HW4 - Evolutionary games

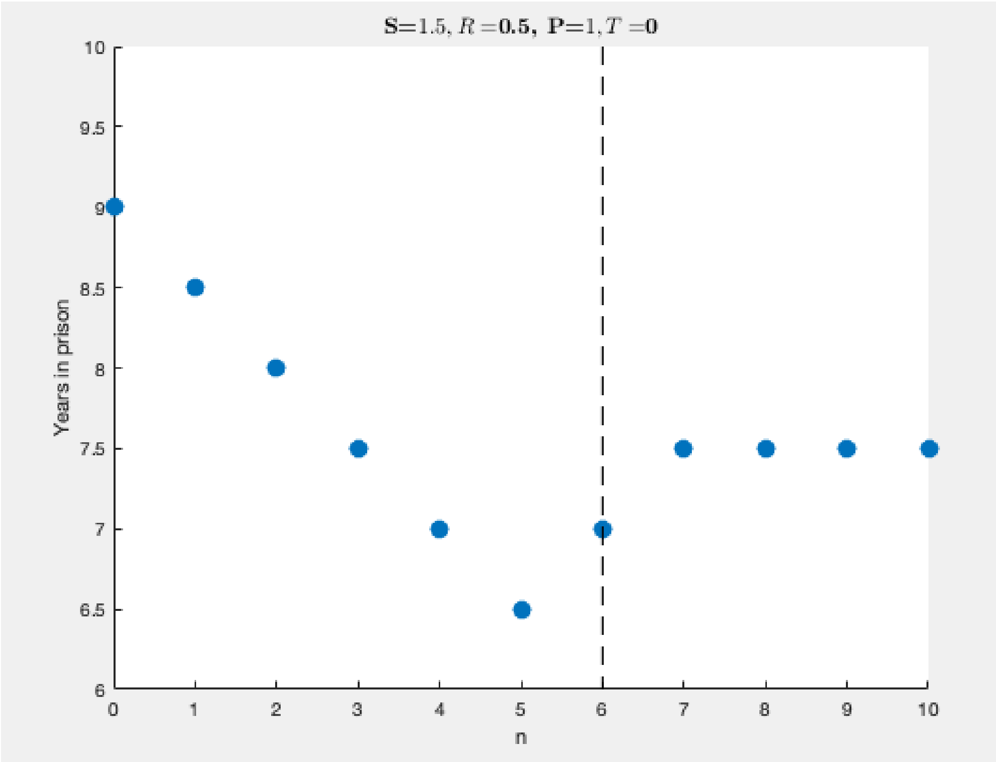
Nicole Adamah 2022

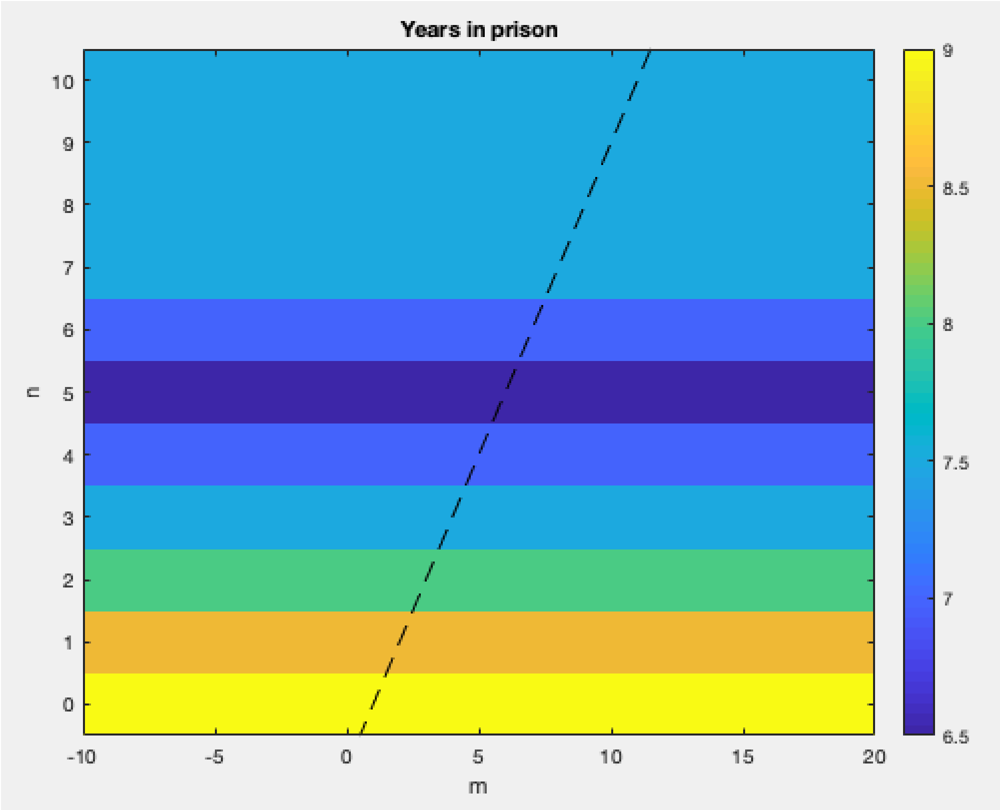
Exercise 13.1. Prisoner's dilemma with multiple rounds. Simulate the prisoner's dilemma for a number of rounds and calculate the accumulated years in prison. Initially, choose the parameters N = 10,T = 0, R = 0.5, P = 1, and S = 1.5.

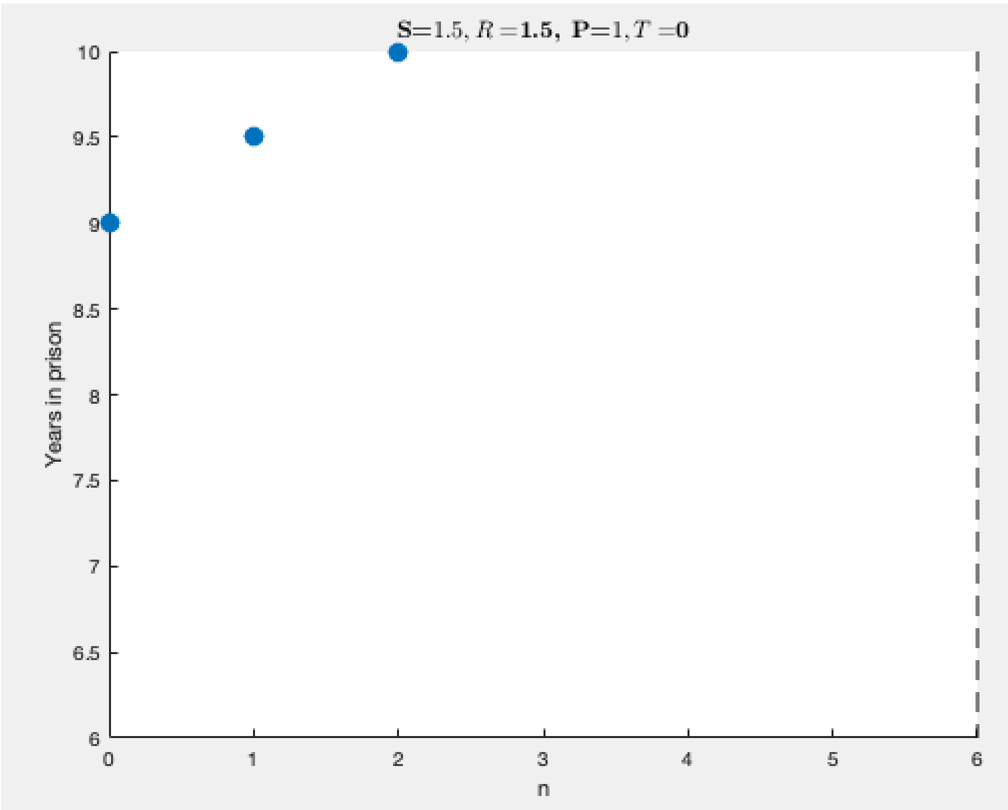
a. Fixing the opponent's strategy to m, show that the best strategy is n = m − 1, as shown in figure 13.3(a).

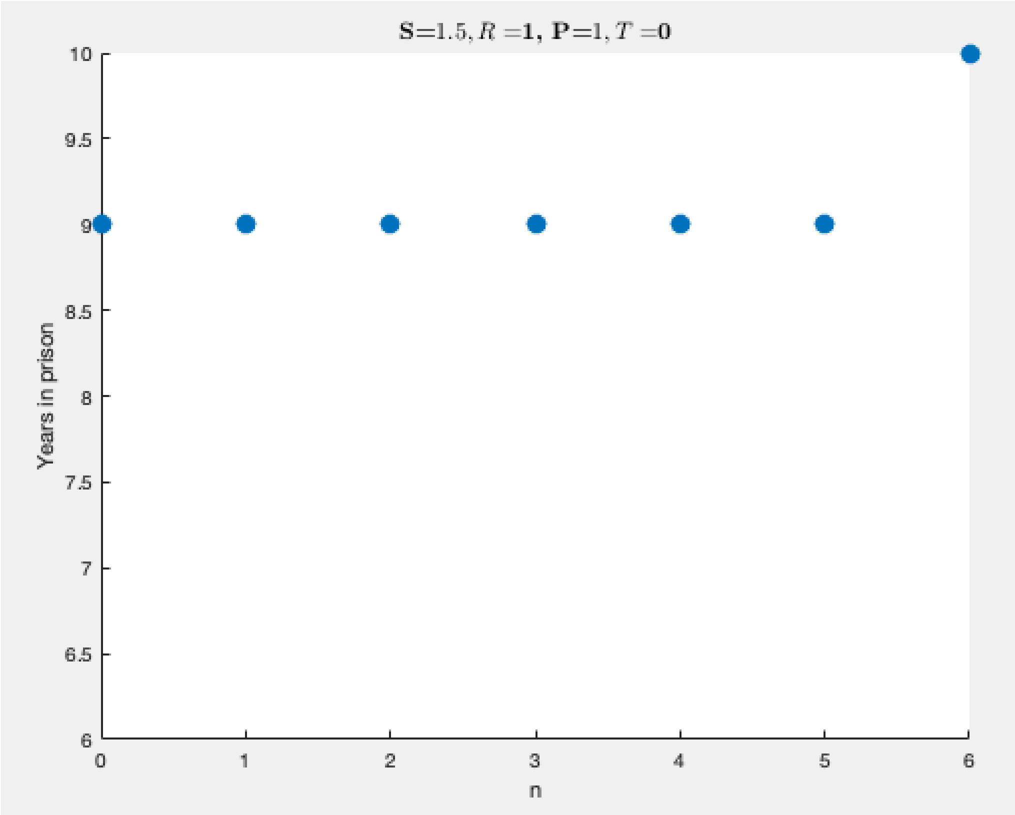
b. Generate a 2D map of years in prison as a function of the player's strategy n and the opponent's strategy m, as shown in figure 13.3(b).

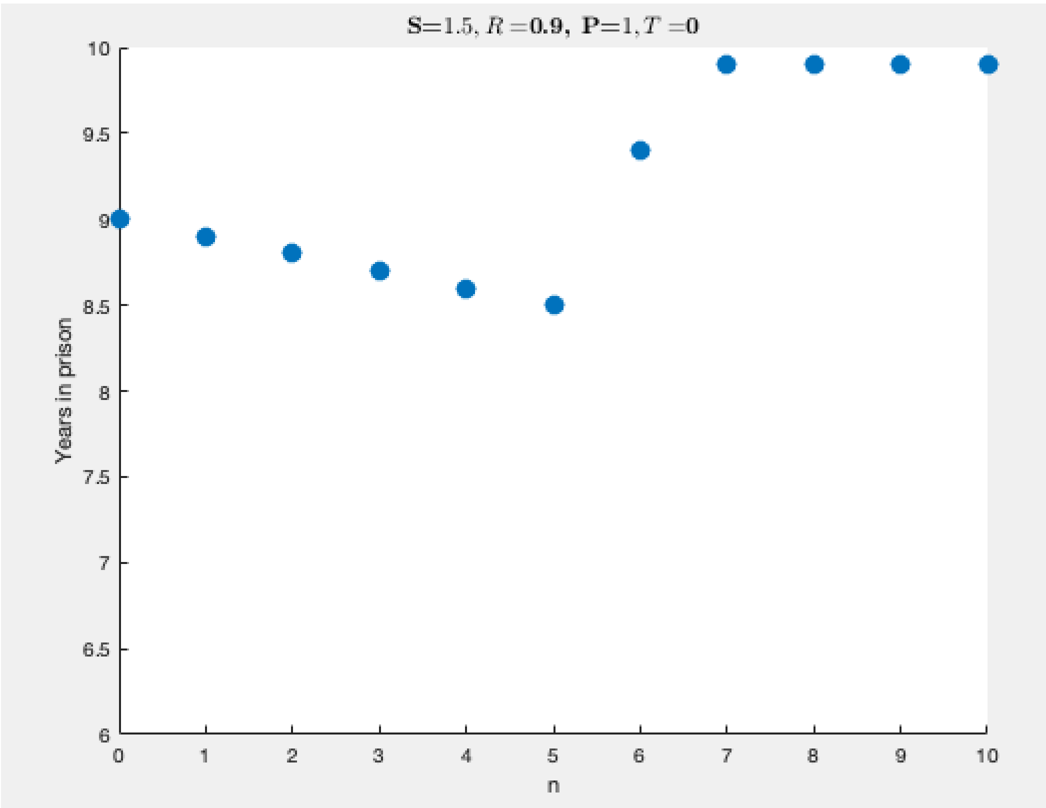
c. While keeping T = 0 and P = 1, play with R and S to see how these parameters affect the results. Show that, as long as the essential conditions of the prisoner's dilemma (T < R < P < S) are satisfied, it is always best to use the strategy m − 1 against a player who uses strategy m.

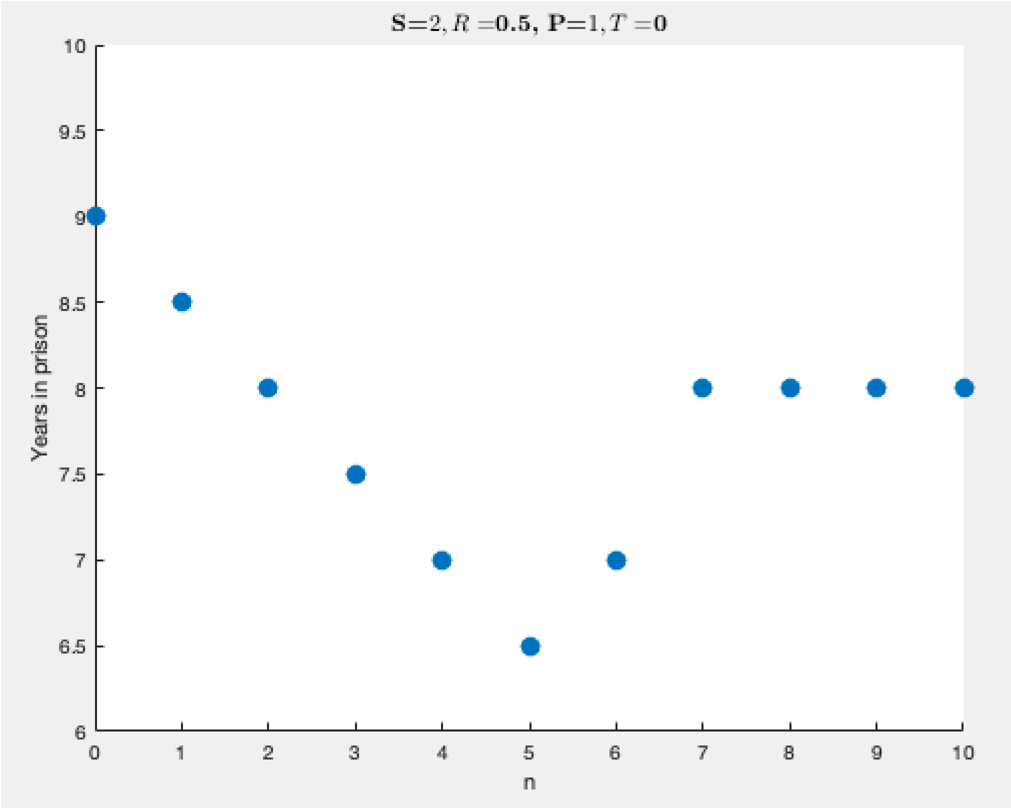


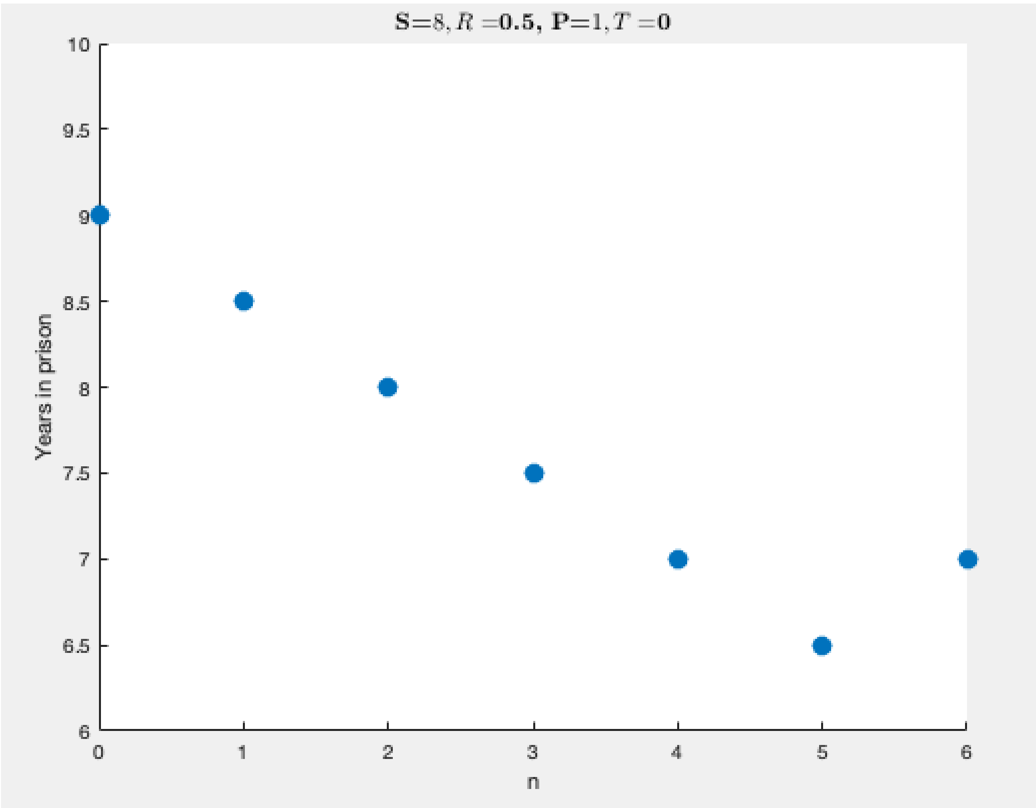

Years in prison
n
m
10
9
8
7
6
5
4
3
2
1
0
-10
-5
0
5
10
15
20
9
8.5
8
7.5
7
6.5

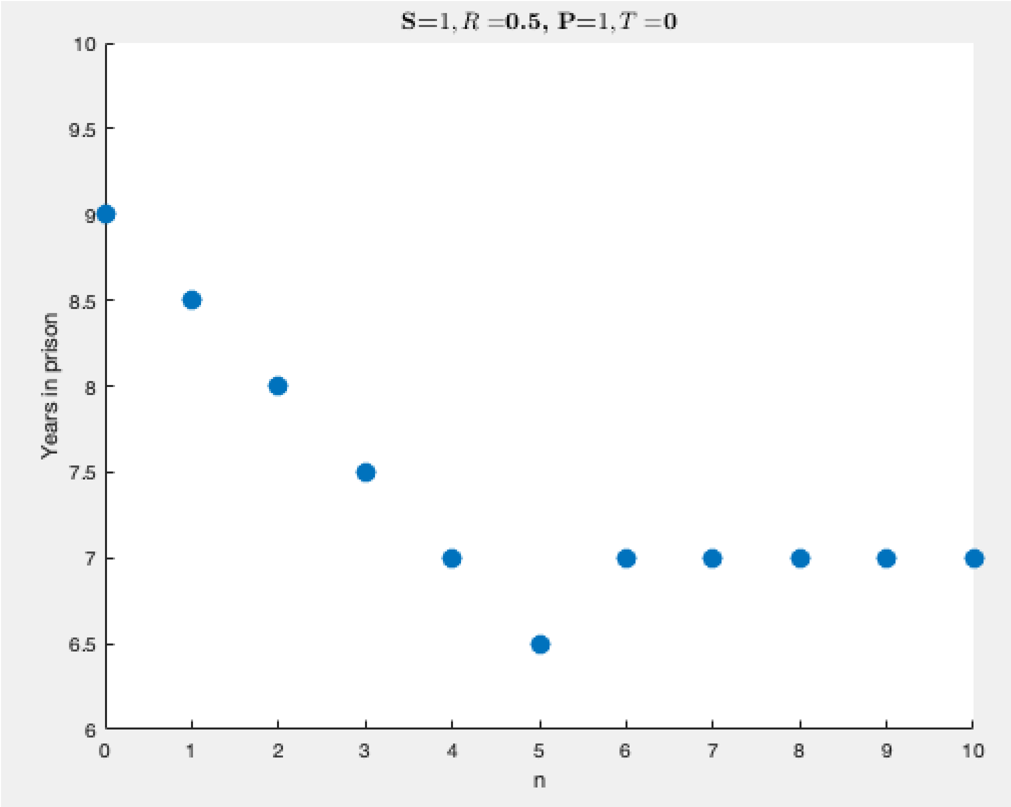

$\mathbf{S}{=}1.5, R=\mathbf{1.5},\ \mathbf{P}{=}1, T=\mathbf{0}$
Years in prison
n
10
9.5
9
8.5
8
7.5
7
6.5
6
0
1
2
3
4
5
6


$\mathbf{S}$=1.5, R =1, $\mathbf{P}$=1, T =0
Years in prison
n

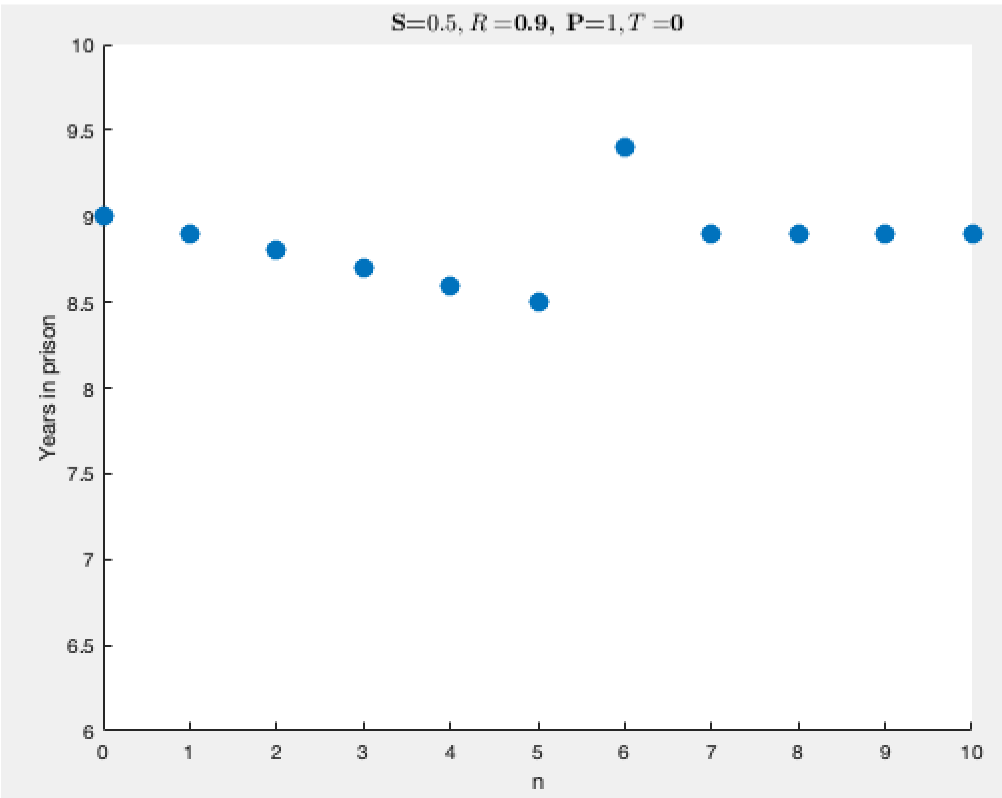

$\mathbf{S}$=1.5, R =0.9, $\mathbf{P}$=1, T =0
Years in prison
n


$\mathbf{S=}2, R=\mathbf{0.5,\ P=}1, T=\mathbf{0}$
Years in prison
n

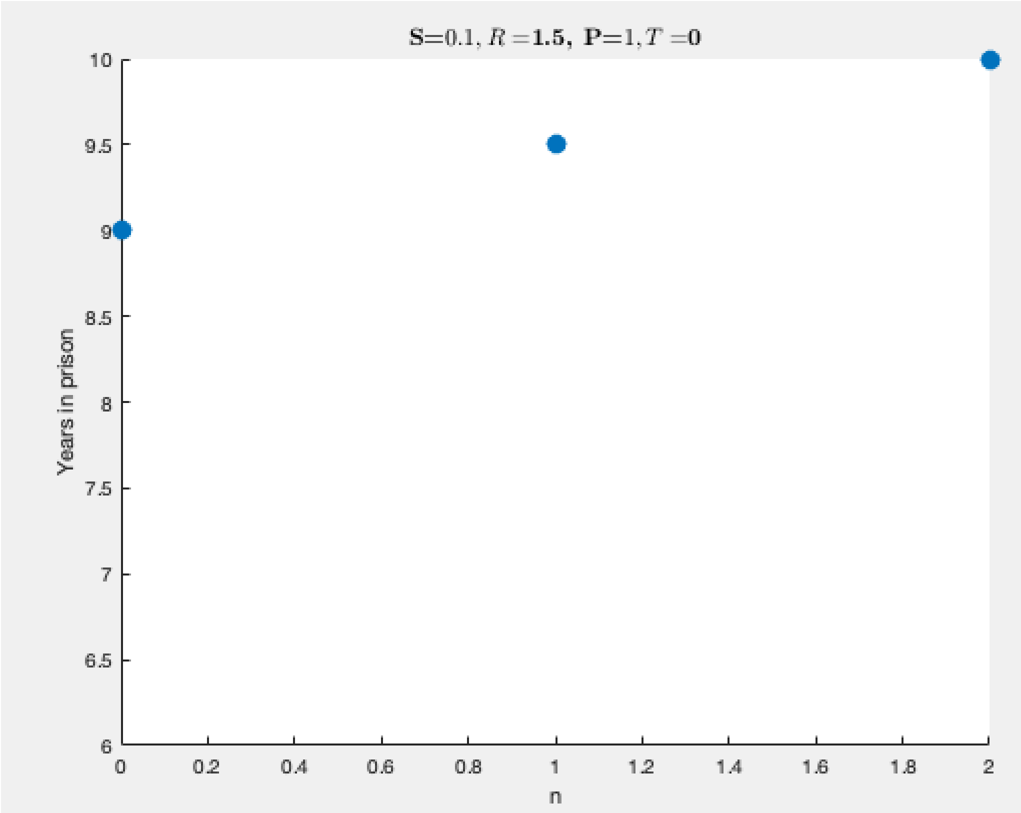

$\mathbf{S=}8, R=\mathbf{0.5,\ P=}1, T=\mathbf{0}$
Years in prison
n

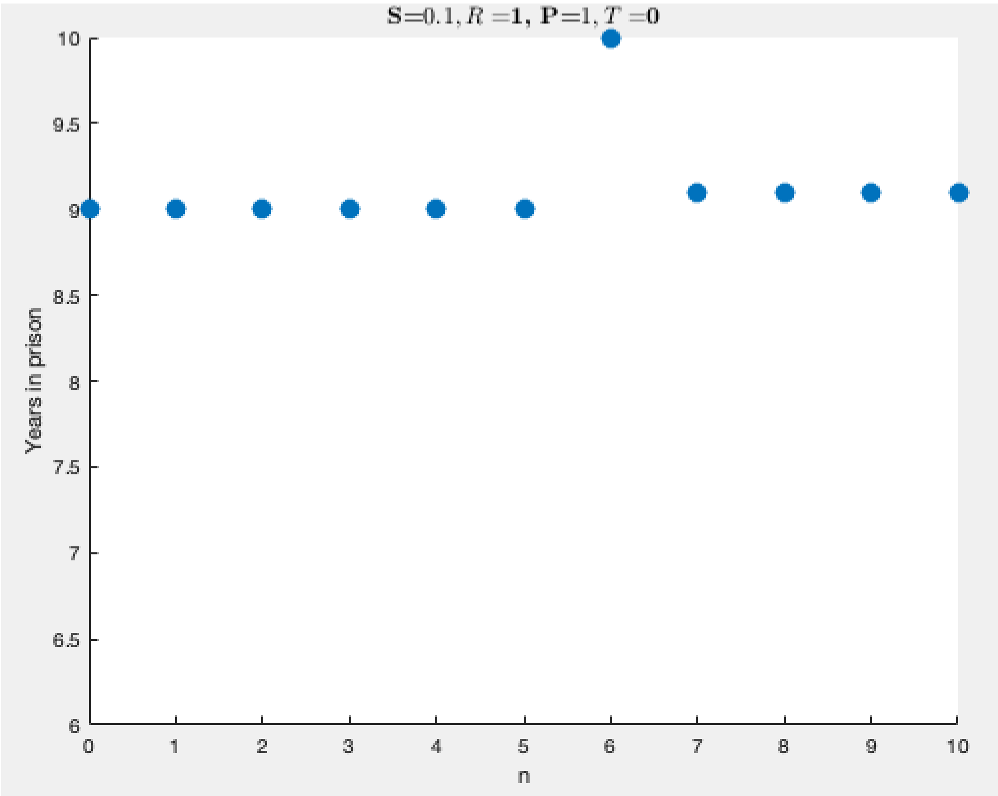


$\mathbf{S}{=}1, R = \mathbf{0.5},\ \mathbf{P}{=}1, T = \mathbf{0}$
Years in prison
n



