基于蒙特卡洛方法随机决策的多人博弈模拟与分析

摘要

近年来群体随机决策系统的研究得到了应用数学、博弈论、控制论、系统论、计算机科学等领域专家学者的广泛关注。相关研究对强化学习、群体智能、智能决策、群体行为预测与分析，多智能体协作等问题的研究具有重要的理论意义和应用价值。而在生活中，博弈论的运用也是屡见不鲜。

基于此，我们在本次实验中，结合部分博弈知识、随机模拟、蒙特卡洛方法等，通过建立合理的模型，对一个多人博弈的过程进行模拟和分析。

首先，我们采用随机产生的 0 和 1 来代表最初两种决策的个体，以模拟初始决策情况。通过将矩阵最后一位数添加到矩阵最前面，将矩阵第一位数添加到矩阵最后形成一个包含 202 个数的矩阵，该矩阵中间 200 位数左右均有邻居，以此模拟首尾相连的环状网络。再通过随机产生-1 和 1 决定每一个个体在博弈阶段结束应该向左邻居还是右邻居学习策略。接着通过所得出的每一个个体的利益计算个体策略调整为所确定的学习对象的策略的概率，再次取得 200 个范围为 0 到 1 的随机数与调整概率相比较，确定是否调整策略（若对应随机数小于等于概率则调整，大于则不调整)。然后对策略矩阵进行调整。我们通过循环实现对 2000 轮的博弈的模拟，并且记录每一轮中策略矩阵中 0 的个数，除以总个体数即 A 决策个体的比例。

其次我们将每一轮 A 决策个体比例绘制为图表，以便能够更加直观的看出我们所模拟过程中 A 决策个体在 2000 轮博弈中所占的比例变化，最后的模拟结果也与我们通过博弈论的部分知识以及自己的推导判断得出的结果相同；同时我们也对在什么轮次 A 决策个体比例达到 0 时进行了研究，并得出了令人信服的结果。

最后我们分优点、缺点对模型进行剖析，进一步验证模型，并给出合理的评价，提出之后的改进方向；与此同时，我们利用该模型对相似的囚徒困境博弈进行了模拟，并得到了理想的结果；我们还撰写了心得体会与总结，并对本实验问题的设计提出了改进意见。

关键词：蒙特卡洛，随机决策，多人博弈

目录

一、问题重述

在某一博弈游戏中，有首尾相连的一个由 200 个节点组成的环状网络（如图 1 所示），其中每一个节点代表一个决策个体，个体之间可以进行博弈决策，其中每个个体可以有两种决策行为，分别为 A 和 B。初始时刻，每个个体等概率地选择 A 和 B 两种决策行为。每一轮博弈决策的过程总体上分成两个阶段：博弈阶段和决策更新阶段。在博弈阶段，每一个个体根据自己的决策行为与其所有的邻居（邻居定义为与该个体直接相连的个体）选择的决策行为进行博弈，并获得相应的效益。当 A 决策个体遇到 A 决策个体时，则其获得效益为 3；当 A 决策个体遇到 B 决策个体时，则其获得效益为 0。当 B 决策个体遇到 A 决策个体时，则其获得效益为 5；当 B 决策个体遇到 B 决策个体时，则其获得效益为 1。环状网络上每个个体 i 都与其所有的邻居进行博弈，在该轮博弈中其获得的收益为 U_i。本轮博弈阶段结束后，每个个体都需要更新自己的决策行为，即每一个个体随机向两位邻居中的一位（j）进行决策学习。个体 i 将自己的决策行为调整为 j 的决策行为的概率：$p=\frac{1}{1+e^{\frac{(U_i-U_j)}{0.5}}}$。

该实验要求用 Matlab 语言编写程序仿真模拟上述环状网络上所有个体 2000 轮的博弈决策过程，并画出选择 A 决策行为个体的比例在 2000 轮博弈决策过程中随时间的变化曲线。

