作业1：规则格子网络雪堆博弈仿真：

雪堆博弈中博弈个体在不同决策下的收益报酬为：，且上述收益关系满足T>R>S>P，假设b为完成任务的收益，c为完成任务付出的成本，雪堆博弈的收益矩阵如下：

|  |  |  |
| --- | --- | --- |
|  | C | D |
| C | b-c/2；b-c/2 | b-c，b |
| D | b，b-c | 0，0 |

令r=c/(2b-c)可得简化模型如下，其中R=1；T=1+r；S=1-r；P=0；

|  |  |  |
| --- | --- | --- |
|  | C | D |
| C | 1，1 | 1-r，1+r |
| D | 1+r，1-r | 0，0 |

每个个体的博弈收益是与所有邻居进行博弈的收益总和。个体采用复制动力学更新规则调整策略，模仿概率为



其中，和分别代表个体和个体的博弈收益，表示规则格子的维度,为博弈情境中最大参数与最小参数 0 的差。

仿真1：合作者密度随博弈次数变化曲线仿真

初始状态下采取C策略的个体比例为：p0 = 0.4 ；

雪堆博弈初始值r = 0.5 ；

博弈次数为 800次；

网格行列数为 100；

20次博弈结果不变时停止仿真 ；

仿真1的博弈收益矩阵如图1.1所示。

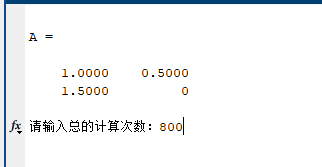


图1.1 仿真1博弈收益矩阵

规则格子网络雪堆博弈合作者密度随博弈次数变化的曲线图如图1.2所示，初始合作者密度为0.4，随着博弈次数的增加，合作者密度在0.345上下浮动。

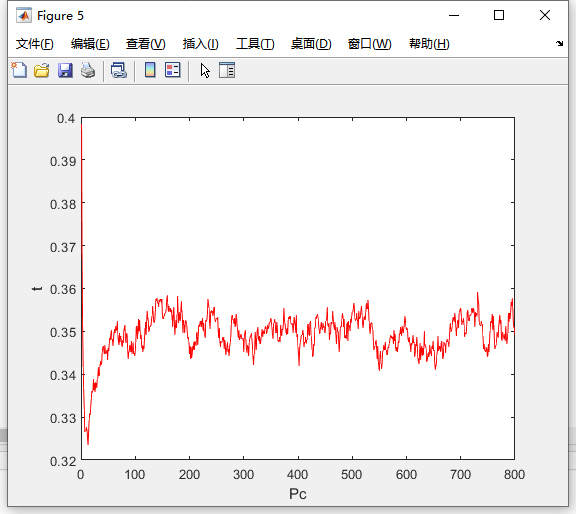


图1.2 合作者密度随博弈次数变化的曲线图

仿真2：初始采用C策略比例p0、雪堆博弈初始值r、合作者密度关系仿真

初始状态下采取C策略的个体比例为：p0 = 0.4：step：0.6，step=0.1 ；

雪堆博弈初始值r = 0.4:step:0.6,step=0.1 ；

博弈次数为 500次；

网格行列数为 100；

20次博弈结果不变时停止仿真 。

在仿真2初始条件设置下，规则格子网络雪堆博弈的初始采用C策略比例p0、雪堆博弈初始值r、合作者密度三维关系曲线图如图2.1所示。初始概率P0为常数的情况下，随着博弈参数r的增大，合作者均衡状态的密度逐渐减小，这是因为r的值越大背叛的收益会变大，导致趋于合作的密度在减小。当模仿概率P0较高时，个体更有可能模仿其邻居的合作策略，这可能会导致合作策略在网络中的传播和稳定。而合作者密度则反映了网络中合作个体的比例，合作者密度较高时，整个网络中的合作行为可能更加普遍和稳定。