$\mathbf{S}{=}0.5, R = \mathbf{0.9},\ \mathbf{P}{=}1, T = \mathbf{0}$
Years in prison
n


S=0.1, R =1.5, P=1, T =0
Years in prison
n


S=0.1, R =1, P=1, T =0
Years in prison
n

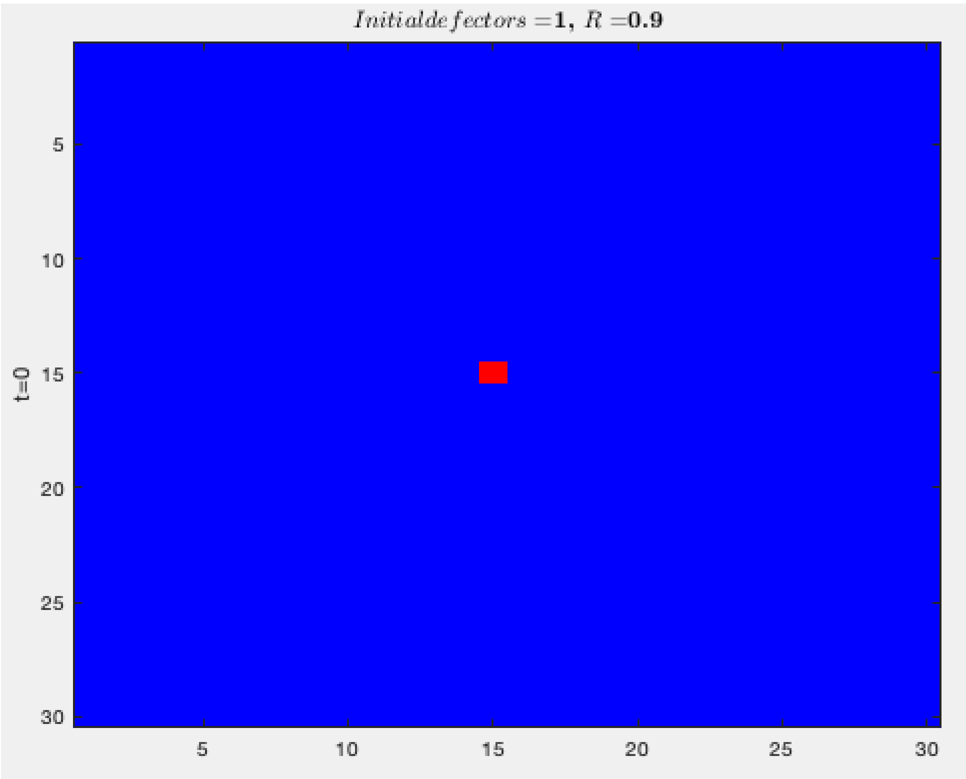
Exercise 13.2. Patterns in evolutionary games. Simulate the prisoner's dilemma on an L × L lattice. Allow only two strategies: always cooperate (n = N) and always defect (n = 0). Assume $\mu = 0$. Make sure that you choose a very small value of L at first to try your code and visualize its results. Then, increase L to a larger value (between 20 and 100). As usual, use T = 0 and P = 1. In addition, you can fix $S = 1.5$ and just play with the parameter R.

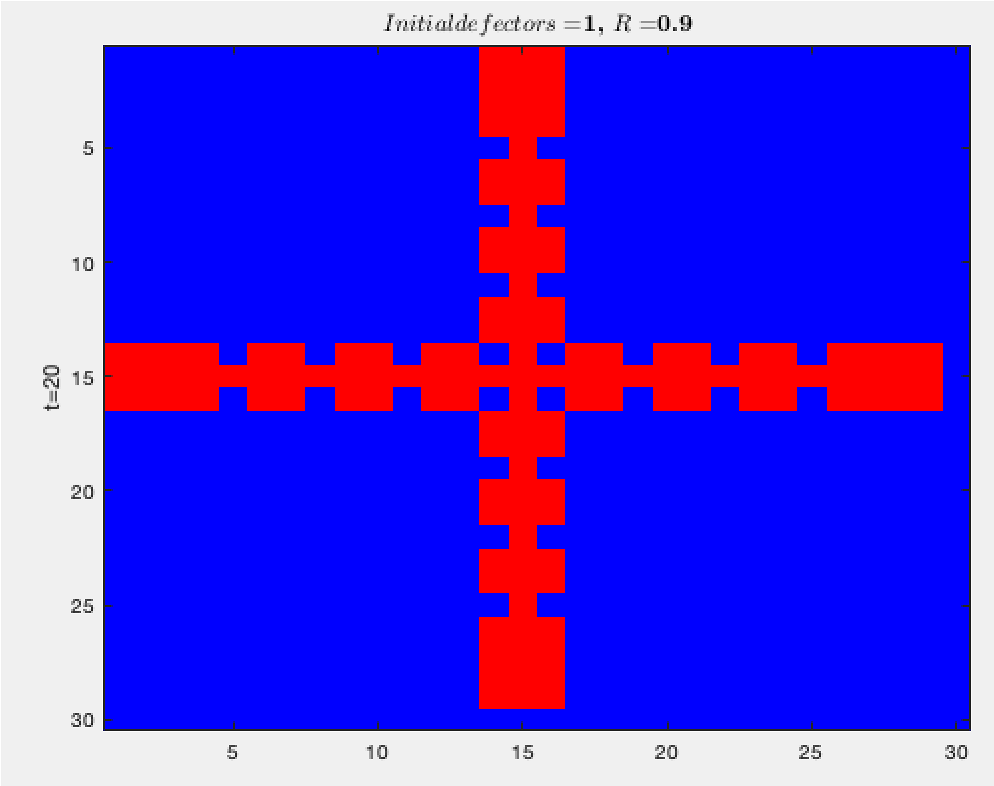
a. Initialize a lattice full of cooperators and place a single defector in the middle. Discover the range of R for which the defecting behavior only spreads in a line pattern in all directions, as shown in figure 13.5(b). What happens for other values of R?

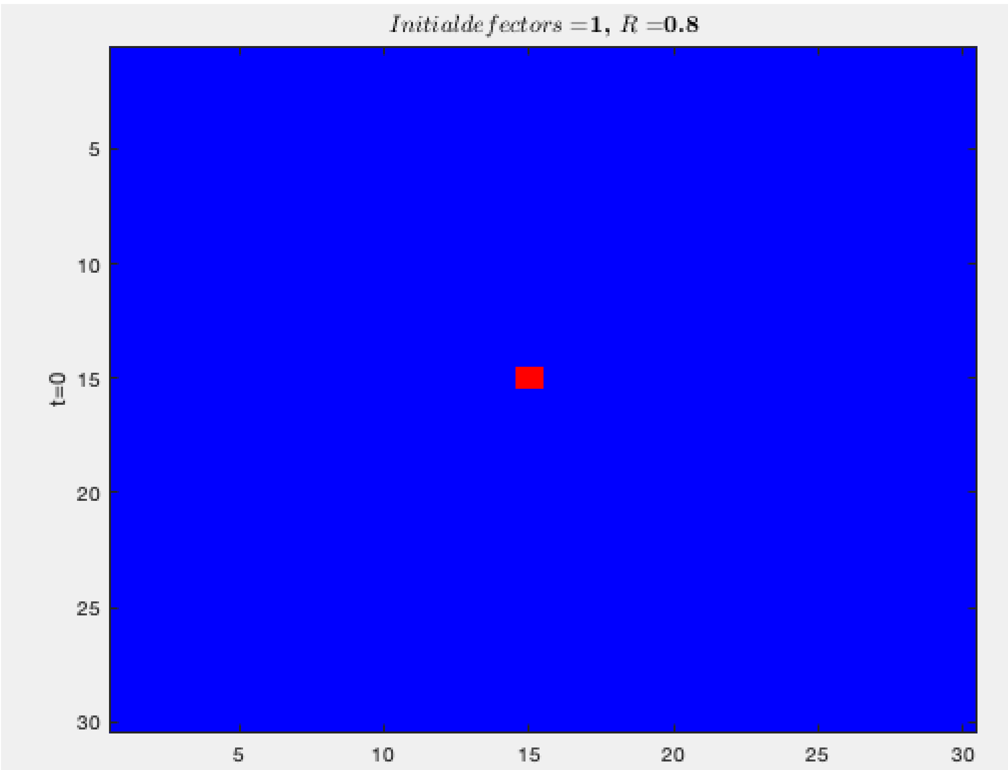
b. Play with the number of initial defectors and simulate the system. Observe different pattern formations similar to those shown in figures 13.5(c)–(h).

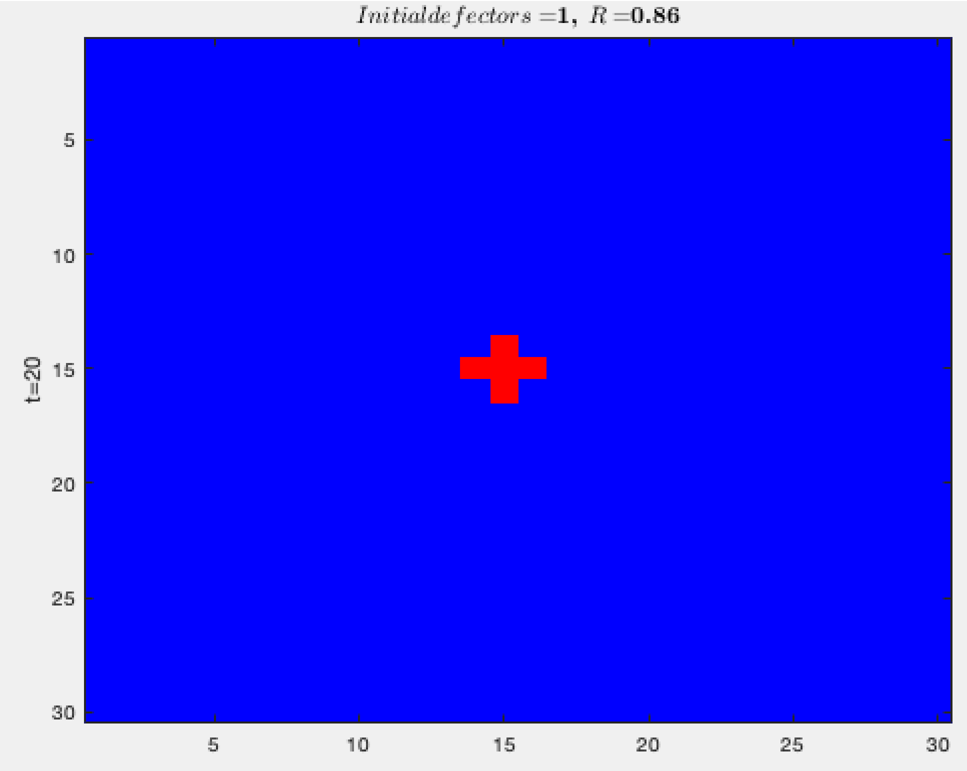
c. What happens if you place a single cooperator in a lattice of defectors?

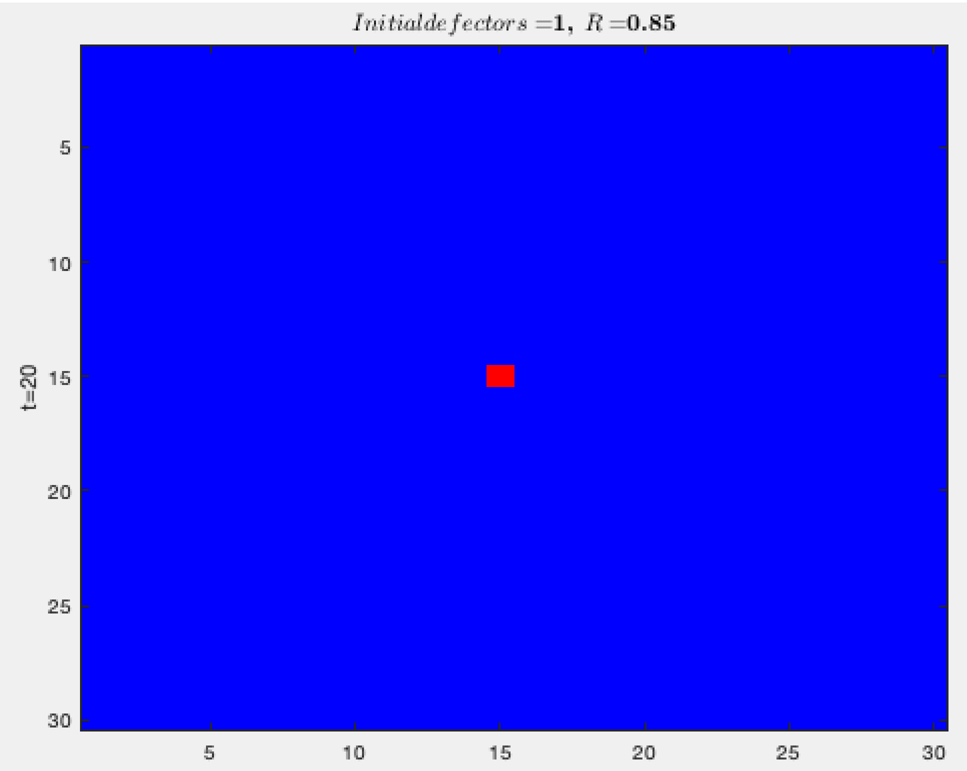
d. Find the ranges of R for which a cluster of cooperators in a background of defectors vanishes, stays stable, or spreads.

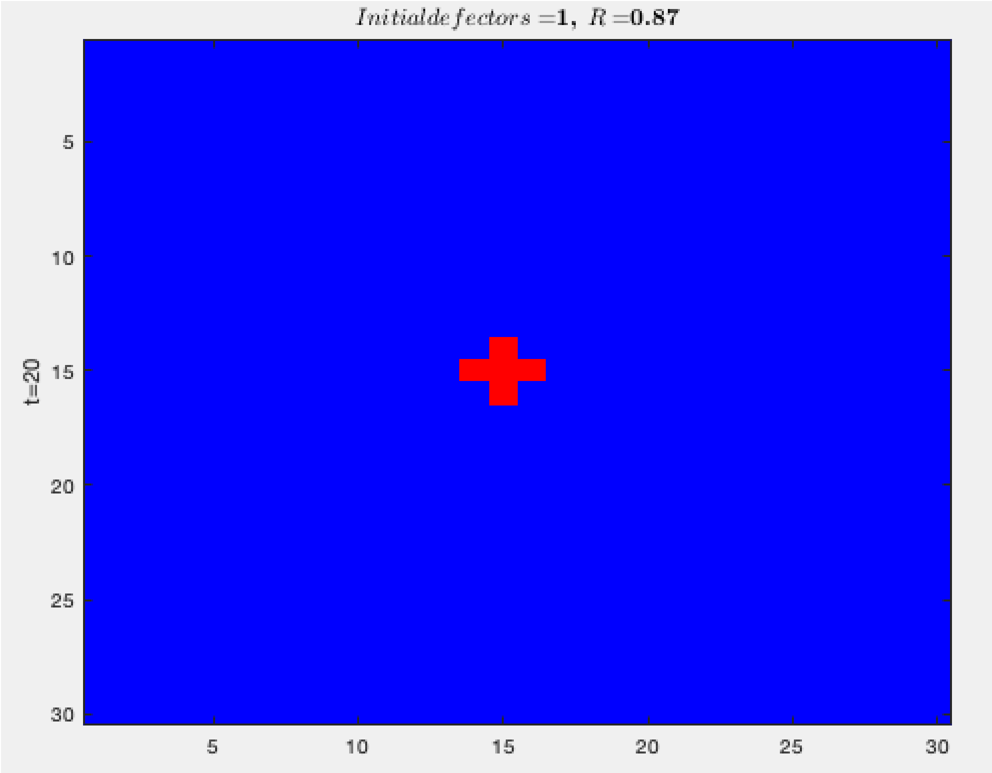



$Initialdefectors = \mathbf{1},\ R = \mathbf{0.9}$
t=20
5
10
15
20
25
30
5
10
15
20
25
30

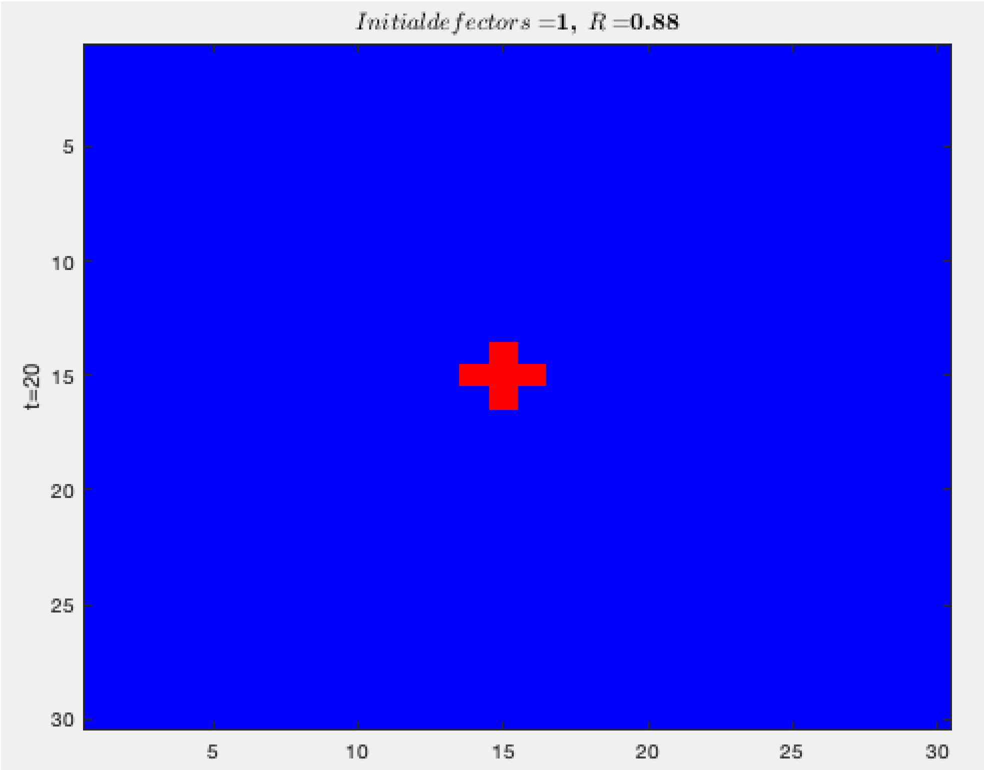

$Initialdefectors = \mathbf{1},\ R = \mathbf{0.8}$
t=0
5
10
15
20
25
30
5
10
15
20
25
30


$Initial defectors = \mathbf{1},\ R = \mathbf{0.86}$
t=20
5
10
15
20
25
30
5
10
15
20
25
30

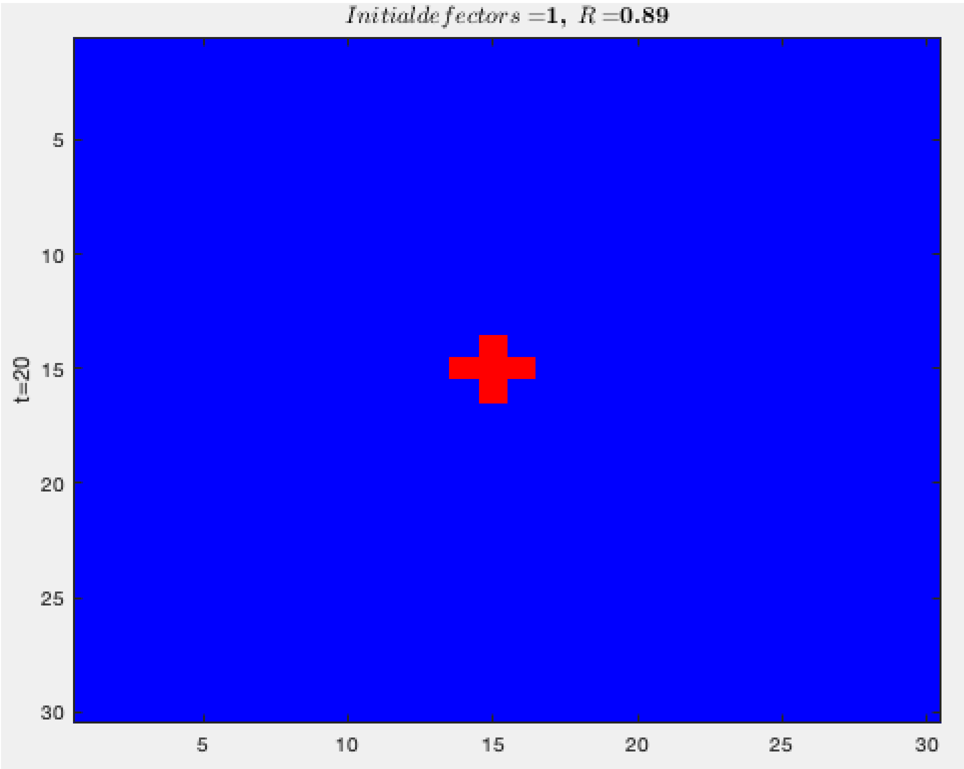

$Initial defectors = \mathbf{1},\ R = \mathbf{0.85}$
t=20
5
10
15
20
25
30
5
10
15
20
25
30

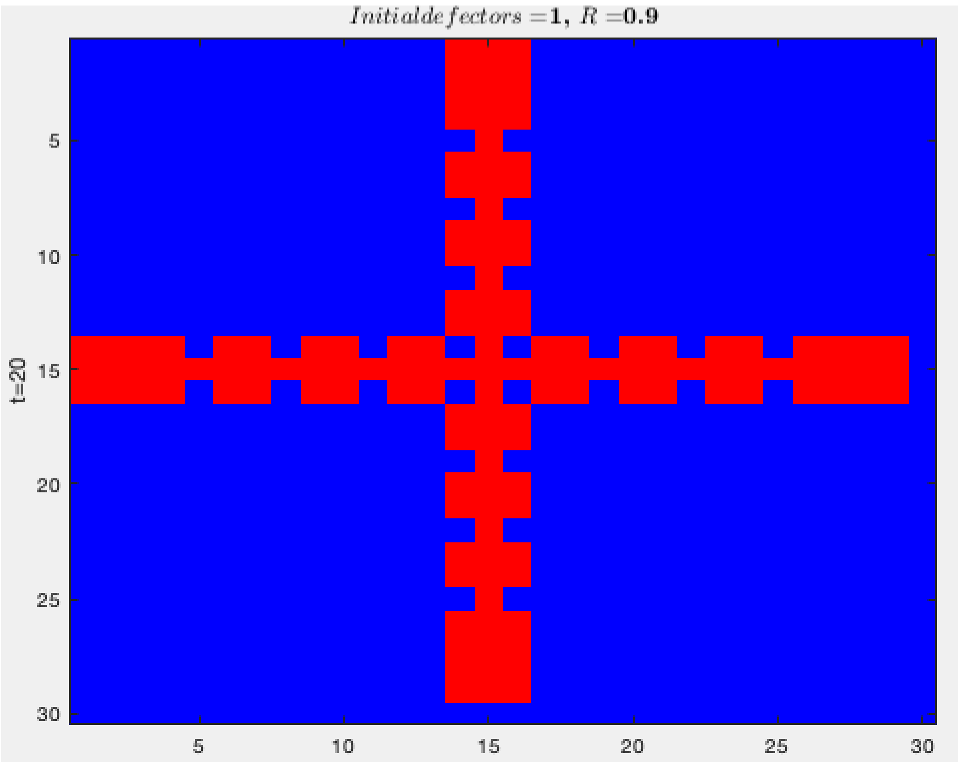


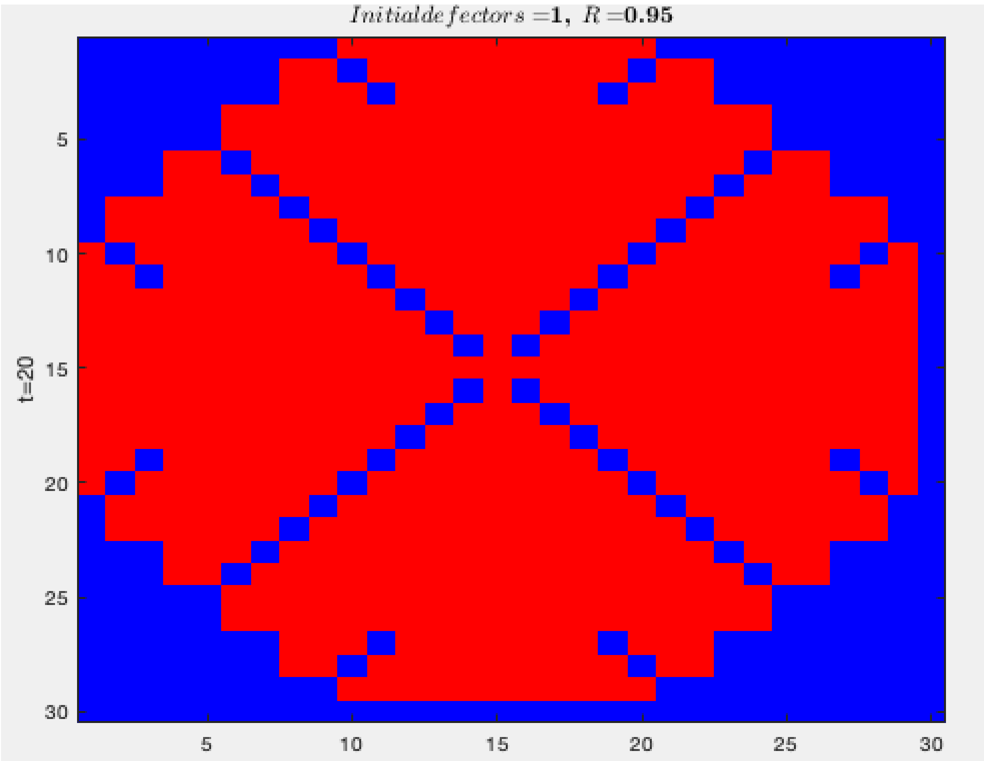
$Initial defectors = \mathbf{1},\ R = \mathbf{0.87}$
t=20
5
10
15
20
25
30