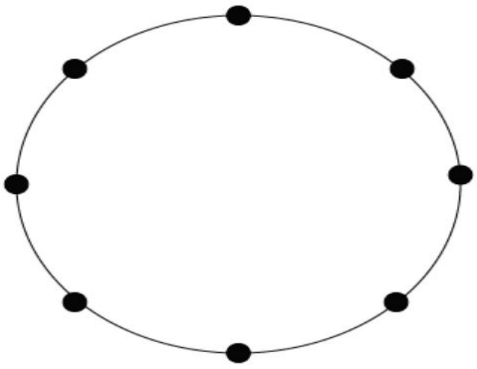


图 1　环状网络结构示意图

二、问题分析

总体而言，我们需要用 Matlab 语言编写程序仿真模拟题中环状网络上所有个体 2000 轮的博弈决策过程，并画出选择 A 决策行为个体的比例在 2000 轮博弈决策过程中随时间的变化曲线。将目标分解可以得到以下几点。

2.1 博弈阶段

博弈本意是：下棋。引申义是：在一定条件下，遵守一定的规则，一个或几个拥有绝对理性思维的人或团队，从各自允许选择的行为或策略进行选择并加以实施，并从中各自取得相应结果或收益的过程。有时候也用作动词，特指对选择的行为或策略加以实施的过程。[2]

而本题中 200 个个体都有着一定的策略，每个个体与其一个邻居的不同策略组合将导致不同的收益。我们需要设计一种算法将两个策略输入转化成一个收益输出，可供选择的有使用 if-else 结构，以及构造一个二元函数将四种策略组合映射成四个收益，其中策略可用 0 和 1 来表示。每个个体与两个邻居分别博弈所获的收益之和作为本轮该个体的总收益。

2.2 决策更新

当一轮博弈结束，每个个体将获得一个总收益，此时该个体需要更新决策以准备参与下一轮博弈。决策更新可分为两步：学习对象选择，以及选择是否更换学习对象的决策。

每个个体等概率选择一个邻居，此时可考虑结合 rand 和 round，并使用线性运算，使得(0,1)上的随机数均匀映射到-1 和 1，分别表示向前一个邻居学习或是向后一个邻居学习。

但学习并不一定意味着一定将决策改变为对方的决策，而是按照一定概率发生改变，具体概率是 $p=\frac{1}{1+e^{\frac{(U_i-U_j)}{0.5}}}$。在一定概率下是否改变为对方决策，可以用 rand 生成的随机数是否处于给定区间来表示。

2.3 题目背景

本题与博弈论关系密切：研究一场博弈的纳什均衡[3]本来就是博弈论的内容之一。

假设每个个体都是理性的决策者，每个人都希望采取对自己最优的策略，在博弈中获得最大收益。而决策改变概率 $p=\frac{1}{1+e^{\frac{(U_i-U_j)}{0.5}}}$ 也印证了这一点——对方收益高于自己时，个体会有转变为对方策略的趋势；对方收益低于自己时，个体会趋向于不转变策略。故本次模拟实验正是对一群理性人在累次博弈中达到纳什均衡的模拟，模拟结果也应与实际情况相符。就本题而言，无论对方是 A 决策个体还是 B 决策个体，自己选择 B 的收益都会比自己选择 A 的收益高，故纳什均衡应该是所有个体都转变为 B 决策个体，模拟结果也应该是 A 个体比例最终降为 0。

三、 模型假设

1. 第一轮等概率产生两种决策。
2. 假设 Matlab 中随机数表示的策略能代表真实个体随机判断。
3. 假设 Matlab 中循环过程真实反映每一轮博弈决策过程。

四、 定义与符号说明

4.1 名词解释

模拟环状结构：由于环状结构收尾相连，因此我们将矩阵最后一项添加到矩阵最前，将矩阵原本的第一项添加于最后，这样得到的新矩阵从第二项开始左右都有相邻者，以此模拟环状结构。

4.2 符号说明

符号	说明
start	初始时所有个体的决策矩阵（0 代表 A 决策个体，1 代表 B 决策个体）
numberoa	每一轮 A 决策个体的数目矩阵
ratio	每一轮 A 决策个体数比例
strategy	将 start 模拟为环状结构得到的矩阵
U	每个个体的效益矩阵
porenew	决策更新概率，与代码中函数 renew 对应
profit	a 与 b 博弈 a 所获效益，与代码中函数 game 对应
act	用于判定是否更改策略