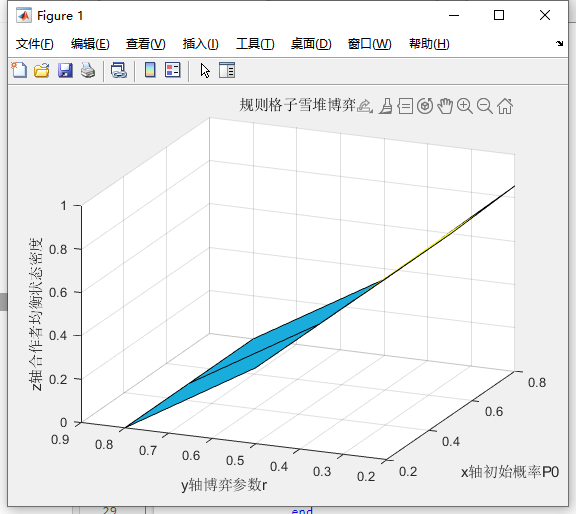


图2.1 初始采用C策略比例p0、雪堆博弈初始值r、合作者密度曲线图

作业2（仿真3）：WS小世界上的雪堆博弈

网络模型：以最近邻耦合网络k=4为基础，以概率P进行断边重连；

雪堆博弈参数为r；

按照轮盘赌的方式选择进行博弈的邻居节点，连接度越高的节点被选中的概率越大；

博弈策略更新规则为：



仿真代码如下：

% Snowdrift博弈模型模拟与可视化

N = 100; % 节点数量

numIterations = 100; % 迭代次数

% 参数设置

ks = [4, 5]; % 邻居数量

rs = [0, 0.5]; % 适应度变化率

ps = [0, 0.1]; % 断边重连概率

subplotIdx = 1;

figure;

for k = ks

for r = rs

for p = ps

% 生成小世界网络

net = WattsStrogatz(N, k, p);

% 初始化节点合作者策略

U = randi(10, 1, N);

cooperatorDensity = zeros(1, numIterations);

% 模拟演化过程

for t = 1:numIterations

for i = 1:N

neighborIdx = chooseNeighbor(net, i);

diffU = U(neighborIdx) - U(i);

P\_ij = diffU / ((1 + r) \* max(1, diffU));

if rand() < P\_ij

U(i) = U(i) + 1;

end

end

cooperatorDensity(t) = sum(U > 5) / N;

end

% 绘制合作者密度随时间的变化曲线

subplot(length(ks), length(rs) \* length(ps), subplotIdx);

plot(1:numIterations, cooperatorDensity);

xlabel('Iterations');

ylabel('Cooperator Density');

title(sprintf('k=%d, r=%.1f, p=%.1f', k, r, p));

subplotIdx = subplotIdx + 1;

end

end

end

% 辅助函数：选择邻居节点

function neighborIdx = chooseNeighbor(net, nodeIdx)

neighbors = find(net(nodeIdx, :));

neighborIdx = neighbors(randi(length(neighbors)));

end

% Watts-Strogatz小世界网络生成函数

function G = WattsStrogatz(N, k, p)

G = zeros(N, N);

for i = 1:N

for j = 1:k/2

idx1 = mod(i+j-1, N) + 1;

idx2 = mod(i-j-1, N) + 1;

G(i, idx1) = 1;

G(i, idx2) = 1;

end

end

for i = 1:N

for j = 1:N

if G(i, j) == 1 && rand() < p

newNeighbor = randi(N);

while newNeighbor == i || G(i, newNeighbor) == 1

newNeighbor = randi(N);

end

G(i, j) = 0;

G(i, newNeighbor) = 1;

end

end

end

end

仿真结果图如图3.1和3.2所示：



图3.1 雪堆博弈中合作者密度随迭代次数变化的曲线图

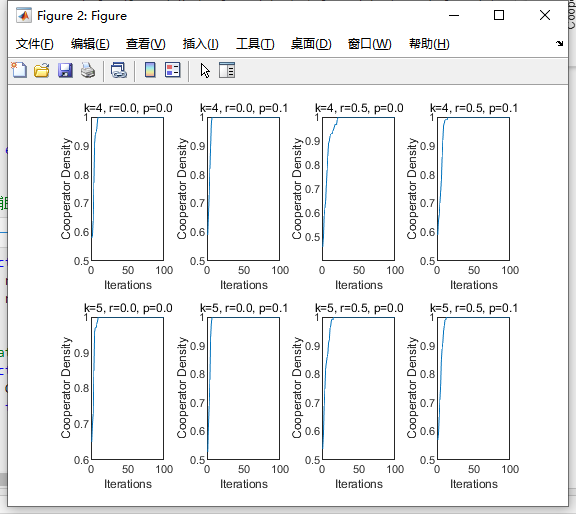


图3.2 小世界雪堆博弈中不同参数下合作者密度的演进变化图