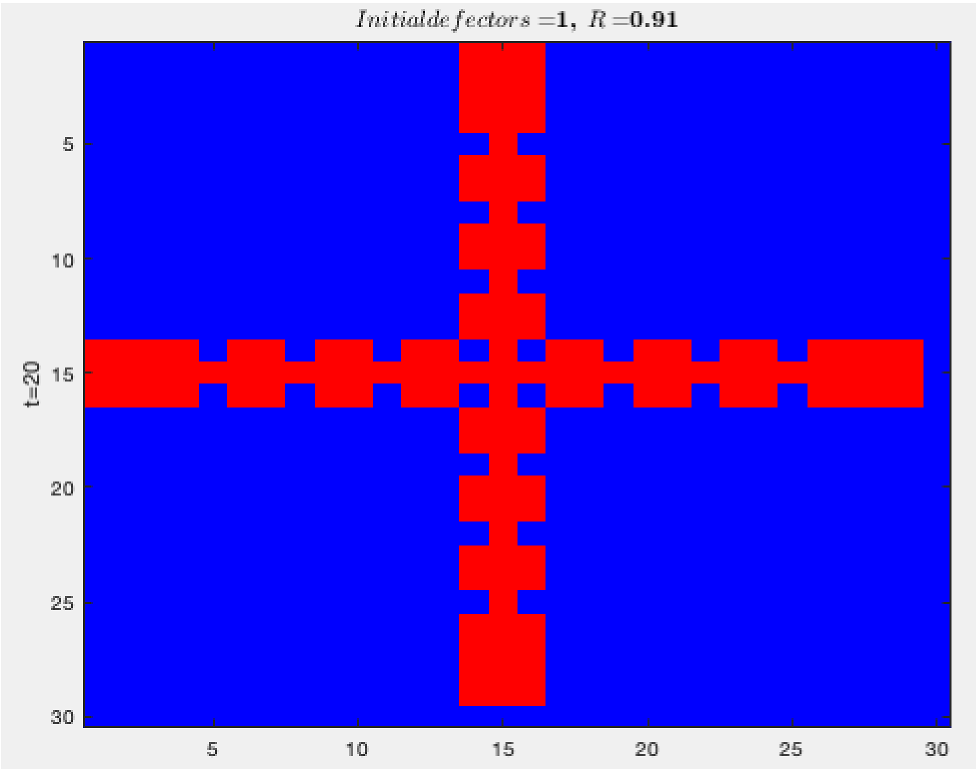


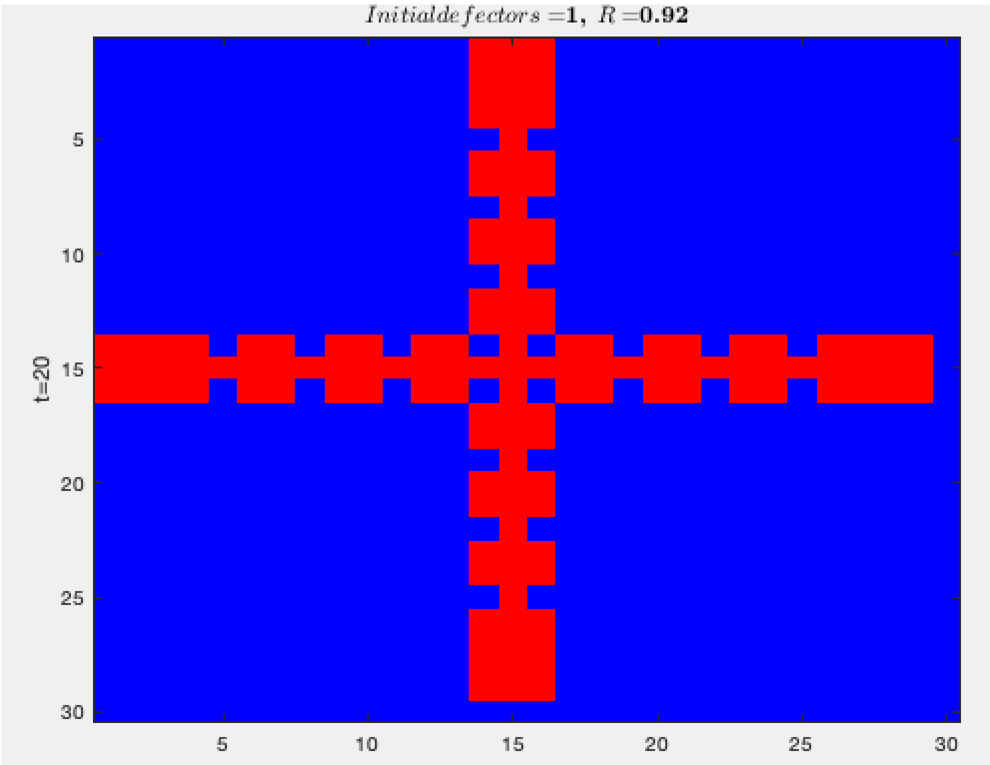
$Initial defectors = \mathbf{1},\ R = \mathbf{0.88}$
t=20
5
10
15
20
25
30


$Initial defectors = \mathbf{1},\ R = \mathbf{0.89}$
t=20
5
10
15
20
25
30
5
10
15
20
25
30

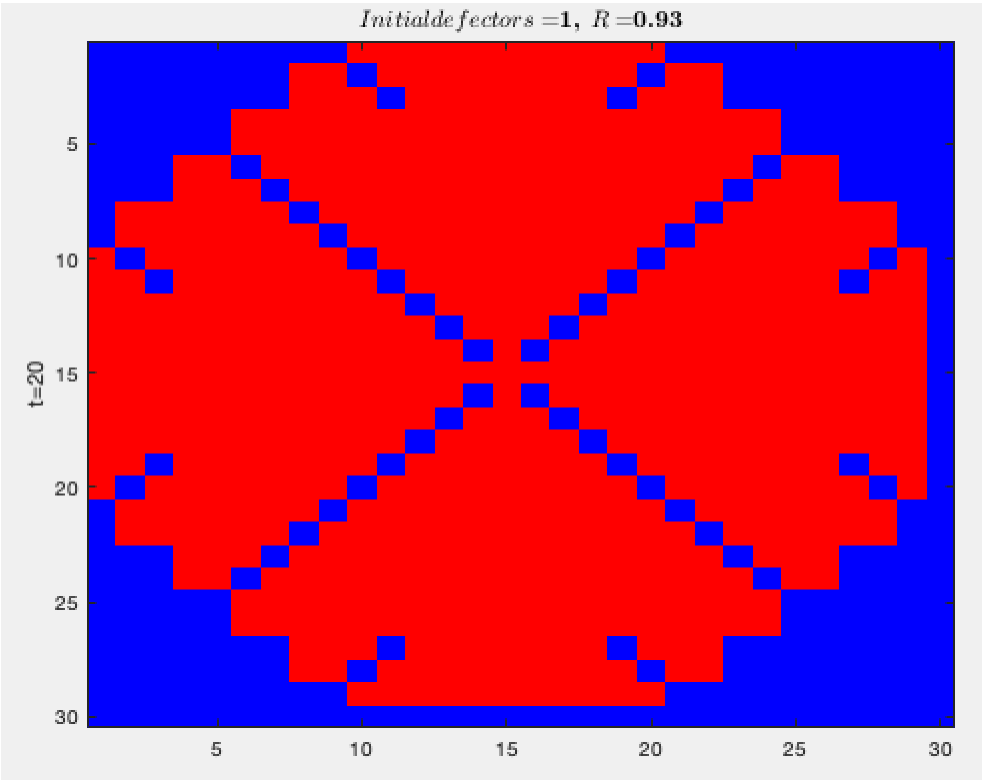

$Initial defectors = \mathbf{1},\ R = \mathbf{0.9}$
t=20
5
10
15
20
25
30
5
10
15
20
25
30


$Initial defectors = \mathbf{1},\ R = \mathbf{0.95}$
t=20
5
10
15
20
25
30
5
10
15
20
25
30

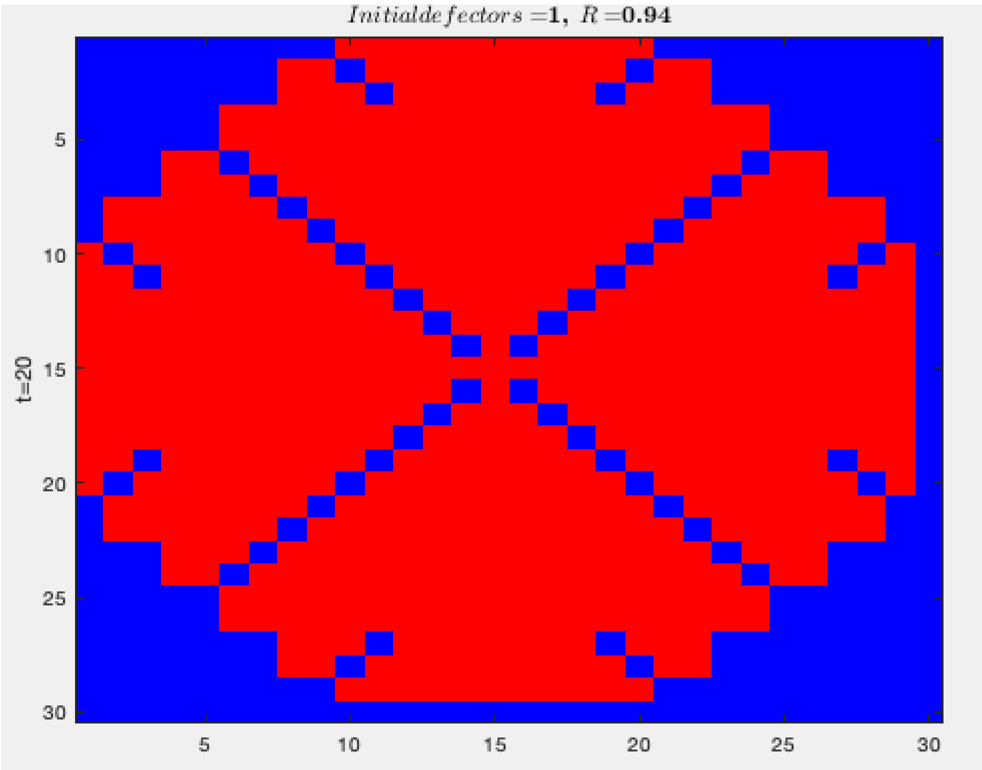

$Initial defectors = \mathbf{1},\ R = \mathbf{0.91}$
t=20
5
10
15
20
25
30
5
10
15
20
25
30



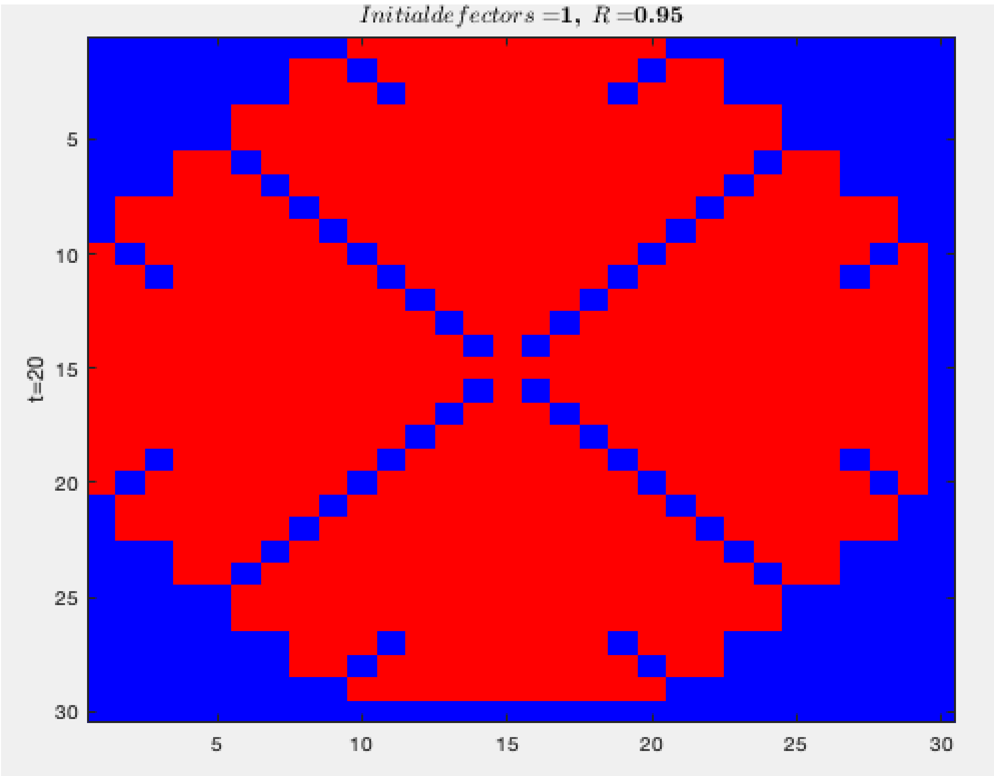
$Initial defectors = \mathbf{1},\ R = \mathbf{0.92}$
t=20
5
10
15
20
25
30



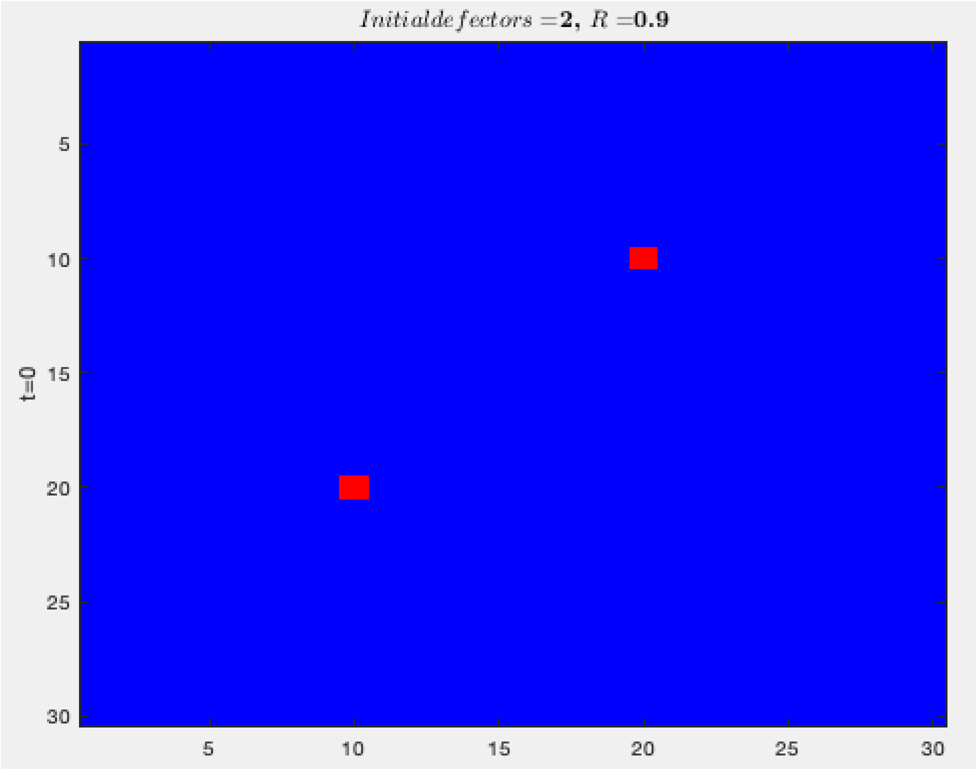
$Initial defectors = \mathbf{1},\ R = \mathbf{0.93}$
t=20
5
10
15
20
25
30

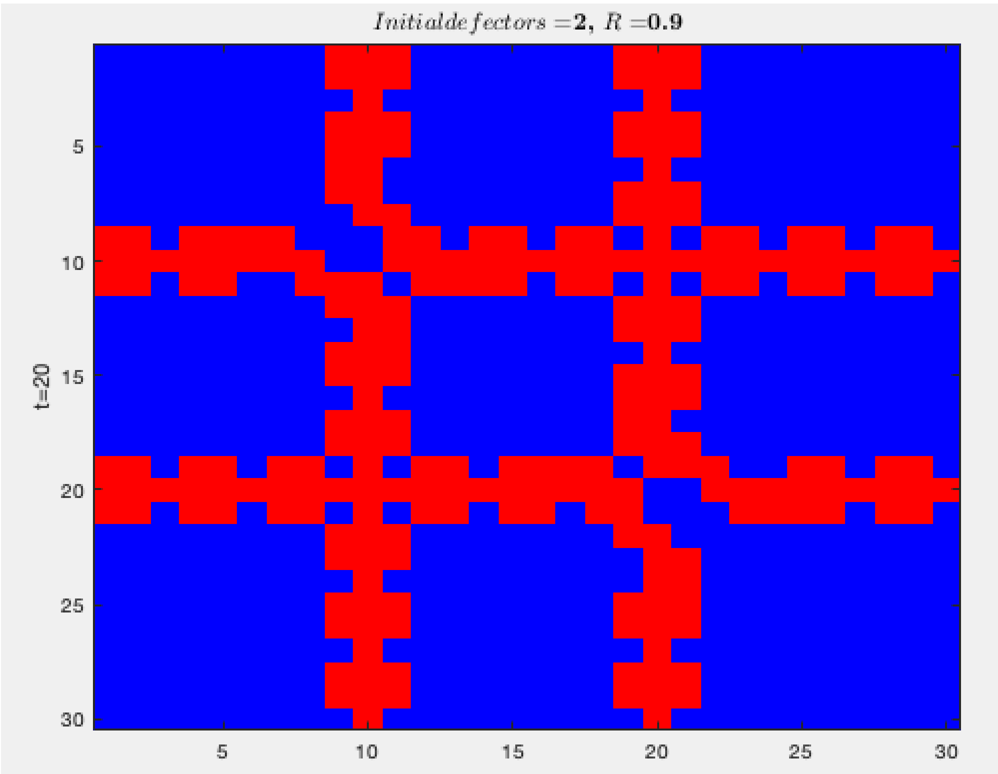


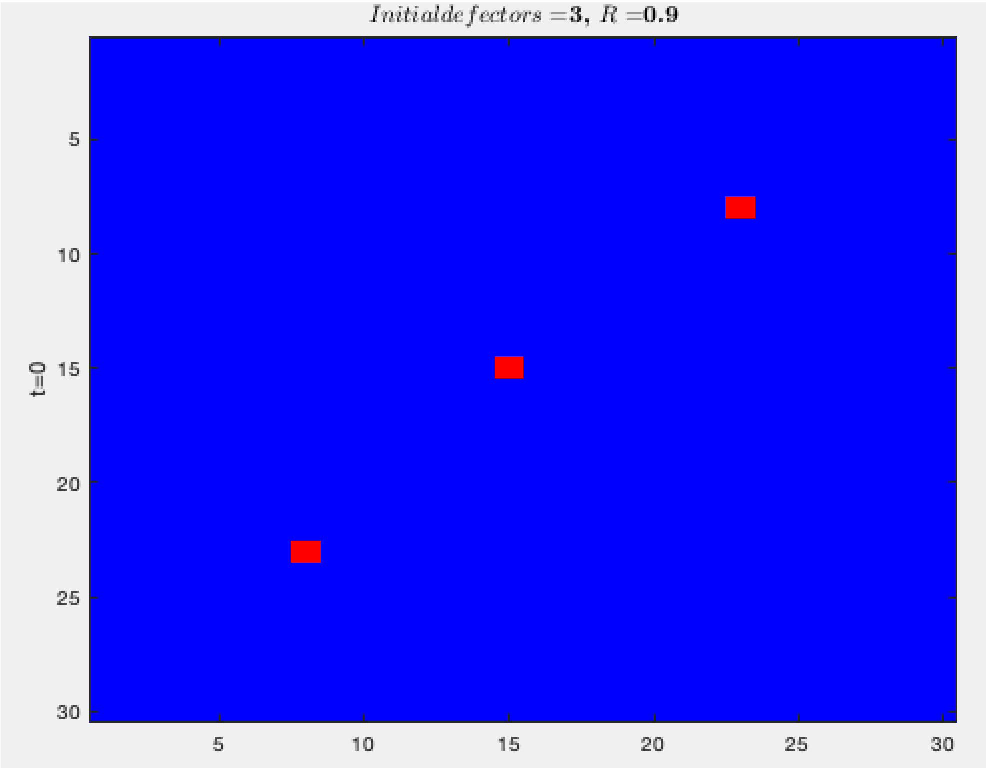
$Initial defectors = \mathbf{1},\ R = \mathbf{0.94}$
t=20
5
10
15
20
25
30
5
10
15
20
25
30



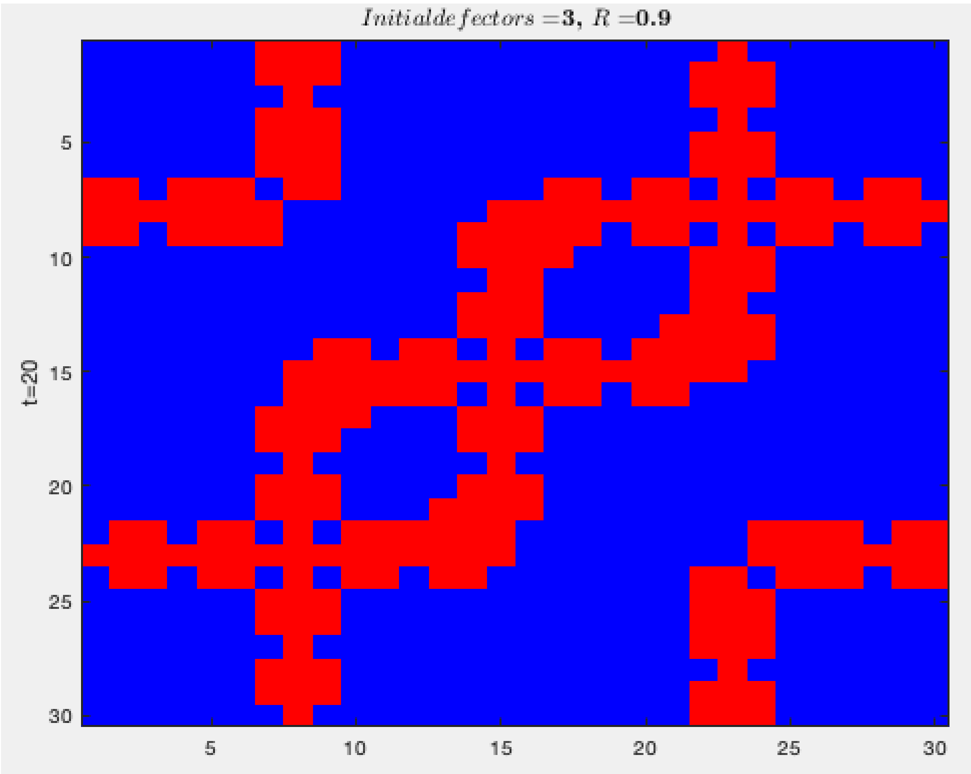
$Initial defectors = \mathbf{1},\ R = \mathbf{0.95}$
t=20
5
10
15
20
25
30
5
10
15
20
25
30