五、 模型的建立与求解

5.1 建模前准备

1. 分析如何模拟环状结构
2. 分析如何设计随机向一位邻居进行策略学习
3. 分析如何判定是否更改决策
4. 确定是否更改决策后如何具体实现决策更改的过程

5.2 模型建立

1. 初始时刻等概率选择决策行为

我们令随机数 0 为 A 决策个体，随机数 1 为 B 决策个体。之后用 ratio 来统计初始 A 决策个体的比例，同时建立关于决策的环状结构。

strategy =[start(200),start,start(1)]

2. 建立相邻个体博弈后效益的函数

profit = 3 + 2a - 3b - a.*b

函数建立方法：考虑设计一个能将四种策略组合映射成四个收益的二元函数，其形式为 $profit = C_1 + C_2a + C_3b + C_4 ab$，将四种策略组合和其对应的四个收益代入以求得参数。

3. 建立策略调整概率函数

porenew = 1./(1+exp((a-b)/0.5))

4. 进行博弈和决策阶段

首先将决策环状结构代入效益函数，得到不同个体的博弈效益数组 U，同时建立效益的环状结构。（U = [U(200),U,U(1)]）

之后建立不同个体的学习对象数组，（通过随机数-1 和 1 代表前后邻居）。

代入策略调整概率函数后，根据随机数组，我们判断如果随机数小于调整概率则调整决策，反之则不调整，同时令 0 代表不调整决策，1 代表调整决策。

得到新一轮的决策环状结构后，统计新一轮的 A 决策个体的比例。

以此进行循环，得到 2000 轮的相关数据。

5. 利用 plot 函数输出图像

5.3 模型求解

根据 5.2 建立模型可以写出求解程序，详见附件一。多次运行所写程序，都得到一条 L 型曲线，且在不到 20 轮时，A 决策个体比例已经降至 0，这与我们 2.3 的分析是一致的。所求 A 决策行为个体的比例在 2000 轮博弈决策过程中随时间的变化曲线见下图 2。这里仅展示其中一次的运行结果，更多 A 决策行为个体比例随时间的变化曲线见附件三。

