our police

Blue. Say we loop through our hole N × N grid and place 20 quiets and 10 police at random we will end up with something like Figure 3

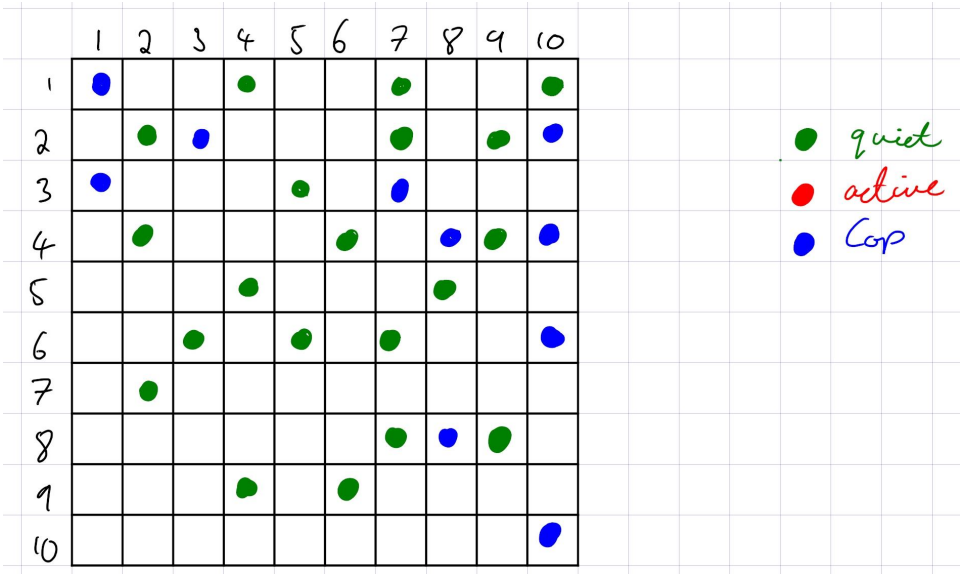

Figure 3: Grid with Agents

Great! That's our hole grid set up. But also note that each individual quiet has 2 attributes to them (risk level and Hardship) which is easier to see on a 10 × 10 example run on our code in Matlab as shown in figure 4

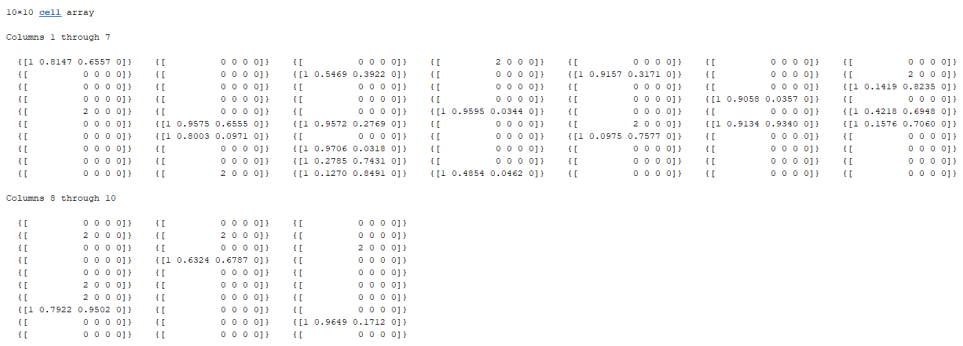

Figure 4: Grid on Matlab

Notice how our model stores the risk level and Hardship by making our board a cell array and storing multiple values for each cell. Also see how all of the cops have no hardships or risk, This is because they do not have the attribute to become an active meaning giving them such values will be redundant.

4.1 Moving

In the model every agent will only move one space (Cell) at a time how they move is shown in figure 5 where the arrows on the diagram on the left show the direction in which each agent will move and the diagram on the left shows the grid after each person has moved

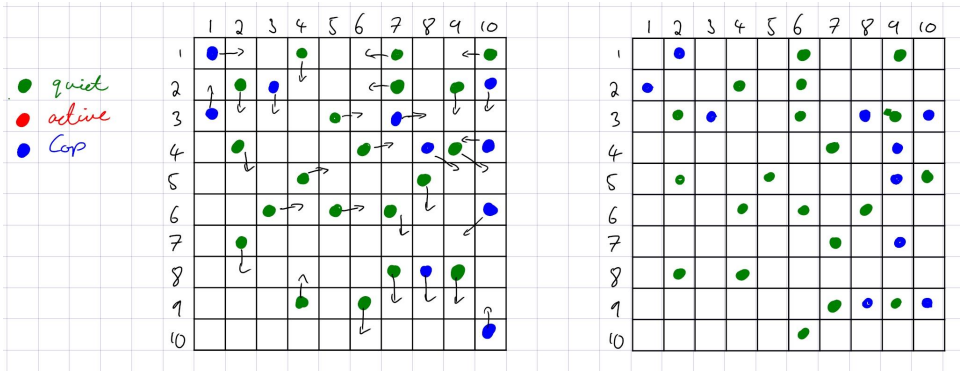

Figure 5: French riot

see how the police in column 10 row 4 (10,4) moves to space (9,4) which is already occupied by a quiet? This happens due to how the model loops through the grid. This is done from the top left of the grid to the bottom right (like a book) meaning that the quiet in space (9,4) has already moved to (10,5) before the police makes his move into the space the quiet was just in.

4.2 Quiets and Actives The Descension to Riot

Just as before our model loops though the hole board, first starting form the top left and ending up on the bottom right. Whenever we land on a quiet we take its risk and hardship as well as searching in a M $\times$ M grid around said quiet to collect the number of A's and P's within its vision, as shown in the mathematical model.

To explore this further lets take figure 6 as an example

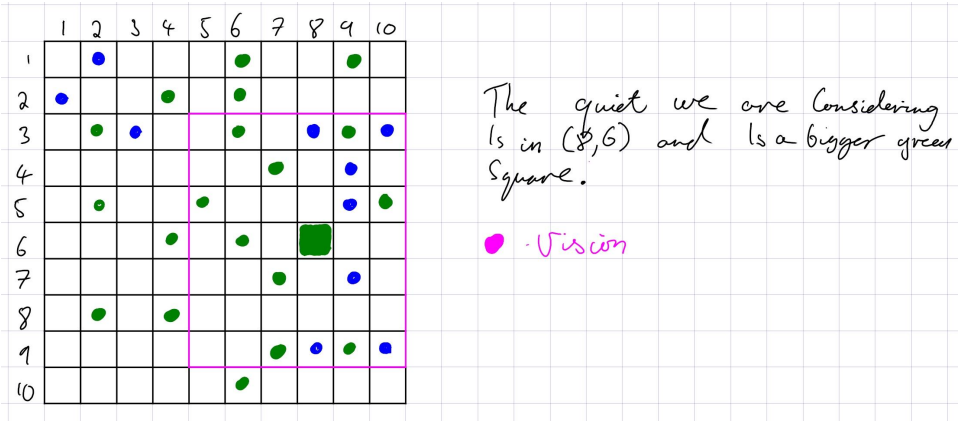

Figure 6: Vision Example for Single Case

B Now looking at what data we have for our particular case in (figure 6) we have C = 7 and Q = 9 not including the quiet we are currently considering.

For this particular quiet to become active we need our agent rule to be satisfied. for this we want N to be small so that the left hand side is greater than the right hand side, looking at N

$$N = RPJ \tag{4}$$

Where R is the level of risk aversion they will take given to each non police agent, higher level of risk aversion the less likely they are to take risks. We can see that if R increases N will clearly increase as well making the agent less likely to become active. Just as increasing R makes an agent less likely to riot, using the same logic we can see that decreasing R makes an agent

more likely to riot.

Now looking at P the other factor in calculating N

$$P = 1 - EXP(-k(C/A)_v) \tag{5}$$

We have C and A from figure 6 which we sub into (5) to get get P = 0.9999999 for our specific case, which makes sense as there are no A's within the vicinity of our Q and there are many C's meaning the probability of arrest should be high. Also see that as C $->\infty$ P $->$ 1 (less likely to become active) and as A $->\infty$ P $->$ 0 (more likely to become active). looking at (4) again we can see that greater P values creates a greater N value making our agent less likely to become active, the more likely someone is to be arrested the less likely they will become active

say in our particular case J = 5 which is a smallish jail sentence which should mean a rioter is more likely to riot then our N = R$\times$4.9999995 and using this and substituting G from (1) we obtain

$$G - N = H(L-1) - R \times 4.9999995 \tag{6}$$

Say we want our Q to become active we want (6) to be Large such that it is $>$ T meaning we want H(L-1) the total grievance of this agent to be large and R $\times$ 4.9999995 to be small. First for H(L-1) to be large we need want the hardship of our specific individual to be large and the legitimacy of our government to be low, Note that just having a high level of hardship does not mean our Q will become A, we need both a high hardship AS WELL as a low perceived legitimacy of the government.

4.3 Police Arresting

C's move just the same way as the Q's and A's, However they cannot become A. The main characteristics C's have is the ability to arrest this is done by them just happening to walking about and searching similarly to Q's and A's "scanning" the area around them. If an A is within a C's vision it then arrests them putting them into "jail". The jail is a matrix storing all actives that have been caught by police. What an arrest would look like in my model can be shown in figure 7 where on the left a C is next to an A and in the next the A has been put into jail. The jail sentence J is set to 5 in this example meaning it takes 5 iterations for our agents to be set free from our jail. The Jail inmate's number next to him is the time he has spent in jail which goes down by 1 after the turn has taken place

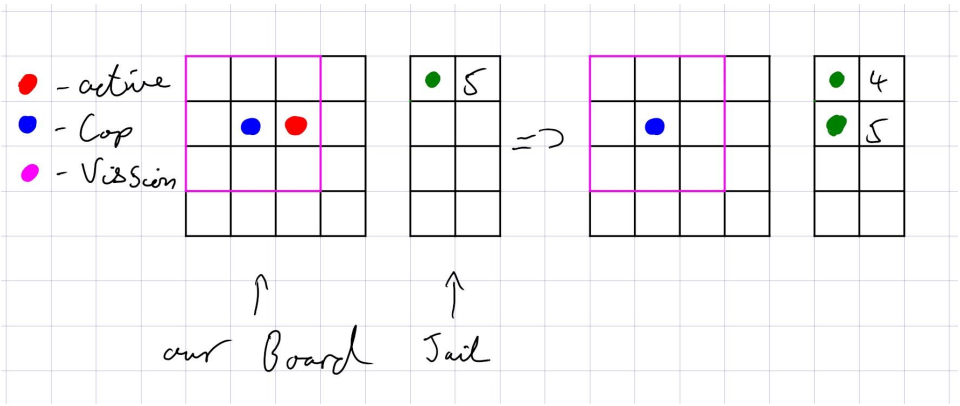

Figure 7: Arresting Example

4.4 Smoke

For my smoke a created another empty N × N array same as figure 4. But for this one each cell has 2 values the first being a 1 or a 0 representing if the square is occupied by smoke or no smoke respectively. A simple example of this is seen in figure 8 where the blue C sees an A within its pink vision. It then promptly throws smoke at the A

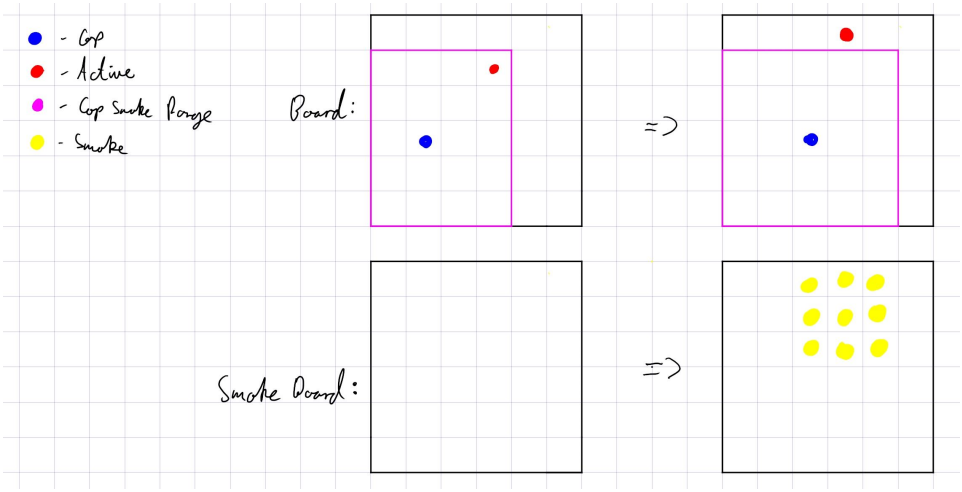

Figure 8: French riot

5 Results From my Model 1

In this section I run my model but change the variables of the model such as the C/A density, the vision of agents and the perceived legitimacy to see if I find any trends. sections 5.1 up to 5.4 are nearly identical to the results of Epstein's model showing that my model works as intended. 5.5 and 5.6 are not on any paper that I have found.

Each iteration of my model consists of:

1 - all agents moving,

2 - non police agents deciding whether or not to riot

3 - police agents arresting one active within its vision

5.1 Deceptive Behaviour

This is the case where a quiet is green when C are not near by but when cops leave the area they become active. This is due to the C/A ratio changing. As in when a C is near by the probability of arrest increases making G-N smaller. If smaller than T then they turn Q. The same applies for if a cop leaves there vision that they have a higher chance to become active.

Epstein compares this behaviour to Mao Zedong's directive where revolutionary's must "swim like fish in the sea" describing how revolutionary's must hide in the crowd and not distinguish themselves from the quiets waiting for the right time to start a revolution.