$Initial defectors = \mathbf{2},\ R = \mathbf{0.9}$
t=0
5
10
15
20
25
30

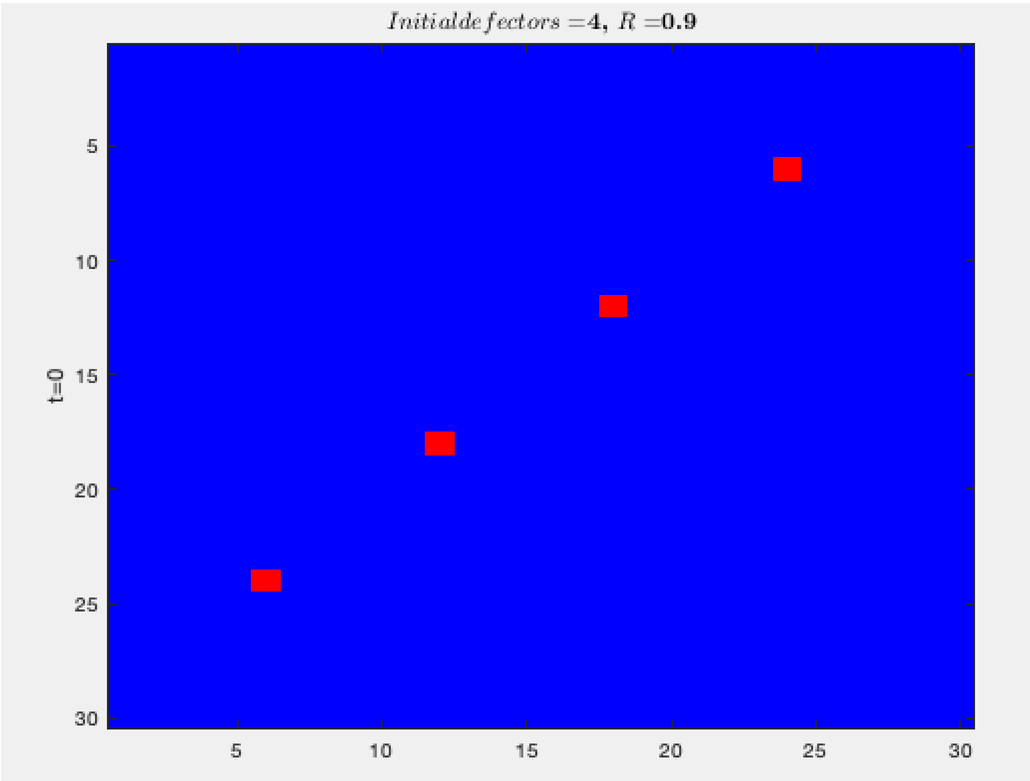

$Initial defectors = \mathbf{2},\ R = \mathbf{0.9}$
t=20
5
10
15
20
25
30



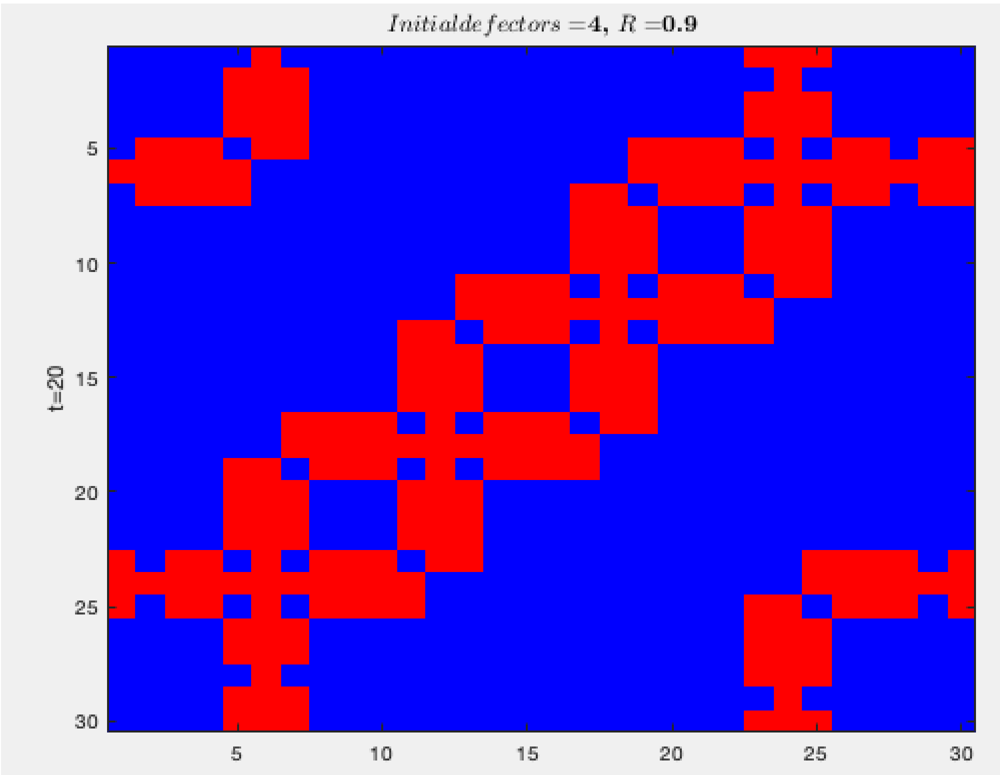
$Initial defectors = \mathbf{3},\ R = \mathbf{0.9}$
t=0
5
10
15
20
25
30
5
10
15
20
25
30



$Initial defectors = \mathbf{3},\ R = \mathbf{0.9}$
t=20
5
10
15
20
25
30
5
10
15
20
25
30



$Initial defectors = \mathbf{4},\ R = \mathbf{0.9}$
t=0



$Initial defectors = \mathbf{4},\ R = \mathbf{0.9}$
t=20

Exercise 13.3. Defectors vs cooperators. Simulate the prisoner's dilemma on an L × L lattice using only two strategies: always cooperate (n = N) and always defect (n = 0). Use a small but non-zero mutation rate, such as μ = 0.01. As usual, use T = 0 and P = 1. In addition, you can fix S = 1.5 and just play with the parameter R.

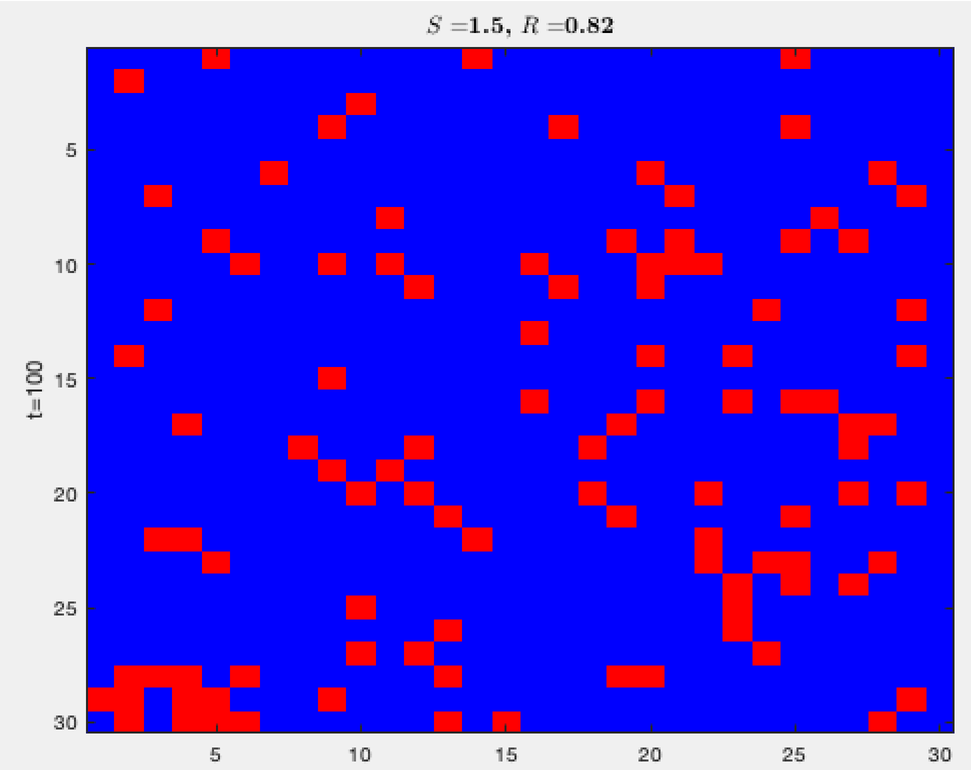
a. With R = 0.82, show that cooperation dominates, as shown in figure 13.6(a).

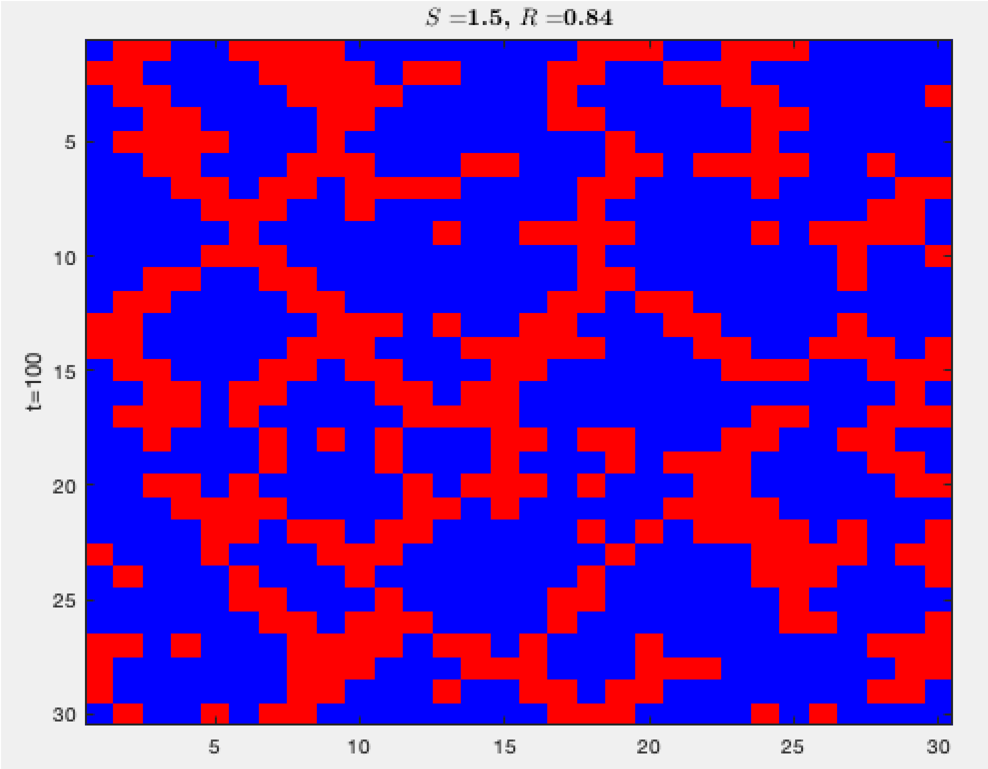
b. With R = 0.84, show that cooperation coexists with defection, as shown in figure 13.6(b).

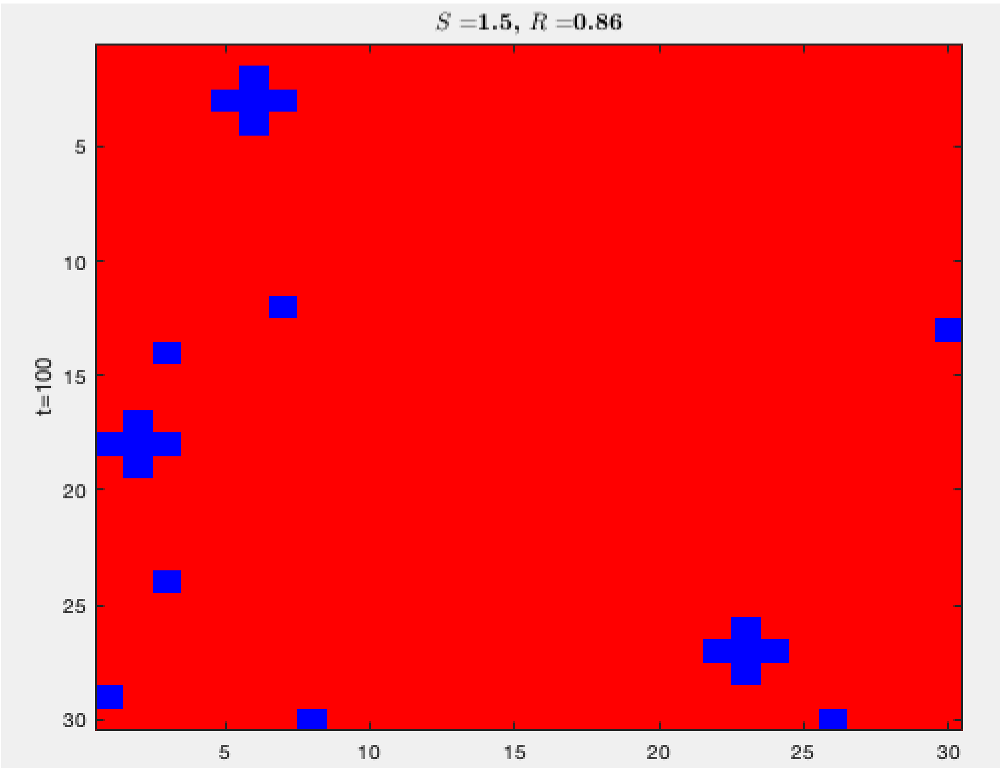
c. With R = 0.86, show that cooperation vanishes (figure 13.6(c)).

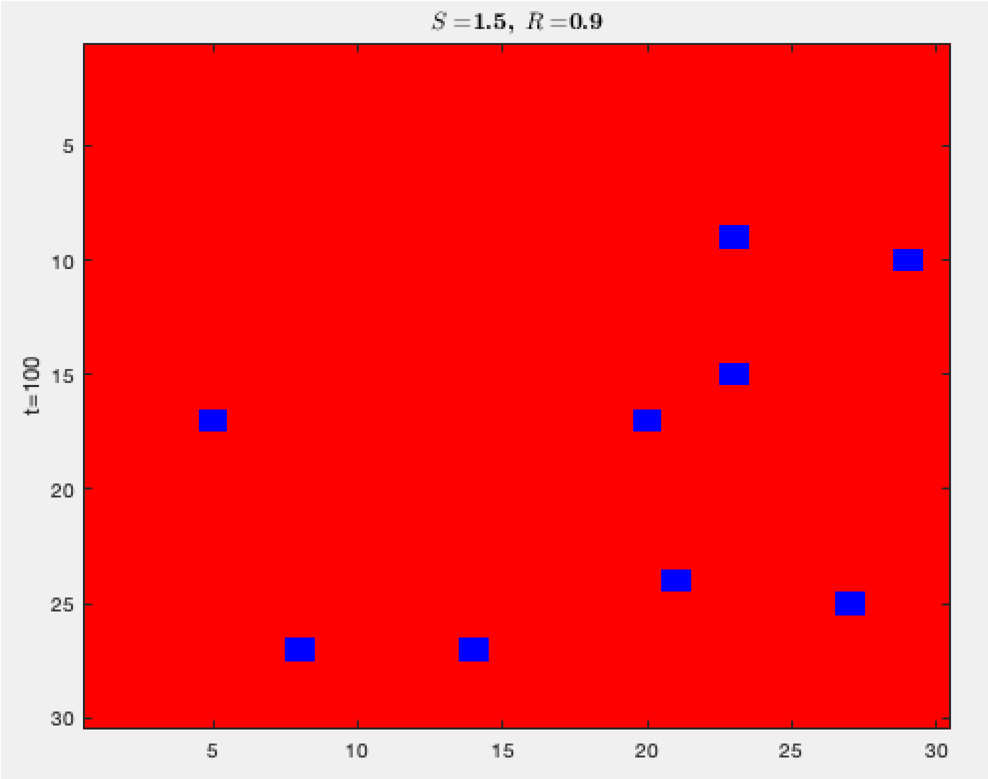
d. Repeat your simulations for a range of R to show that this behavior is critical. Determine the critical values of R that lead to each regime.

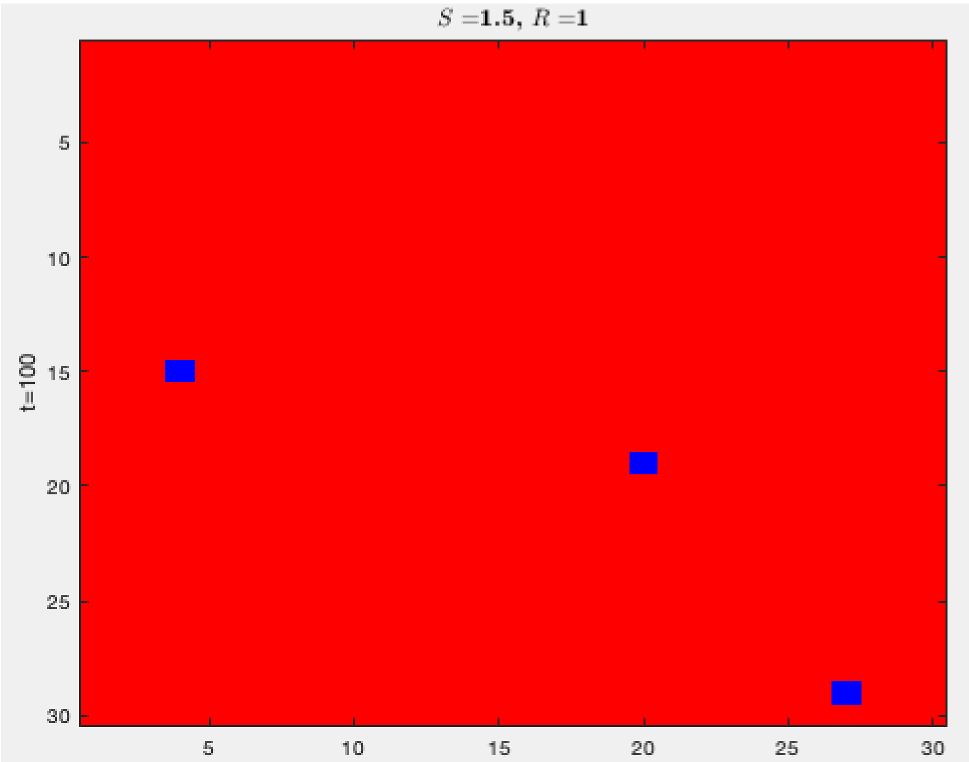
e. Repeat the same analysis, fixing the value of R and varying S.