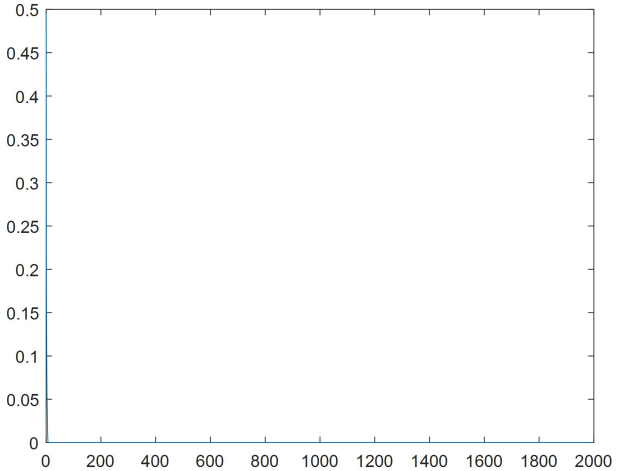


图 2　A 决策行为个体比例随时间的变化曲线

记录多次运行结果中 A 决策个体比例达到 0 时的轮次，统计其不同轮次数的频次，见下图。

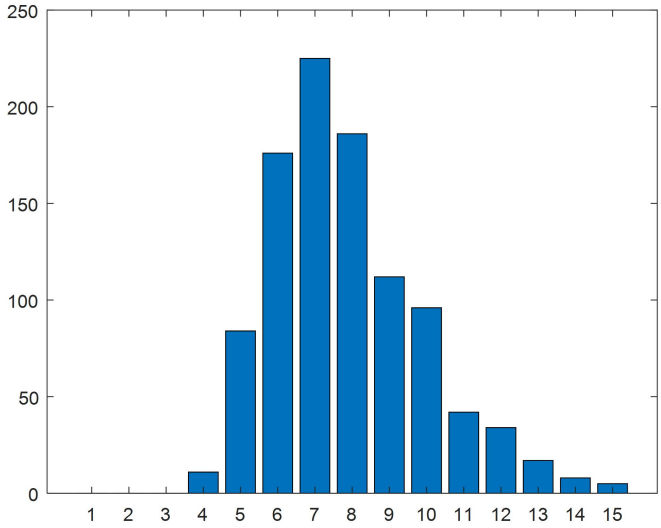


图 3　A 决策个体比例达到 0 时的轮次统计

可见多数时候在 7 轮左右时所有个体已经变成了 B 决策个体，说明我们的模拟是稳定的；且轮次呈正态分布，符合实际。

六、 模型的评价与推广

6.1 模型优点

利用计算机随机模拟方法，避免了过多次的真实实验过程，节约了时间，也节省了人力、物力。

省却了繁复的数学推导和演算过程，简单快速地得到所需图像，便于理解。

其中的代码充分利用 matlab 的数组运算功能，尽量避免较多循环；采用函数运算，使代码简洁有序。

6.2 模型缺点

计算量巨大，且准确性的提高速度较慢。

计算机模拟产生的是伪随机数，可能导致错误结果。

6.3 模型改进

1. 可以继续优化代码，如去掉不必要的语句，用矩阵运算代替循环，减少代码行数，以减小时间复杂度。

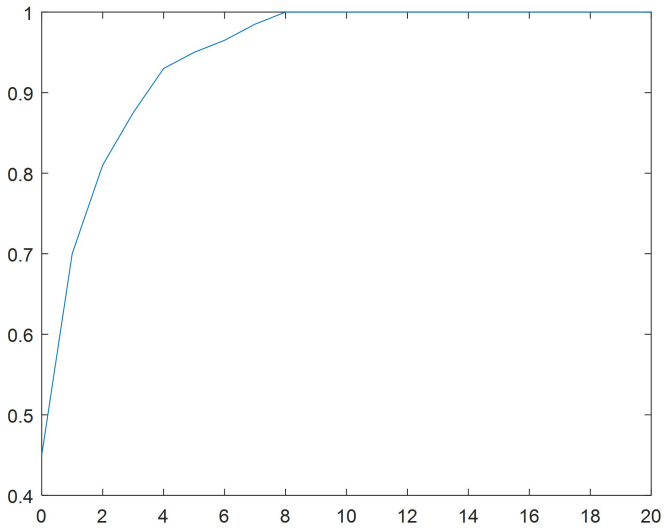
2. 注意代码的层次、缩进，增加代码可读性。

6.4 模型推广

1. 对于一些复杂的随机变量，不能从数学上得到它的概率分布时，可以采用计算机的随机模拟得到近似的解答。

2. 本模型也可用于求一些解析方法或者常规数学方法难解问题的低精度解，或用于对其他算法的验证。

3. 本程序可以用于其他多种博弈过程的模拟，只需要改变少量代码即可。下面展示囚徒困境的博弈模拟。



上图为一次 20 轮囚徒困境模拟，代码详见附件二。可以看到，在 10 轮以内已经达到均衡，所有人均选择了坦白。

七、 心得体会与总结

通过本次综合性实验，我们感受到了 Matlab 相对于 C 语言在处理矩阵以及整块运算处理数学问题上的优越性；也感受到了在随机模拟实验中 Matlab 的运用之便捷。不仅仅对于这一个实验题目，在随机模拟，博弈论还有生活中的各个方面的问题，我们都可以通过 Matlab 对问题进行分析处理。

在本次实验中，我们小组通力合作，分工明确，完成了本次实验，也感受到了一个团体中合作的重要性。同时我们认为一个团体中每一位成员确实是可以提出自己不同的看法，集思广益，但最后确定一个方案即统一意见后，每一位成员都需要有团队精神，互帮互助，通力合作才能把事情做好。

此外在我们完成本次实验的过程中，我们不断去尝试，去讨论，去查阅资料，也更加感受到了这门课程的魅力，更深入的理解了随机系统建模过程，也更加熟悉了模拟算法设计和系统仿真的步骤。同时这也帮助我们了解了博弈论的相关知识和人工智能领域的相关前沿课题研究。