5.2 Snowball Effect

If a small group of quiets go active then this can lead to even more quiets going active, we observe this in figure 10 where area's where there is a low density in cops leads to quiet's becoming active, which in turn decreases the C/A ratio giving a lesser probability of arrest meaning even people with a high level of risk aversion have a higher chance of joining in the riot on the next iteration.

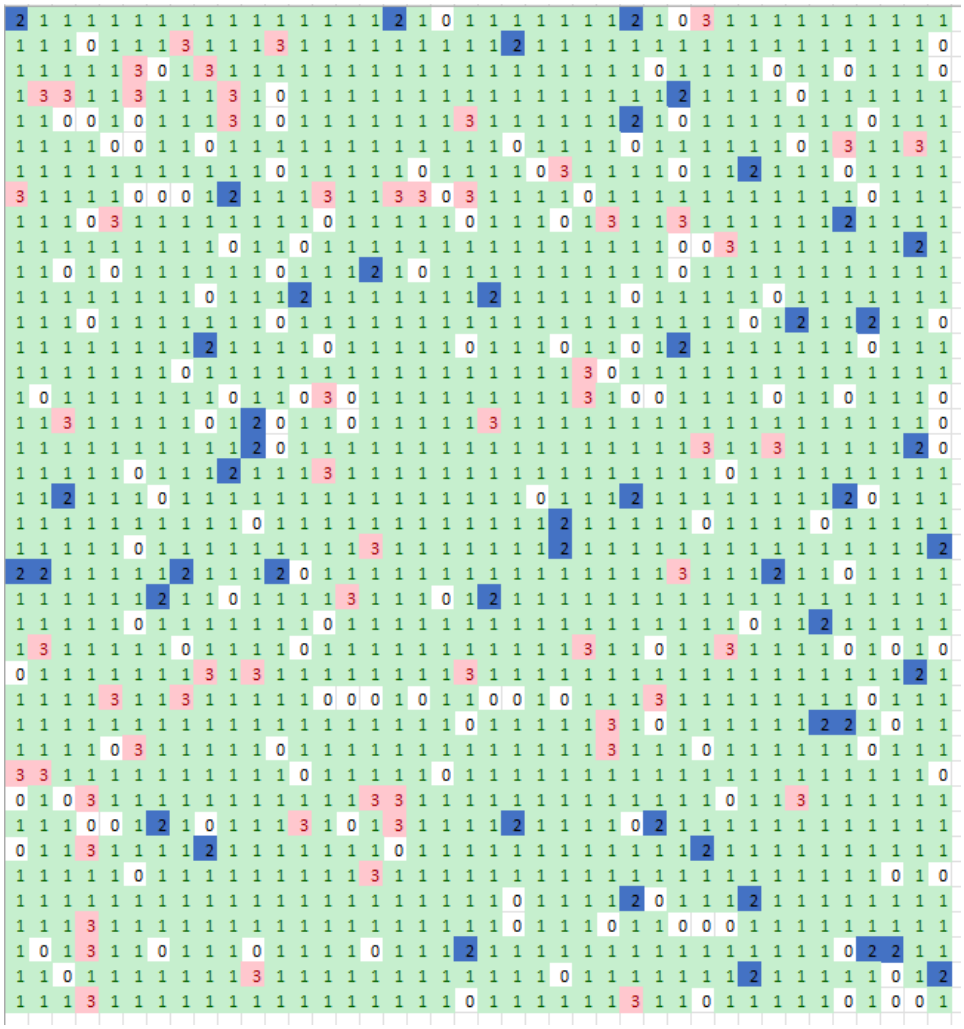

Figure 9: C=1400,Q=50,V=2,T = 0.1, L=0.89

This can be illustrated even further by decreasing our T value shown in figure 11.

To be the first to riot you have to have a very low risk aversion or very high hardship. But the more people that join the riot the lower the probability of arrest is and the risk involved is much lower.

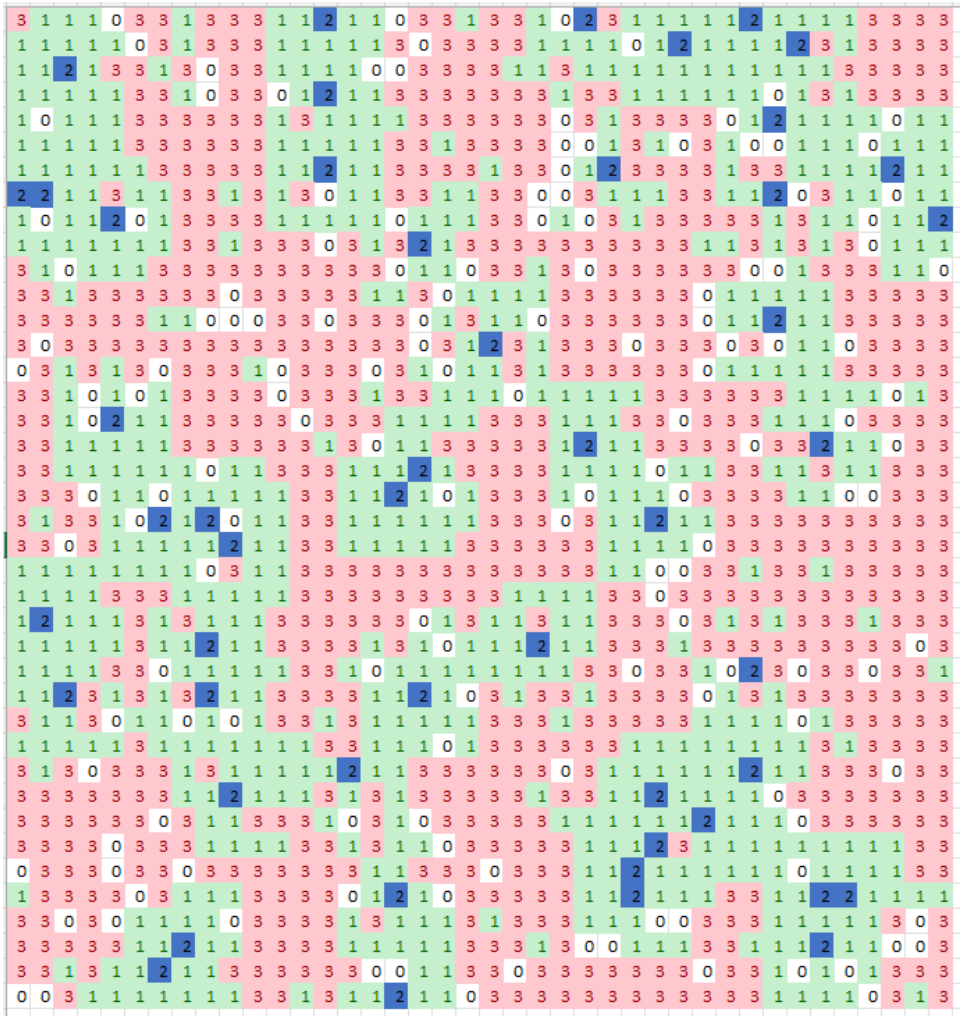

Figure 10: C=1400,Q=50,V=2,T = 0.01, L=0.89

5.3 Effect of Legitimacy

Initial conditions: C = 118, Q = 1120, v = 6(all agents), T = 0.1

To explore the effect of legitimacy the model is run the model starting with L = 0.9 and decreasing our legitimacy by 0.01 each iteration of the model. Looking at figure 12 We see the yellow legitimacy line is straight and constantly decreasing, the number of rioters is horizontal running all the way along the x-axis so the number of rioters = 0 at every iteration and the number of prisoners steadily increases throughout. The number of rioters is never above 1 this is because a small decrease in legitimacy only makes a few quiets want to come active and the get arrested straight away not allowing the snowball effect to take place.

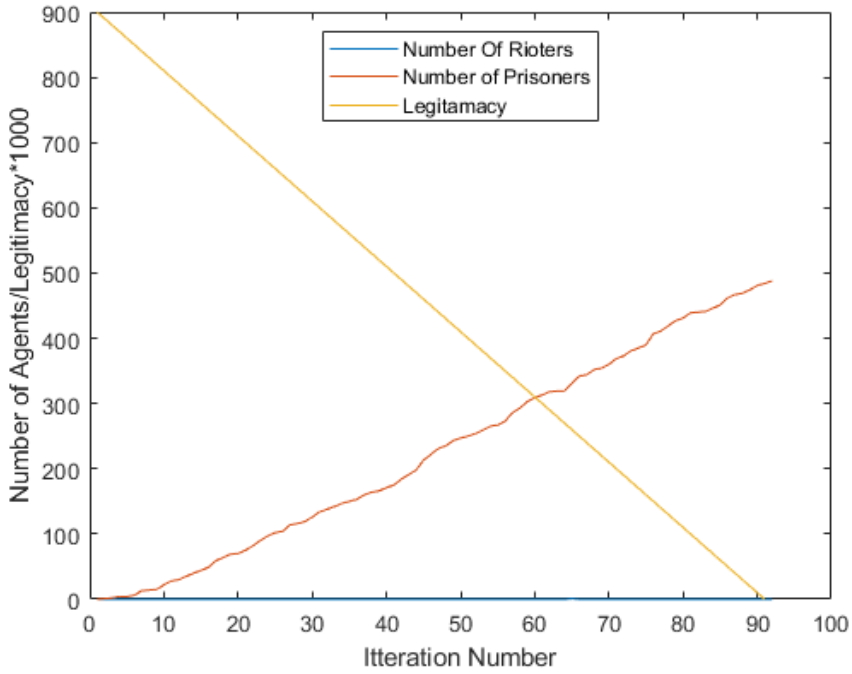

Figure 11: Slow Decrease in Legitimacy

Now (figure 13) instead of a slow decrease in legitimacy over time we

have a constant high L = 0.9 for 77 iterations and then a sudden drop to L = 0.7 for a further 43 iterations. There is stark difference between the figures 12 and 13. 15 has a massive outburst in actives revolting against the government they are in with a significantly smaller change in legitimacy.

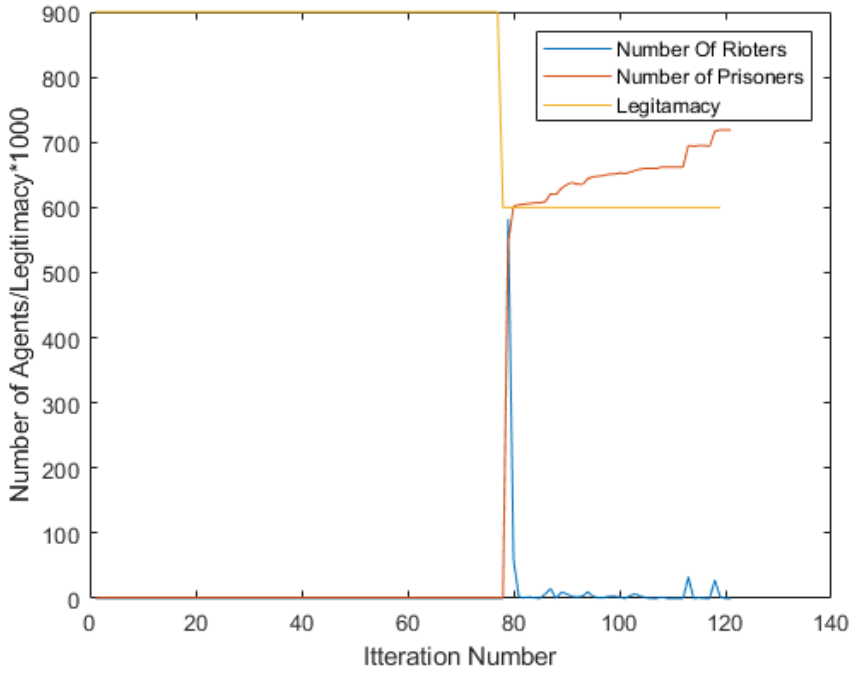

Figure 12: Effect of a Sudden Drop in Legitimacy

The reason that figure 15 has a sudden revolt is due to the sudden decrease in L creating a larger amount of active which depress local C/A ratios which mean that in the next iteration not only will they have a lower L but also the lower C/A making the likelihood for one agent to riot much greater (snowball effect)

This result gives insight in tack-ticks which can be used for somebody to create a revolution. It shows that slowly releasing negative information from a government will not cause a riot or any real change. However releasing large amounts of negative information about a government at one time will create an even greater impact, even if the value of legitimacy the government decreases by a smaller amount than the total decrease in legitimacy in the slower method. Also note that releasing information is not the only way to lower legitimacy.

Epstein compared this effect to a few real life situation's such as Mao Zedong who would isolate himself himself in the mountains for a big reappearance. Or even the return of leaders such as Vladimir Lenin and Ruhollah Khomeini after being exiled would come back with a big bang. Possibly this could even represent "triggering events" where the legitimacy of a government suddenly drops for instance an assassination.