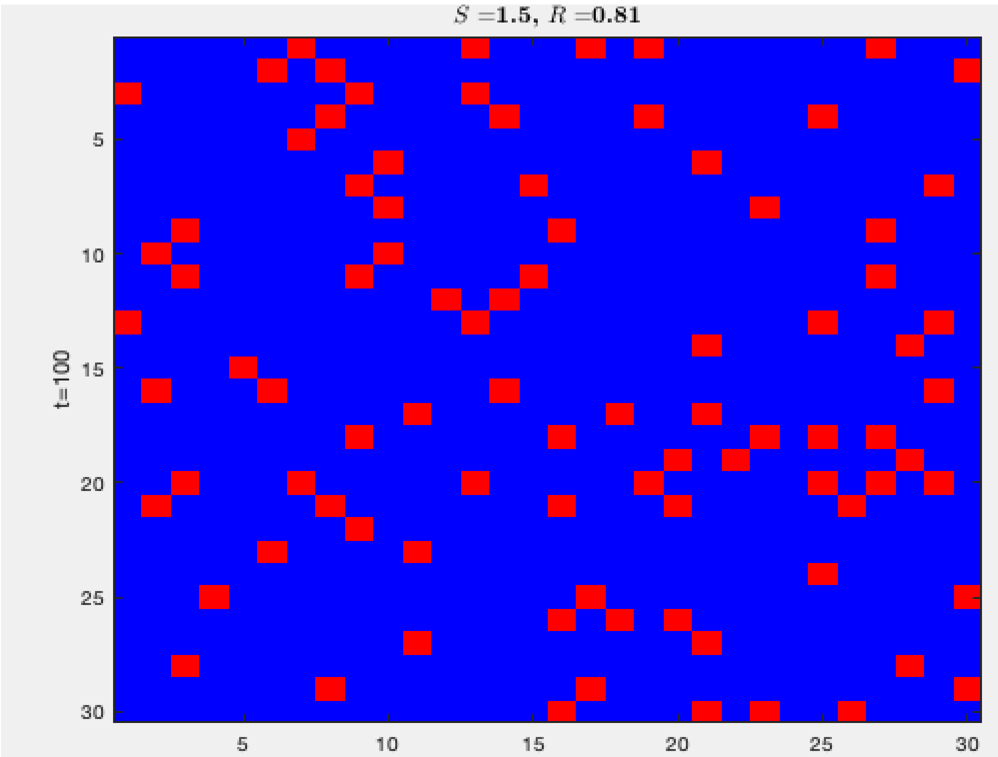


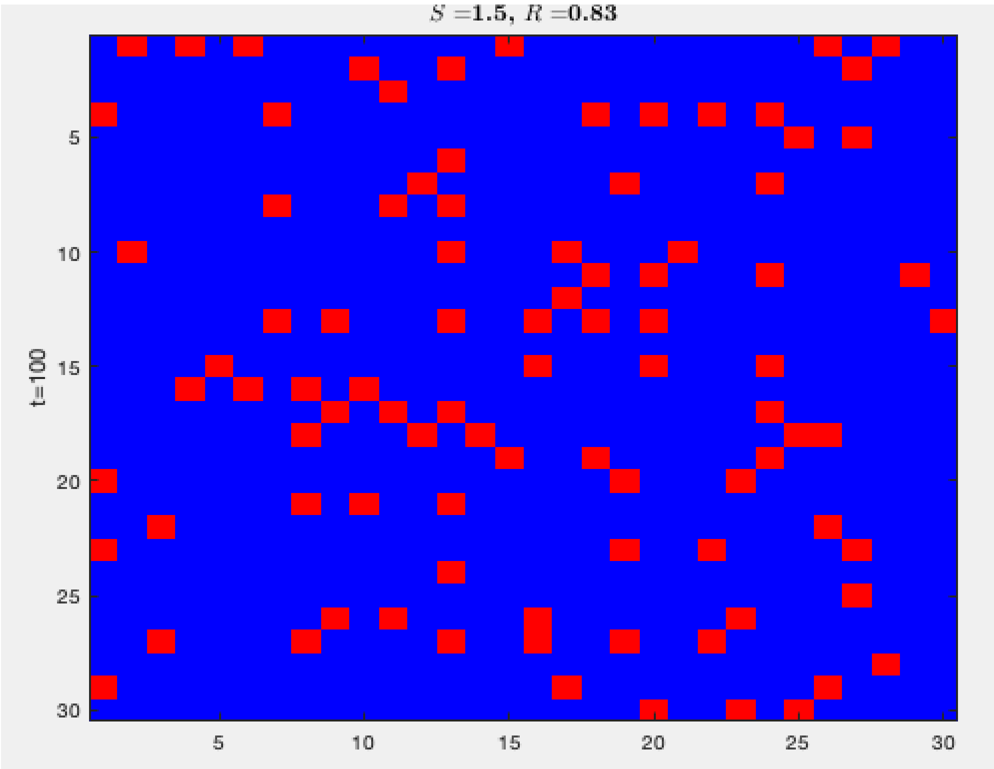

S =1.5, R =0.84
t=100
5
10
15
20
25
30

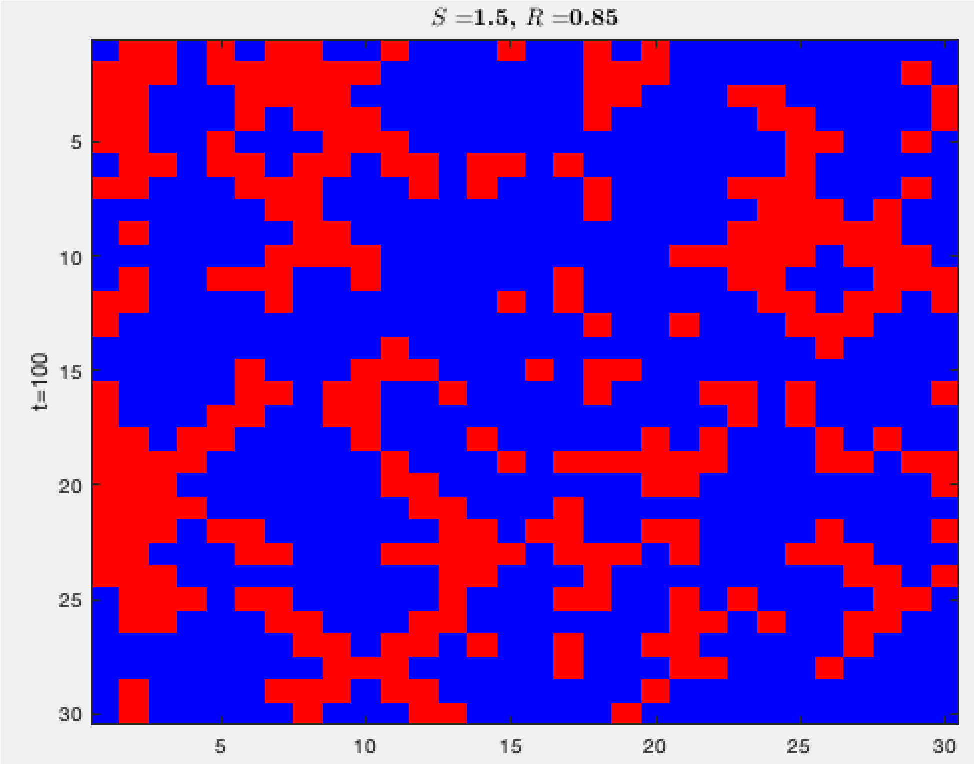

S =1.5, R =0.86
t=100
5
10
15
20
25
30

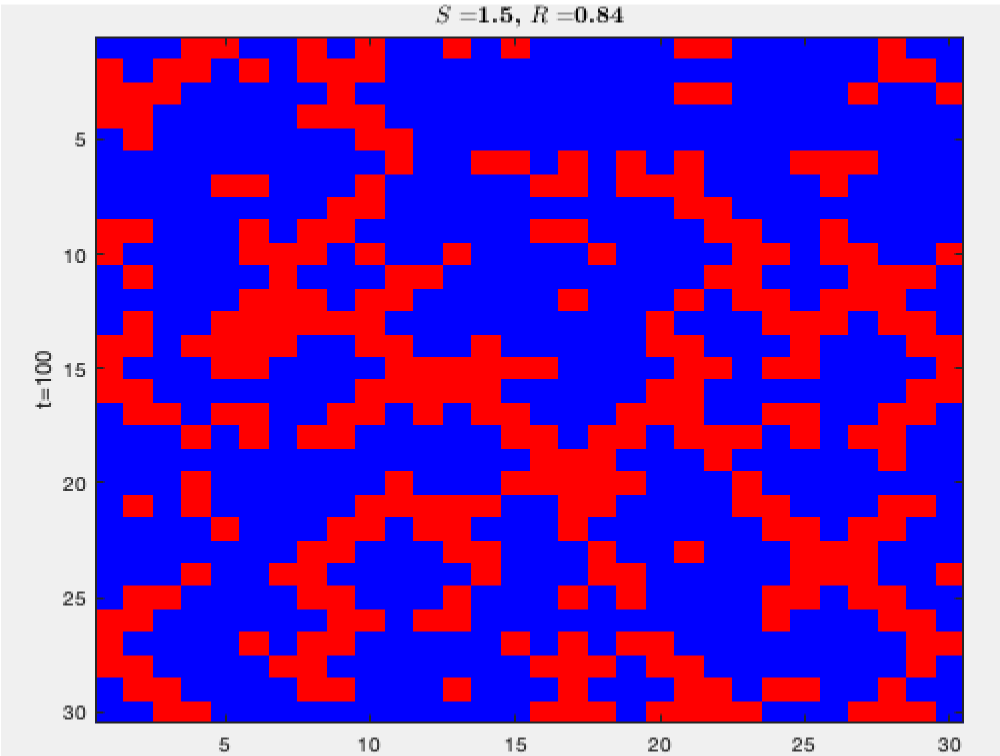

$S = \mathbf{1.5},\ R = \mathbf{0.9}$
t=100
5
10
15
20
25
30
5
10
15
20
25
30

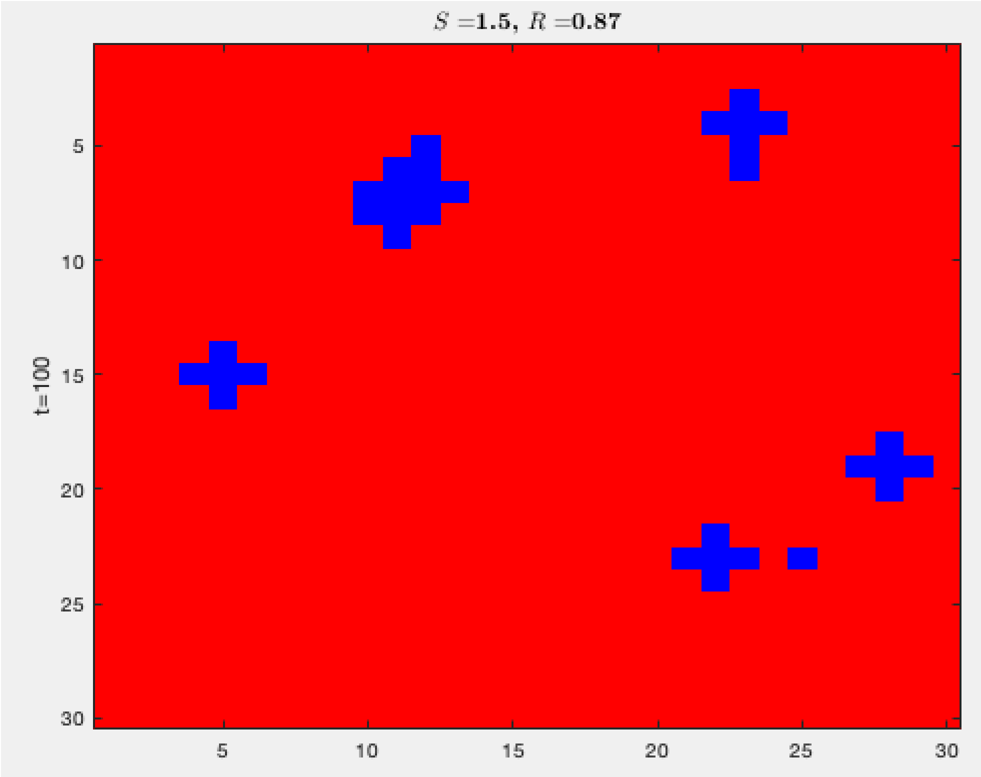

$S = \mathbf{1.5},\ R = \mathbf{1}$
t=100
5
10
15
20
25
30
5
10
15
20
25
30


S =1.5, R =0.81
t=100

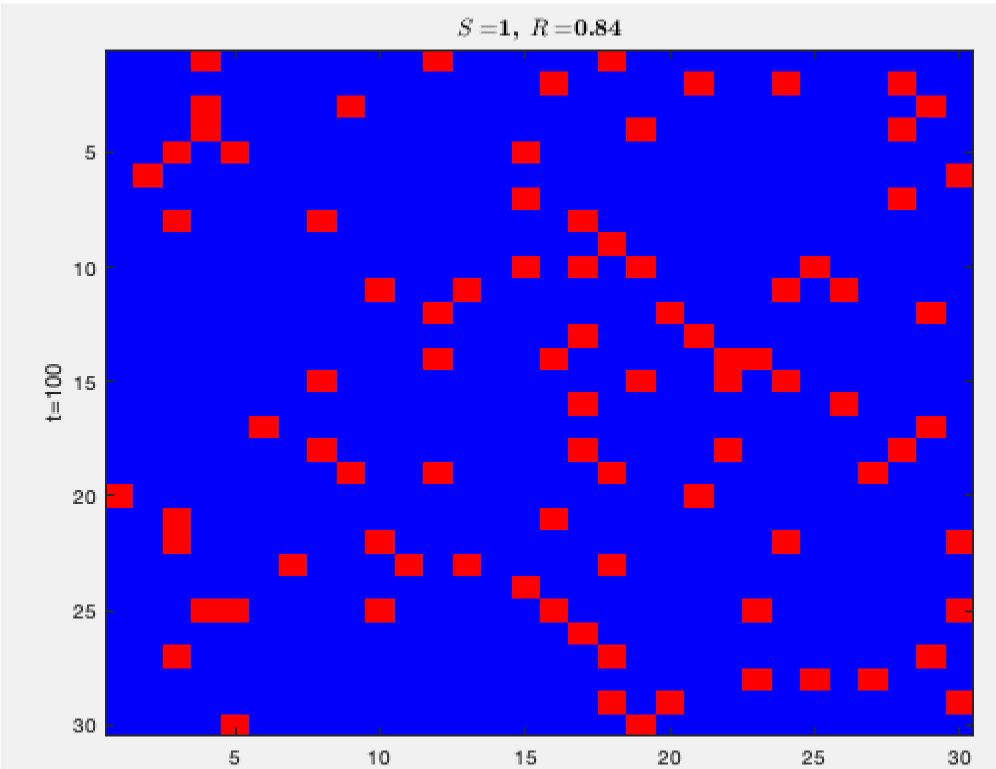

S =1.5, R =0.83
t=100


S =1.5, R =0.85
t=100
5
10
15
20
25
30
5
10
15
20
25
30

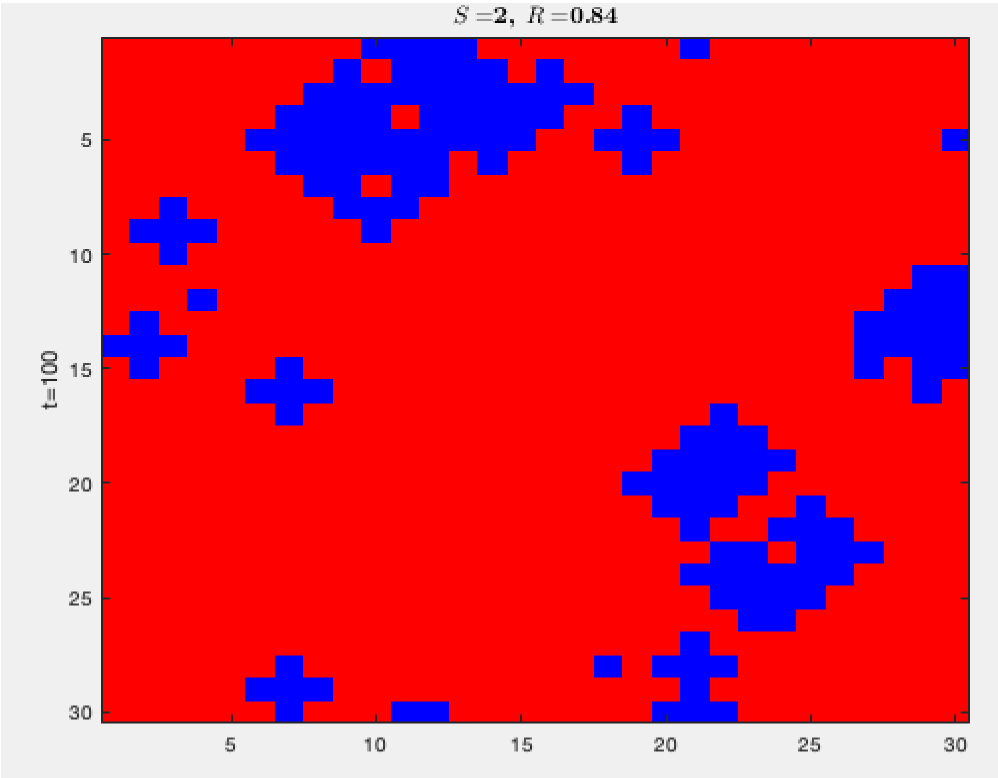

S =1.5, R =0.84
t=100
5
10
15
20
25
30
5
10
15
20
25
30

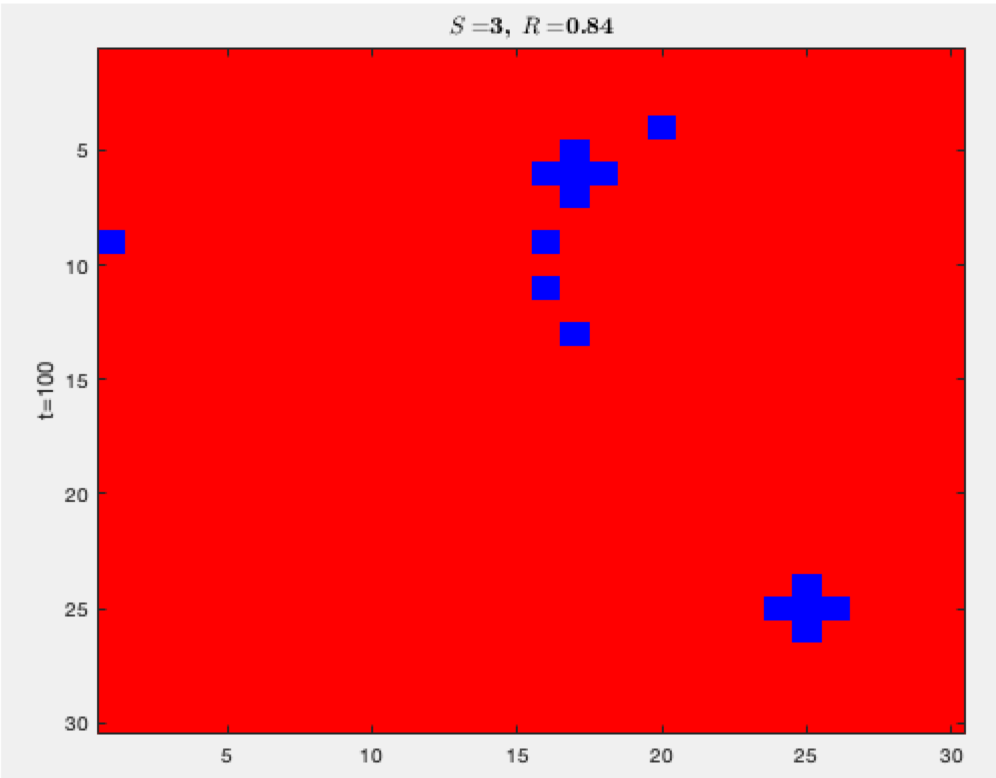


$S = \mathbf{1.5},\ R = \mathbf{0.87}$
t=100
5
10
15
20
25
30

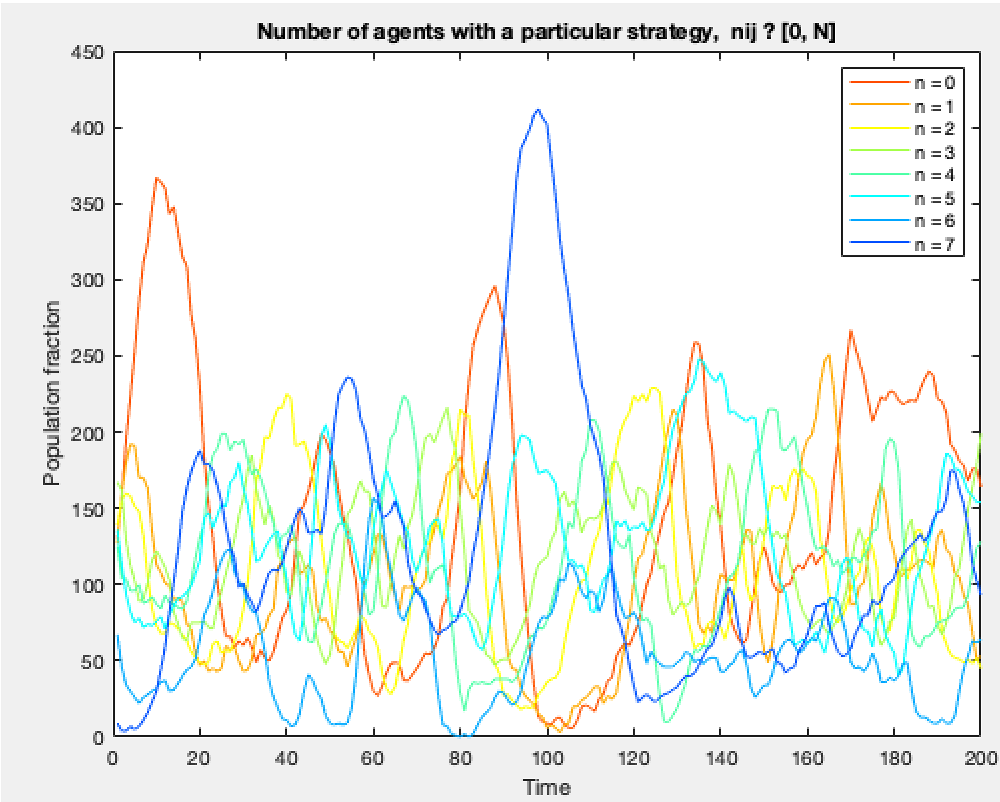


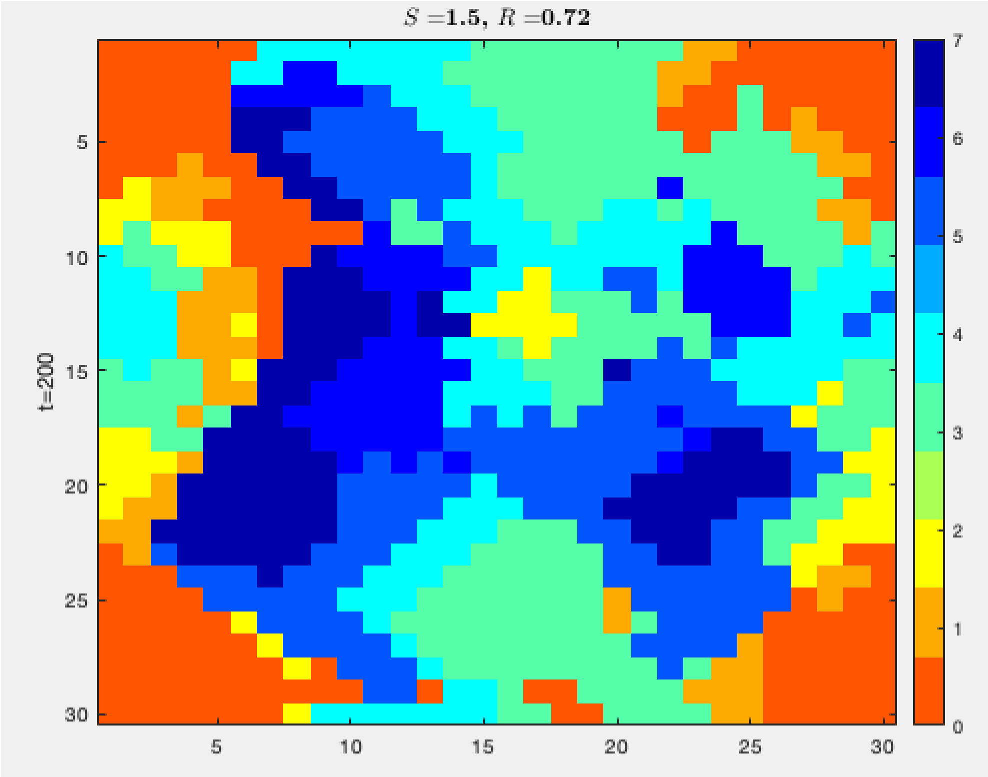
$S = \mathbf{1},\ R = \mathbf{0.84}$
t=100
5
10
15
20
25
30

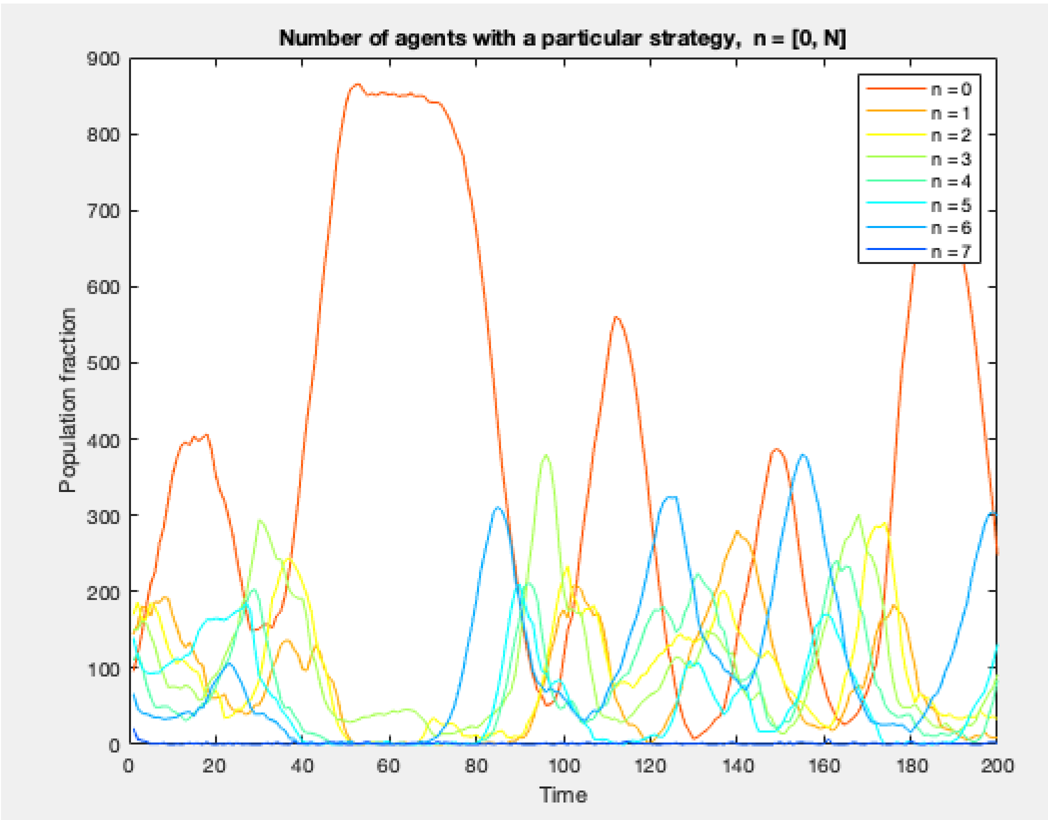

$S = \mathbf{2},\ R = \mathbf{0.84}$
t=100
5
10
15
20
25
30

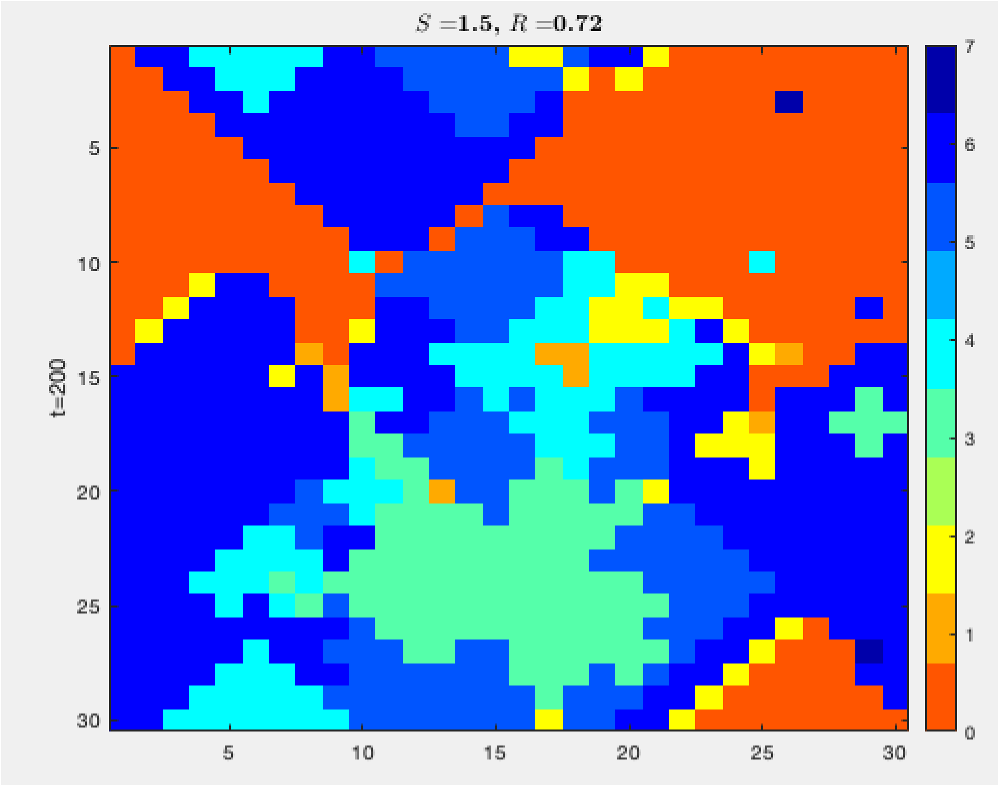

$S = \mathbf{3},\ R = \mathbf{0.84}$
t=100
5
10
15
20
25
30

Exercise 13.4. Evolutionary games on a lattice with multiple strategies. Simulate a seven-round prisoner's dilemma on an L × L lattice. Allow all strategies ($0 \leqslant n \leqslant 7$). Use a small but non-zero mutation rate, such as $\mu = 0.01$. As usual, use $T = 0$ and $P = 1$. Depending on the parameters R and S, multiple regimes can be observed.
a. Fix S = 1.5 and play only with parameter R. Observe what happens within the lattice as time evolves. Show that different strategies may emerge and propagate, as shown in figure 13.7(a).
b. Show that there are three main regimes that are similar to those identified in exercise 13.3. However, this time, the regime in which different strategies coexist can have different dominant populations, depending on the value of R. Discover the parameters and the population distribution.
c. Discuss the results of your simulation.What do these numerical outcomes tell us about the evolution of cooperation? Which strategies are stable evolutionary strategies? - higher R -> strategy 0 and lower R -> strategy N

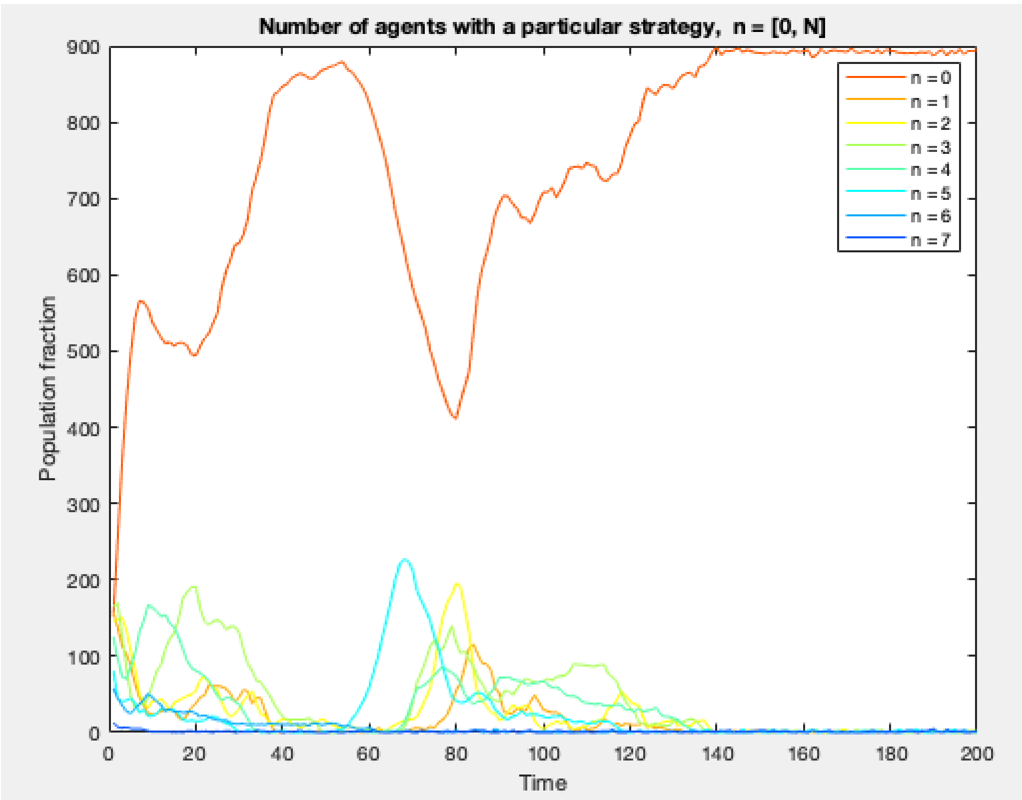



S =1.5, R =0.72
t=200
5
10
15
20
25
30
0
1
2
3
4
5
6
7

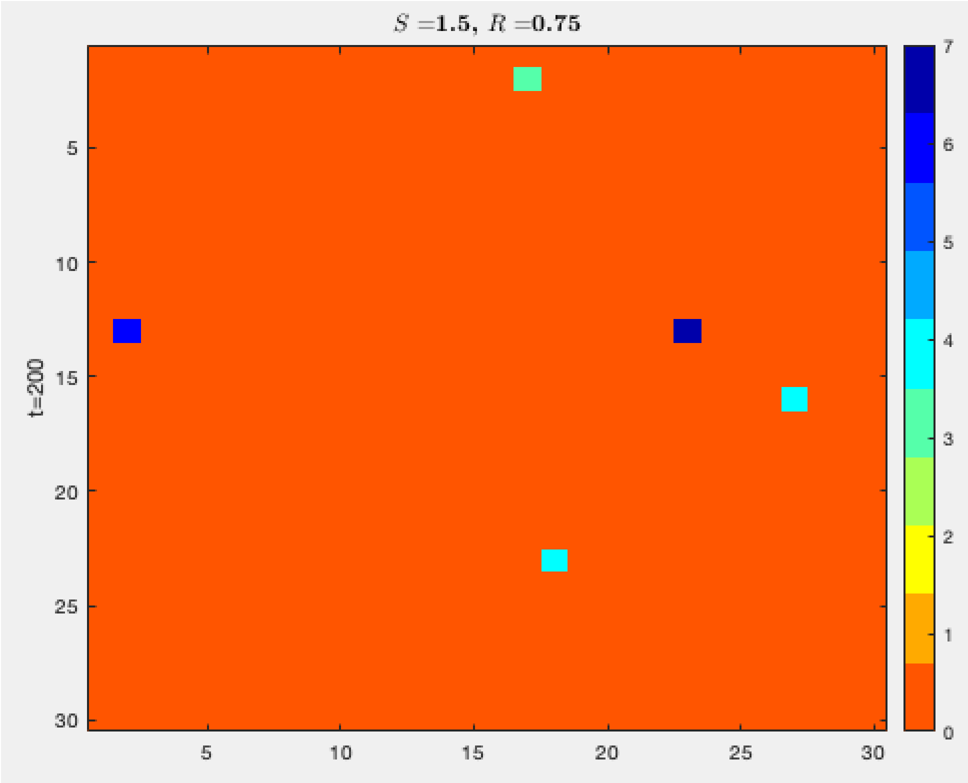

Number of agents with a particular strategy, n = [0, N]
n = 0
n = 1
n = 2
n = 3
n = 4
n = 5
n = 6
n = 7
Population fraction
Time
0
100
200
300
400
500
600
700
800
900
20
40
60
80
100
120
140
160
180
200