八、 对本实验问题的设计提出改进意见

本问题新颖，内容贴近科技前沿，在加深对数学实验课程相关内容的理解和掌握的同时，帮助学生了解人工智能领域的相关前沿课题研究。但希望做出如下改进：

1. 使题目描述更精确，不引起歧义。如“根据自己的决策行为”可以指明是“根据自己上一轮的决策行为”。这一点在小组讨论中引起了分歧。

2. 可以适当加大题目难度。我组程序一中的有效代码不超过 25 行，所用时间也不算太多。可以尝试使博弈策略及与收益的关系复杂化等等。

3. 可以考虑增加实验任务，加入更具体的博弈案例进行模拟。如囚徒困境，市场进入阻扰博弈等等。

4. 可以尝试将题目写成数模题目的形式，以便学生更好的完成论文写作。

5. 论文模板可以只保留二级标题，三级及以下标题几乎无用。

九、 参考文献

[1] 电子科技大学数学实验课程组，《数学实验与数学建模基础》

[2] 百度百科 博弈 https://baike.baidu.com/item/博弈/4669968?fr=aladdin 2020.6.17 17：00

[3] MBA 智库百科 纳什均衡 https://wiki.mbalib.com/wiki/纳什均衡 2020.6.17 10：54

十、 附件清单

附件一：程序一，模拟 2000 轮博弈的第一个 Matlab 程序

附件二：程序二，模拟 20 轮囚徒困境博弈的 Matlab 程序

附件三：更多 A 决策行为个体的比例在 2000 轮博弈决策过程中随时间的变化曲线

十一、 附件

附件一：程序一，模拟 2000 轮博弈的第一个 Matlab 程序

```
%初始博弈策略
start = rand(1,200);
start(start <=0.5) = 0;    %A 决策个体
start(start > 0.5) = 1;    %B 决策个体
numberoa = sum(1-start);    %初始 A 决策个体数(number of A)
```

```
ratio = numberoa/200    %初始 A 决策个体数比例
strategy = [start(200),start,start(1)];    %环状结构，便于矩阵运算

for i = 1:2000
    %一轮收益
    U = game(strategy(2:201),strategy(1:200)) + game(strategy(2:201),strategy(3:202));
    U = [U(200),U,U(1)];    %环状结构，便于矩阵运算
    %策略更新准备
    leaobject=2*round(rand(1,200))-1+[2:201];    %学习对象(learning object)随机选择
(邻居)
    %策略调整概率
    porenew = renew(U(2:201),U(leaobject));
    %是否调整策略
    act = rand(1,200);
    act(act < 1-porenew)=0;    %不调整策略
    act(act >=1-porenew)=1;    %调整策略
    %策略更新
    strategy(2:201)=act.*strategy(leaobject)+(1-act).*strategy(2:201);
    %环状结构更新
    strategy(1) = strategy(201);
    strategy(202) = strategy(2);
    ratio = [ratio,sum((1-strategy(2:201)))/200];    %添加此轮 A 决策个体比例
end
plot(0:2000,ratio)


function profit = game(a,b)
    profit = 3 - 3*b + 2*a - a.*b;
    %容易知道 a 与 b 博弈后收益满足上式
end

function porenew = renew(a,b)
    porenew = 1./(1+exp((a-b)/0.5));
    %策略调整概率,possibility of renewing
end
```

注：

上述红色 game 函数为计算效益的函数，我们也可以用循环来代替，程序如下。

```
    for i=2:201
        if R(i)==0&&R(i-1)==0
            U(i-1)=U(i-1)+3;
        elseif R(i)==0&&R(i-1)==1
            U(i-1)=U(i-1);
```

```
        elseif R(i)==1&&R(i-1)==0
            U(i-1)=U(i-1)+5;
        elseif R(i)==1&&R(i-1)==1
            U(i-1)=U(i-1)+1;
        end
    end
    for i=2:201
        if R(i)==0&&R(i+1)==0
            U(i-1)=U(i-1)+3;
        elseif R(i)==0&&R(i+1)==1
            U(i-1)=U(i-1);
        elseif R(i)==1&&R(i+1)==0
            U(i-1)=U(i-1)+5;
        elseif R(i)==1&&R(i+1)==1
            U(i-1)=U(i-1)+1;
        end
    end
```