5.4 Police Reductions

Initial conditions: C = 118, Q = 1120, v = 6(all agents), T = 0.1, L = 0.86,

It would be though under an oppressive regime where there are loads of police enforcing the regime on their citizens that decreasing the oppressiveness by taking away police could result in a happier population. However as our model depicts in figure 16 we see that the people wait around as the government reduce the number of police until there seems like there

are too few police to resist. This is very different to the gradual decrease in legitimacy shown above. In fact the small revolution that breaks out in figure 16 closely resembles a sudden decrease in legitimacy. This shows that for the model, the perceived legitimacy of the government is fundamentally different to the density of police in the model

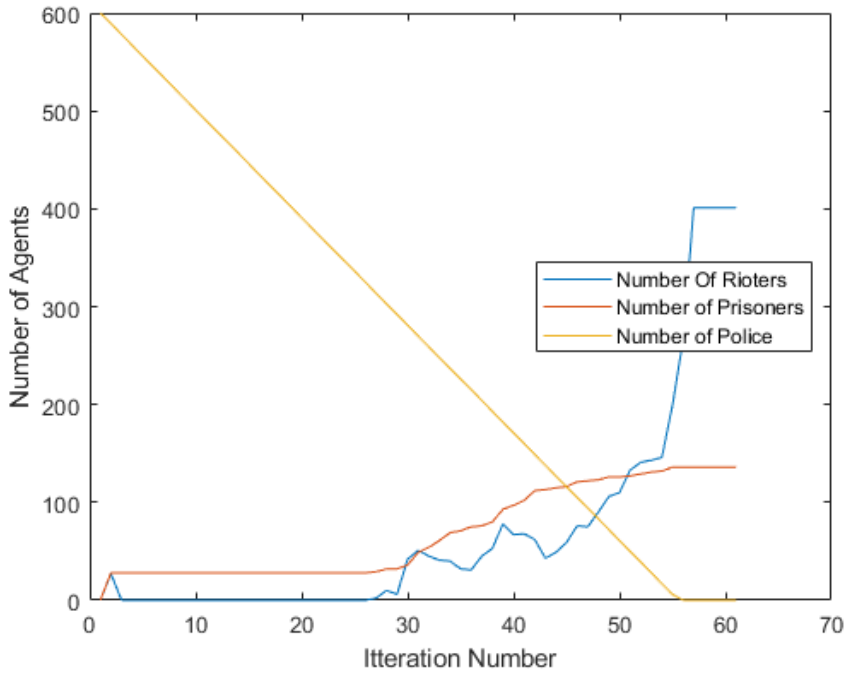

Figure 13: C=600,Q=800,v=2,L=0.4,T=0.1

Epstein (2002) compares what we see in the model to the French Russian and Iranian revolutions.

5.5 Simulating a 'Riot'

For my model I will simulate some sort of riot by placing all of my non police agents on one side of the board being the riot, and then the police on the other side of the board shown in figure 17

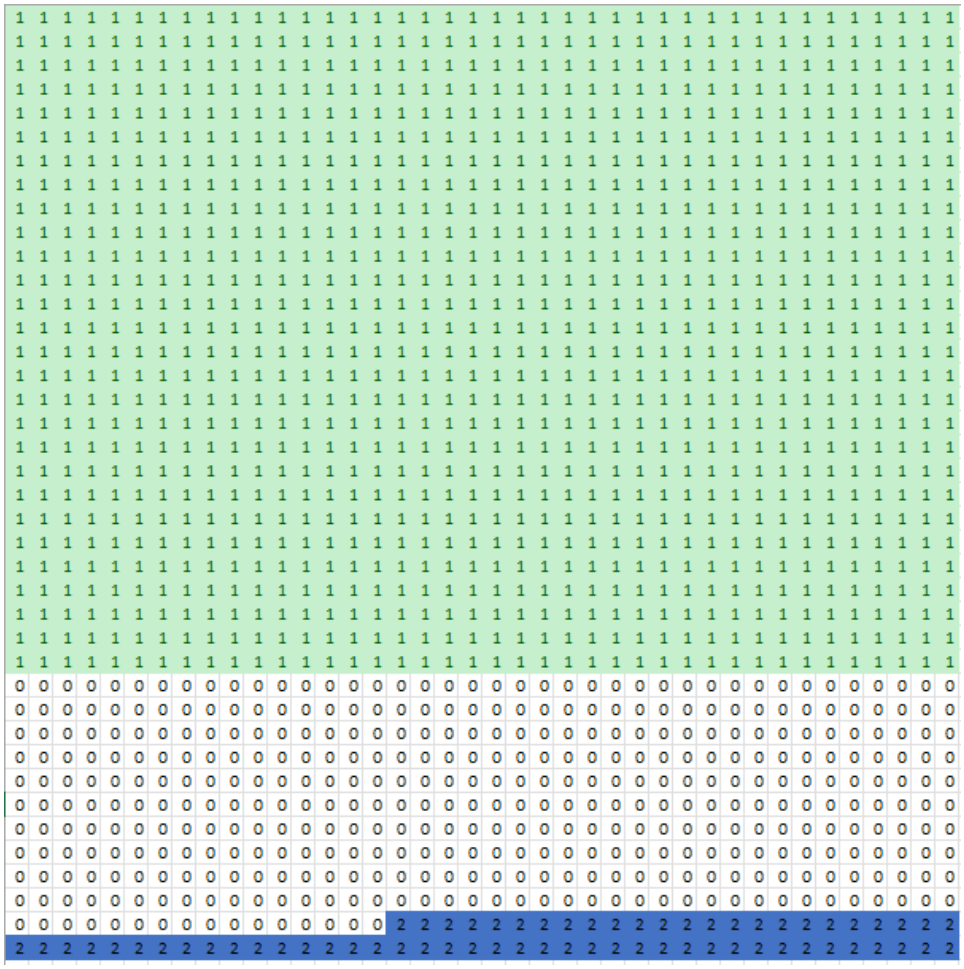

Figure 14: C=64,Q=1120,v=2,T=0.1,L=0.82

The model is then run with this initial layout until there are no actives

left. The jail sentence is set to be infinite, as usually when somebody is detained from a riot they don't get back into the riot. The other initial conditions we have are C=64, Q=1120, vision = 6 (police and non police), T=0.1, L=0.82

from this new setup we get the results in figure 18 where we have the blue line showing how the number of rioters starting out in a "riot" formation decreases with time. We also have a red line across the x axis showing the number of rioters for random placement which we did in our previous examples. We then have the purple line showing the number of people removed from the "riot" as well as the yellow line showing the same but for the random placement.

We can clearly see that when we place our agents in a riot formation they start of with a far greater number of actives due to a very low C/A ratio as there are no police whatsoever in the riot crowd. we then see that it takes far longer for the number of rioters in our riot to reach 0. We also see a far greater total number of non police agents going into jail when they are randomly distributed throughout the grid.

This is a great example of how with correct policing and spacial awareness can affect whether or not a riot will get out of hand.

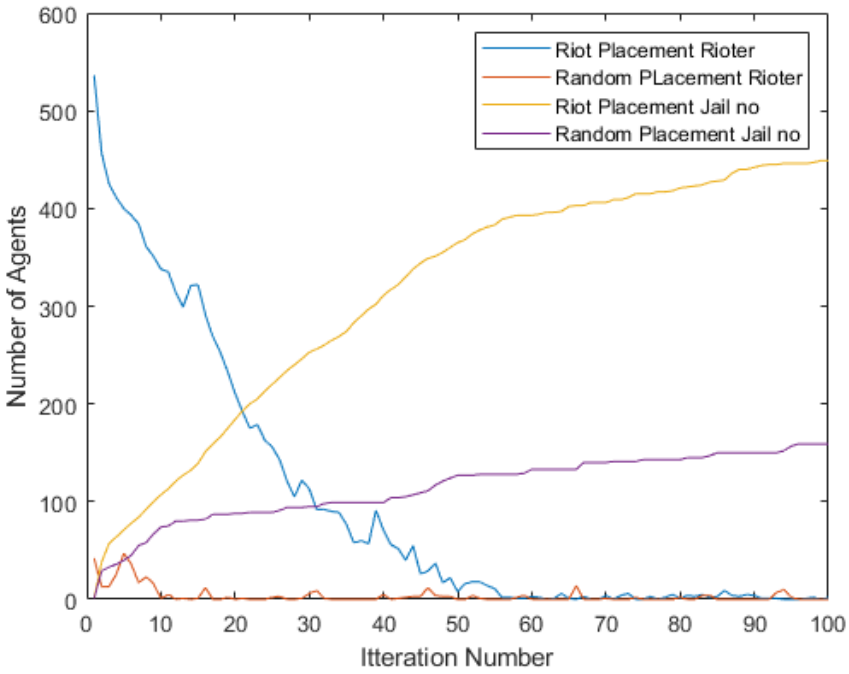

Figure 15: Riot Placement Vs Random Placement

5.6 The Effect of Smoke

Initial Conditions: C = 118, Q = 1120, v = 6(all agents), T = 0.1, L = 0.86, smoke size = 6 (7 × 7 grid), smoke duration = 40, jail length = infinite

First of all we simulate the exact same riot as we simulated before, except now we introduce our concept of smoke, from this we get figure 17

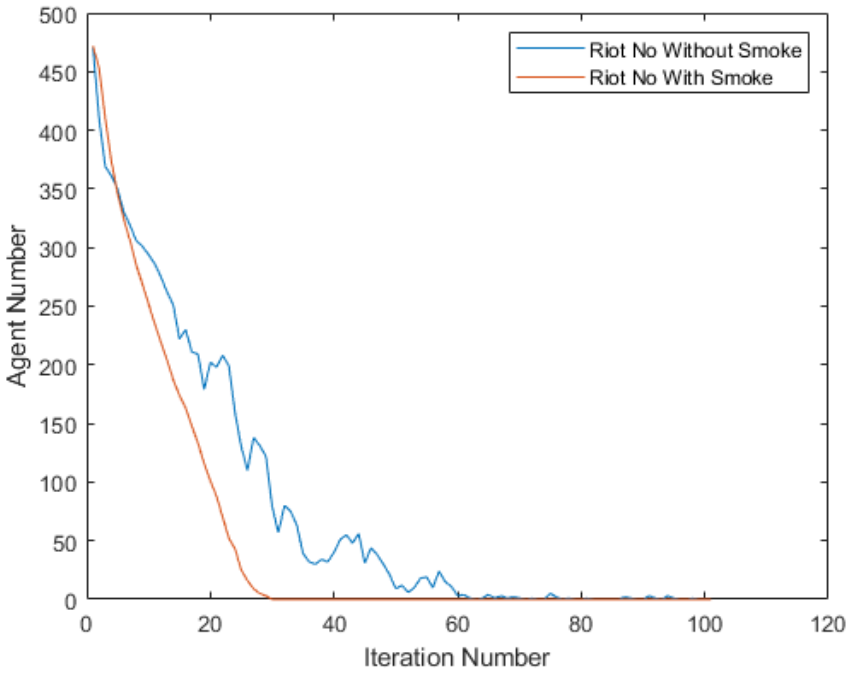

Figure 16: The Riot Layout

from the figure we see that with and without smoke both start at about the same number of initial rioters. However looking at further iterations shows how the use of smoke Helps subdue our riot and makes the riot come to a halt at about 30 iterations. Whilst on the other had the model without the smoke takes a total of about 60 iterations to come to a halt! this is a big jump in the length of time in which it takes for the riot to stop, just by introducing smoke.

5.7 Model Analysis

Despite this model being pretty basic, it still manages to produce very interesting results which can be compared historical real world events. We managed to observe individual deceptive behaviour in which agents pretend to not be rebellious whilst they wait for the best time to strike as well how legitimacy and the density of police effects A's

The main concepts that we observe in our model are described as "tipping points" by episteme (ref), these are moments in a regime when a large amount of people rebel and start a riot or even a revolution. We see a few such tipping points firstly from the snowball effect where we see that time is of the essence for a government to suppress a few active agents before more and more join them. We then see our second tipping point occur when there is a sudden drop in perceived legitimacy causing a large outburst in active agents rebelling against their government. Our final tipping point is found when we slowly decrease the amount of police of an oppressive scheme where agents are being deceptive the hole time until they see an opportunity when there are far fewer police about to stop them.

Along with these observations we then created a riot and saw how the effect of obscuring vision can affect how quickly a riot can be stopped!

6 Ethnic group model

Now I create the second model from the Epstein paper where we introduce two ethnic groups, a purple group and a green group. A few changes are made but the mathematical model is much the same for this, however now when one person from one group becomes active they will now take an agent from the opposite group off the board. More attributes are also added to our agents one being cloning, where each agent has a 1/20 chance to reproduce another agent of the same group and hardship in an empty space next to them. Age is also introduced where age is the number of iterations one agent can survive, we pick each agents maximum iteration they can reach from U(0,maximum age) where the maximum age is 200.