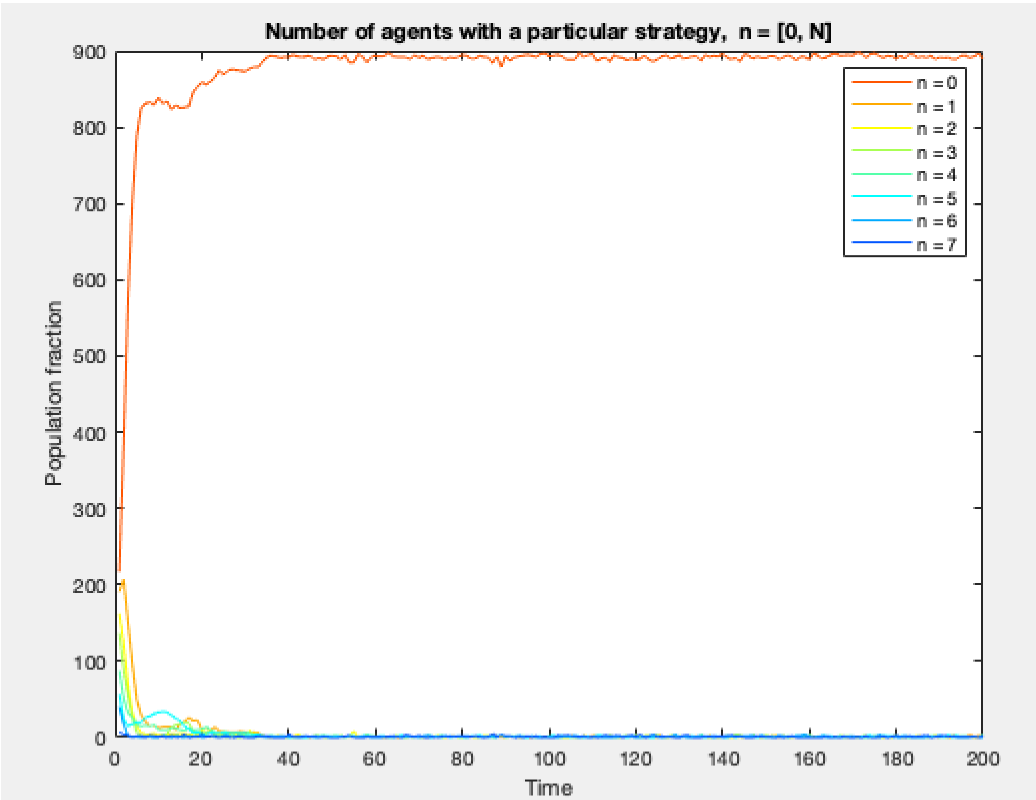
S =1.5, R =0.72
t=200



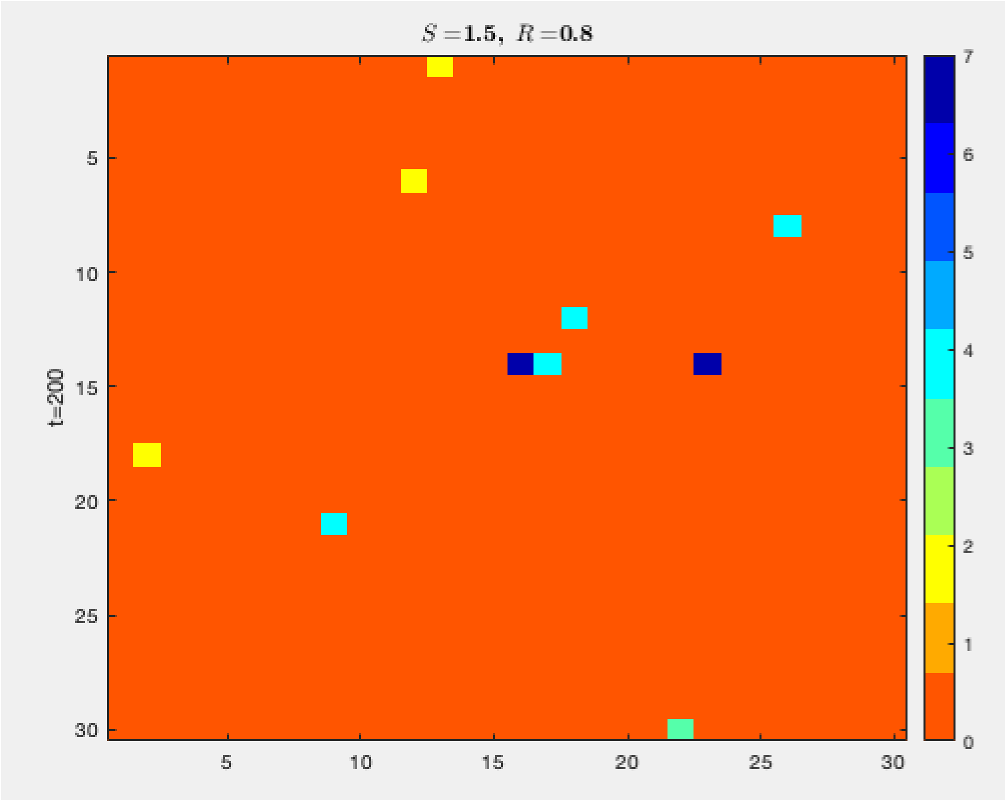
Number of agents with a particular strategy, n = [0, N]
Population fraction
Time
n = 0
n = 1
n = 2
n = 3
n = 4
n = 5
n = 6
n = 7



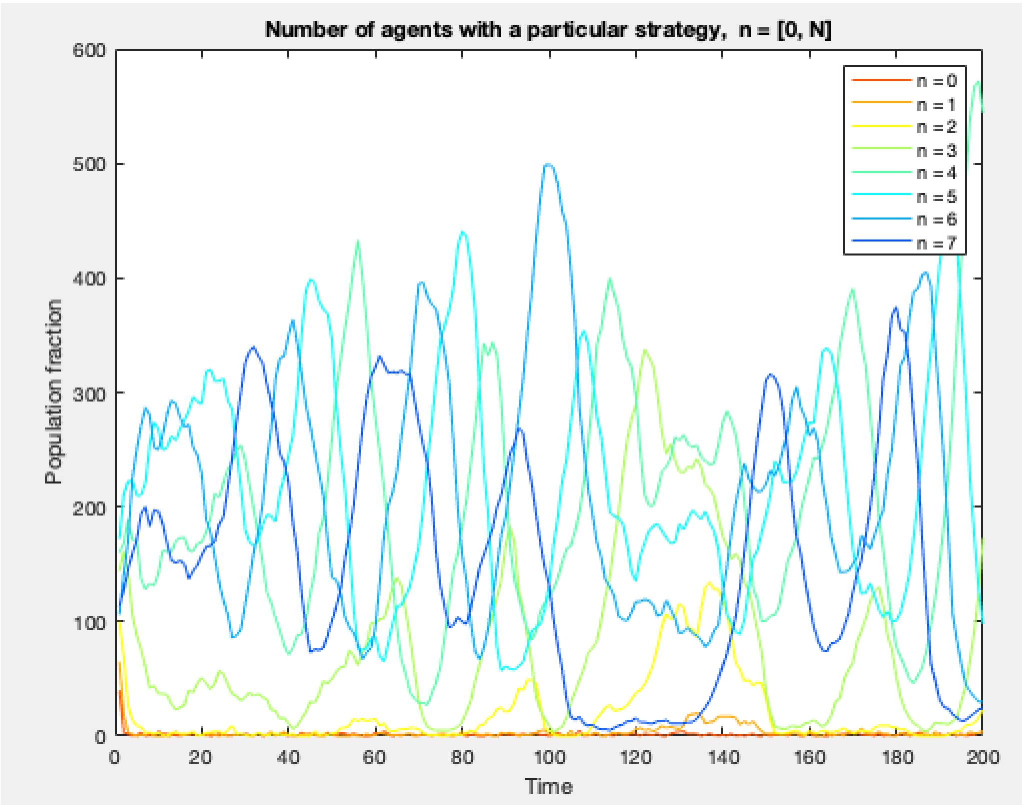
S =1.5, R =0.75
t=200



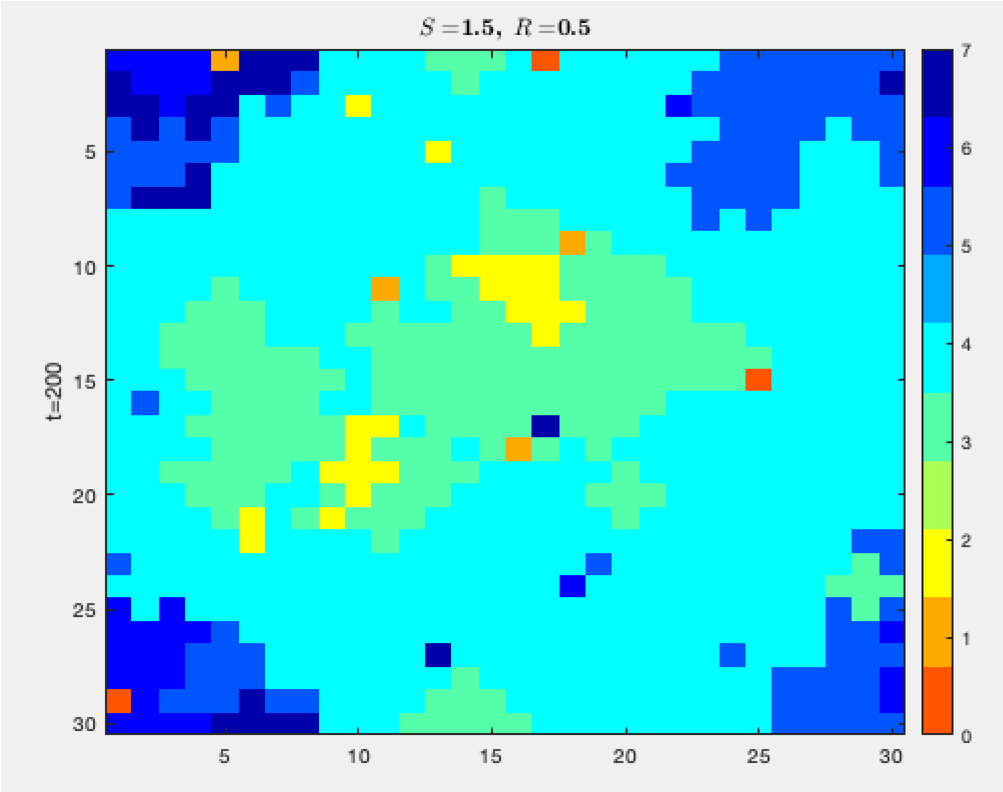
Number of agents with a particular strategy, n = [0, N]
Population fraction
Time
n = 0
n = 1
n = 2
n = 3
n = 4
n = 5
n = 6
n = 7



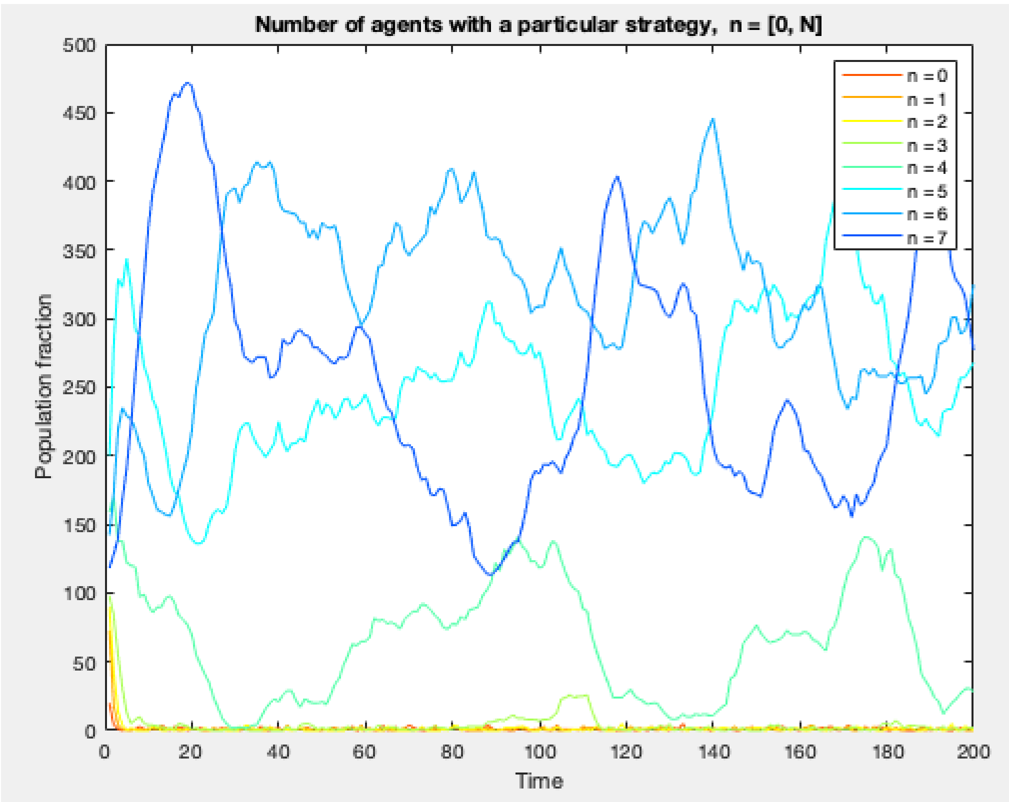
$S = 1.5, \; R = 0.8$
t=200



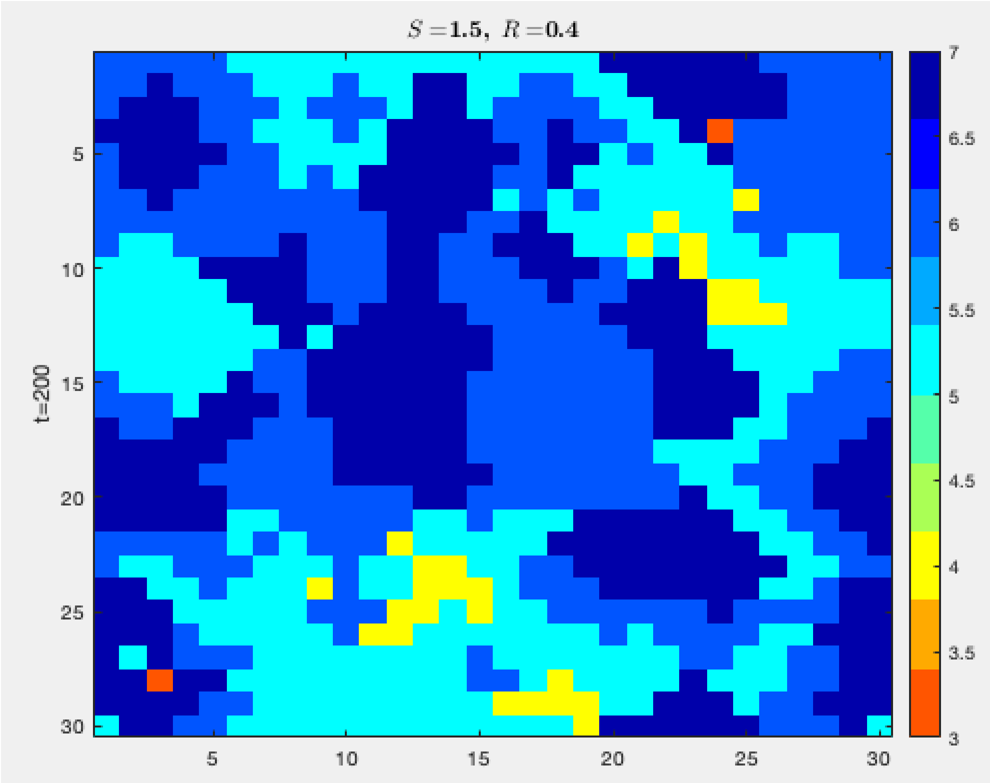
Number of agents with a particular strategy, n = [0, N]
Population fraction
Time
n = 0
n = 1
n = 2
n = 3
n = 4
n = 5
n = 6
n = 7

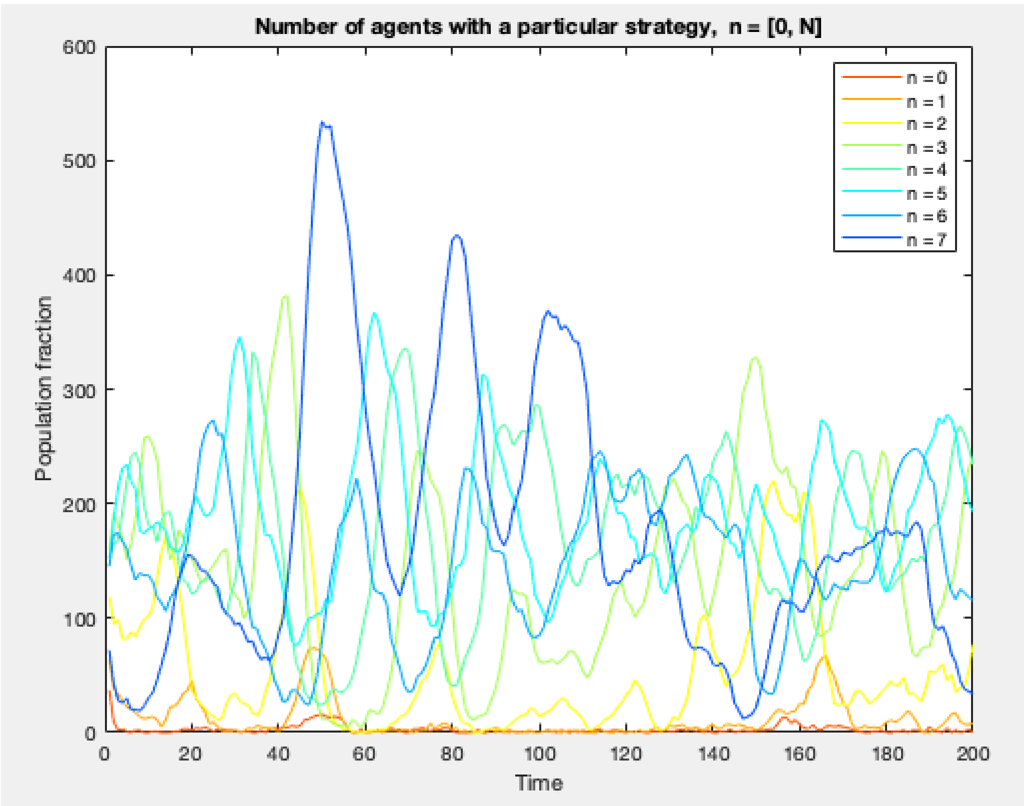


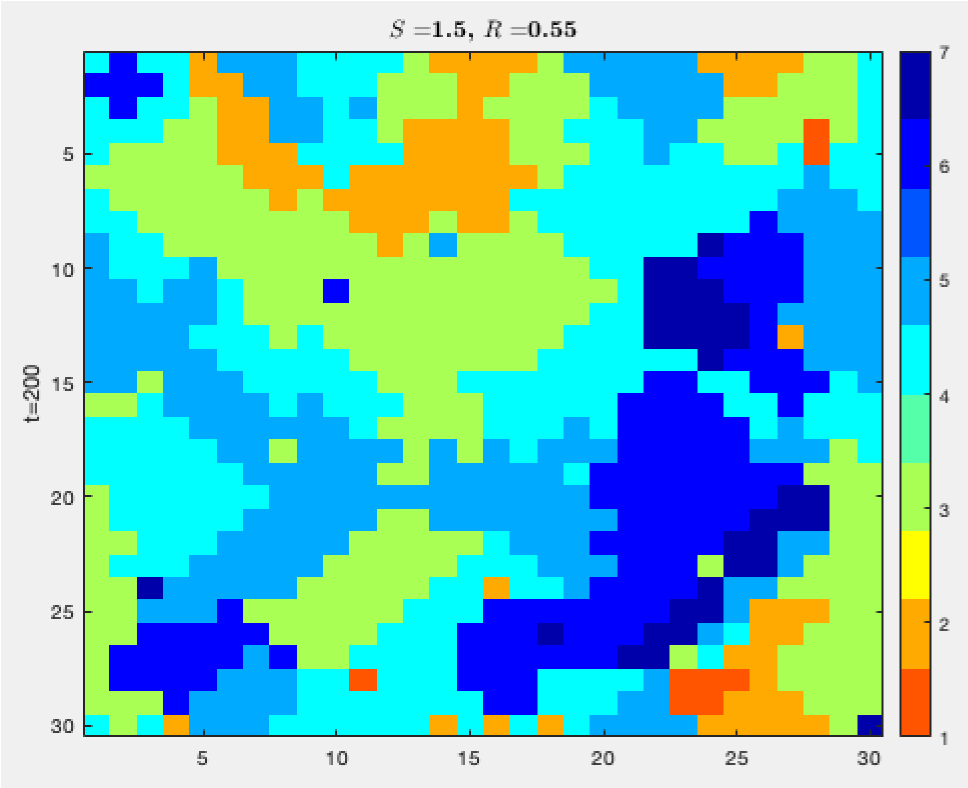
$S = 1.5,\ R = 0.5$
t=200



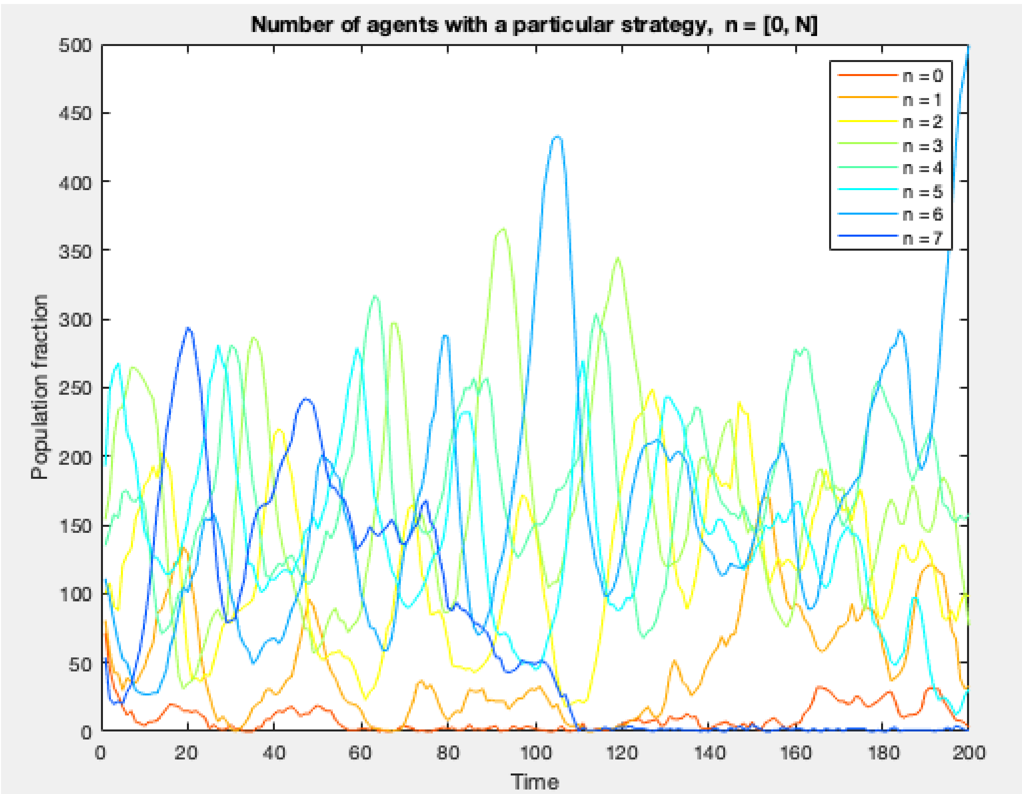
Number of agents with a particular strategy, n = [0, N]
Population fraction
Time
n = 0
n = 1
n = 2
n = 3
n = 4
n = 5
n = 6
n = 7


$S = \mathbf{1.5},\ R = \mathbf{0.4}$
t=200

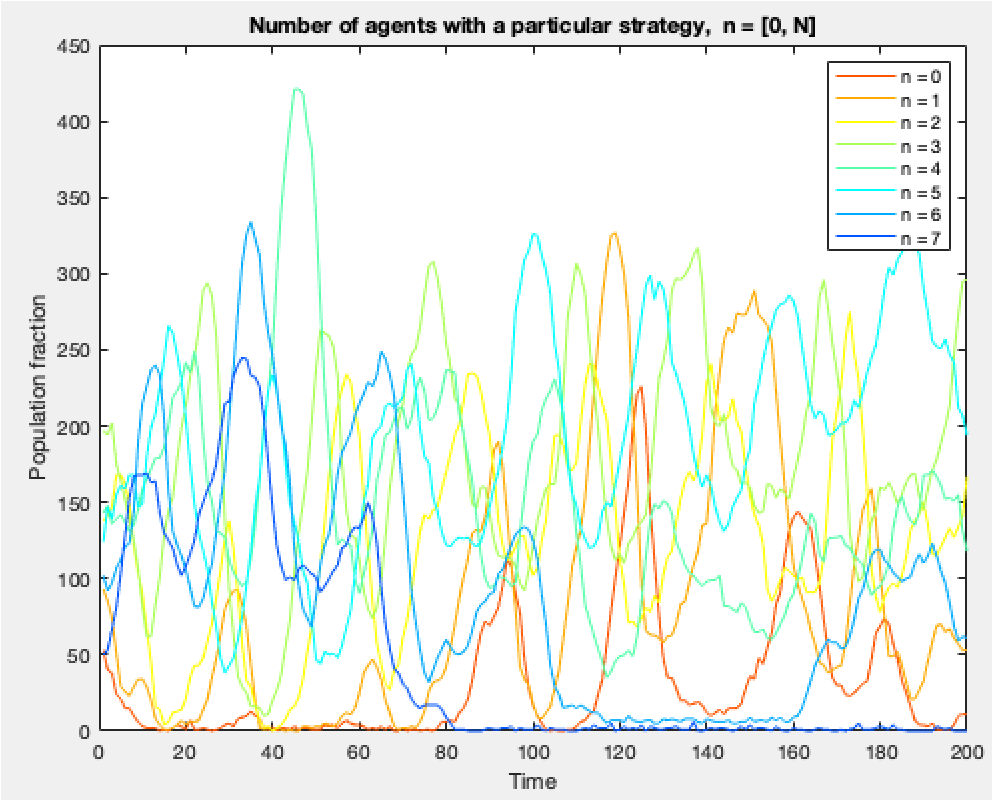

Number of agents with a particular strategy, n = [0, N]
n = 0
n = 1
n = 2
n = 3
n = 4
n = 5
n = 6
n = 7
Population fraction
Time