程序二：程序二，模拟 20 轮囚徒困境博弈的 Matlab 程序

```
%初始博弈策略
start = rand(1,200);
start(start <=0.5) = 0;   %A 决策个体
start(start > 0.5) = 1;   %B 决策个体
numberoa = sum(1-start);   %初始 A 决策个体数(number of A)
strategy = [start(200),start,start(1)];   %环状结构，便于矩阵运算

for i = 1:20
    %一轮收益
    U = game(strategy(2:201),strategy(1:200)) + game(strategy(2:201),strategy(3:202));
    U = [U(200),U,U(1)];   %环状结构，便于矩阵运算
    %策略更新准备
    leaobject=2*round(rand(1,200))-1+[2:201];   %学习对象(learning object)随机选择
(邻居)
    %策略调整概率
    porenew = renew(U(2:201),U(leaobject));
    %是否调整策略
    act = rand(1,200);
    act(act < 1-porenew)=0;   %不调整策略
    act(act >=1-porenew)=1;   %调整策略
    %策略更新
    strategy(2:201)=act.*strategy(leaobject)+(1-act).*strategy(2:201);
    %环状结构更新
    strategy(1) = strategy(201);
```

```
    strategy(202) = strategy(2);
    numberoa = [numberoa,sum((1-strategy(2:201)))];   %添加此轮 A 决策个体数
end
plot(0:20,numberoa)

function profit = game(a,b)
    profit = -8 + 8*b - 2*a + a.*b;
    %容易知道 a 与 b 博弈后收益满足上式
end

function porenew = renew(a,b)
    porenew = 1./(1+exp((a-b)/0.5));
    %策略调整概率,possibility of renewing
end
```