6.1 Peace and Harmony

initial conditions: C = 0, Q = 560 purple and 560 green, v = 2, T = 0.1, L = 0.9

Here we set legitimacy to a high L = 0.9. As such nobody really rebels and the population of each group multiplies filling the board as shown

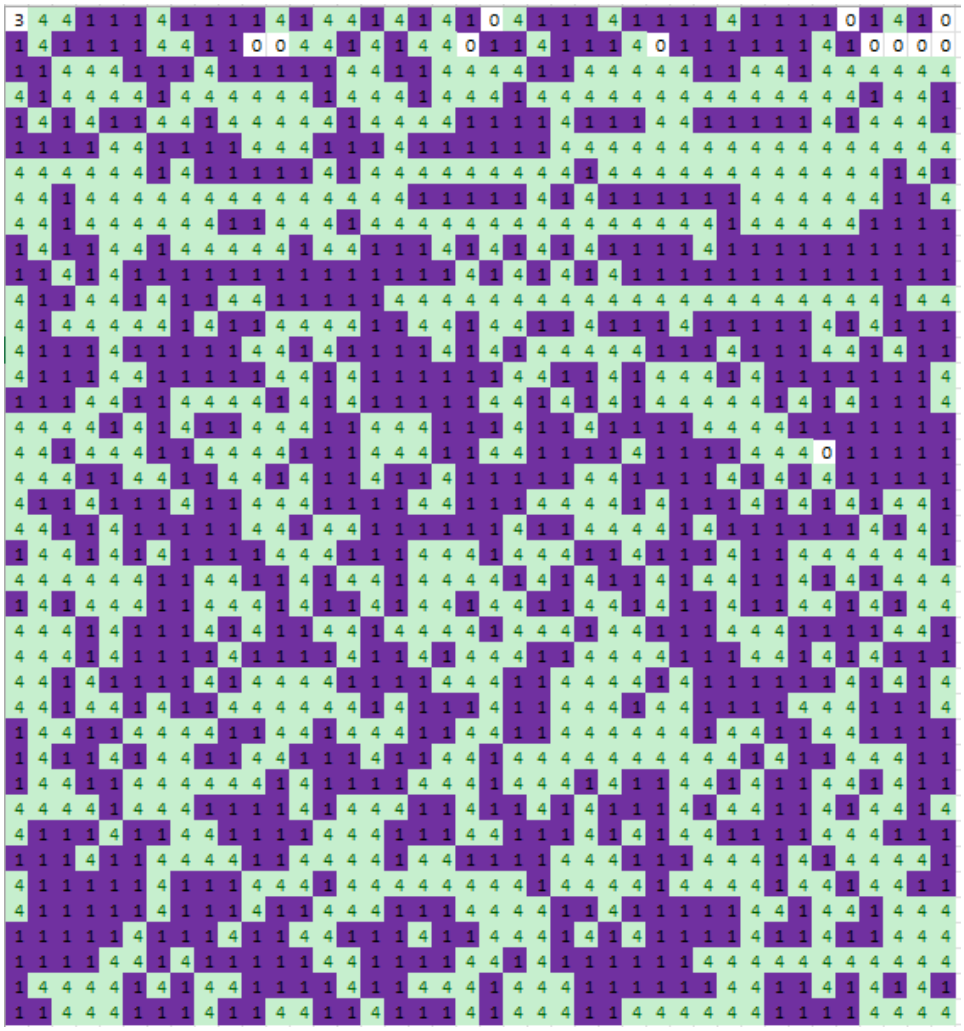

Figure 17:

6.2 Ethnic Cleansing

initial conditions: C = 0, Q = 560 purple and 560 green, v = 2, T = 0.1, L = 0.8

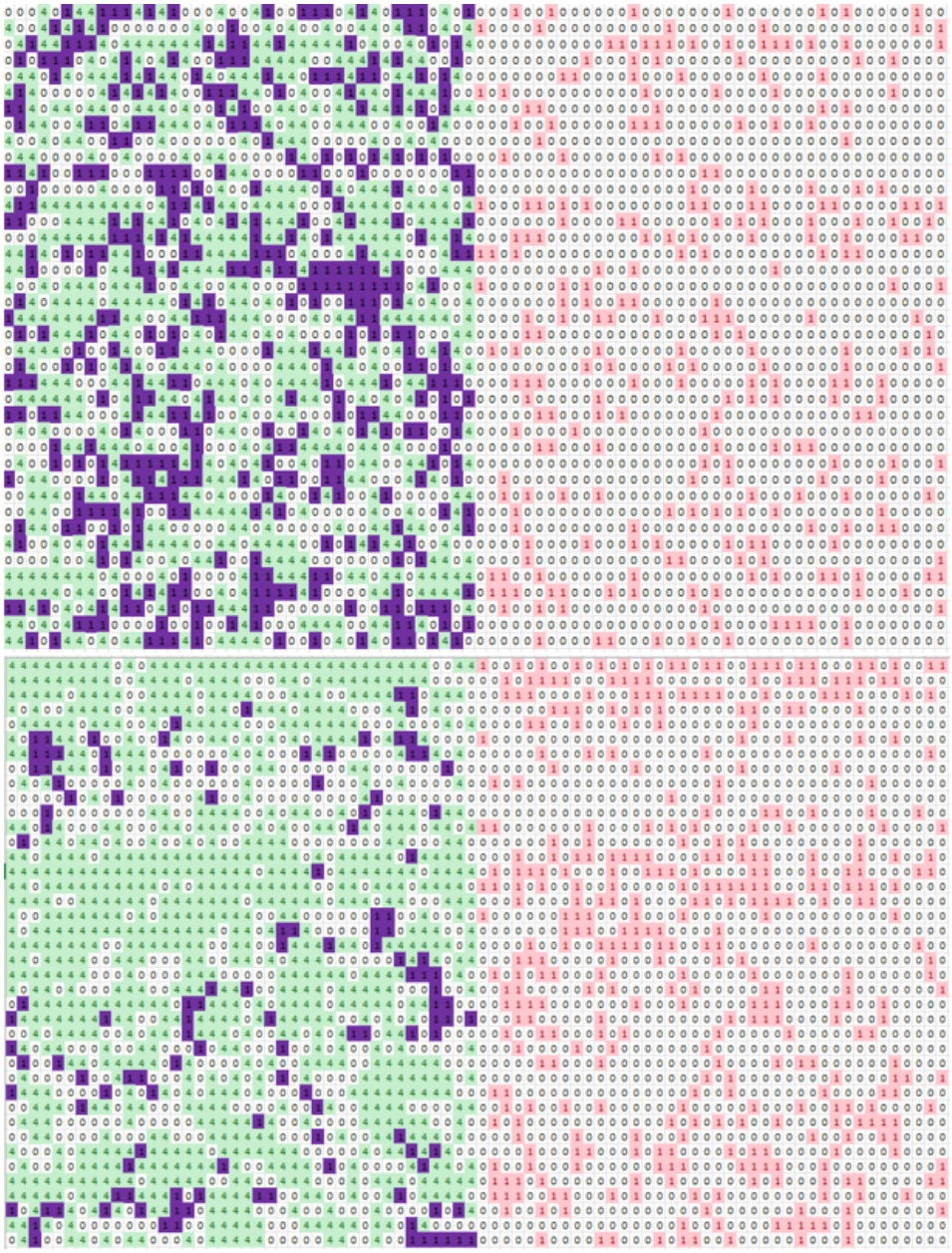

Figure 18:

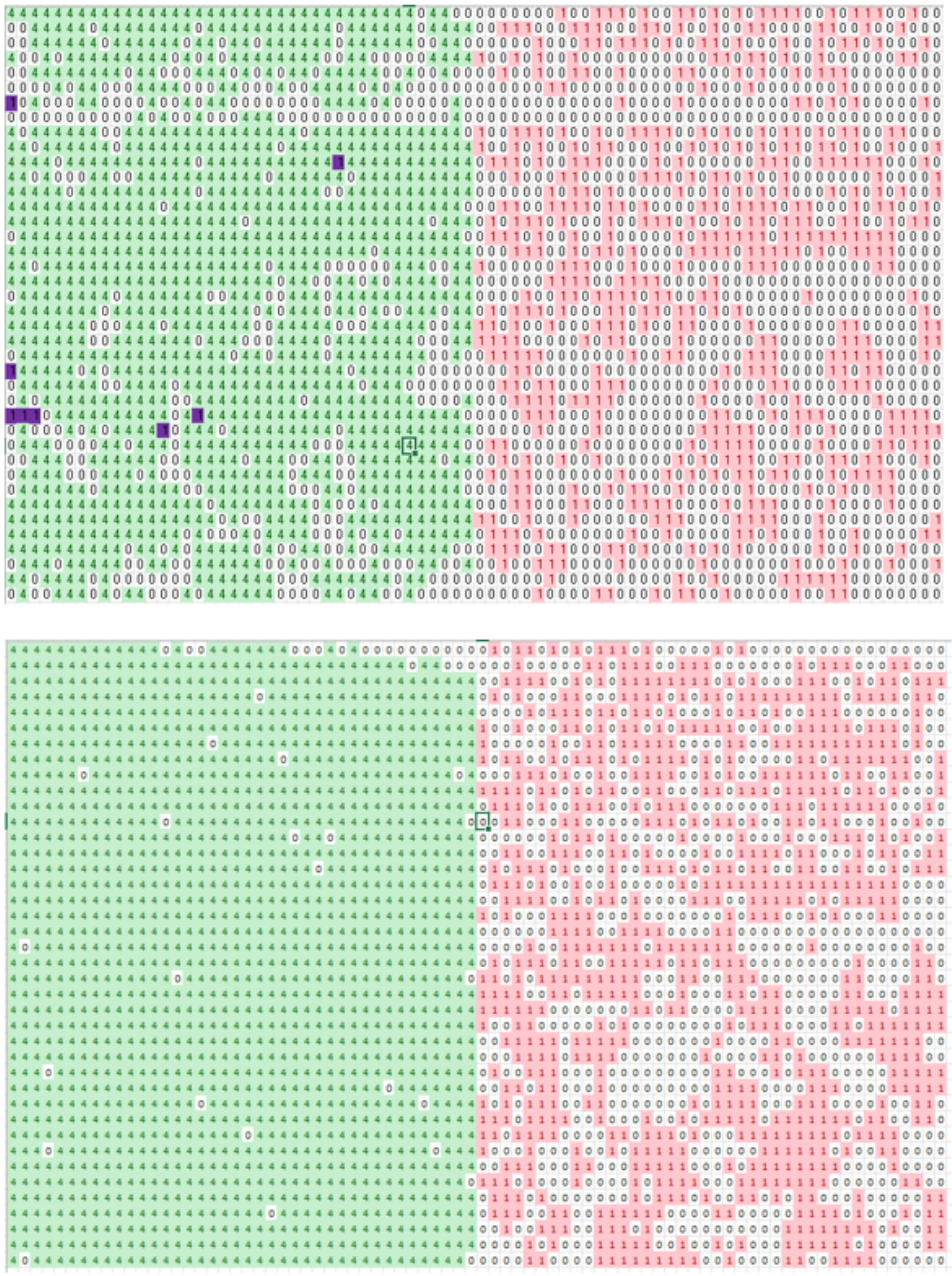

Figure 19:

from the above images we can see our green group from the model multiplying over 30 iterations and slowly removing all of the purple group. Over a long period of time (30 iterations) we always see one group completely get rid of the other group.

This can be compared to "competitive exclusion" which is the idea of two species being in a confined space both competing for resources and in the end one species may get rid of the other taking all the space for its own expansion and more resources. However when a predator is introduced to regulate the growth of the species then both can end up surviving for a longer period of time.

6.3 Effect of Police Density on Extinction Times

Police in my model can be seen as predators, they themselves can not be taken away by the prey (ethnic groups) however they themselves can take them off the board, even if it is a short period of time. This in turn helps keep the ethnic groups alive for longer than what they usually would be, However their coexistence is not peaceful. Both groups are still taking each other off the board and rioting.

In order to see the effect that police have on the length of time that the groups can exist together without wiping each other out I run the grid changing the density of police on the grid starting out with 0 P's on the board and moving up to 150 adding 3 each time. The model also runs 50 times at each P density. This gives figure 21

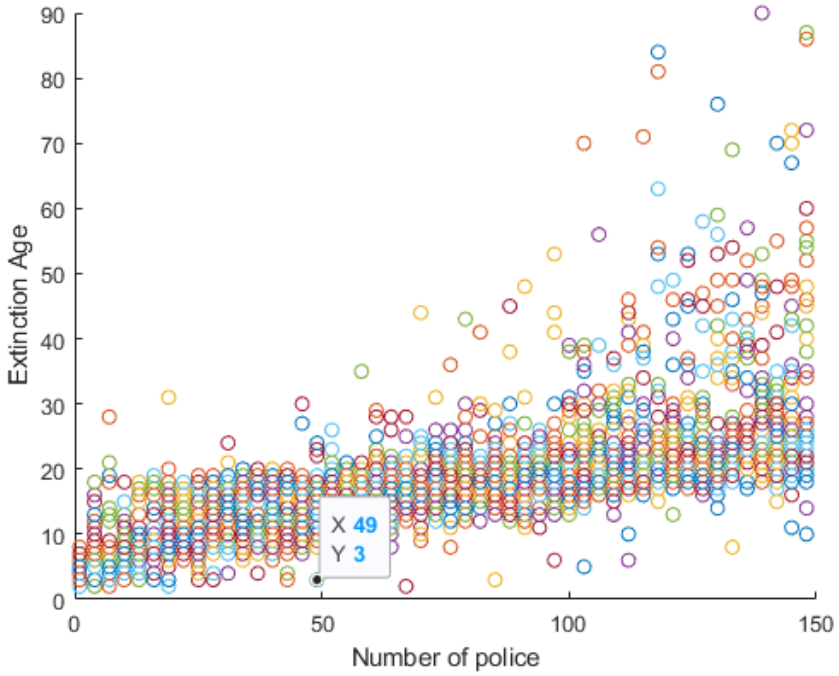

Figure 20:

We can see a trend in this graph, it is clear that as we increase the density of P's on the grid that the length of time our ethnic groups live for increases. Further more if we find the average extinction time for each density we get figure 22

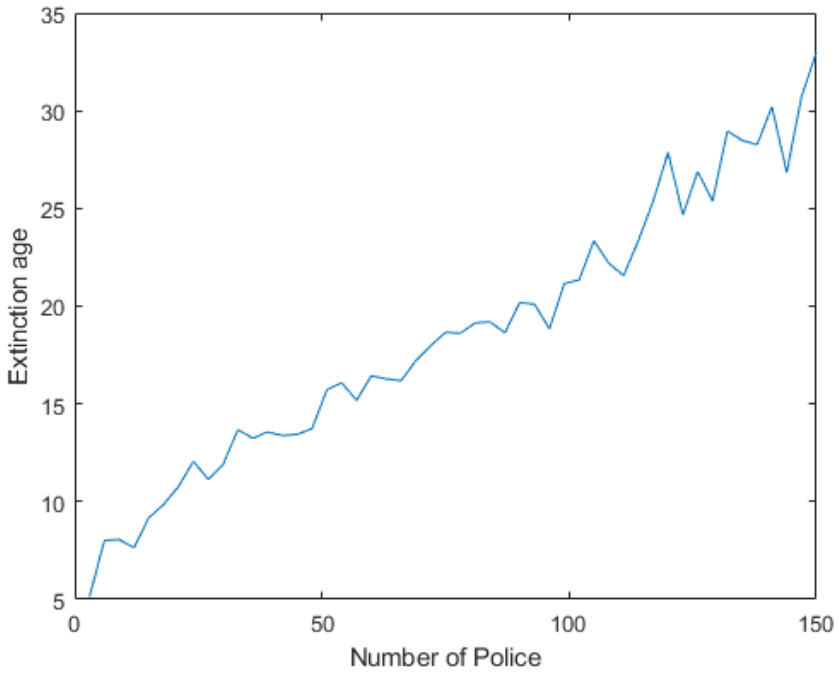

Figure 21:

This makes it very clear that there is a linear relationship between cop density and extinction time as we see the extinction age increasing from beginning to end.