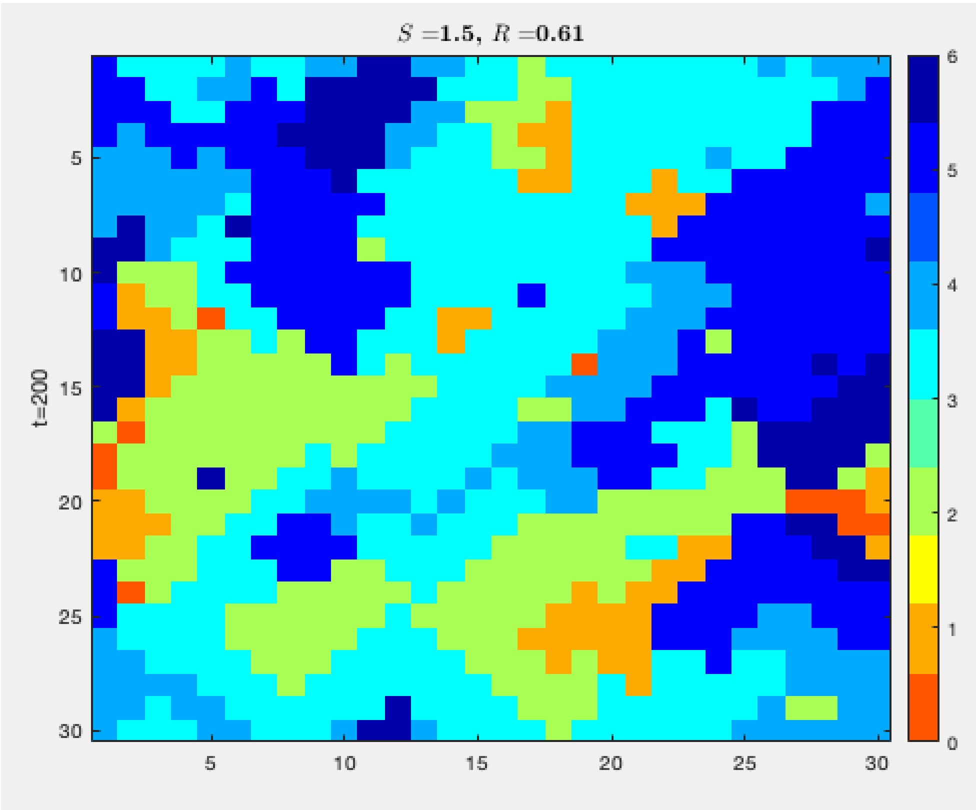


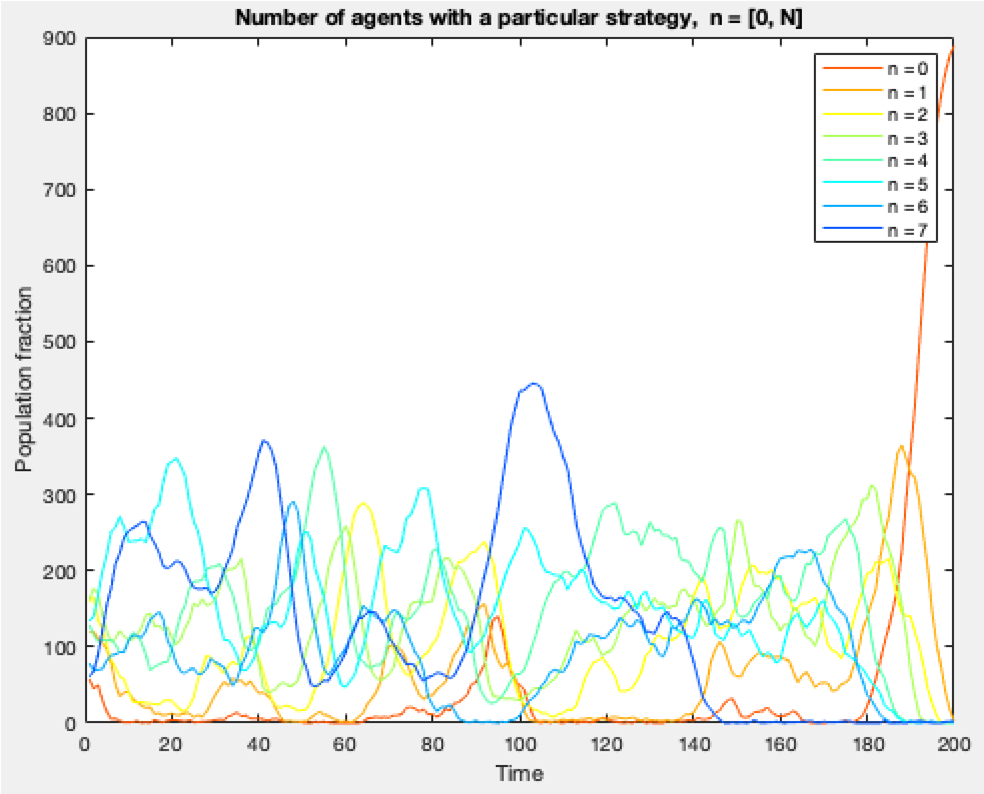
S =1.5, R =0.55
t=200

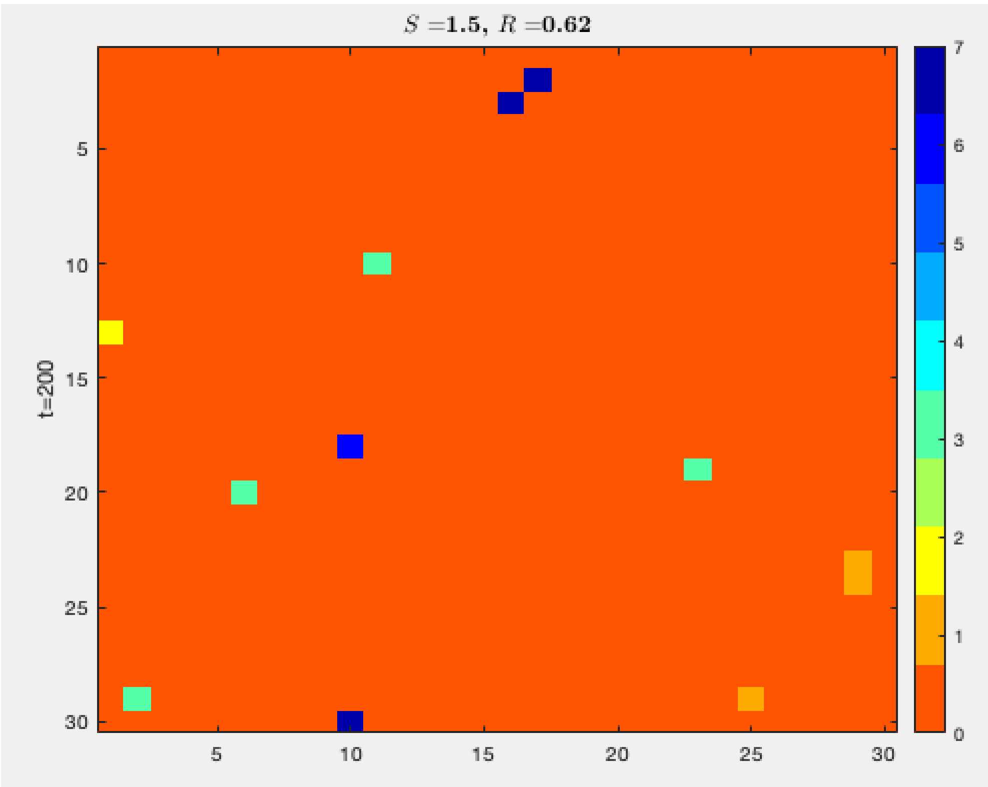


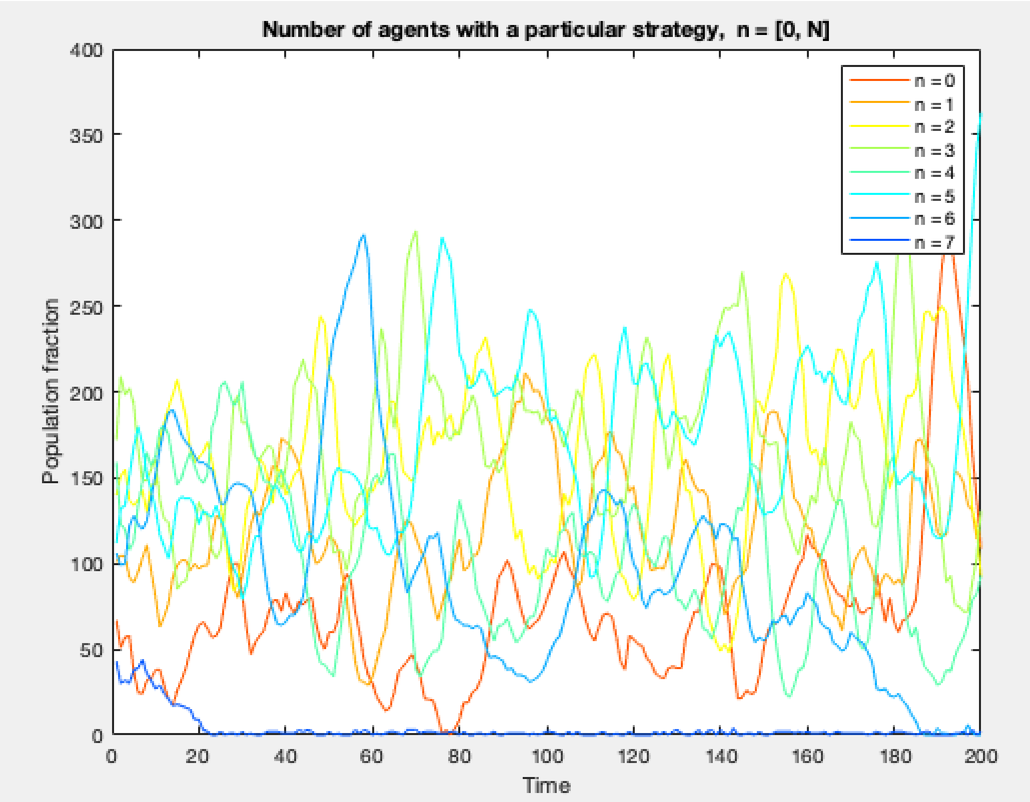
Number of agents with a particular strategy, n = [0, N]
Population fraction
Time
n = 0
n = 1
n = 2
n = 3
n = 4
n = 5
n = 6
n = 7

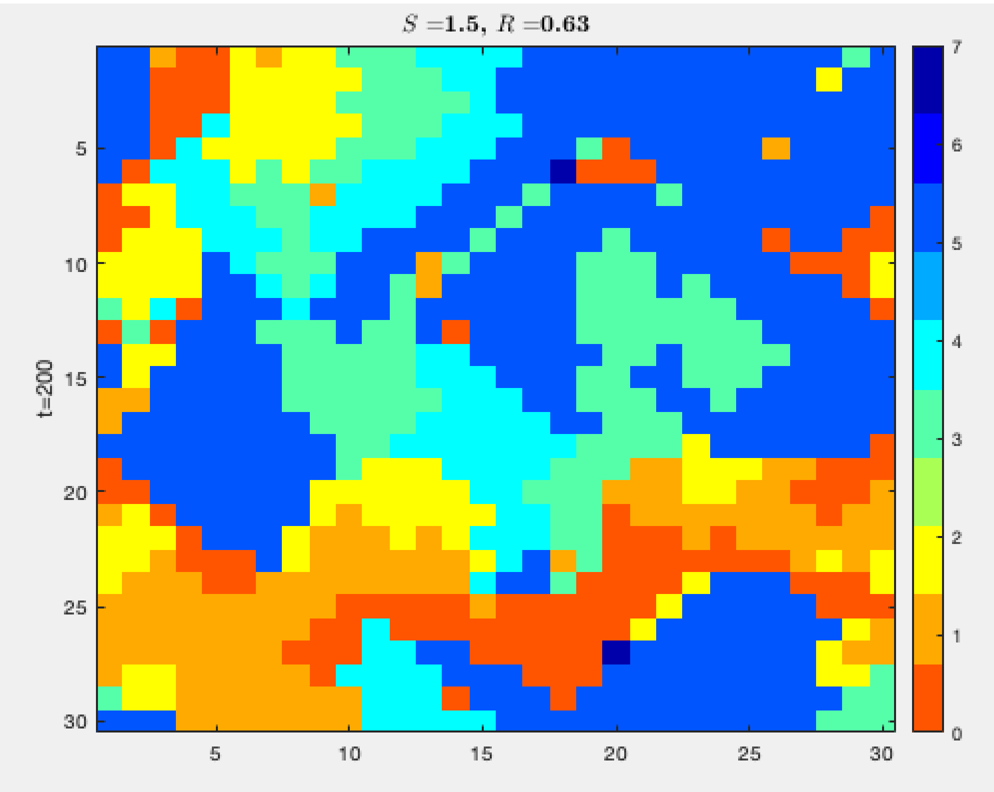

Number of agents with a particular strategy, n = [0, N]
n = 0
n = 1
n = 2
n = 3
n = 4
n = 5
n = 6
n = 7
Population fraction
Time

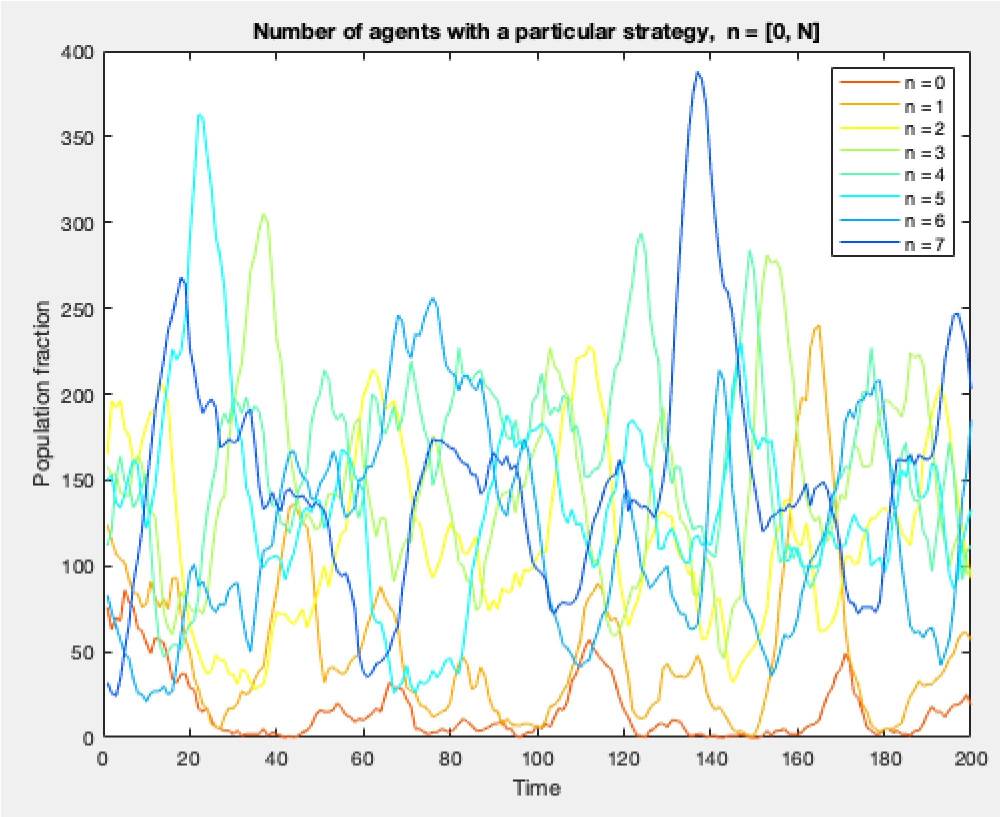

$S = 1.5$, $R = 0.61$
t=200


Number of agents with a particular strategy, n = [0, N]
Population fraction
Time
n = 0
n = 1
n = 2
n = 3
n = 4
n = 5
n = 6
n = 7

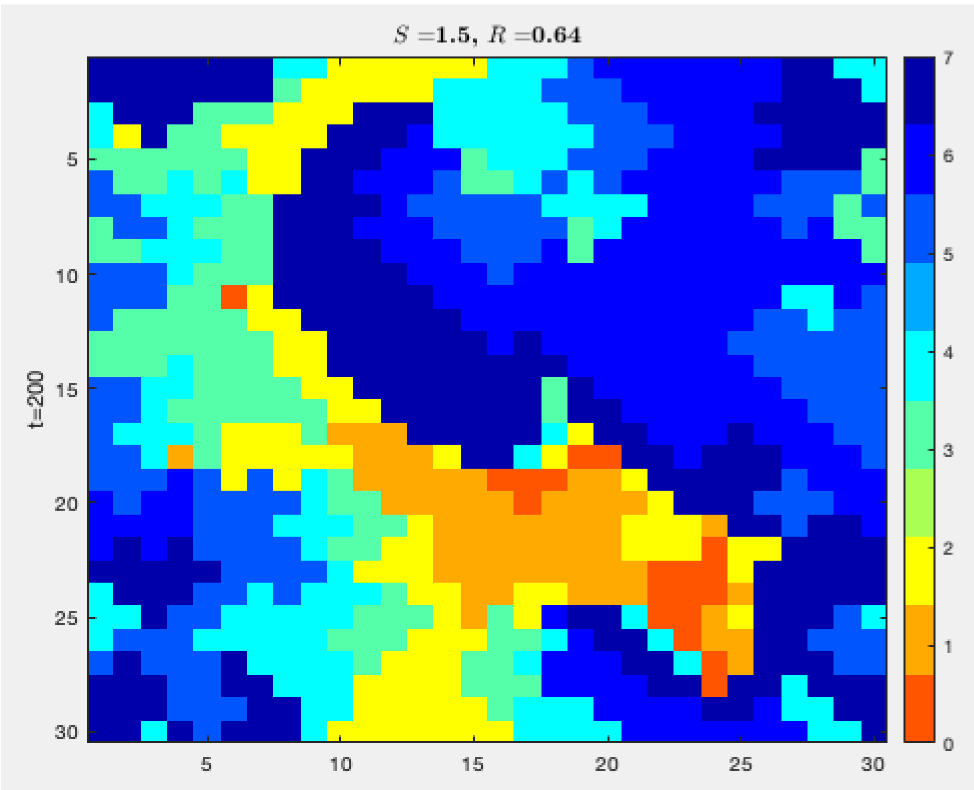

S =1.5, R =0.62
t=200


Number of agents with a particular strategy, n = [0, N]
n = 0
n = 1
n = 2
n = 3
n = 4
n = 5
n = 6
n = 7
Population fraction
Time
0
20
40
60
80
100
120
140
160
180
200
0
50
100
150
200
250
300
350
400

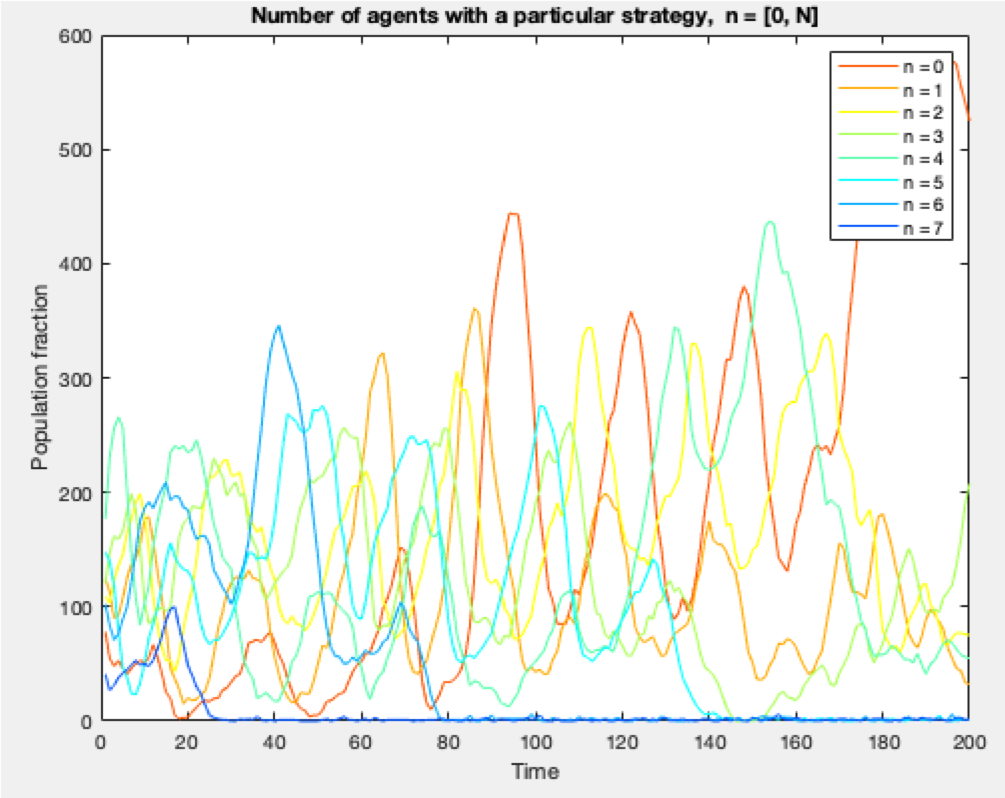

S =1.5, R =0.63
t=200
5
10
15
20
25
30
0
1
2
3
4
5
6
7

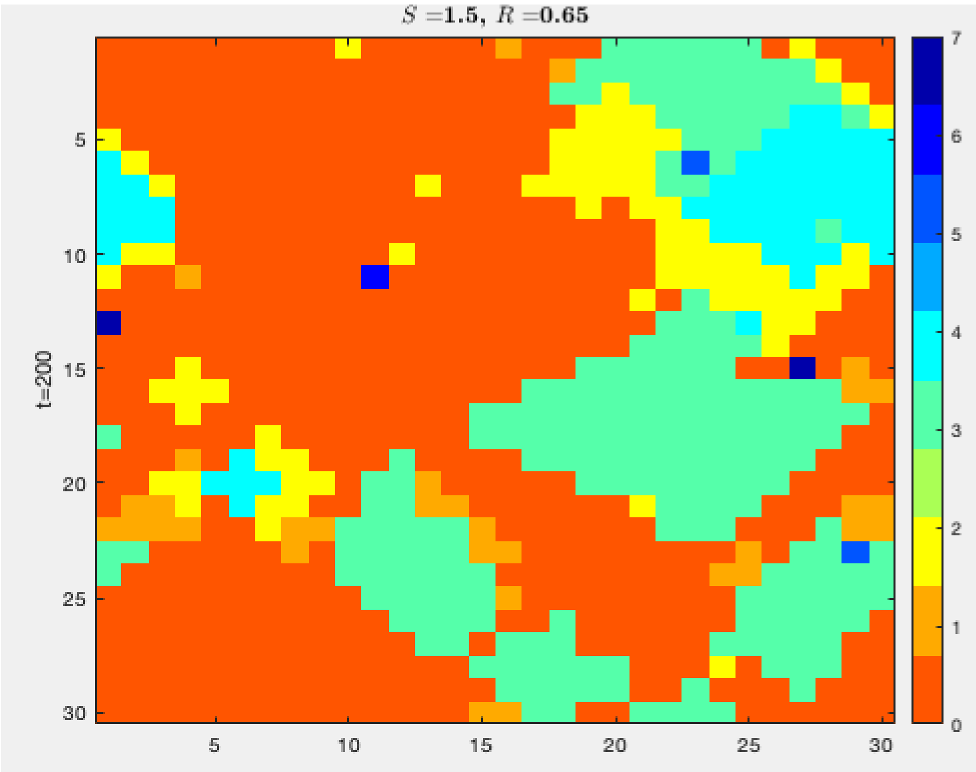


Number of agents with a particular strategy, n = [0, N]
Population fraction
Time
n = 0
n = 1
n = 2
n = 3
n = 4
n = 5
n = 6
n = 7



$S = 1.5$, $R = 0.64$
t=200

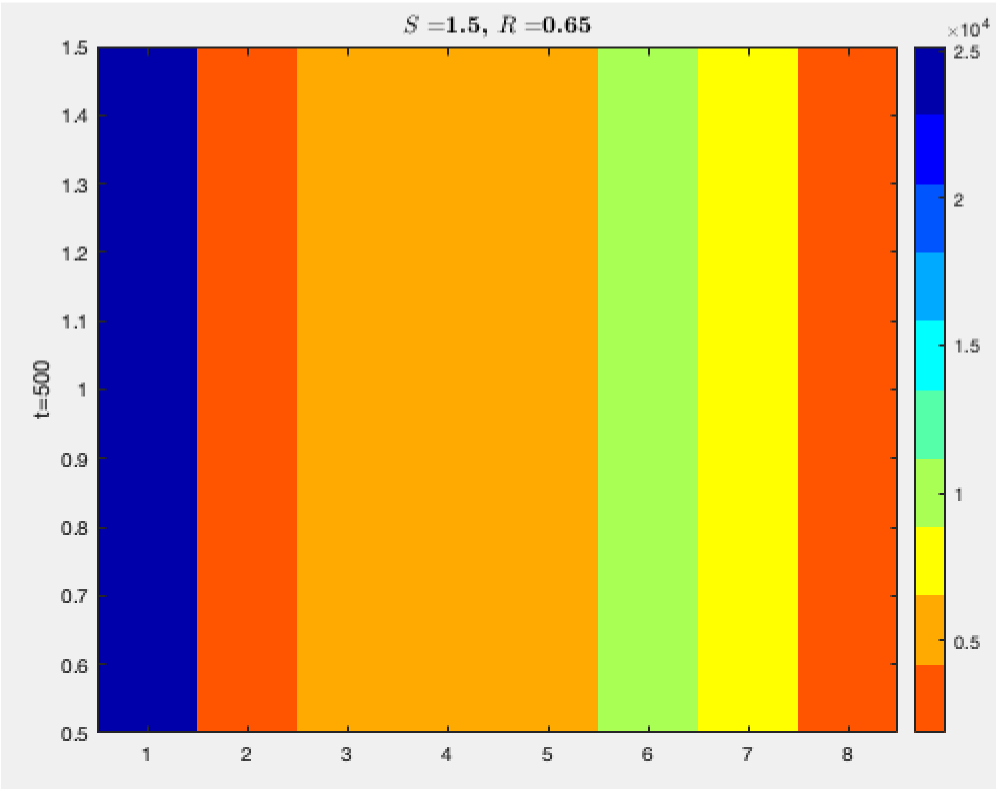

Number of agents with a particular strategy, n = [0, N]
Population fraction
Time
n = 0
n = 1
n = 2
n = 3
n = 4
n = 5
n = 6
n = 7