附件三：更多 A 决策行为个体的比例在 2000 轮博弈决策过程中随时间的变化曲线

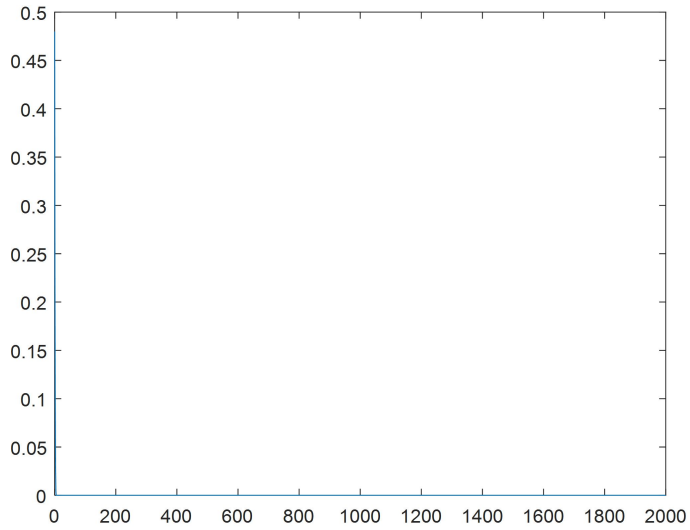


图 4　A 决策行为个体比例随时间的变化曲线（第 5 轮降为 0）

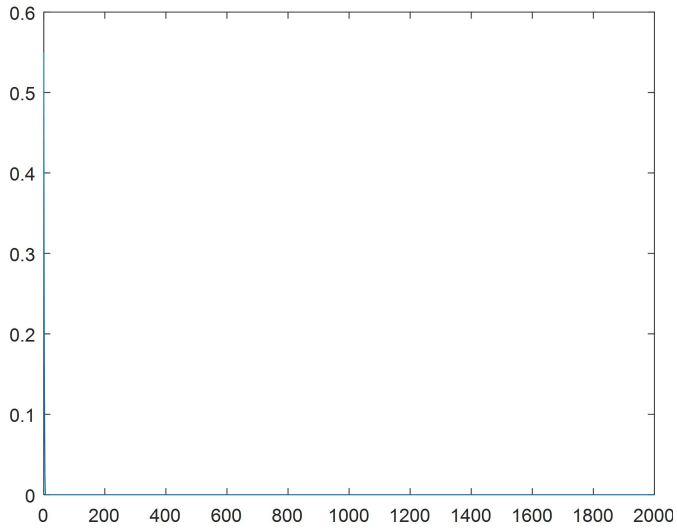


图 5　A 决策行为个体比例随时间的变化曲线（第 6 轮降为 0）

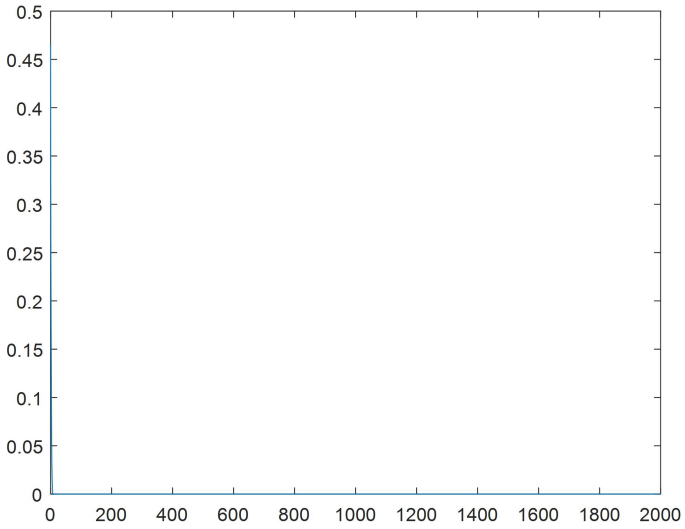


图 6　A 决策行为个体比例随时间的变化曲线（第 7 轮降为 0）

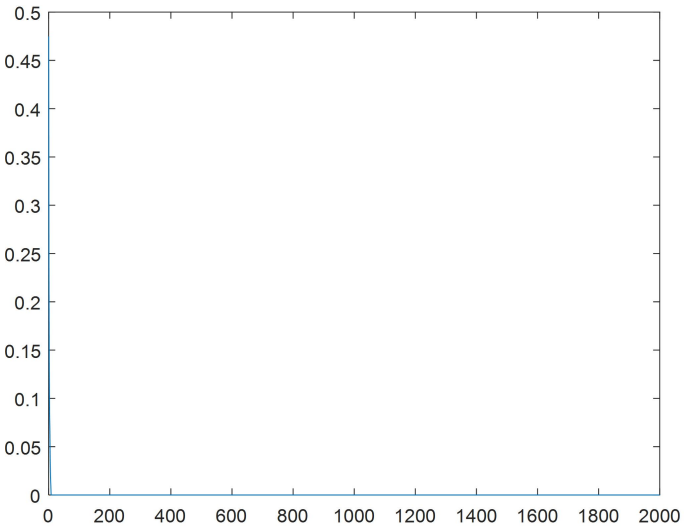


图 7　A 决策行为个体比例随时间的变化曲线（第 8 轮降为 0）