7 Model 2 Evaluation

With a high legitimacy L=0.9 two ethnic groups live together peacefully in my model, but when this legitimacy is reduced to L = 0.8 we observe ethnic cleansing take place. The addition of police can help prolong the length of time two ethnic groups live together before one wipes the other out and

adding more police increases this time even more however it seemed that ethnic cleansing still took place even with very high cop density's

8 Conclusions

Agent based models may not be quite as complex as an actual riot or a revolution. However studying them can Help us understand what variables may effect a revolution and how smoke can be used to bring a riot to a quick end.

References

Joshua M Epstein. Modeling civil violence: An agent-based computational approach. *Proceedings of the National Academy of Sciences*, 99(suppl_3): 7243–7250, 2002.

A Tocqueville. The old regime and the french revolution (trans: Gilbert s) doubleday. *Garden City*, 1856.

9 Appendices (Matlab Code)

here is my code for my riot simulation

```
clc
clear all
%%creating paramaters
%
x=6;

% police_n = 64;

crowd_n = 1120;
police_n = 118;
L = 0.82;
T = 0.1;


age_max = 120;

grid_size = 40;
vision_l =6;
P_Vis = 6;
smoke_size = 6;
smoke_L = 40;
no=100   ;
```

```
23
24 %number of iterations
25 jail = {[9,9,9,-1,-1]};
26 jail_L=15 ;
27 %legitimacy of goverment
28
29
30 % T is a value which if G - N > T a active begins to
       riot (xpected utility
31 % of not expressing there private grievance)
32
33 JustMakeGraphXD = RiotW_Smoke_VS_Without(grid_size,L,
       vision_l,T,crowd_n, police_n, no,smoke_size,smoke_L
       ,P_Vis,jail,jail_L);
34 %% [group_green, group_blue, police] = create_ethnic(
       crowd_n,police_n,age_max);
35 %% grid = rand_initiallize_grid_ethnic(crowd_n,
       group_green, group_blue, police_n, police,
       grid_size);
36 %% grid = ethnic_agents(grid,grid_size,L,vision_l,T,
       P_Vis,age_max);
37 %% [jail,grid] = policemen(grid, grid_size,jail,
       jail_L,P_Vis);
38 %%
```

```
39 %% [jail,grid] = policemen(grid, grid_size,jail,
       jail_L,P_Vis);
40 %% [jail,grid] = policemen(grid, grid_size,jail,
       jail_L,P_Vis);
41 %%
42 %% jail_keys4 = count_jail_keys(jail,4)
43 %% jail_keys1 = count_jail_keys(jail,1)
44 %% one = count_keys(grid,1)
45 %% four = count_keys(grid,4)
46
47
48 % [group_green, group_blue, police] = create_ethnic(
       crowd_n,police_n,age_max);
49 % grid = rand_initiallize_grid_ethnic(crowd_n,
       group_green, group_blue, police_n, police,
       grid_size);
50 % grids = {};
51 % active_keys = {};
52 % gridkeys = {};
53 % purpleno = zeros(1,100);
54 % greenno = zeros(1,30);
55 %
56 % jailno = zeros(1,100);
57 %
58 %
```

```
% purpleno(1,1) = count_keys(grid,1);
% greenno(1,1) = count_keys(grid,4);
% policeno(1,1) = count_keys(grid,2);
%
%
% grid_keys(grid,grid_size);
% grids = {grid};
% for i = 1:300
%     i
% grid = ethnic_agents(grid,grid_size,L,vision_l,T,
    P_Vis,age_max);
% [jail,grid] = policemen(grid, grid_size,jail,jail_L,
    P_Vis);
% purpleno(1,i+1) = count_keys(grid,1);
% greenno(1,i+1) = count_keys(grid,4);
% policeno(1,i+1) = count_keys(grid,2);
% jailno(1,i+1) = length(jail);
%
% grids{end+1} = grid;
%
% end
%
% plot(purpleno,'DisplayName','purple no')
% hold on
% plot(greenno, 'DisplayName','green no')
```

```
% plot(jailno, 'DisplayName', 'jail no')
% xlabel('Itterations')
% ylabel('Number of Agents')
% hold off
%
% legend

% for i = 1:30
% active_keys{end+1} = activekeys(grids{i},grid_size);
%
% gridkeys{end+1} = grid_keys(grids{i},grid_size);
%
% end
%
% A = {}
% for i = 1:30
%     gridkeys{i}
%  A{end+1} = [gridkeys{i} active_keys{i}];
% end


% filename = 'ethnic_cleanse_keys.xlsx';
% writematrix(gridkeys,filename)
%
```

```
107 % filename2 = 'ethnic_cleanse_Actives.xlsx';
108 % writematrix(active_keys,filename2)
109
110
111
112
113
114 % [grids,agentno,jailno,riotno] = run_till_no_nosmoke(
      grid_size,L,vision_l,T,crowd_n, police_n, no,jail,
      jail_L,P_Vis,2);
115 % plot(riotno)
116 % plot(sriotno,'displayname','Rioters with smoke')
117 % hold on
118
119 % hold off
120 % legend
121 % sgrid = smoke_grid_init(grid_size);
122
123
124
125
126
127 %  smokeriotno = run_till_no_smoke(grid_size,L,
      vision_l,T,crowd_n, police_n,no,sgrid,smoke_size);
```

```
128 %  riotno = run_till_no_nosmoke(grid_size,L,vision_l,T
    ,crowd_n, police_n, no);
129 %  plot(1:length(riotno),riotno,1:length(smokeriotno),
    smokeriotno)
130
131
132
133
134 %%my functions
135 %{
136
137 run through a riot until all rioters have been arested
138 %}
139
140 function [grid,riotno,jailno] = effect_of_Police(
    grid_size,L,vision_l,T,crowd_n, police_n, no,
    smoke_size,smoke_L,P_Vis,jail)
141     jail_L=10000000;
142     %create the grid
143     [crowd,police] = create_crowd_police(crowd_n,
        police_n);
144     grid  = rand_initiallize_grid(crowd_n, crowd,
        police_n, police, grid_size);
145         riotno = zeros(no,1);
146         jailno = zeros(no,1);
```

```
147         policeno = zeros(no,1);
148         policeno(1,1) = count_keys(grid,2);
149         riotno(1,1) = 0;
150         jailno(1,1) = 0;
151         count = 2;
152     for i = 1:no
153         grid = agents_nosmoke(grid,grid_size,L,
                vision_l,T);
154         riotno(count,1)=count_keys(grid,3);
155
156         grid = removeC(grid,grid_size);
157         [jail,grid] = policemen(grid, grid_size,jail,
                jail_L,P_Vis);
158
159
160
161
162         policeno(count,1) = count_keys(grid,2);
163         jailno(count,1) = length(jail);
164         count = count +1;
165     end
166     plot(riotno)
167     hold on
168     plot(jailno)
169     plot(policeno)
```

```
170      legend({'Number Of Rioters','Number of Prisoners',
             'Number of Police'},'Location','east')
171      xlabel('Itteration Number')
172      ylabel('Number of Agents')
173      hold off
174  end
175
176
177
178  function [Ls,riotno,jailno] = effect_of_legitamcy(
         grid_size,L,vision_l,T,crowd_n, police_n, no,
         smoke_size,smoke_L,P_Vis,jail)
179      jail_L = 10000000000;
180      Ls = 0.90:-0.01:0.0;
181      %create the grid
182      [crowd,police] = create_crowd_police(crowd_n,
             police_n);
183      grid  = rand_initiallize_grid(crowd_n, crowd,
             police_n, police, grid_size);
184          riotno = zeros(length(Ls),1);
185          jailno = zeros(length(Ls),1);
186          riotno(1,1) = 0;
187          jailno(1,1) = 0;
188          count = 2;
189      for L = Ls
```

```
190             grid = agents_nosmoke(grid,grid_size,L,
                    vision_l,T);
191
192             [jail,grid] = policemen(grid, grid_size,jail,
                    jail_L,P_Vis);
193
194
195
196             riotno(count,1)=count_keys(grid,3);
197
198             jailno(count,1) = length(jail);
199             count = count +1;
200         end
201         plot(riotno)
202         hold on
203         plot(jailno)
204         plot(Ls*1000)
205         legend({'Number Of Rioters','Number of Prisoners',
                'Legitamacy'},'Location','north')
206         xlabel('Itteration Number')
207         ylabel('Number of Agents/Legitimacy*1000')
208         hold off
209     end
210
```

```
211 function [Ls,riotno,jailno] =
    sudden_effect_of_legitamcy(grid_size,L,vision_l,T,
    crowd_n, police_n, no,smoke_size,smoke_L,P_Vis,jail
    ,jail_L)
212     Ls =[0.9*ones(1,77),0.6*ones(1,42)];
213     %create the grid
214     [crowd,police] = create_crowd_police(crowd_n,
            police_n);
215     grid  = rand_initiallize_grid(crowd_n, crowd,
            police_n, police, grid_size);
216         riotno = zeros(length(Ls),1);
217         jailno = zeros(length(Ls),1);
218         riotno(1,1) = 0;
219         jailno(1,1) = 0;
220         count = 2;
221     for i = 1:77
222         L = 0.9;
223         %grid = agents_nosmoke(grid,grid_size,L,
                vision_l,T);
224
225         [jail,grid] = policemen(grid, grid_size,jail,
                jail_L,P_Vis);
226
227
228
```

```
229         riotno(count,1)=count_keys(grid,3);
230
231         jailno(count,1) = length(jail);
232         count = count +1;
233     end
234     for i = 78:120
235         L = 0.7;
236         grid = agents_nosmoke(grid,grid_size,L,
                vision_l,T);
237         riotno(count,1)=count_keys(grid,3);
238         [jail,grid] = policemen(grid, grid_size,jail,
                jail_L,P_Vis);
239
240
241
242
243
244         jailno(count,1) = length(jail);
245         count = count +1;
246
247     end
248     plot(riotno)
249     hold on
250     plot(jailno)
251     plot(Ls*1000)
```

```
252     legend({'Number Of Rioters','Number of Prisoners',
            'Legitamacy'},'Location','northeast')
253     xlabel('Itteration Number')
254     ylabel('Number of Agents/Legitimacy*1000')
255
256     hold off
257
258
259 end
260
261
262
263
264 function [sgrid,riotno] = run_till_no_smoke(grid_size,
        L,vision_l,T,crowd_n, police_n, no,smoke_size,
        smoke_L,P_Vis,jail,jail_L,type)
265     %create the grid
266     if type ==1
267         [crowd,police] = create_crowd_police(crowd_n,
                police_n);
268         grid = place_riot(crowd_n, crowd, police_n,
                police, grid_size);
269     end
270     if type ==2
```

```
271         [crowd,police] = create_crowd_police(crowd_n,
                police_n);
272         grid  = rand_initiallize_grid(crowd_n, crowd,
                police_n, police, grid_size);
273     end
274     %create the smoke grid
275     sgrid = smoke_grid_init(grid_size);
276     %make agents rioters
277     grid = agents_smoke(grid,grid_size,L,vision_l,T,
            sgrid);
278     riotno = zeros(no,1);
279     agentno = zeros(no,1);
280
281     count = 1;
282     riotno(count,1)=count_keys(grid,3);
283
284     for i = 1:no
285
286         count = count+1;
287
288
289         [jail, sgrid, grid] = policemen_smoke(grid,
                grid_size,smoke_size,smoke_L,sgrid,jail,
                jail_L,P_Vis);
290         sgrid
```

```
291             grid = agents_smoke(grid,grid_size,L,vision_l,
                    T,sgrid);
292
293             riotno(count,1)=count_keys(grid,3);
294             agentno(count-1,1) = count_keys(grid,1);
295
296
297
298         end
299         plot(riotno)
300 end
301
302
303
304 function riotno = run_till_no_nosmoke_smoke(grid_size,
        L,vision_l,T,crowd_n, police_n, no,smoke_size,jail,
        jail_L)
305         %create the grid
306         [crowd,police] = create_crowd_police(crowd_n,
                police_n);
307         grid  = rand_initiallize_grid(crowd_n, crowd,
                police_n, police, grid_size);
308         %create the smoke grid
309         sgrid = smoke_grid_init(grid_size);
310         %make agents rioters
```

```
311     grid = agents_smoke(grid,grid_size,L,vision_l,T,
            sgrid);
312     riotno = zeros(no,1);
313     agentno = zeros(100,1);
314
315     count = 1;
316     riotno(count,1)=0;
317
318     for i = 1:no/2
319
320         count = count+1;
321
322
323         grid = policemen(grid, grid_size,jail,jail_L);
324         grid = agents_nosmoke(grid,grid_size,L,
                vision_l,T);
325
326         riotno(count,1)=count_keys(grid,3);
327         agentno(count-1,1) = count_keys(grid,1);
328
329     end
330     for i = 1:no/2
331
332         count = count+1;
333
```

```
334
335             [jail, sgrid, grid] = policemen_smoke(grid,
                    grid_size,smoke_size,smoke_L,sgrid,jail,
                    jail_L,P_Vis);
336
337             grid = agents_smoke(grid,grid_size,L,vision_l,
                    T,sgrid);
338
339             riotno(count,1)=count_keys(grid,3);
340             agentno(count-1,1) = count_keys(grid,1);
341
342
343
344         end
345
346 end
347
348
349
350 function [grids,agentno,jailno,riotno] =
        run_till_no_nosmoke(grid_size,L,vision_l,T,crowd_n,
         police_n, no,jail,jail_L,P_Vis,type)
351         if type ==1
352             [crowd,police] = create_crowd_police(crowd_n,
                    police_n);
```

```
353         grid = place_riot(crowd_n, crowd, police_n,
                police, grid_size);
354     end
355     if type ==2
356         [crowd,police] = create_crowd_police(crowd_n,
                police_n);
357         grid  = rand_initiallize_grid(crowd_n, crowd,
                police_n, police, grid_size);
358     end
359
360     %grid = agents_nosmoke(grid,grid_size,L,vision_l,T
            );
361     riotno = zeros(no,1);
362     agentno = zeros(no,1);
363     jailno = zeros(no,1);
364     count = 1;
365     grids = {grid};
366     for i = 1:no
367
368
369         [jail,grid] = policemen(grid, grid_size,jail,
                jail_L,P_Vis);
370
371         grid = agents_nosmoke(grid,grid_size,L,
                vision_l,T);
```

```
372
373
374             riotno(count,1)=count_keys(grid,3);
375             agentno(count,1) = count_keys(grid,1);
376             jailno(count,1) = length(jail);
377             grids{end+1} = grid;
378             count = count+1;
379
380 %             numciv = count_keys(grid,1)
381 %             numriot = count_keys(grid,3)
382 %             total = numciv+numriot
383
384         end
385         plot(jailno)
386         plot(riotno)
387 end
388
389 function plot = run_ethnic(crowd_n, group_green,
        group_blue, police_n, police, grid_size,age_max,L,
        vision_l,T,P_Vis,jail,jail_L)
390         [group_green, group_blue, police] = create_ethnic(
            crowd_n,police_n,age_max);
391         grid = rand_initiallize_grid_ethnic(crowd_n,
            group_green, group_blue, police_n, police,
            grid_size);
```

```
392     grids = {};
393     active_keys = {};
394     gridkeys = {};
395     purpleno = zeros(1,100);
396     greenno = zeros(1,30);
397
398     jailno = zeros(1,100);
399
400
401     purpleno(1,1) = count_keys(grid,1);
402     greenno(1,1) = count_keys(grid,4);
403     policeno(1,1) = count_keys(grid,2);
404
405
406     grid_keys(grid,grid_size);
407     grids = {grid};
408     for i = 1:300
409         i
410     grid = ethnic_agents(grid,grid_size,L,vision_l,T,
            P_Vis,age_max);
411     [jail,grid] = policemen(grid, grid_size,jail,
            jail_L,P_Vis);
412     purpleno(1,i+1) = count_keys(grid,1);
413     greenno(1,i+1) = count_keys(grid,4);
414     policeno(1,i+1) = count_keys(grid,2);
```

```
415         jailno(1,i+1) = length(jail);
416
417         grids{end+1} = grid;
418
419         end
420
421         plot(purpleno,'DisplayName','purple no')
422         hold on
423         plot(greenno, 'DisplayName','green no')
424         plot(jailno, 'DisplayName', 'jail no')
425         xlabel('Itterations')
426         ylabel('Number of Agents')
427         hold off
428
429         legend
430 end
431
432
433
434 function extinction_age = run_until_extinction(crowd_n
        ,police_n,age_max,grid_size,L,vision_l,T,P_Vis,jail
        ,jail_L)
435         p=0;
436
```

```
437       [group_green, group_blue, police] = create_ethnic(
              crowd_n,police_n,age_max);
438       grid = rand_initiallize_grid_ethnic(crowd_n,
              group_green, group_blue, police_n, police,
              grid_size);
439       for i = 1:15000
440           grid = ethnic_agents(grid,grid_size,L,vision_l
                  ,T,P_Vis,age_max);
441           [jail,grid] = policemen(grid, grid_size,jail,
                  jail_L,P_Vis);
442           if or(and(count_keys(grid,1) ==0,
                  count_jail_keys(jail,1)==0), and(count_keys
                  (grid,4)==0,count_jail_keys(jail,4) == 0))
443               extinction_age = i;
444               p=1;
445               break
446           end
447       end
448       if p==0
449       extinction_age = 15000
450       end
451   end
452
453   function extinction_age = affect_police_ethnic(crowd_n
          ,age_max,grid_size,L,vision_l,T,P_Vis,jail,jail_L)
```

```
454         police_n = 0:3:150;
455         extinction_age = zeros(50,50);
456         run_until_extinction(crowd_n,0,age_max,
                grid_size,L,vision_l,T,P_Vis,jail,jail_L);
457
458         for j = police_n
459             for i = 1:50
460                 j
461                 i
462                 extinction_age(i,(j/3)+1) =
                        run_until_extinction(crowd_n,i,
                        age_max,grid_size,L,vision_l,T,
                        P_Vis,jail,jail_L);
463                 extinction_age(i,(j/3)+1)
464             end
465         end
466 end
467
468 function jorwi= random_place_riot_place(grid_size,L,
        vision_l,T,crowd_n, police_n, no,jail,jail_L,P_Vis)
469         %riot
470         jail_L = 10000;
471         [grids,agentno,jailno,riotno]=
                run_till_no_nosmoke(grid_size,L,vision_l,T,
                crowd_n, police_n, no,jail,jail_L,P_Vis,1)
```

```
472             %rand
473             [grids,ragentno,rjailno,rriotno]=
                    run_till_no_nosmoke(grid_size,L,vision_l,T,
                    crowd_n, police_n, no,jail,jail_L,P_Vis,2)
474             plot(riotno,'DisplayName','Riot Placement
                    Rioter')
475             hold on
476             plot(rriotno,'DisplayName','Random PLacement
                    Rioter')
477             plot(jailno,'DisplayName','Riot Placement Jail
                     no')
478             plot(rjailno,'DisplayName','Random Placement
                    Jail no')
479             ylabel('Number of Agents')
480             xlabel('Itteration Number')
481             hold off
482             legend
483 end
484
485 function JustMakeGraphXD = RiotW_Smoke_VS_Without(