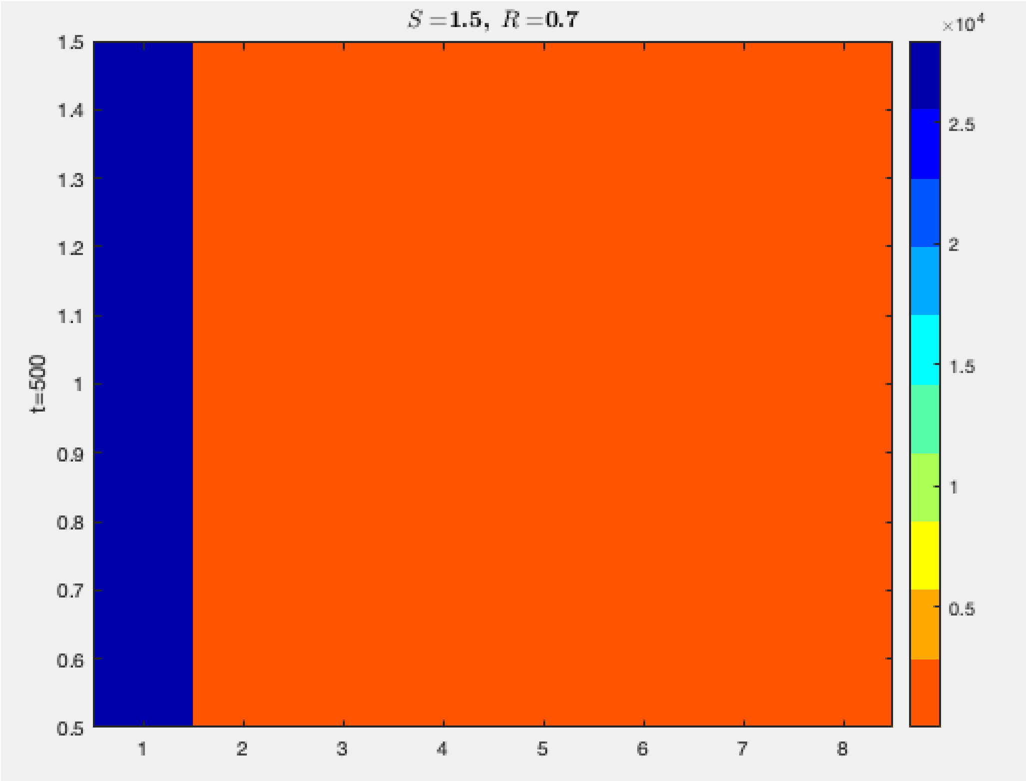

S = 1.5, R = 0.65
t=200

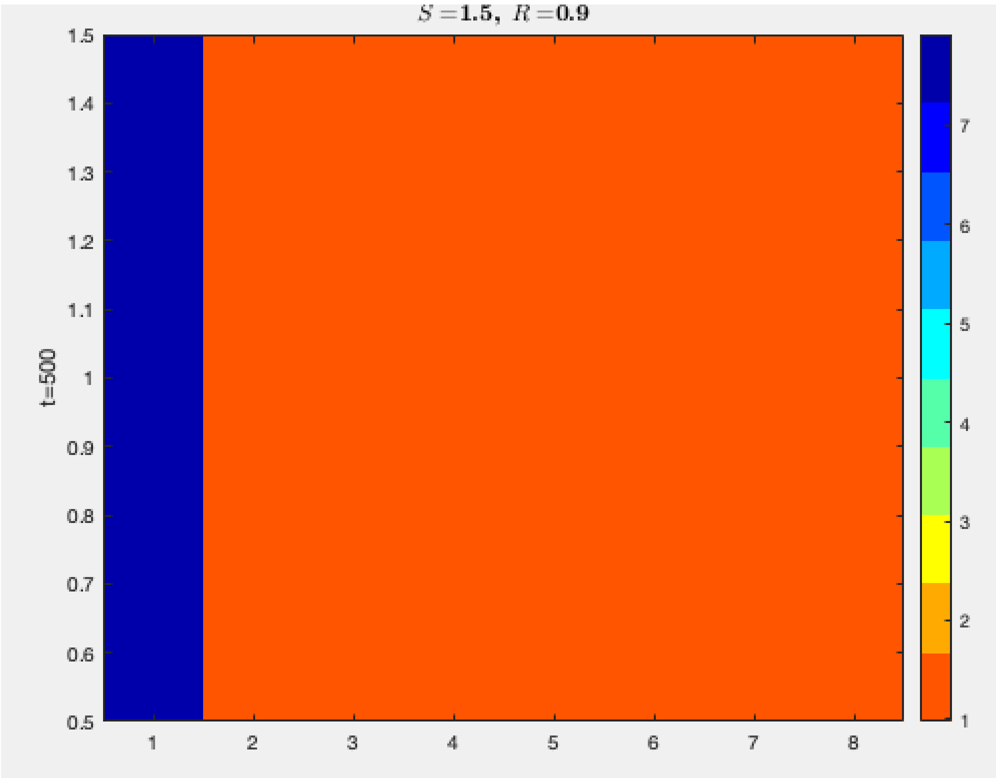
Exercise 13.5. Two-dimensional phase map of evolutionary games. For the case of multiple allowed strategies, the dynamics does not only depend on one parameter, or a simple mathematical relation between the two parameters R and S. Instead, the output behavior is rather complex and depends on both parameters.

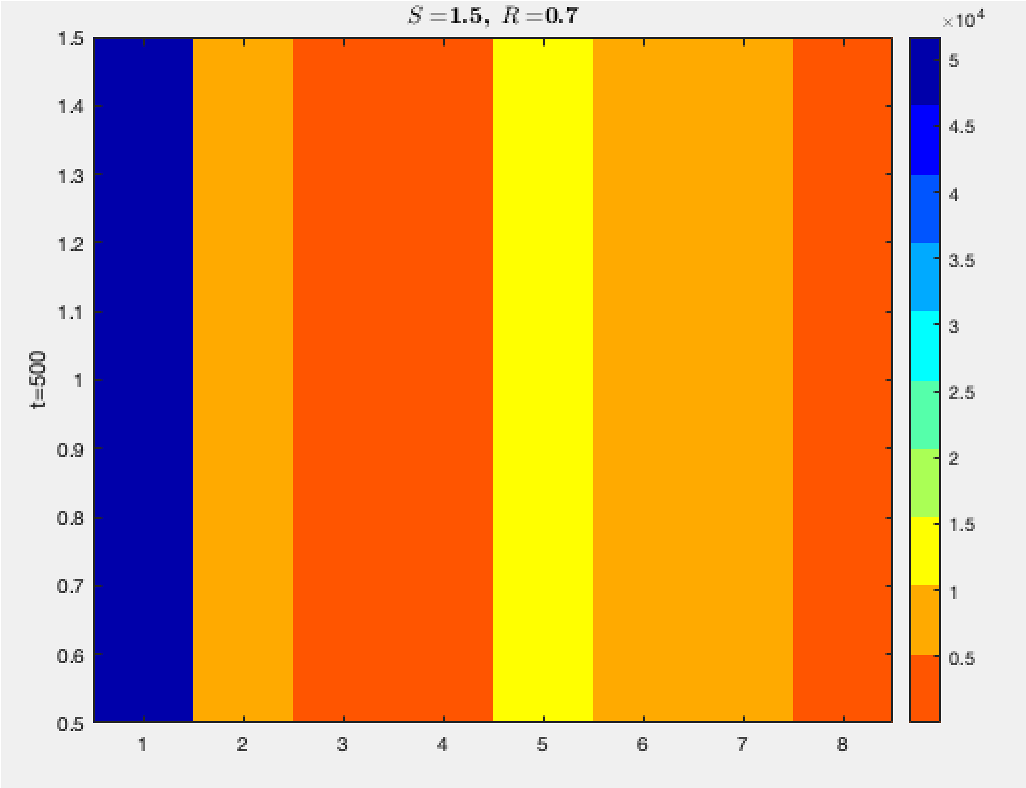
a. Simulate the evolutionary games with different values of R and S. For every simulation, record at least 500 steps and omit the first 100 steps in the analysis. Calculate the variance of the population for each strategy σ_n .

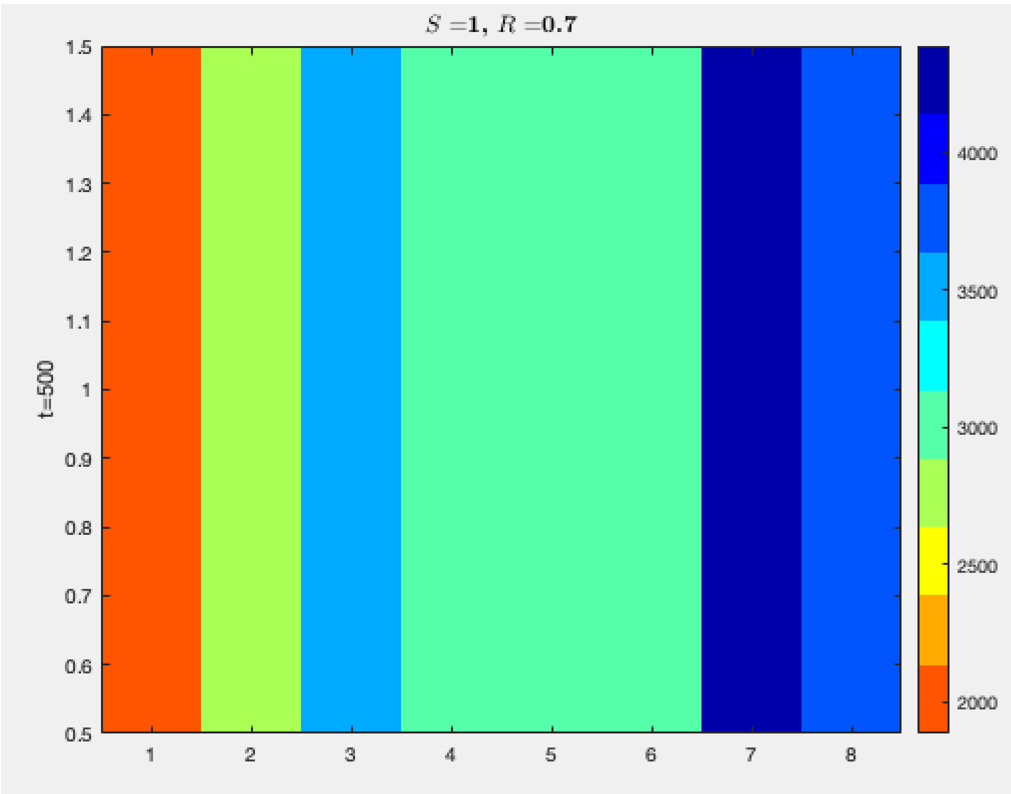
b. Determine a reasonable threshold for this parameter and create a phase diagram for the output behavior as a function of R and S.

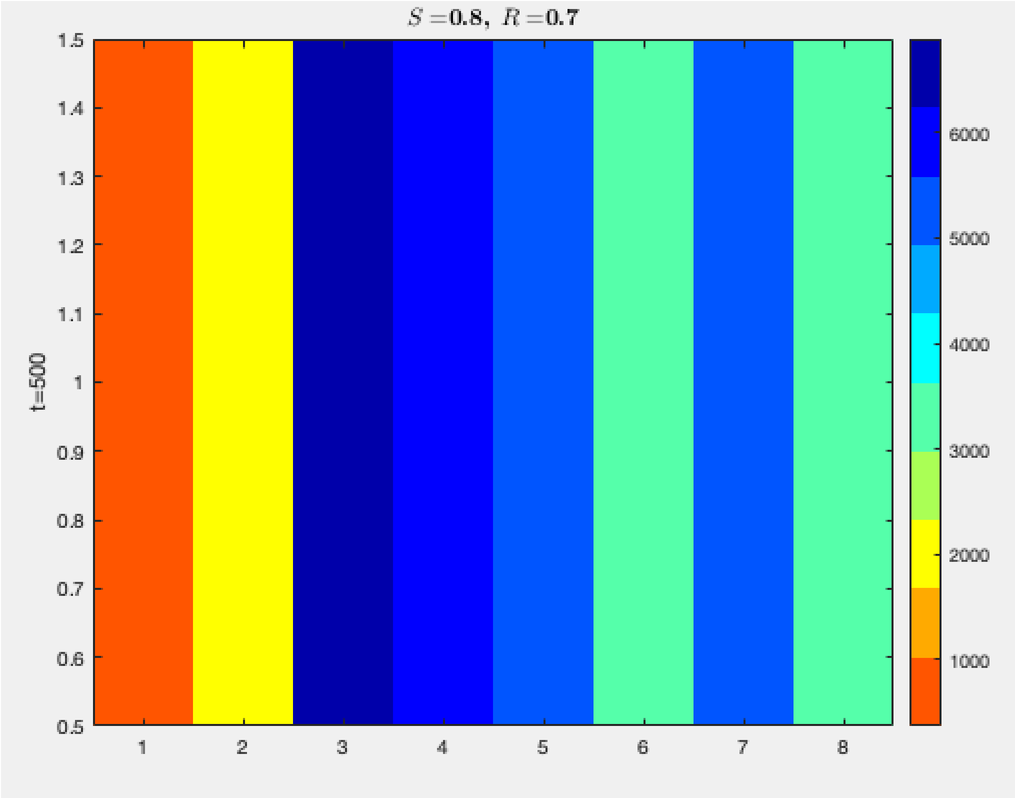


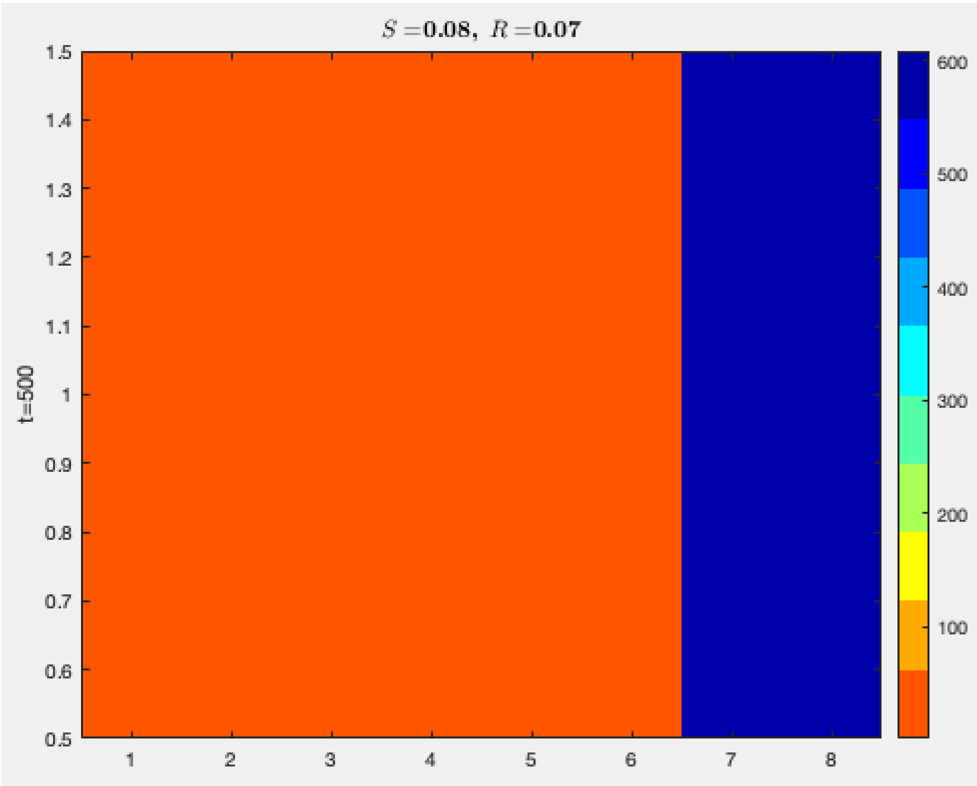

$S = 1.5, \ R = 0.7$
t=500
1.5
1.4
1.3
1.2
1.1
1
0.9
0.8
0.7
0.6
0.5
1
2
3
4
5
6
7
8
$\times 10^4$
2.5
2
1.5
1
0.5


$S = 1.5, \ R = 0.9$
t=500
1.5
1.4
1.3
1.2
1.1
1
0.9
0.8
0.7
0.6
0.5
1
2
3
4
5
6
7
8
7
6
5
4
3
2
1


$S = \mathbf{1.5},\ R = \mathbf{0.7}$
t=500
$\times 10^4$
1.5
1.4
1.3
1.2
1.1
1
0.9
0.8
0.7
0.6
0.5
1 2 3 4 5 6 7 8
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5


$S = \mathbf{1},\ R = \mathbf{0.7}$
t=500
1.5
1.4
1.3
1.2
1.1
1
0.9
0.8
0.7
0.6
0.5
1 2 3 4 5 6 7 8
4000
3500
3000
2500
2000


$S = 0.8,\ R = 0.7$
t=500
1.5
1.4
1.3
1.2
1.1
1
0.9
0.8
0.7
0.6
0.5
1
2
3
4
5
6
7
8
6000
5000
4000
3000
2000
1000


$S = 0.08,\ R = 0.07$
t=500
1.5
1.4
1.3
1.2
1.1
1
0.9
0.8
0.7
0.6
0.5
1
2
3
4
5
6
7
8
600
500
400
300
200
100

```matlab
%Prisoner's dilemma 13.2 & 13.3
%Nicole Adamah 2022
close all
clear all
clc
%% Prisoner's dilemma with multiple rounds.

N = 10;
T = 0;%the player that defects while the other cooperates will be punshied with T years
S = 1.5;%The other player that cooperates while the other will be punished with S years
R = 0.5;% both players cooperates
P = 1; % both players betray
m = 6;% other player's strategy
%T < R < P < S
n_range = 0:N;
years_in_prison = zeros(length(n_range),1);

for n=0:length(n_range)-1
    m = m;% other player's strategy, amount of ones(1=cooperate)
    if n < m
        years_in_prison(n+1) = (n)*R + (N-1-n)*P + (1)*T; %player 1 defects, player 2 cooperates

    elseif n > m
        years_in_prison(n+1) = (m)*R + (N-1-m)*P + (1)*S; %Player 2 defects, Player 1 cooperates.

    elseif n==m
        years_in_prison(n+1) = (m)*R + (N-m)*P;
    end
end
%% 11.1a
f1 = figure;
scatter(n_range,years_in_prison,100,'filled');
x = [6 6]; y = [0 12];
line(x, y,'Color', 'black', 'LineStyle', '--')
ylim([6 10])
xlabel('n')
ylabel('Years in prison')
title(['\bf{S=$' num2str(S) ', R=$' num2str(R) ', P=$' num2str(P) ', T=$' num2str(T)  '}'],'FontSize',12,'Interpreter','Latex')
%% 11.1b
f2 = figure;
imagesc(0:N, 0:N, years_in_prison)
set(gca,'YDir','normal')
hold on;
colorbar
x = [-1 12]; y = [-2 11];
line(x, y, 'Color', 'black', 'LineStyle', '--')
xlabel('m')
ylabel('n')
title('Years in prison')
```

```
%Prisoner's dilemma 13.2 & 13.3
%Nicole Adamah 2022
close all
clear all
clc
%% PARAMETERS
R = 0.07;
S = 0.08;
P = 1;
T=0;
timesteps = 500;
L=30;
N = 7;
mu = 0.01;
d=7;%number of defectors, nr 5 is for 11.2c, nr 6 is for 11.3 and nr 7 for 11.4
lattice=Initialize(L, N, d);
initialLattice = lattice;
data = zeros(timesteps, N+1);%for 11.4
%% SIMULATION
for t = 1:timesteps
    lattice_years = zeros(L);
    for i = 1:L
        for j = 1:L
%       Interaction with the four nearest von Neumann neighbors (top, bottom, left, right). Wrap around if at edges
            if (i == 1)
                topNeighbor=lattice(L,j);
            else
               topNeighbor = lattice(i-1,j);
            end
            if (i == L)
                bottomNeighbor=lattice(1,j);
            else
                bottomNeighbor = lattice(i+1,j);
            end
            if (j == 1)
               leftNeighbor=lattice(i,L);
            else
               leftNeighbor = lattice(i,j-1);
            end
            if (j == L)
               rightNeighbor=lattice(i,1);
            else
               rightNeighbor = lattice(i,j+1);
            end
            currentAgent=lattice(i,j);
            myNeighbors=[topNeighbor bottomNeighbor leftNeighbor rightNeighbor];
            years=[];
                for k=1:4
                    years(k)=PrisonersModel1(currentAgent,T,S,R,P,myNeighbors(k),N);

                end

            lattice_years(i, j) = sum(years);
        end
    end

    % Update strategy
    newLattice = lattice;
    for i = 1:L
        for j = 1:L
            if (i == 1)
                topNeighbor1=lattice_years(L,j);
            else
               topNeighbor1 = lattice_years(i-1,j);
            end
            if (i == L)
                bottomNeighbor1=lattice_years(1,j);
            else
                bottomNeighbor1 = lattice_years(i+1,j);
            end
```

```
        if (j == 1)
          leftNeighbor1=lattice_years(i,L);
        else
          leftNeighbor1 = lattice_years(i,j-1);
        end
        if (j == L)
          rightNeighbor1=lattice_years(i,1);
        else
          rightNeighbor1 = lattice_years(i,j+1);
        end
        currentAgent1 = lattice_years(i,j);
        all=[topNeighbor1, bottomNeighbor1, leftNeighbor1, rightNeighbor1,currentAgent1 ];

        optima=min(all);%change to max for 11.2c
        idx=find(all==optima);
        if length(idx)>1   % If two scores tie randomly select one of them
           idx1=idx(randi([1 length(idx)]));
        else
           idx1=idx;
        end
        % Update original lattice with best strategies
        if (idx1 == 1)
           if (i-1== 0)
              newLattice(i, j) = lattice(L, j);
           else
              newLattice(i, j) = lattice(i - 1, j);
           end

        elseif (idx1 == 2)
           if (i+1== L+1)
              newLattice(i, j) = lattice(1, j);
           else
              newLattice(i, j) = lattice(i + 1, j);
           end

        elseif (idx1 == 3)

           if (j-1== 0)
              newLattice(i, j) = lattice(i, L);
           else
              newLattice(i, j) = lattice(i, j - 1);
           end

        elseif (idx1 == 4)
           if (j+1 == L+1)
              newLattice(i, j) = lattice(i, 1);
           else
              newLattice(i, j) = lattice(i, j + 1);
           end
        elseif(idx1 == 5)
              newLattice(i, j) = lattice(i, j);
        end
      end
    end
    lattice = newLattice;
%   Mutate with probability mu
    lattice = mutate(lattice, 2, L, N, mu); % change between 2 and 3 depding on problem 11.2 or 11.3
    idxData = zeros(1, N+1);
    for i = 1:N+1
       idxData(i) = sum(lattice(:) == i-1);
    end
    data(t, :) = idxData;
end
%% PLOT 11.2
% figure;
% imagesc(initialLattice);
% colormap([1 0 0;0 0 1]);
% title(['\bf{$Initial defectors=$' num2str(d) ', $R=$' num2str(R)  '}'],'FontSize',12,'Interpreter','Latex')
% ylabel('t=0');
%
```

```
% figure;
% imagesc(lattice);
% colormap([1 0 0;0 0 1]);
% title(['\bf{$Initial defectors=$' num2str(d) ', $R=$' num2str(R)  '}'],'FontSize',12,'Interpreter','Latex')
% ylabel('t=20');
%% PLOT 11.3
% figure;
% imagesc(lattice);
% colormap([1 0 0;0 0 1]);
% title(['\bf{$S=$' num2str(S) ', $R=$' num2str(R)  '}'],'FontSize',12,'Interpreter','Latex')
% ylabel('t=100');
%% PLOT 11.4
% figure;
% imagesc(initialLattice);
% colormap(flipud(jet(N+1)))
% colorbar
% title(['\bf{$S=$' num2str(S) ', $R=$' num2str(R)  '}'],'FontSize',12,'Interpreter','Latex')
% ylabel('t=0');
%
% figure;
% imagesc(lattice);
% colormap(flipud(jet(10)))
% colorbar
% title(['\bf{$S=$' num2str(S) ', $R=$' num2str(R)  '}'],'FontSize',12,'Interpreter','Latex')
% ylabel('t=200');
% figure
% for i = 1:N+1
%     plot(1:1:timesteps, data(:,i), 'color', colors(i,:))
%     hold on
% end
% legend('n = 0', 'n = 1', 'n = 2', 'n = 3', 'n = 4','n = 5', 'n = 6', 'n = 7')
% xlabel('Time')
% ylabel('Population fraction')
% title('Number of agents with a particular strategy,  n = [0, N]')
%% PLOT 11.5
data = data(101:500, :);
v=var(data);
figure;
imagesc(v);
set(gca, 'YDir', 'normal')
colormap(flipud(jet(10)))
colorbar
title(['\bf{$S=$' num2str(S) ', $R=$' num2str(R)  '}'],'FontSize',12,'Interpreter','Latex')
ylabel('t=500');
%% FUNCTIONS
%Prison dilemma
function years_in_prison1 = PrisonersModel1(n, T,S,R,P,m,N)
years_in_prison1 = 0;
for i=1:N
    if n < m
        years_in_prison1 = (n)*R + (N-1-n)*P + (1)*T; %player 1 defects, player 2 cooperates

    elseif n > m
        years_in_prison1 = (m)*R + (N-1-m)*P + (1)*S; %Player 2 defects, Player 1 cooperates.

    elseif n==m
        years_in_prison1 = (m)*R + (N-m)*P;
    end
end
end
%% Mutation
function latticeM = mutate(lattice, mut, L, N, mu)
if (mut==2)
     for i = 1:L
        for j = 1:L
           if (rand() < mu)
              lattice(i, j) = randi(N+1) - 1;
           end
        end
     end
```

```
    latticeM=lattice;
end
if (mut==3)
   for i = 1:L
      for j = 1:L
         if (rand() < mu)
            if (rand() < 0.5)
               lattice(i, j) = 0;
            else
               lattice(i, j) = N;
            end
         end
      end
    latticeM=lattice;
   end
end
end


%% Initialize the lattice with defectors
function InitialLattice = Initialize(L,N, defector)

% Initialization.
if (defector == 1)
   lattice = N*ones(L);
   lattice(ceil(L/2), ceil(L/2)) = 0;
   InitialLattice = lattice;
end
if (defector == 2)
   lattice = N*ones(L);
   lattice(ceil(L/3), ceil(2*L/3)) = 0;
   lattice(ceil(2*L/3), ceil(L/3)) = 0;
   InitialLattice = lattice;
end
if (defector == 3)
   lattice = N*ones(L);
   lattice(ceil(L/2), ceil(L/2)) = 0;
   lattice(ceil(L/4), ceil(3*L/4)) = 0;
   lattice(ceil(3*L/4), ceil(L/4)) = 0;
   InitialLattice = lattice;
end
if (defector == 4)
   lattice = N*ones(L);
   lattice(ceil(L/5), ceil(4*L/5)) = 0;
   lattice(ceil(2*L/5), ceil(3*L/5)) = 0;
   lattice(ceil(3*L/5), ceil(2*L/5)) = 0;
   lattice(ceil(4*L/5), ceil(L/5)) = 0;
   InitialLattice = lattice;
end
if (defector==5)
   lattice = zeros(L);
   lattice(ceil(L/2), ceil(L/2)) = N;
   InitialLattice = lattice;
end
if (defector == 6)
   strategy = [0, N];
   r = randi([1, 2], L);
   lattice = strategy(r);
   InitialLattice = lattice;
end
if (defector == 7)
   lattice = randi([0, N], L);
   InitialLattice = lattice;
end

end
```