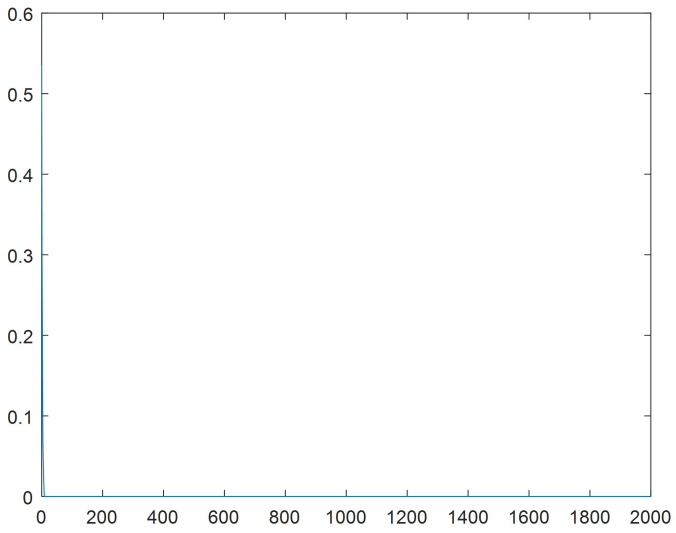


图 8　A 决策行为个体比例随时间的变化曲线（第 9 轮降为 0）

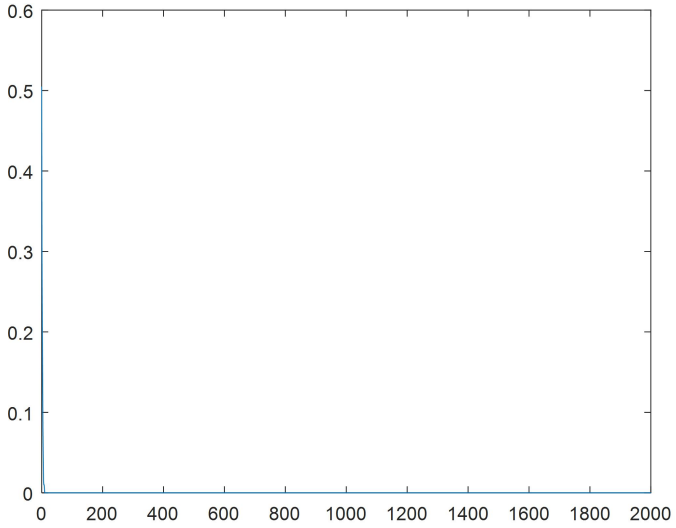


图 9　A 决策行为个体比例随时间的变化曲线（第 10 轮降为 0）

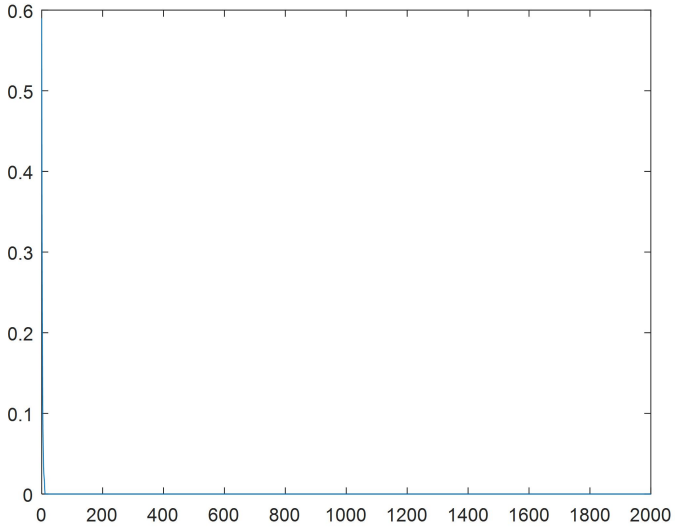


图 10　A 决策行为个体比例随时间的变化曲线（第 11 轮降为 0）

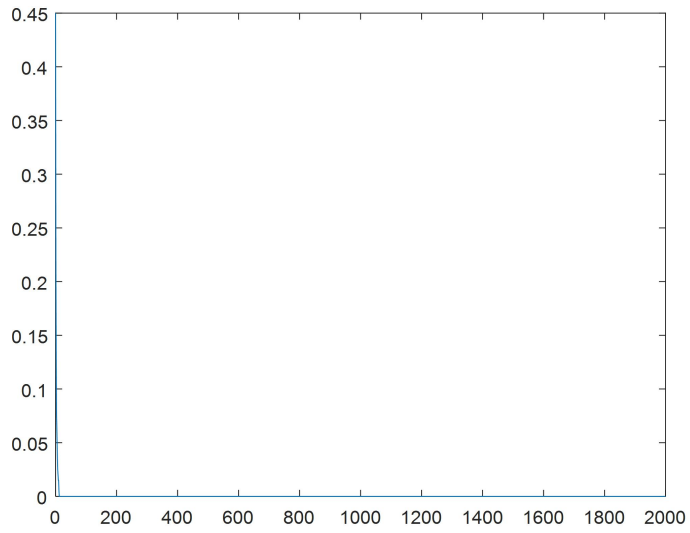


图 11　A 决策行为个体比例随时间的变化曲线（第 12 轮降为 0）

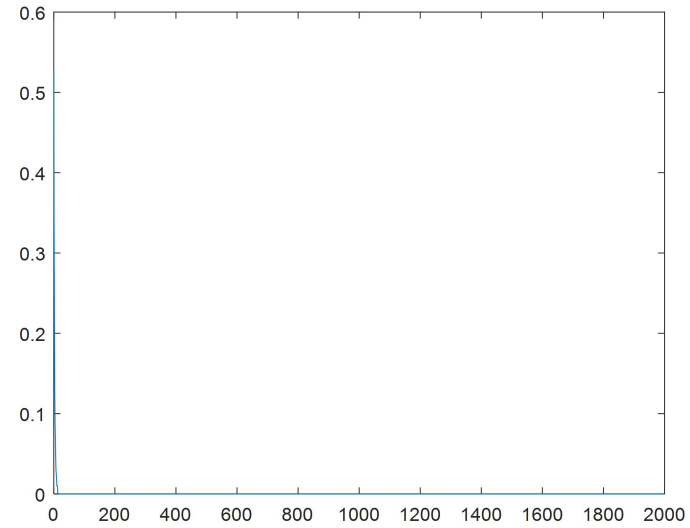


图 12　A 决策行为个体比例随时间的变化曲线（第 13 轮降为 0）