        grid_size,L,vision_l,T,crowd_n, police_n, no,
        smoke_size,smoke_L,P_Vis,jail,jail_L)
486      jail_L = 9999999;
487             [grids,agentno,jailno,riotno] =
                    run_till_no_nosmoke(grid_size,L,vision_l,T,
```

```
                crowd_n, police_n, no, jail, jail_L, P_Vis, 1);
488             [sgrid, Priotno] = run_till_no_smoke(grid_size
                    , L, vision_l, T, crowd_n, police_n, no,
                    smoke_size, smoke_L, P_Vis, jail, jail_L, 1);
489             plot(riotno, 'DisplayName', 'Riot No Without
                    Smoke')
490             hold on
491             plot(Priotno, 'DisplayName', 'Riot No With Smoke
                    ')
492             hold off
493             legend
494             JustMakeGraphXD = 0;
495 end
496 %% agents
497 function grid = agents_smoke(grid, grid_size, L, vision_l
        , T, sgrid)
498     grid = move_rand(1, grid_size, grid);
499     grid = move_rand(3, grid_size, grid);
500     grid = agent_to_riot_smoke(grid, grid_size, L,
            vision_l, T, sgrid);
501 end
502
503 function grid = agents_nosmoke(grid, grid_size, L,
        vision_l, T)
504     grid = move_rand(1, grid_size, grid);
```

```
505       grid = move_rand(3,grid_size, grid);
506       grid = agent_to_riot_nosmoke(grid,grid_size,L,
              vision_l,T);
507   end
508
509
510   function [jail, sgrid, grid] = policemen_smoke(grid,
          grid_size,smoke_size,smoke_L,sgrid,jail,jail_L,
          P_Vis)
511       grid = move_rand(2, grid_size, grid);
512       sgrid = smoke(grid,grid_size,smoke_size,sgrid,
              smoke_L);
513       sgrid = updatesmoke(sgrid);
514       [jail,grid] = riot_to_arrest(grid_size, grid,jail,
              jail_L,P_Vis);
515       [jail,grid] = update_jail(jail,grid,grid_size);
516   end
517
518
519
520   function [jail,grid] = policemen(grid, grid_size,jail,
          jail_L,P_Vis)
521       grid = move_rand(2, grid_size, grid);
522       [jail,grid] = update_jail(jail,grid,grid_size);
```

```
523         [jail,grid] = riot_to_arrest(grid_size, grid,jail,
                jail_L,P_Vis);
524
525
526 end
527
528
529 function grid = ethnic_agents(grid,grid_size,L,
        vision_l,T,P_Vis,age_max)
530         grid = move_rand(1, grid_size, grid);
531         grid = move_rand(4,grid_size, grid);
532         grid = ethnic_group_to_riot(grid,grid_size,L,
                vision_l,T);
533         grid = kill(grid,grid_size,P_Vis);
534         grid = cloneing(grid_size,grid,age_max);
535         grid = ageing(grid,grid_size);
536 end
537
538
539 %% model 1
540
541 %{
542 grievance():
543         -Hardship G (physical or economic deprivation)
                drawn from U(0,1) ]
```

```
544       uniform distrobution on 0,1
545
546       -Legitamacy L percived legitamacy of a regime.
              This will be arbritary
547       number from 0,1
548   %}
549   function G  = grievance(H,L)
550       a = 1-L;
551       G = H*a;
552   end
553
554   %{
555   arrest_probability():
556       -constant k set such that P = 0.9 when C=A=1
557
558       -active rioters A
559
560       -cops in the area C
561   C/A changes for every rioter depending on how many C's
          and A's are within
562   the vision of each crowd member within a set vision
563   %}
564   function P = arrest_probability(k,C,A)
565       P = 1 - exp(-(k*(C/A)));
566   end
```

```
567
568
569 %{
570 create_crowd_police()
571 -crowd_n:   number of people in crowd
572 -police_n:  number of police
573
574 -crowd:     our crowd with h r vals and key 1
575 -police:    our police with h r vals and key 2
576
577 creates the crowd aswell as the police such that:
578 -in the crowd each member has a number 1 in row1 to
      represent that they
579 are non rioters aswell as a hardship (H) row2 and a
      level of
580 risk they are willing to take (R) in row3. form:
581 [0,h1,r1; 0,h2,r2; ...; 0,hn,rn]
582
583 -in the police we have our key as 2 instead of 1 and
      we have H=R=0 as all
584 the police do is arrest and have no R H
585 %}
586 function [crowd,police] = create_crowd_police(crowd_n,
      police_n)
```

```
587     %take our H and R randomly from the uniform
            distrobution beween 0-1
588     H = unifrnd(0,1,1,crowd_n)';
589     R = unifrnd(0,1,1,crowd_n)';
590     %make a list of 1's from each person in the crowd
591     crowdkey = ones(1,crowd_n)';
592     %put crowd in the form
593     crowd = [crowdkey,H, R, zeros(1,crowd_n)'];
594     %same but police
595     police = [2*ones(1,police_n)',zeros(1,police_n)',
            zeros(1,police_n)',zeros(1,police_n)'];
596 end
597
598 %{
599  rand_initiallize_grid()
600 -crowd_n:    number of people in crowd
601 -crowd:      from create_crowd_police()
602 -ploice_n:   number of police
603 -police:     from create_crowd_police
604 -grid_size: how big we want our grid
605
606 -grid:       our initialized grid
607 %}
608
```

```
609 function grid  = rand_initiallize_grid(crowd_n, crowd,
        police_n, police, grid_size)
610
611     %first start our grid out as being a grid of zeros
            for our given grid
612     %size
613     for i = 1:grid_size
614         for j = 1:grid_size
615             grid{i,j} = [0,0,0,0];
616         end
617     end
618
619
620
621 %     now we put in our crowd, distrobuted randomly
622 %     throughout the grid
623     while crowd_n ~= 0
624
625         i = randi(grid_size,1);
626         j = randi(grid_size,1);
627         if grid{i,j}(1) == 0
628             grid{i,j}=crowd(crowd_n,1:4);
629             crowd_n = crowd_n - 1;
630         end
631     end
```

```
632
633
634     %now we put the desired number of plolice randomly
            into our grid
635     while police_n ~=0
636         i = randi(grid_size,1);
637         j = randi(grid_size,1);
638         if grid{i,j}(1) == 0
639             grid{i,j}=police(police_n,1:4);
640             police_n = police_n - 1;
641         end
642     end
643 end
644
645 %{
646 grid:       the grid
647 grid_size:  size of the grid
648 vision_l:   length of the vision
649
650 v:          vision at point i j
651
652 takes in the grid and a point of the grid, it then
      returns a subsection
653 ofthe grid around this point
654 %}
```

```
655 function v = vision(i,j,grid, vision_l,grid_size)
656
657     w = i - vision_l;
658     e = i + vision_l;
659     s = j + vision_l;
660     n = j - vision_l;
661     if w < 1
662         w = 1;
663     end
664
665     if e > grid_size
666         e = grid_size;
667     end
668
669     if s >grid_size
670         s = grid_size;
671     end
672
673     if n < 1
674         n = 1;
675     end
676     dns = s-n+1;
677     dew = e-w+1;
678     v = grid(w:e,n:s);
679
```

```
680      %v = reshape({grid(w:e,n:s)},[dew,dns]);
681
682  end
683
684
685
686  %{
687
688
689  takes in the grid and checks the number of rioters
        around each agent with
690  the number of police and uses equations to evaluate if
         the agent becomes a
691  rioter aswell
692  %}
693
694
695  function grid = agent_to_riot_nosmoke(grid,grid_size,L
        ,vision_l,T)
696      %loop through each space in the grid
697      for i = 1:grid_size
698      for j = 1:grid_size
699          %if we land on an agent
700          if grid{i,j}(1)==1 || grid{i,j}(1) == 3
701              %find its grievance number
```

```
702             G  = grievance(grid{i,j}(2),L);
703
704             %find its risk probablility
705             R  = grid{i,j}(3);
706
707             %find the vision at that point
708             v = vision(i,j, grid, vision_l, grid_size)
                    ;
709
710             %count number of police in vision
711             C=count_keys(v,2);
712
713             %count number of actives(rioters) in
                    vision
714             A=count_keys(v,3)+1;
715
716             %with our C and A we can now find the
                    probability for our guy
717             %at i j to be arrested
718             P = arrest_probability(2.3,C,A);
719
720             %our agents net risk
721             N = R*P;
722
```

```
723                 %expected utility of publicly expressing
                        ones private grievanc
724                 gmn = G-N;
725
726                 if gmn >T
727                     grid{i,j}(1) = 3;
728                 else
729                     grid{i,j}(1) = 1;
730                 end
731
732             end
733
734         end
735         end
736         %grid;
737 end
738
739
740
741
742
743 %{
744
745 turns agents into rioters taken into acount the smoke
746 %}
```

```
747 function grid = agent_to_riot_smoke(grid,grid_size,L,
        vision_l,T,sgrid)
748     %loop through each space in the grid
749     for i = 1:grid_size
750     for j = 1:grid_size
751         %if we land on an agent
752         if grid{i,j}(1)==1
753             %find its grievance number
754             G  = grievance(grid{i,j}(2),L);
755
756             %find its risk probablility
757             R  = grid{i,j}(3);
758
759             %find the vision at that point
760             v = vision(i,j, grid, vision_l, grid_size)
                    ;
761
762             %count number of police in vision
763             C=count_keys(v,2);
764
765             %count number of actives(rioters) in
                    vision excluding those
766             %with smoke
767             A=count_keys_smoke(v,3,sgrid)+1;
768
```

```
769                 %with our C- and A we can now find the
                        probability for our guy
770                 %at i j to be arrested
771                 P = arrest_probability(-log(0.1),C,A);
772
773                 %our agents net risk
774                 N = R*P;
775
776                 %expected utility of publicly expressing
                        ones private grievanc
777                 gmn = G-N;
778
779                 if gmn >T
780                     grid{i,j}(1) = 3;
781                 else
782                     grid{i,j}(1) = 1;
783                 end
784
785             end
786
787         end
788         end
789         %grid;
790 end
791
```

```
792
793
794
795
796
797
798 %{
799 grid:    the matrix you want to count the keys in
800 key:     the key you want to count
801
802 n:       the count
803
804 counts the number of a given key in a given matrix
805 %}
806 function n = count_keys(grid,key)
807     n=0;
808     [r,c] = size(grid);
809     for x = 1:r
810     for y = 1:c
811         grid{x,y};
812         if grid{x,y}(1)==key
813             n=n+1;
814         end
815
816     end
```

```
817         end
818     end
819
820     %{
821
822     count keys in matrix excluding the ones with smoke on
        them
823     %}
824
825
826     function n = count_keys_smoke(grid,key,sgrid)
827         n=0;
828         [r,c] = size(grid);
829         for x = 1:r
830         for y = 1:c
831
832             if grid{x,y}(1)==key && sgrid{x,y}(1) == 0
833                 n=n+1;
834             end
835
836         end
837         end
838     end
839
840     %{
```

```
841 grid_size:  size of grid
842 grid:       current grid
843
844 grid:       updated grid
845
846 looks at every police person and arrestsa rioter if it
        is near by
847 %}
848
849 function [jail,grid] = riot_to_arrest(grid_size, grid,
      jail,jail_L,P_Vis)
850     %for every (i,j)th place on the grid
851
852     for i = 1:grid_size
853         for j = 1:grid_size
854             %if the (i,j)th place is a policeman
855             one = 0 ;
856
857                 if grid{i,j}(1) == 2
858                 %check every square 3*3 around the
                        police man
859
860
861                         for n = -P_Vis:P_Vis
862                             for m = -P_Vis:P_Vis
```

```
863                 %assighn new (i,j)th
                        position we are
                        considering
864                 positioni = i + n;
865                 positionj = j + m;
866                 %check point is on
                        grid and it is a
                        rioter
867                 if positioni<=
                        grid_size &&
                        positionj <=
                        grid_size &&
                        positioni > 0 &&
                        positionj > 0 &&
                        grid{positioni,
                        positionj}(1) == 3
868                     %arrest that
                            rioter
869
870
871                 % grid{positioni,
                        positionj}(1)=1;
872                   l = grid{
                          positioni,
                          positionj};
```

```
873                                 sentence = randi(
                                        jail_L);
874                                 l(4)=sentence;
875
876
877                                 jail{end+1} = l;
878
879                                 grid{positioni,
                                        positionj} =
                                        [0,0,0,0];
880                                 %return so we only
                                        arest 1 person
                                        per turn
881                                 one =1;
882                                 break
883
884                             end
885
886                         end
887                         if one ==1
888                             break
889                         end
890
891                 end
892
```

```
893                 end
894             end
895         end
896
897
898
899 end
900
901
902
903
904
905
906
907
908 %{
909 crowd_trype:     the type of crowd you want to move
910 grid_size:       the size of the grid
911 grid:            the current grid
912
913 grid:            the grid after all the agents of a
        certain type have moved
914 %}
915
916 function grid = move_rand(crowd_type, grid_size, grid)
```

```
917         v = 0;
918         %for every (i,j)th place on the grid
919         for i = 1:grid_size
920             for j = 1:grid_size
921                 %while we reach a place with the correct
                        crowd type, and we
922                 %have made a certain number of attempts v
923                 while grid{i,j}(1) == crowd_type && v <
                        1000
924                     %create a random number -1 to 1 for n-
                            s plane and w-e plane
925                     %and take the absolute value to get
                            rid of negatives
926                     directionns = i + randi([-5,5],1);
927                     directionwe = j + randi([-5,5],1);
928                     v = v + 1;
929                     %check if the new position is inside
                            of the grid and not =0
930                     if directionns<=grid_size &&
                            directionwe <= grid_size &&
                            directionns > 0 && directionwe > 0
931                         %if that space is empty (=0)
932                         if grid{directionns,directionwe
                                }(1) == 0
```

```
933                             %move that person to that
                                    location
934                             grid{directionns,directionwe}
                                    = grid{i,j};
935                             %remove that person from where
                                     he was previously
936                             grid{i,j} =  [0,0,0,0,0];
937                         end
938                     end
939                 end
940             end
941         end
942
943 end
944
945
946
947 %%smoke section
948
949 function sgrid = smoke(grid,grid_size,smoke_size,sgrid
        ,smoke_L)
950     %go through each square on grid
951     for i = 1:grid_size
952         for j = 1:grid_size
953
```

```
954
955             if grid{i,j}(1) == 2
956                 %throw smoke at fools
957                 sgrid = throwsmoke(i,j,grid,smoke_size,
                        grid_size,sgrid,smoke_L);
958             end
959         end
960     end
961 end
962
963
964
965
966 function sgrid = throwsmoke(i,j,grid,smoke_size,
        grid_size,sgrid,smoke_L)
967     L = length(smoke_size);
968     %search around the police officer for rioters
969     for n = -L:L
970         for m = -L:L
971             %assighn new (i,j)th position we are
                    considering
972             positioni = i + n;
973             positionj = j + m;
974
975
```

```
976             %check point is on grid and it is a rioter
977             if positioni<=grid_size && positionj <=
                    grid_size && positioni > 0 && positionj
                    > 0 && grid{positioni,positionj}(1) ==
                    3
978                 %create an square area of length
                        vision_l around our rioter
979                 w = positioni - smoke_size;
980                 e = positioni + smoke_size;
981                 s = positionj + smoke_size;
982                 north = positionj - smoke_size;
983                 %make each point inside of the grid
984                 if w < 1
985                     w = 1;
986                 end
987
988                 if e > grid_size
989                     e = grid_size;
990                 end
991
992                 if s >grid_size
993                     s = grid_size;
994                 end
995
996                 if north < 1
```

```
997                         north = 1;
998                     end
999
1000                    for k = w:e
1001                        for o = north:s
1002
1003                            sgrid{k,o}(1) = 1;
1004                            sgrid{k,o}(2) = smoke_L;
1005                        end
1006                    end
1007
1008
1009                end
1010            end
1011        end
1012    end
1013
1014
1015
1016
1017    function sgrid = smoke_grid_init(grid_size)
1018        sgrid = {};
1019        for i = 1:grid_size
1020            for j = 1:grid_size
1021                sgrid{i,j} = [0,0];
```

```
1022            end
1023        end
1024    end
1025
1026
1027
1028
1029    function sgrid = updatesmoke(sgrid)
1030        N = length(sgrid);
1031        for i = 1:N
1032            for j = 1:N
1033                if sgrid{i,j}(2)~=0
1034                    sgrid{i,j}(2) = sgrid{i,j}(2) -1;
1035                end
1036                if sgrid{j,i}(2) ==0 && sgrid{j,i}(1)==1
1037                    sgrid{j,i}(1)=0;
1038                end
1039            end
1040        end
1041    end
1042
1043
1044    function grid = removeC(grid,grid_size)
1045        n = 0;
1046
```

```
1047            for i = 1:grid_size
1048                for j = 1:grid_size
1049                    if grid{i,j}(1) == 2
1050                        grid{i,j} = [0,0,0,0];
1051                        n = n+1;
1052                        if n == 11
1053                            return
1054                        end
1055                    end
1056
1057                end
1058
1059            end
1060
1061
1062    end
1063
1064    function grid = place_riot(crowd_n, crowd, police_n,
        police, grid_size,grid)
1065        for i = 1:grid_size
1066            for j = 1:grid_size
1067                grid{i,j} = [0,0,0,0];
1068            end
1069        end
1070
```

```
1071            for i = 1:grid_size
1072                for j = 1:grid_size
1073                    grid{i,j} = crowd(crowd_n,1:4);
1074                    if crowd_n == 1
1075                        grid = place_riotsquad(police_n,
                                police, grid_size,grid);
1076                        return
1077                    end
1078                    crowd_n = crowd_n -1;
1079                end
1080            end
1081
1082    end
1083
1084
1085    function grid = place_riotsquad(police_n, police,
        grid_size,grid)
1086         for i = grid_size:-1:1
1087                for j = grid_size:-1:1
1088                    grid{i,j} = police(police_n,1:4);
1089                    if police_n == 1
1090                        return
1091                    end
1092                    police_n = police_n -1;
1093                end
```

```
1094        end
1095    end
1096
1097
1098
1099
1100
1101    function gridkeys = grid_keys(grid,grid_size)
1102        gridkeys = zeros(grid_size,grid_size);
1103        for i = 1:grid_size
1104            for j = 1:grid_size
1105                gridkeys(i,j) = grid{i,j}(1);
1106            end
1107        end
1108    end
1109
1110    function activekeys = activekeys(grid,grid_size)
1111        activekeys = zeros(grid_size,grid_size);
1112        for i = 1:grid_size
1113            for j = 1:grid_size
1114                activekeys(i,j) = grid{i,j}(4);
1115            end
1116        end
1117    end
1118    function jail_keys = count_jail_keys(jail,key)
```

```
1119        L = length(jail);
1120        jail_keys = 0;
1121        for i = 1:L
1122            if jail{i}(1) == key
1123                jail_keys = jail_keys +1;
1124            end
1125        end
1126    end
1127
1128
1129    %%
1130    %{
1131    this is the section for the ethnic model part
1132
1133    I need to:
1134     - create the groups
1135     - put groups on board
1136     - make it so you can distinguish between different
            groups being active
1137     - let groups murder each other
1138     - cloneing
1139     - old age death
1140    %}
1141
1142
```

```
1143 %{
1144 this function takes in the number of police and people
         in each ethnic group
1145 and gives back them,
1146
1147 %}
1148
1149
1150 function [group_green, group_blue, police] =
       create_ethnic(crowd_n,police_n,age_max)
1151     %take our H and R randomly from the uniform
           distrobution beween 0-1
1152     Hgreen = unifrnd(0,1,1,crowd_n)';
1153     Rgreen = unifrnd(0,1,1,crowd_n)';
1154
1155     Hblue = unifrnd(0,1,1,crowd_n)';
1156     Rblue = unifrnd(0,1,1,crowd_n)';
1157     %make a list of 1's from each person in the crowd
1158     crowdkey_blue = ones(1,crowd_n)';
1159     crowdkey_green = 4*ones(1,crowd_n)';
1160     %put crowd in the form
1161     group_green = [crowdkey_blue,Hgreen, Rgreen, zeros
           (1,crowd_n)',randi([0,age_max],1,crowd_n)'];
1162
```

```
1163        group_blue = [crowdkey_green,Hblue,Rblue,zeros(1,
                crowd_n)',randi([0,age_max],1,crowd_n)'];
1164
1165
1166        %same but police
1167        if police_n ==0
1168            police = [];
1169
1170        else
1171            police = [2*ones(1,police_n)',zeros(1,police_n
                )',zeros(1,police_n)',zeros(1,police_n)',
                zeros(1,police_n)'];
1172
1173        end
1174    end
1175
1176    %{
1177    randomly place all agents on grid for ethnic group
          simulations
1178    %}
1179
1180    function grid  = rand_initiallize_grid_ethnic(crowd_n,
          group_green, group_blue, police_n, police,
          grid_size)
1181
```

```
1182      %first start our grid out as being a grid of zeros
              for our given grid
1183      %size
1184      for i = 1:grid_size
1185          for j = 1:grid_size
1186              grid{i,j} = [0,0,0,0,0];
1187          end
1188      end
1189
1190
1191
1192  %       now we put in our crowd, distrobuted randomly
1193  %       throughout the grid
1194      p = crowd_n;
1195      while p ~= 0
1196
1197          i = randi(grid_size,1);
1198          j = randi(grid_size,1);
1199          if grid{i,j}(1) == 0
1200              grid{i,j}=group_green(crowd_n,1:5);
1201              p = p - 1;
1202          end
1203      end
1204
1205      while crowd_n ~= 0
```

```
1206
1207            i = randi(grid_size,1);
1208            j = randi(grid_size,1);
1209            if grid{i,j}(1) == 0
1210                grid{i,j}=group_blue(crowd_n,1:5);
1211                crowd_n = crowd_n - 1;
1212            end
1213        end
1214
1215
1216        %now we put the desired number of plolice randomly
                into our grid
1217        while police_n ~=0
1218            i = randi(grid_size,1);
1219            j = randi(grid_size,1);
1220            if grid{i,j}(1) == 0
1221                grid{i,j}=police(police_n,1:5);
1222                police_n = police_n - 1;
1223            end
1224        end
1225    end
1226
1227
1228    %{
1229    now I code the murder
```

```
1230 %}
1231
1232 function grid = kill(grid,grid_size,P_Vis)
1233     %for every (i,j)th place on the grid
1234
1235     for i = 1:grid_size
1236         for j = 1:grid_size
1237             %if the (i,j)th place is
1238             two= 0;
1239
1240             if grid{i,j}(1) == 1 && grid{i,j}(4) == 1
1241             %check every square 3*3 around the police
                    man
1242
1243
1244                     for n = -2:2
1245                         for one = -2:2
1246                             %assighn new (i,j)th
                                    position we are
                                    considering
1247                             positioni = i + n;
1248                             positionj = j + one;
1249                             %check point is on grid
                                    and it is a rioter of
1250                             %the opposite group
```

```
1251                    if positioni<=grid_size &&
                            positionj <= grid_size
                            && positioni > 0 &&
                            positionj > 0 && grid{
                            positioni,positionj}(1)
                            == 4
1252                        %remove from grid :)
1253                        grid{positioni,
                                positionj} =
                                [0,0,0,0,0];
1254                        %return so we only
                                arest 1 person per
                                turn
1255                        two=1;
1256                        break
1257
1258                    end
1259
1260                end
1261                if two==1
1262                    break
1263                end
1264
1265            end
1266
```

```
1267            else
1268
1269                if grid{i,j}(1) == 4 && grid{i,j}(4)
                       == 1
1270                %check every square 3*3 around the
                       police man
1271
1272
1273                    for n = -2:2
1274                        for one = -2:2
1275                            two = 0;
1276                            %assighn new (i,j)th
                                   position we are
                                   considering
1277                            positioni = i + n;
1278                            positionj = j + one;
1279                            %check point is on
                                   grid and it is a
                                   rioter
1280                            if positioni<=
                                   grid_size &&
                                   positionj <=
                                   grid_size &&
                                   positioni > 0 &&
                                   positionj > 0 &&
```

```
                                    grid{positioni,
                                    positionj}(1) == 1
1281                                %remove from grid
                                        :)
1282                                grid{positioni,
                                        positionj} =
                                        [0,0,0,0,0];
1283                                %return so we only
                                         arest 1 person
                                         per turn
1284                                two=1;
1285                                break
1286
1287                            end
1288
1289                        end
1290                        if two==1
1291                            break
1292                        end
1293
1294                    end
1295
1296                end
1297            end
1298        end
```

```
1299      end
1300
1301  end
1302
1303
1304  function grid = ethnic_group_to_riot(grid,grid_size,L,
          vision_l,T)
1305      %loop through each space in the grid
1306      for i = 1:grid_size
1307      for j = 1:grid_size
1308          %if we land on an agent
1309          if  grid{i,j}(1) == 1 || grid{i,j}(1) == 4
1310              %find its grievance number
1311              G  = grievance(grid{i,j}(2),L);
1312
1313              %find its risk probablility
1314              R  = grid{i,j}(3);
1315
1316              %find the vision at that point
1317              v = vision(i,j, grid, vision_l, grid_size)
                      ;
1318
1319              %count number of police in vision
1320              C=count_keys(v,2);
1321
```

```
1322                %count number of actives(rioters) in
                        vision
1323                A=count_keys(v,3)+1;
1324
1325                %with our C and A we can now find the
                        probability for our guy
1326                %at i j to be arrested
1327                P = arrest_probability(2.3,C,A);
1328
1329                %our agents net risk
1330                N = R*P;
1331
1332                %expected utility of publicly expressing
                        ones private grievanc
1333                gmn = G-N;
1334
1335                if gmn >T
1336                    grid{i,j}(4) = 1;
1337                else
1338                    grid{i,j}(4) = 0;
1339                end
1340
1341            end
1342
1343        end
```

```
1344        end
1345        %grid;
1346    end
1347
1348    function grid = cloneing(grid_size,grid,age_max)
1349        %loop throuh the grid
1350        for i = 1:grid_size
1351            for j = 1:grid_size
1352                %if the grid is one of the ethnic groups
1353                if grid{i,j}(1) == 1 || grid{i,j}(1) == 4
1354                    %create a 1 in 20 chance to clone
1355                    prob = randi(20);
1356                    %if we get 1/20
1357                    if prob == 1
1358                        %check spaces next to character
                                being cloned
1359                        one = 0;
1360                        for x = -1:1
1361                            for y = -1:1
1362                                position_x = i + x;
1363                                position_y = j + y;
1364                                %check the position is
                                        within the grid and is
1365                                %empty
```

```
1366                                    if position_x > 0 &&
                                           position_x < grid_size
                                           +1 && position_y > 0 &&
                                            position_y < grid_size
                                            + 1 &&  grid{
                                           position_x, position_y
                                           }(1) == 0
1367                                        grid{position_x,
                                               position_y} = [grid
                                               {i,j}(1),grid{i,j
                                               }(2), unifrnd(0,1)
                                               ,0,randi([0,age_max
                                               ])];
1368                                        one = 1;
1369                                        break
1370                                    end
1371                                end
1372 %                                  if one ==1
1373 %                                      break
1374 %                                  end
1375                             end
1376                         end
1377
1378                     end
1379                 end
```

```
1380        end
1381    end
1382
1383
1384    function grid = ageing(grid,grid_size)
1385        %loop through the grid
1386        for i = 1:grid_size
1387            for j = 1:grid_size
1388                %check is someone ahs died of old age
1389                if grid{i,j}(5) == 0 && or(grid{i,j}(1)
                        ==1, grid{i,j}(1)==4)
1390                    grid{i,j} = [0,0,0,0,0];
1391                %if theyre not dead make them age one time
1392                else
1393                    if or(grid{i,j}(1)==1, grid{i,j}(1)
                            ==4)
1394                        grid{i,j}(5) = grid{i,j}(5) - 1;
1395                    end
1396                end
1397            end
1398        end
1399    end
1400
1401
1402
```

```
1403
1404 function [jail,grid] = riot_to_arrest_ethnic(grid_size
       , grid,jail,jail_L,P_Vis)
1405     %for every (i,j)th place on the grid
1406
1407     for i = 1:grid_size
1408         for j = 1:grid_size
1409             %if the (i,j)th place is a policeman
1410             one = 0 ;
1411
1412                 if grid{i,j}(1) == 2
1413                 %check every square 3*3 around the
                        police man
1414
1415
1416                         for n = -P_Vis:P_Vis
1417                             for m = -P_Vis:P_Vis
1418                                 %assighn new (i,j)th
                                        position we are
                                        considering
1419                                 positioni = i + n;
1420                                 positionj = j + m;
1421                                 %check point is on
                                        grid and it is a
                                        rioter
```

```
1422                    if positioni<=
                           grid_size &&
                           positionj <=
                           grid_size &&
                           positioni > 0 &&
                           positionj > 0 &&
                           grid{positioni,
                           positionj}(4) == 1
1423                        %arrest that
                               rioter
1424
1425
1426                       % grid{positioni,
                             positionj}(1)=1;
1427                        l = grid{
                               positioni,
                               positionj};
1428                       sentence = randi(
                               jail_L);
1429                      l(4)=sentence;
1430
1431
1432                      jail{end+1} = l;
1433
```

```
1434                                    grid{positioni,
                                            positionj} =
                                            [0,0,0,0,0];
1435                                    %return so we only
                                            arest 1 person
                                            per turn
1436                                    one =1;
1437                                    break
1438
1439                                end
1440
1441                            end
1442 %                              if one ==1
1443 %                                  break
1444 %                              end
1445
1446                        end
1447
1448                end
1449            end
1450        end
1451
1452
1453
1454 end
```

here is the code for my jail

```
1
2
3  function [jail,grid] = update_jail(jail,grid,grid_size
       )
4      %store the length of our jail in the beginning
5      if isempty(jail) == 0
6          %jail1 = jail
7          %create a list for everyone leaving
8          leavers = {};
9          %create a list of people leaving
10
11         for i = 1:length(jail)
12             if jail{i}(4) == 0
13
14                 leavers{end+1} = jail{i};
15
16             end
17         end
18
19         v = 1;
20         %put people back onto the board
21         count = length(leavers);
22         while count ~= 0 && v ~= 100000
23             i = randi(grid_size,1);
```

```
24              j = randi(grid_size,1);
25              v=v+1;
26              if grid{i,j}(1) == 0
27                  %leavers{count}
28                  grid{i,j}= [leavers{count}];
29                  count = count - 1;
30              end
31              %assume grid is full if v = 10000 so empty
                    the jail
32
33          end
34          %jail3 = jail
35          %create count
36          count = 1;
37          %whilst there are people needing to leave the
                jail
38          while count_leavers(jail) ~=0
39
40
41              %remove them from the jail
42
43              if jail{count}(4)==0
44                  jail(count)=[];
45                  leavers(1) = [];
46                  count = count -1;
```

```
47                  end
48                  count=count+1;
49              end
50              %jail4 = jail
51              %jail = jail(not(cellfun(@(x)isequal(x([1])
                    ,[2]),jail)));
52              N = length(jail);
53
54
55              for i = 1:N
56
57
58                      jail{i}(4) = jail{i}(4) -1;
59              end
60          end
61
62      end
63
64
65
66
67
68
69      %
70      % function [jail,grid] = update_jail1(jail,grid,
```

```
    grid_size)
71 %     %store the length of our jail in the beginning
72 %     if isempty(jail) == 0
73 %         %get the length of the jail
74 %         N = length(jail);
75 %         %for every ith element from 1-the length of
    the jail
76 %         for i = 1:N
77 %             %if the sentnce is =0
78 %             if jail{i}(4)==0
79 %                 q=1
80 %                 while q ~= 0
81 %                     %find a place on the grid
82 %                     for m =1:grid_size
83 %                         for n = 1:grid_size
84 %
85 %                             %if the space is
    free
86 %                             if grid{n,m}(1)==0
87 %                                 %add the current
     ith jail member to
88 %                                 %that space
89 %                                 grid{n,m} = jail
    {i};
90 %                                 %change the
```

```
first number of the guy to
91 %                                          %5
92 %                                          jail(i) =
{[5,0.9,0.9,-1]};
93 %                                          %stop the while
loop if we find a space
94 %                                          q=q-1
95 %                                          break
96 %
97 %                                      end
98 %                                      break
99 %                                  end
100 %                                 break
101 %                         end
102 %                         break
103 %                         end
104 %                     end
105 %
106 %
107 %                 end
108 %           i=1;
109 %           while count_leavers(jail) ~=0 && isempty(
jail) == 0
110 %
111 %                     if jail{i}(1)==5
```

```
112 %                       jail(i)=[];
113 %                   end
114 %               i=i+1;
115 %           end
116 %
117 %           %jail = jail(not(cellfun(@(x)isequal(x([1])
        ,[2]),jail)));
118 %           N = length(jail);
119 %
120 %
121 %           for i = 1:N
122 %
123 %
124 %                   jail{i}(4) = jail{i}(4) -1;
125 %           end
126 %       end
127 % end
128
129
130
131 function count = count_leavers(jail)
132     n = length(jail);
133     count = 0;
134     for i = 1:n
135         if jail{i}(4) == 0
```

```
136                 count = count +1;
137             end
138         end
139 end
```